Speaker Identification with Feature Extraction and Feature Matching

Bassel Alkhatib, Mohammad Madian Kamal Eddin
Syrian Virtual University
{drbasselalkhatib@gmail.com, k.madian123@gmail.com}

ABSTRACT: *Speech processing and voice modelling activities are important where the speaker identification plays a role. The critical security systems depend on the speaker definition application. The main issue is the large scale voice recognition. Rapid and new techniques use computational intelligence for speaker database construction. For speaker identification attempts have been initiated for the establishment of variable-based systems and the development of new methodologies. These activities include the process of recognizing the actors who use speaking with the characteristics extracted from the speech's waves like pitch, tone and frequency. The speaker's models are created and saved in the system environment and used to verify the identity required by people accessing the systems, which allows access to various services that controlled by voice. These processes include two components, the first part is the feature extraction and the second part is the feature matching.*

Keywords: Speaker Identification, MFCC, Vector Quantization, Recording and Signal Processing

Received: 4 April 2020, Revised 19 July 2020, Accepted 9 August 2020

DOI: 10.6025/stj/2020/9/2/43-55

1. Introduction

The sound is a signal that contains a tone or several tones that used to communicate between humans or any living organism , through which they expresses what they wants to say or do consciously or unconsciously, and the sense caused by those vibrations is called hearing. Sound is the basis of many of the experiences acquired by man. In the past, man was not only dependent on the sounds he made out of his throat, but also on the sounds of drums and instruments that make jingling and crackling. The speed of sound in the center of a normal antenna estimated at 343 m/s or 1224 km/h. The speed of the sound related to the hardness factor and the density of the material in which the sound is moving. The audio signal is constantly changing, to simplify things, assume that on short time scales the sound signal does not change much (when we say it does not change, we mean statistically constant, obviously the samples change continuously even on short time scales). There is a difference between recognizing voices (recognizing who is speaking) and recognizing speech (recognizing what said). These terms are often confused, and "voice recognition" used for both. In addition, there is a difference between the act of authentication (referred to as verifying speaker authentication) and identification.

A speaker's understanding would simplify the task of translating speech into systems that trained on a person's voice or used

to authenticate or verify the identity of the speaker and consider it as part of a security operation.

The main objective of voice-based systems is to customize security operations to suit the needs of users and to contribute to the development of security operation performance. It is therefore critical to create individual files based on the analysis of the speaker's voices. These data should use and effectively exploited in the security environment. Artificial intelligence techniques are useful for several reasons; including the ability to develop and mimic decision-making processes, many artificial intelligence techniques used by security systems based on physical properties such as Gaussian Mixture Models (GMM), Hidden Markov models (HMM), Artificial Neural Network (ANN) and Vector quantization (VQ).

The goal of this paper is to build a security system that identifies the users from the unique physical properties of each sound and moves from traditional methods to more rigorous and highly reliable user identification patterns. For that reason digital processing speech signal and the selection of the voice recognition algorithm is very important for fast and accurate automatic voice recognition system. In order to recognize the speaker in the speaker identification system, the signal must pass through several phases, which we will present briefly:

1. Recording & signal processing
2. Feature Extraction
3. Feature Matching

The following figure shows the system's structure:

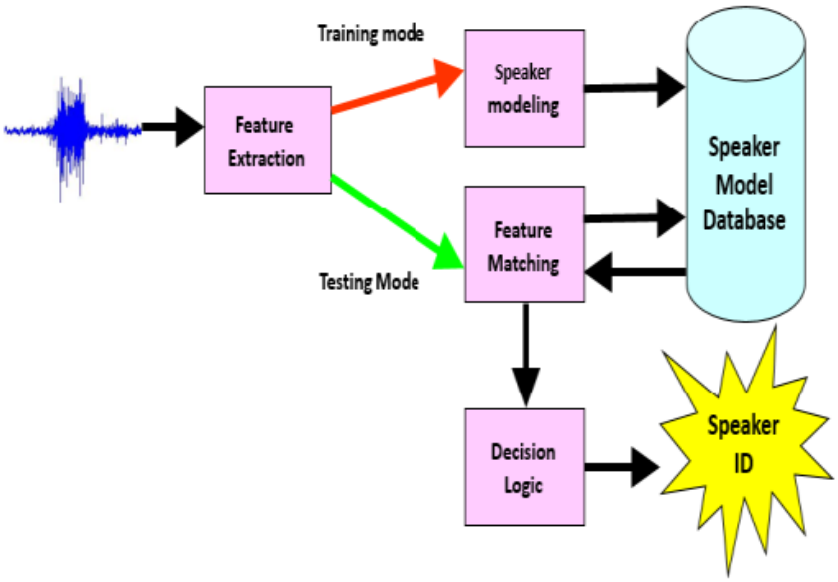

Figure 1. Speaker identification system Basic structure

2. Speaker Identification

The speaker identification systems usually require several operations and phases in which the voice signal must pass through to reach the result. There are two classes of these systems and our system classified as text-independent identification systems.

