Abstract—This paper presents an approach to speaker recognition using frequency spectral information with Mel frequency for the improvement of speech feature representation in a Vector Quantization codebook based recognition approach. The Mel frequency approach extracts the features of the speech signal to get the training and testing vectors. The VQ Codebook approach uses training vectors to form clusters and recognize accurately with the help of LBG algorithm.

Key words: Speaker Recognition, MFCC, Mel Frequencies, Vector Quantization.

I. INTRODUCTION

Speech is the most natural way of communicating. Unlike other forms of identification, such as passwords or keys, speech is the most non-intrusive as a biometric. A person’s voice cannot be stolen, forgotten or lost, therefore speaker recognition allows for a secure method of authenticating speakers.

In speaker recognition system, an unknown speaker is compared against a database of known speakers, and the best matching speaker is given as the identification result. This system is based on the speaker-specific information which is included in speech waves.

Earlier systems used the traditional approaches but in this paper, we intend to increase the efficiency and accuracy of the system by making use of a new approach which would include the fusion neural networks and clustering algorithms.

The general process of speaker identification involves two stages:

1) Training Mode
2) Recognition Mode

In the training mode, a new speaker (with known identity) is enrolled into the system’s database. In the recognition mode, an unknown speaker gives a speech input and the system makes a decision about the speaker’s identity.

II. BLOCK DIAGRAM

The process of real time speaker identification system consists of two main phases. During the first phase, speaker enrolment, speech samples are collected from the speakers, and they are used to train their models. The collection of enrolled models is also called a speaker database. In the second phase, identification phase, a test sample from an unknown speaker is compared against the speaker database. Both phases include the same first step, feature extraction, which is used to extract speaker dependent characteristics from speech. The main purpose of this step is to reduce the amount of test data while retaining speaker discriminative information. Then in the enrolment phase, these features are modelled and stored in the speaker database. This process is represented in Fig. 1.

In the identification step, the extracted features are compared against the models stored in the speaker database. Based on these comparisons the final decision about speaker identity is made. This process is represented in Fig. 2.

III. PRE-PROCESSING

Speech is recorded by sampling the input which results in a discrete time speech signal. Pre-processing is a technique used to make discrete time speech signal more amenable for the process that follows. The pre-processing techniques are used to enhance feature extraction. They include pre-emphasis, framing and windowing. Pre-emphasis is extensively explained below whereas frame blocking and windowing are explained in further parts.

A. Pre-emphasis

Pre-emphasis is a technique used in speech processing to enhance high frequencies of the signal. There are two main
factors driving the need for pre-emphasis. Firstly, the speech signal generally contains more speaker specific information in the higher frequencies than the lower frequencies. Secondly, pre-emphasis removes some of the glottal effects from the vocal tract parameters. For voiced sounds which have a steep roll-off in the high frequency region, the glottal source has an approximately −12dB/octave slope. However, when the acoustic energy radiates from the lips, this causes a roughly +6dB/octave boost to the spectrum. As a result, a speech signal when recorded with a microphone from a distance, has approximately −6dB /octave slope downward compared to the true spectrum. Therefore, by applying pre-emphasis, the spectrum is flattened, consisting of formats of similar heights. The spectrum of unvoiced sound is already flat therefore there is no reason to pre-emphasize them. This allows feature extraction to focus on all aspects of the speech signal.

Pre-emphasis is implemented as a first-order Finite Impulse Response (FIR) filter defined as:

$$H(z) = 1 - \alpha z^{-1}$$

(1.1)

Generally $\alpha$ is chosen to be between 0.9 and 0.95. We have used $\alpha=0.95$.

**B. Frame Blocking**

To prevent the occurrence of aliasing effect, frame blocking is used to convert the continuous speech signal into frames of desired sample length. In this step the continuous speech signal is blocked into frames of N samples, with adjacent frames being separated by M such that M < N. The first frame consists of the first N samples. The second frame begins M samples after the first frame, and overlaps it by N - M samples. Similarly, the third frame begins 2M samples after the first frame (or M samples after the second frame) and overlaps it by N - 2M samples. This process continues until all the speech is accounted for within one or more frames. Typical values for N and M are N = 256 (which is equivalent to ~ 30 mSec windowing and facilitate the fast radix-2 FFT) and M = 100. [1]

**C. Windowing**

The signal obtained after frame blocking has signal discontinuities at the beginning and end of each frame. To minimize these signal discontinuities blocking is used. In blocking, the spectral distortion is minimized by using a window to taper the signal to zero at the beginning and at the end of each frame. The different windows available for this process include rectangular window, triangular window, hanning window, hamming window, etc. If we define the window as $w(n)$, $0 \leq n \leq N-1$, where $N$ is the number of samples in each frame, then the result of windowing is the signal as in Equation 1.2

$$y(n) = x(n) \cdot w(n)$$

(1.2)