The first phase in such systems are to train the system environment on the new voices to form knowledge and create a reference models from these voices, where each speaker must provide samples of his speech so that the system can create the reference model for the speaker. It consists of two main parts. Part 1: consists of processing the speech sample provided by

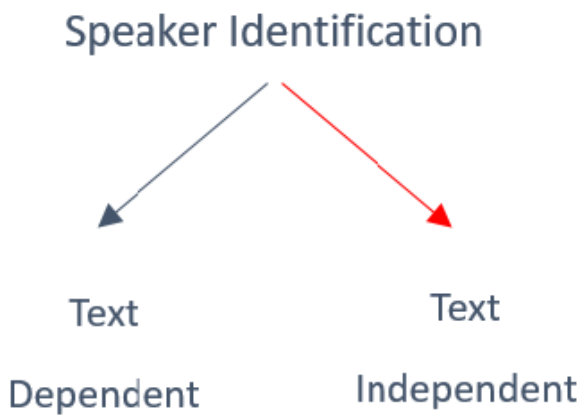

Figure 2. Speaker recognition types

the speaker to condense and summarize the properties of the acoustic tract, Part II: Collecting the data of each speaker together in one matrix that can be easily processing. The second phase is the test phase where it is reflect the structure of the training phase. First, analyze the input signal, and then compare the data stored in the Codebook. Difference between the input speech and the stored one used to make the decision in the system.

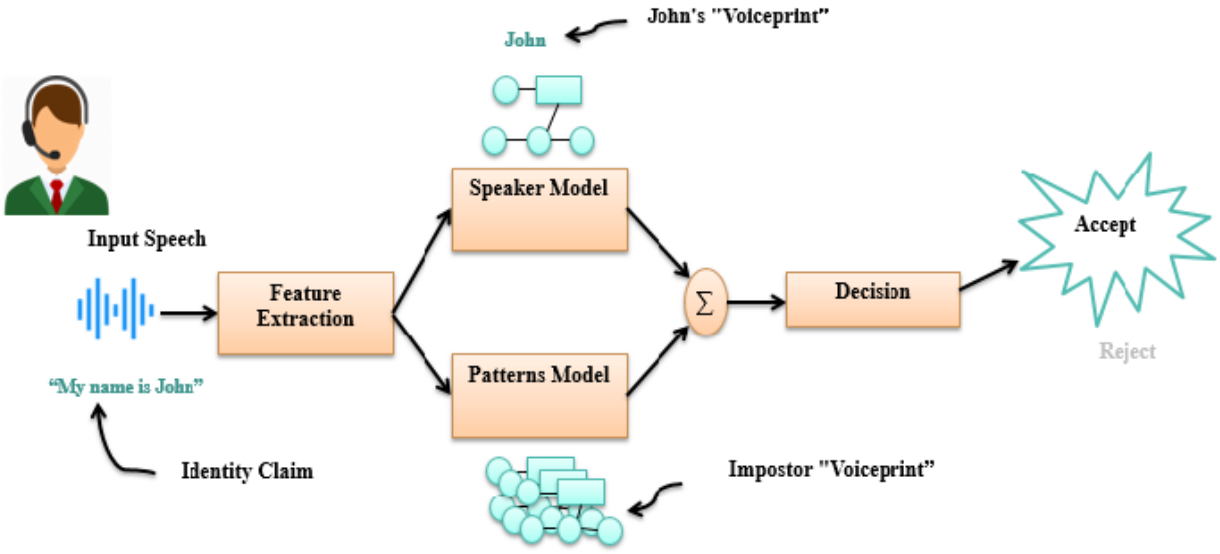

Figure 3. Speaker identification

The nature of the system determines the choice of technology used in the application. The system that we develop classified as text-independent speaker identification system, whose task is to determine who is speaking, regardless of what the speaker is saying.

Generally, all speaker recognition systems have two basic units: extract features and match these features. The first part is the extract features which extract a small amount of data from the voice signal that can be used later to represent each user. The matching feature includes the actual procedure for determining who is speaking by comparing features extracted from the input speech with forms saved or known by the system, which already explained.

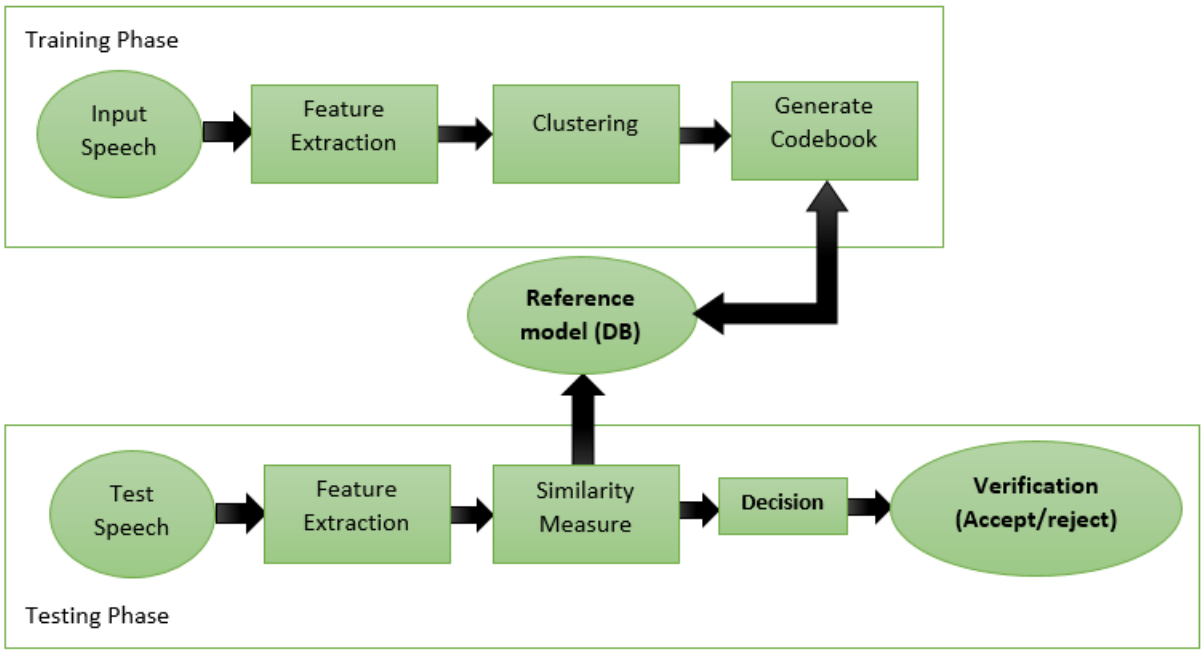

Figure 4. Speaker's identification block diagram

3. Recording & Signal Processing

3.1. Mean Correction

The first process that can be applied is to modify the signal values according to the average as the purpose of this process is to reduce the effect of any continuous frequency [10] produced by the recording devices. We select a certain threshold from the mean and subtract it from the signal values; this process does not change the shape of the signal but modifies the frequencies slightly as follows:

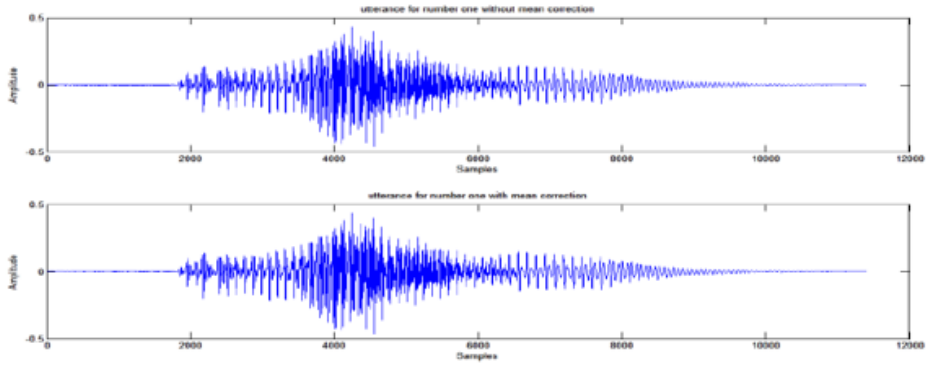

Figure 5. Number one with a threshold of 0.05

As shown in the figure, there is no clear difference between the two signals, the primary objective of this process is to try to mitigate the impact of the continuous frequency that produced by the recording devices.

3.2. Speech Boundary Detection

Often, moments of silence may pass before and after recording. These moments may affect to the quality of the sound sample in distinguishing the content of this sample. Therefore, these static samples removed from the signal and the samples contain

ing the operative sound information must be removed. Because of that, we have used the short-term energy measure and this method is one of the most widely used in edge detection algorithms because it give us the power to distinguish between sounds and silence, as this method relies on signal energy.

$$E_{log} = \sum_{i=1}^{n} \log\left(s(i)^2\right)$$

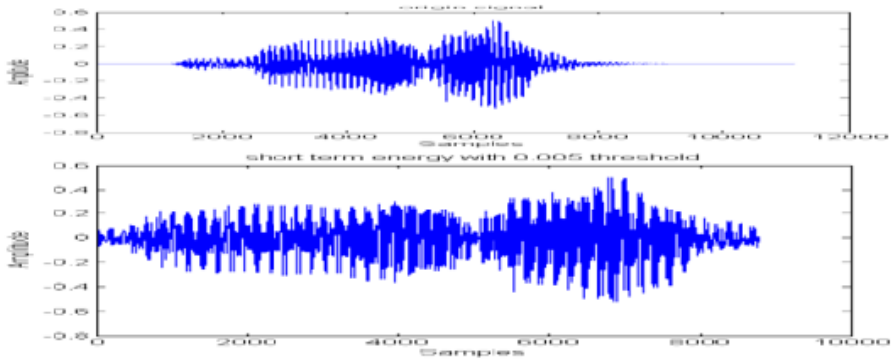

Figure 6. The signal after cutting the silence edges

4. Feature Extraction

At this stage, techniques that contribute to the feature extraction of audio signal that help to distinguish the content of the signal from others are applied. These features used to train a probabilistic system or a neural network to distinguish speech content. The first step in any system for speaker identification is the features extraction, which aim to identify speech signal components that are good for identifying language content and ignoring all other objects that carry information like background noise and emotion. The main point of understanding voices is that the audio forms including the tongue, teeth, etc. filter the sounds generated by the human, and this shape determines the resulting sound. If we can accurately determine the shape, this should give us a precise representation of the sound produced. The shape of the audio channel shown by the short spectrum energy (envelope) and the MFCC can represent this state accurately. The design of the system and the environment required from the speaker to pass through deferent operation to recognize the voice. Moreover, the first one is getting the input speech through the microphone from the user speech signal and then apply the process steps on that signal like pre-emphasis, framing, windowing, Mel-Cepstrum analysis (Feature Extraction) and vector quantization (Feature Matching) of the speech.