Typically the Hamming window is used for the windowing process, which has the form as in Equation 1.3:

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

(1.3)

**IV. FEATURE EXTRACTION**

The amount of data, generated during the speech production, is quite large while the essential characteristics of the speech process change relatively slowly and therefore, they require less data. According to these matters feature extraction is a process of reducing data while retaining speaker discriminative information. [2]

In order to create a speaker profile, the speech signal must be analyzed to produce some representation that can be used as a basis for such a model. In speech analysis this is known as feature extraction. Feature extraction allows for speaker specific characteristics to be derived from the speech signal, which are used to create a speaker model. The speaker model uses a distortion measure to determine features which are similar. This places importance on the features extracted, to accurately represent the speech signal. Feature extraction phase consists of transforming the speech signal in a set of feature vectors called parameters. [3]

The aim of this transformation is to obtain a new representation which is more compact, less redundant, and more suitable for statistical modeling and calculation of distances. Most of the speech parameterizations used in speaker recognition systems relies on cepstral representation of the speech signal. [4]

A wide range of possibilities exist for parametrically representing the speech signal for the speaker recognition task, such as Linear Prediction Coding (LPC), Mel-Frequency Cepstrum Coefficients (MFCC), and others. MFCC is perhaps the best known and most popular, and these will be used in this project.

**V. MEL-FREQUENCY CEPSTRAL COEFFICIENTS**

MFCC’s are based on the known variation of the human ear’s critical bandwidths with frequency; filters spaced linearly at low frequencies and logarithmically at high frequencies have been used to capture the phonetically important characteristics of speech. This is expressed in the mel-frequency scale, which is linear frequency spacing below 1000 Hz and a logarithmic spacing above 1000 Hz. [1]

![Fig. 3: Block diagram of MFCC processor](image)

A block diagram of the structure of an MFCC processor is given in Figure 2. The speech input is typically recorded at a sampling rate above 10000 Hz. This sampling frequency was chosen to minimize the effects of aliasing in the Analog-to-digital conversion. These sampled signals can capture all frequencies up to 5 kHz, which cover most energy of sounds that are generated by humans. As been discussed previously, the main purpose of the MFCC processor is to mimic the behaviour of the human ears. In addition, rather than the speech waveforms themselves, MFCC’s are shown to be less susceptible to mentioned variations. Figure 2 shows the block diagram of the MFCC processor.
A. Fast Fourier Transform

All this while the computations have been carried out in the time domain. But since for MFCC, we need the samples in the frequency domain, we use the Fast Fourier Transform method to convert each frame of N samples from time domain into the frequency domain. The FFT is a fast algorithm to implement the Discrete Fourier Transform (DFT) which is defined on the set of N samples \( \{x_n\} \) as in Equation 2.1:

\[
X_n = \sum_{k=0}^{N-1} x_k e^{-2\pi jkn/N} \quad 0 \leq n \leq N - 1
\] (2.1)

The spectrum obtained after the fast Fourier transform is used for mel-frequency warping.

B. Mel-Frequency Warping

MFCCs are typically computed by using a bank of triangular-shaped filters, with the centre frequency of the filter spaced linearly for frequencies less than 1000 Hz and logarithmically above 1000 Hz. The bandwidth of each filter is determined by the centre frequencies of the two adjacent filters and is dependent on the frequency range of the filter bank and number of filters chosen for design. But for the human auditory system it is estimated that the filters have a bandwidth that is related to the centre frequency of the filter. Further it has been shown that there is no evidence of two regions (linear and logarithmic) in the experimentally determined Mel frequency scale.

Recent studies on the effectiveness of different frequency regions of the speech spectrum for speaker recognition frequency-scale warping method provided better performance than standard Mel scale filter bank. [4]

The frequency-scale warping is implemented by using the bilinear transform method. A range of warping functions can be achieved by using the bilinear transform technique. It is extremely flexible and a wide range of warping functions can be obtained by suitably fixing a “warping factor” for the given sampling frequency. We use this framework to carry out an experimental study toward determining the optimal choice of frequency-scale warping for the speaker recognition task.