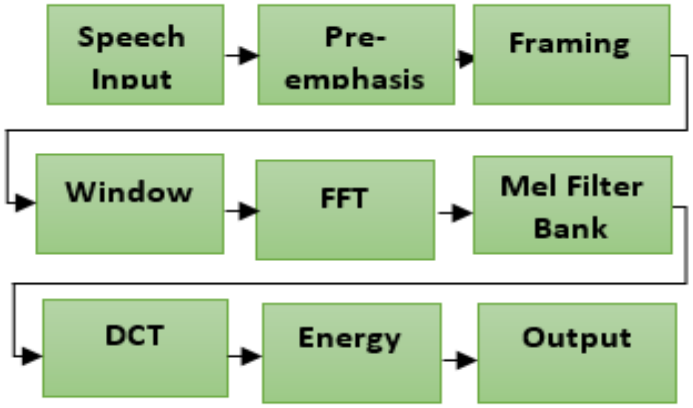

Figure 7. Block diagram of MFCC

Why MFCC's steps are necessary: The audio signal is constantly changing, to simplify things, assume that on short time scales the sound signal does not change much (when we say it does not change, we mean statistically constant, obviously the samples change continuously even on short time scales). That is why we have split the audio signal into clips (20-40ms); if the section is much shorter, we do not have enough samples to get the spectral estimate that is reliable, if the signal changes much longer throughout the section. The next step is to calculate the energy spectrum for each segment. From here, we have obtained a Mel filter-bank, and then we calculate the Mel filter-bank.

Why would we use a logarithm rather than a cube root: The logarithm allows us to use subtraction, the channel normalization [12], and to imitate human cognition of sound because experiments have shown that humans recognize sounds on a logarithmic scale [11].

4.1. Pre-emphases

Pre-emphases refers to focusing the filtering process on higher frequencies and its purpose is to balance the spectrum of the sounds so the signal sent to a high pass filter:

$$y(n) = x(n) - \alpha^* x(n-1)$$

Where $y(n)$ is the output signal and the value of α is usually between 0.9 and 1.0. The Z transform of this equation given by:

$$H(Z) = 1 - \alpha * z - 1$$

The goal of pre-emphasis is to raise the frequencies of high frequencies versus low frequency frequencies, to increase the predictability of the sounds represented by these frequencies [1, 2].

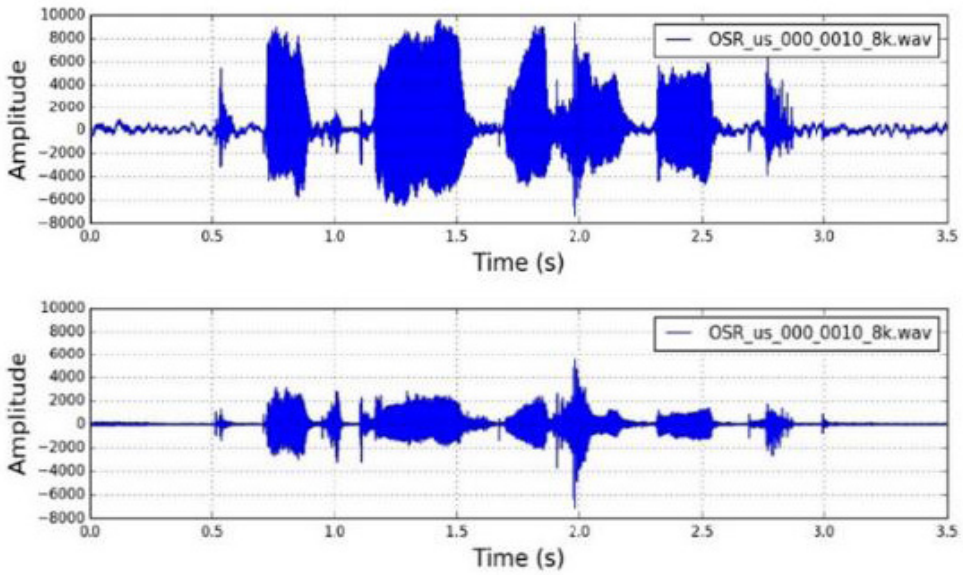

Figure 8. The signal before and after Pre-emphasis

4.2. Frame Blocking & Windowing

The speech signal is continuous over time or semi-static slowly. For stable audio properties, speech must examined within an enough short periods. Therefore, speech analysis should always be performed on short parts where the speech signal is assumed be static. Short-term spectral measurements usually performed over 20 milliseconds, and progress every 15 milliseconds. Increasing the time window every 15 milliseconds enables tracking the temporal characteristics of individual speech sounds, and the analysis window of 30 milliseconds is usually sufficient to provide good spectral analysis of these sounds, while short enough to solve important temporal characteristics [2]. If the sample rate is 16 kHz and the frame size is 320 sample, then the frame duration is 320/16000 = 0.02 = 20 *Ms*. if the overlap is 160 points, then the frame rate is 16000/ (320-160) = 100 frames per second. In general, Hamming windows used to surround the signal. This done to enhance the harmonics,

facilitate the edges and reduce the edge effect with the FFT calculation on the signal and to reduce the effect of discontinuity introduced by the framing process by attenuating the values of the samples at the beginning and end of each frame [3, 4].

$$Y(n) = x(n)W(n)$$

Hamming window that used for speaker recognition task defined as:

$$W(n) = 0.54 - 0.46\ cos\ (\frac{2\pi}{N-1})$$

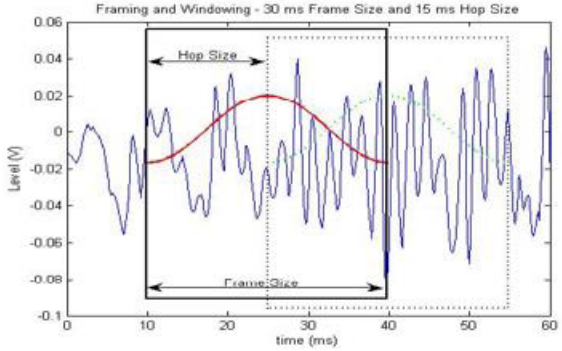

Figure 9. Framing & windowing

4.3. Fast Fourier Transform FFT

FFT is a process of converting time domain into frequency domain. To obtain the magnitude frequency response of each frame we perform FFT. By applying FFT, the output is a spectrum or periodogram. [5].

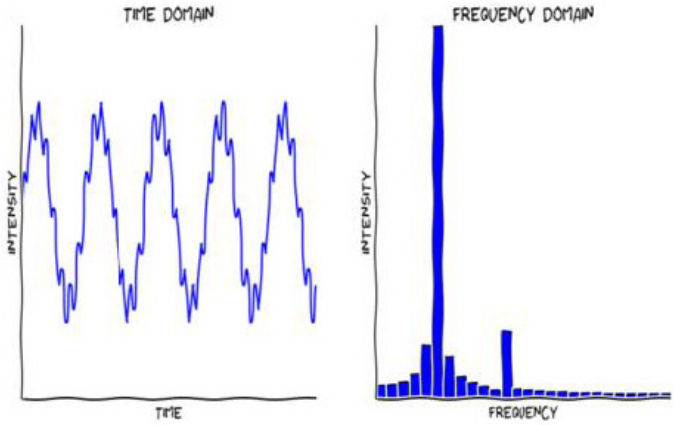

Figure 10. Fast Fourier Transform

4.4. Mel-Frequency Wrapping

It calculated by passing a Fourier signal that converted through a set of high pass filters known as a Mel-filter bank. The Mel is a unit of measurement based on the frequency of the human's ears. Does not correspond in linear scale to the physical frequency of the tone, and it seems that the human hearing system does not recognize linear vibrations. The Mel scale is approximately a linear spacing below 1 kHz and a logarithmic distance above 1 kHz and series of triangular band pass filters that designed to simulate the band pass filtering believed to occur in the audible system. Therefore, we can use the following approximate formula to compute the Mel for a given frequency in Hz [5].

$$Mel(f) = 2595 * log(1+ f/700)$$

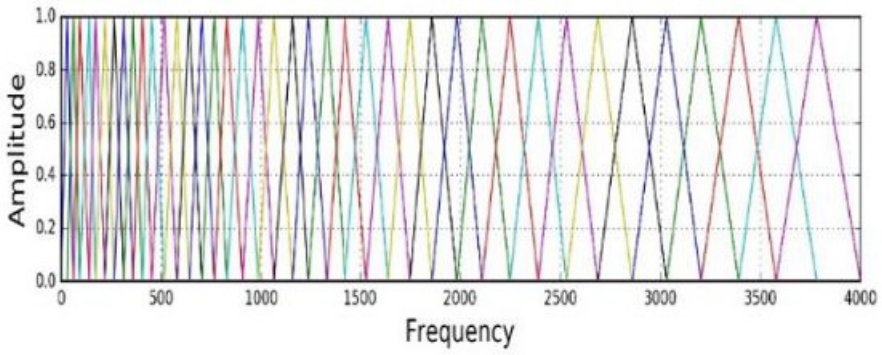

Figure 11. Mel-filter bank

4.5. Discrete Cosine Transform

Since the audio channel is soft, the energy levels at adjacent points tend to be interconnected where DCT applied to converted Mel frequency coefficients resulting in a set of Cepstral Coefficients. Before DCT calculation, the Mel spectrum typically represented on a logarithmic scale this result in a signal in the Cepstral domain with the frequency peak corresponding to signal vibration and a number of formulas representing low frequency peaks. Since most signal information represented by the first few MFCC transactions the system can be more accurate by extracting transactions that ignore or interrupt the higher-order DCT components. Finally, in this step, DCT applied to the output of the N triangular band pass filters to obtain L Mel-scale Cepstral Coefficients. The formula for DCT is:

$$C(n) = \sum Ek * \cos\left(n * (k - 0.5) * \frac{\pi}{40}\right)$$

Where $n = 0, 1, \ldots, N$

Where n is the number of triangular band pass filters, K is the number of Mel-scale Cepstral Coefficients. In this project, there are $n = 40$ and $K = 13$. Since we have performed *FFT, DCT* transforms the frequency domain into a time domain. The obtained features are similar to Cepstrum, thus it referred to as the Mel-scale Cepstral Coefficients, or MFCC.

5. Vector Quantization

At this stage, the features that extracted in the feature extraction phase, the values converted into a form that used as an input for the probabilistic model. Similar values grouped together and given a common value. This called clustering.