C. Cepstrum

After frequency warping, we convert the log Mel spectrum from frequency back into time. The result obtained after the Cepstrum operation is called the Mel Frequency Cepstrum Coefficients (MFCC). Since the Mel spectrum coefficients are real numbers, so are their logarithmic values. Therefore we can convert them back into the time domain by using simple Discrete Cosine Transform. Therefore if we denote those mel power spectrum coefficients that are the result of the last step are \( S_k \), \( k=1, 2, \ldots, K \), we can calculate the MFCC’s, as in Equation 2.4:

\[
\xi_n = \sum_{k=1}^{K} (\log S_k) \cos \left[ \pi \left( k - \frac{1}{2} \right) \frac{n}{K} \right]
\] (2.4)

The cepstral representation of the speech spectrum provides a good representation of the local spectral properties of the signal for the given frame analysis. [3]

VI. FEATURE MATCHING AND SPEAKER RECOGNITION

The state-of-the-art in feature matching techniques used in speaker recognition includes Dynamic Time Warping (DTW), Hidden Markov Modelling (HMM), and Vector Quantization (VQ). In this project, the VQ approach will be used, due to ease of implementation and high accuracy. VQ is a process of mapping vectors from a large vector space to a finite number of regions in that space. Each region is called a cluster and can be represented by its centre called a codeword. The collection of all codeword is called a codebook.
Figure 5 shows a conceptual diagram to illustrate this recognition process. In the figure, only two speakers and two dimensions of the acoustic space are shown. The training vectors obtained are used to build a speech specific VQ codebook for the speaker dictionary. The LBG algorithm is used for clustering a set of L training vectors into a set of M codebook vectors. This algorithm designs an M-vector codebook in stages. [6] It starts first by designing a 1-vector codebook, then uses a splitting technique on the codeword to initialize the search for a 2-vector codebook, and continues the splitting process until the desired M-vector codebook is obtained. This algorithm is based on the nearest-neighbour search procedure which assigns each training vector to a cluster associated with the closest codeword. The centroid of the clusters obtained and the distortion calculates the sum of the distances of all training vectors so as to decide whether the procedure has converged. The point of convergence is used to decide the result of the speaker recognition. [7]

VII. EXPERIMENTAL RESULTS

The database consists of 20 distinct speakers including both male and female speakers. It also contains 50 sound files used for training and testing the Speaker Recognition module. New sound files are recorded in real time for testing the Continuous Speech Recognition module in clean and noisy environments for both multi-speaker and speaker-independent modes. Recognition rate of the trained VQ codebook model is defined as follows:

\[ RR = \frac{N_{total}}{N_{correct}} \times 100 \]  

(3.1)

In the equation 3.1, RR is the recognition rate, \( N_{correct} \) is the number of correct recognition of testing speech samples per digit, and \( N_{total} \) is the total number of testing speech samples per digit [8].

<table>
<thead>
<tr>
<th>Environment</th>
<th>Number of Samples Tested</th>
<th>Number of Samples Recognized Accurately</th>
<th>Recognition Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean</td>
<td>20</td>
<td>19</td>
<td>95</td>
</tr>
<tr>
<td>Noisy</td>
<td>20</td>
<td>16</td>
<td>80</td>
</tr>
</tbody>
</table>

Table 1: Overall Recognition Rate of Proposed Speaker Recognition System

<table>
<thead>
<tr>
<th>Feature Extraction Technique</th>
<th>Feature recognition technique</th>
<th>Recognition rate (%)</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPC</td>
<td>VQ and HMM</td>
<td>62% to 96%</td>
<td>[9]</td>
</tr>
<tr>
<td>MFCC</td>
<td>VQ</td>
<td>70% to 85%</td>
<td>[10]</td>
</tr>
<tr>
<td>MFCC</td>
<td>VQ</td>
<td>88.88%</td>
<td>[2]</td>
</tr>
<tr>
<td>MFCC</td>
<td>VQ</td>
<td>57% to 100%</td>
<td>[11]</td>
</tr>
</tbody>
</table>

Table 2: Comparative Performance of Various Speaker Recognition Researches

VIII. CONCLUSION

The performance measured on the basis of accuracy, time taken to compute the feature recognition, it was observed that the Speaker Recognition System performs well in both clean and noisy environment with both multi-speaker and speaker independent modes. The entire research process was carried out using MATLAB R13 on an Intel i5 powered machine. It is noticed that the recognition results on the clean environment are much higher than the recognition results of the noisy environment.

ACKNOWLEDGMENT

The authors would like to thank the college authorities for providing infrastructure to carry out experimental and research work required. The authors would also like to thank Ms Mani Roja, Associate Professor of the Electronics and Telecommunication Department of Thadomal Shahani Engineering College for reviewing the paper and guiding the authors with their valuable feedbacks.

REFERENCES

[1] Santosh Gaikwad “Feature Extraction Using Fusion MFCC For Continuous Marathi Speech Recognition”. 
[3] Ibrahim Patel, Dr.Y.Srinivasa Rao “Speech Recognition using Hidden Markov Model with MFCC-Sub band technique” 