Until this point, we have obtained the characteristic features of each speech signal in our identification system, but the direct use of these features is not possible for a number of reasons:

1. In Pattern Recognition systems (which our system is a part of) we need to have specific data values that should mark the data point with a value.

2. Because of the large number of data where its values are close, these data can be Summarizing by a table representing an index of these data and its values.

For these reasons, data cannot be direct manipulate, as we have to convert them into the right shape to be suitable for a later processing. This conversion process called indexing. It is a process whereby each data point that represents the speech signal given a value that distinguishes it from other elements so that we use this value to denote this data point and this value becomes the input of subsequent algorithms.

The goal of pattern recognition is to classify a set of data objects into multiple categories or classes so that the objects in a

cluster are similar, but they are quite different from the objects in other clusters. Similarities evaluated based on the values that describe the objects and often involve distance scales. The latest techniques for feature matching used in speaker identification are Hidden Markov Modeling (HMM), Gaussian Mixture Model (GMM) and Vector Quantization (VQ). VQ is a process of identifying vectors from a large vector space to a set of number in the space. Each region called a cluster and represented by its center, which called Codewords. All Codewords called Codebook; VQ is the technique we used in this system because of the fast and high accuracy of the comparison. VQ is a lossy data compression method based on principle of blocks coding. It is a fixed-to-fixed length algorithm [9].

Vector Quantization is one of the most common techniques used in the field of indexing for audio samples, because of its multiple uses in the areas of voice compression and speech recognition, the use of this technique has increased since the start of the use of linear prediction technology LPC in the sixties of the last century. The idea of a Vector Quantization is to give the signal a unique value from a set of values so that two different signals do not have the same value, and the converged and similar signals have the same value.so the process of VQ is limit the repetition of similar objects with a single value represents these objects.

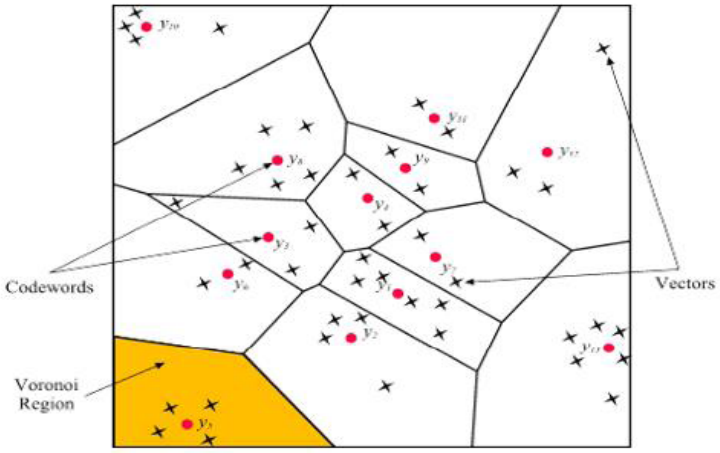

Figure 12. Two Dimensional VQ

The previous figure shows how the algorithm works by distributing data points to clusters so that data within the same cluster is as close as possible and as different as possible from data in other clusters. The following form shows two separated speakers the first speaker is shown as the green circles surrounding the black one, these are the Codevector and the centroid of that speaker (speaker 1), the second speaker is shown as the red triangles surrounding the black one, these are the Codevector and the centroid of that speaker (speaker 2). The distance between the centroid and the code vector called the vector quantization distortion.

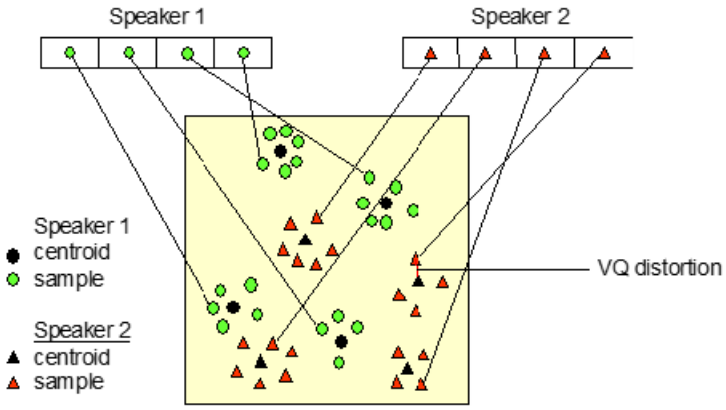

Figure 13. Codebook formation

In the training process from our algorithm, we generate a VQ codebook for each speaker clustering the acoustic vectors of each speaker in the database [7].

The distance of a vector closest to the Codeword of a Codebook is the named VQ-distortion. In the recognition phase, an input expression of an unknown voice is a "vector-quantized" using each trained Codebook and the total VQ distortion is calculated [8]. The speaker in the database with the smallest distortion is the one that will match with the incoming voice.

5.1. LBG Clustering Process

To build the index or Codebook, we need a clustering algorithm that assembles the converged vectors and finds the Codeword that represents them. The best and most widely used algorithms is the LBG algorithm, which divided Training data space into clusters to achieve the following two-optimization conditions:

1. Each training data (vector) must be close to a particular cluster and away from the rest of the clusters. In other words, the vector must belong to only one cluster so that the distance between this vector and the cluster center is smaller than all other distances or other cluster's centers.

2. The center of the cluster selected so that the distance between the center and all vectors is as low as possible. The average error (distance from center) for each cluster reduced. With the known that the center of the cluster is the average of the vector's value that belong to this cluster.

5.2. LBG steps

Initialization: At this point, the number of clusters K selected and initial vectors selected to represent these clusters, as these vectors selected randomly from training data.

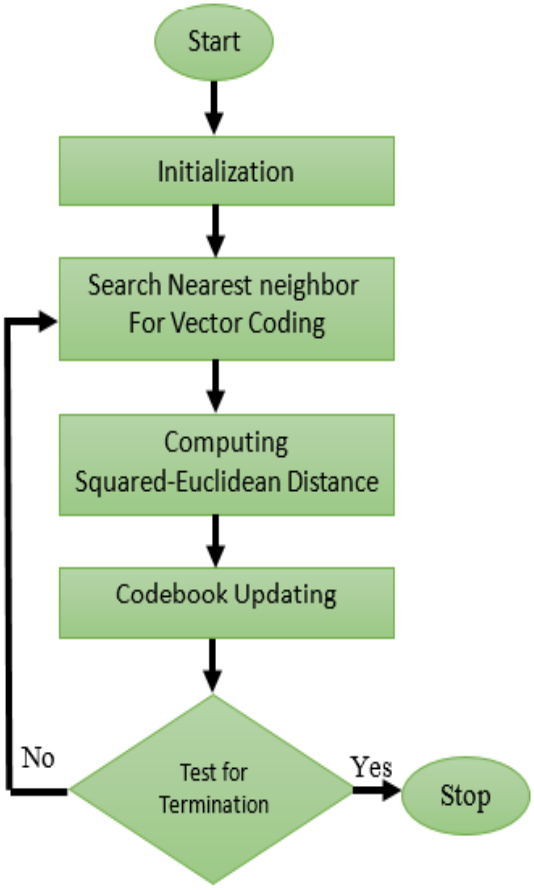

Figure 14. LBG process steps

Search nearest neighbor: Calculate the distance between each vector and the center to find the nearest center for that vector.

Compute Euclidean distance: We calculate the total of the distances for this iteration, which represents the amount of error in this iteration.

Update centroids: We update the centers where the average values of the closest vectors of this center, which resulted from step (2), are calculated. This average is the new center of the cluster.

Termination: Repeat the previous steps (except initialization) to get an error in step 3 below a specified limit previously selected or even reaching to the greatest number of sessions.

At the end of this algorithm, we have obtained the final clusters and the vectors they represent and achieve the best distance and the lowest possible error rate.

Using the VQ, we can generate the Codebook index, as after extracting the training vectors, these vectors become frames and each of these frames has its own attributes, so that we have a matrix in which each column represents a frame. In the beginning we must determine the size of the Codebook index, the number of clusters K in which the frames will be clustered, and then apply the clustering algorithm as shown previously. We configure the index by K vectors selected from the training matrix randomly. We look at each cluster in the training matrix for the nearest vectors from the index. This vector (located in the training matrix) gives the value indicating the location of the cluster that follows it. The distances obtained at this stage combined to represent the total error for this iteration. After the attribution, the average values of the vectors of the same value calculated from the index (the vectors that belong to the same cluster). This mean represents the center of the new cluster, the Codevectors to which we will compare the values. All previous steps repeated until the difference between the old center and the new center is equal to zero. At the end of the algorithm implementation, we have obtained the final positions that will convert the series of frames represented by an audio signal to a series of vectors representing the Codebook to that entered into the next step of matching.

6. Experiment Results and Analysis

As stated above, in this paper, we will experience the building and testing of a speaker recognition system. In order to implement such a system, the user must go through several steps which were described in details in previous sections, from defining the system to the user's voice, which analyzing the physical characteristics of each voice to identify the speaker.

6.1. Enrollment Phase or Training Phase

At this stage, our goal is to train the voice model (training the VQ model on MFCC extracted features) for all speech files in the train phase. After this training step, the system will be aware of the characteristics of each speaker's voice. Then, in the testing phase, the system will be able to determine the speaker's voice and identify the speakers.

6.2. Speech Signal Processing

At this point, the speech signals analyzed and converted to a series of MFCC features using the speech processing steps described earlier. After getting the speech signal, we cut it to overlay (30 MS) frames. The result is a matrix where each column in this matrix is a frame of N samples from the original speech signal. We then apply the steps of Windowing and FFT to convert the signal from the time domain to the frequency domain; this process used in many different applications and referred to Windowed Fourier Transform (WFT) or a short Fourier transform (STFT) Spectrum or Periodogram. The last step in speech processing is to convert the energy spectrum into Cepstrum Coefficients. Thus, we have obtained the matrix of Cepstrum Coefficients that leads to MFCC treatment.

6.3. Building the Model

The result of the last section is that we convert speech signals into features. In this section, we will apply the VQ-based pattern recognition techniques to construct reference models of these features in the training phase. To achieve this, we build the previously described Codebook that trains the VQ models. Comparisons performed by calculating the Euclidean distance between the input speech signal and the models stored in the database.

Euclidean distance between Codebook patterns for some users:

	Sp1	**Sp2**	**Sp3**	**Sp4**	**Sp5**
Sp1	**4.7141**	8.91	10.7859	10.3863	11.4722
Sp2	6.7652	**5.5148**	6.3454	7.4315	7.7702
Sp3	5.1283	5.5148	**3.3465**	4.0861	4.3574
Sp4	3.5836	3.5063	3.2954	**2.404**	3.3519
Sp5	1.404	1.4154	1.3641	1.2587	**1.206**

Table 1. Euclidean space distortion

The result of matching some users in the system:

	Sp1	**Sp2**	**Sp3**	**Sp4**	**Sp5**
Sp1	**Match**	No	No	No	No
Sp2	No	**Match**	No	No	No
Sp3	No	No	**Match**	No	No
Sp4	No	No	No	**Match**	No
Sp5	No	No	No	No	**Match**

Table 2. Matching results

The Codebook for some users that shown in the previous table which we have calculated the Euclidean distance for them.

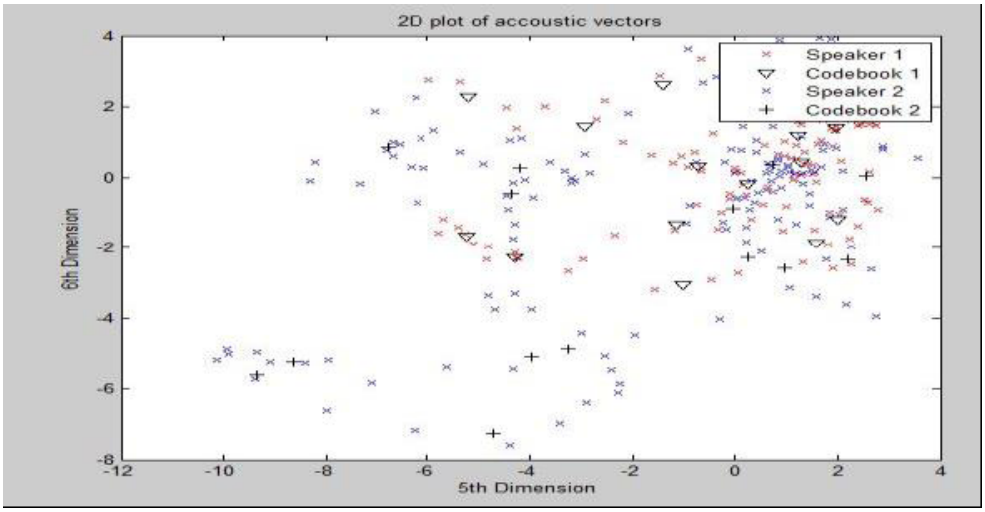

Figure 15. VQ codebook for some users

7. Conclusion

The purpose of this paper is to identify the speaker where a number of techniques have been used together to achieve the desired results from the system. The speaker's voice processed with a number of operations that discussed in detail in the previous sections. Mel Frequency Cepstral Coefficient and Vector Quantization used together to extract the features and matching them. Moreover, both gave good performance and accuracy results. MFCC used to analyze the voice and extract the tone, pitch and frequency features of that voice. VQ used to encode these features and matching the voices but to achieve a better performance and accuracy the training and testing sessions have to be repeat to update the speaker's Codebooks in the environment of the system. Thus, the more the system is used, the faster the recognition of speakers and the more accurate the system becomes.

References

[1] Rabiner, L. R., Schafer, R.W. (1978). *Digital Processing of Speech Signals*, Prentice-Hall, Englewood Cliffs, NJ.

[2] Rabiner, L. R., Juang, B. B. H. (1993). *Fundamentals of Speech Recognition*, Prentice Hall, 1993.

[3] Xiong, X. (2009). *Robust speech features and acoustic models for speech recognition*, PhD. Thesis, 194 p., Nanyang Technological University, Singapore, 2009.

[4] Mitra, S. K. (2011). *Digital signal processing: a computer-based approach*, vol. 1221: McGraw- Hill New York, 2011.

[5] Karpov, E. (2003). *Real Time Speaker Identification*, Master's thesis, Department of Computer Science, University of Joensuu, 2003.

[6] Arun Rajsekhar. G. (2000). Real time speaker recognition using MFCC and VQ, Department of electronics & communication engineering National Institute of Technology. 2000.

[7] Soong, F., Rosenberg, E., Juang, B., Rabine, L. (1987). *A Vector Quantization Approach to Speaker Recognition*, *ATT Technical Journal*, vol. 66, March/April 1987, p 14-26.

[8] Zhong-Xuan., Bo-Ling, Yuan., Yu, Xu Chong-Zh. (1999). *Binary Quantization of Feature Vector for Robust TextIndependent Speaker Identification in IEEE Transactions on Speech and Audio Processing*, 7 (1), January 1999. *IEEE*, New York, NY, U.S.A.

[9] Linde, Y., Buzo, A., Gray, R. (1980). An Algorithm for Vector Quantizer Design, *IEEE Transactions on Communications,* 28, p 84-95, no. 1.

[10] *Strang, Gilbert* (MayJune1994). *Wavelets. American Scientist.* 82(3): 250–255. JSTOR 29775194.

[11] Niemann, H. (2003). *Klassifikation von Mustern*, 2nd ed. Berlin, New York, Tokyo: Springer, 2003.

[12] Davis, S., Mermelstein, P. (1980) *Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences*. *In*: *IEEE Transactions on Acoustics, Speech and Signal Processing*, 28 (4), p 357-366.