

Feature Extraction using MFCC and Classification using GMM

Suchitha T R¹ Bindu A Thomas²

¹Student ²Assistant Professor & HOD

^{1,2}Department of Electronics and Communication Engineering

^{1,2}VVIET- Mysuru

Abstract— Speech processing is emerged as one of the important application area of digital signal processing. Various fields for research in speech processing are speech recognition, speaker recognition, speech synthesis, speech coding etc. The objective of automatic speaker recognition is to extract, characterize and recognize the information about speaker identity. Feature extraction is the first step for speaker recognition. Many algorithms are suggested/developed by the researchers for feature extraction and for feature matching. In this work, the Mel Frequency Cepstrum Coefficient (MFCC) feature has been used for designing a text dependent/independent speaker identification system. The individual Gaussian component of Gaussian Mixture Model (GMM) represents vocal tract configurations that are effective for speaker identification. Gaussian Mixture Modeling(GMM) algorithms are used for generating template and feature matching purpose.

Key words: Feature Extraction, Feature Matching, MFCC, GMM

I. INTRODUCTION

The speech spoken by humans contains a lot of information. The main purpose of speech signal is to recognize the words spoken by humans. But speech signal also contains information about the identity of the speaker. This is the area of Speaker Recognition. While speech recognition is concerned with what is spoken by the speaker, Speaker Recognition mainly deals with extracting the identity of the speaker. It is a very helpful utility in areas where there is a need of remote verification based on biometrics and also where there is a need of inexpensive biometric verification system. Depending upon the application, the general area of Speaker Recognition is divided into two tasks: verification and identification. In verification, the goal is to determine from a voice sample if a person is who he or she claims. In speaker identification, the goal is to determine which one of a group of known voices best matches the input voice sample. Furthermore, in either task the speech can be constrained to be a known phrase (text-dependent) or totally unconstrained (text-independent). Success in both tasks depends on extracting and modeling the speaker-dependent characteristics of the speech signal which can effectively distinguish one talker from another.

Speaker recognition systems contain two main modules: feature extraction and feature matching [2]. Feature extraction is the process of extracting a small amount of data from the voice signal that can be later used to represent each speaker. Feature matching involves the actual procedure to identify the unknown speaker by comparing extracted feature from his/her voice input with the ones from a set of known speakers.

The speaker recognition systems are presented in two phases – training phase and testing phase. In the training phase, each registered speaker has to provide samples of their speech so that the system can build or train a reference

model for that speaker. In the testing phase, the input speech is matched with stored reference models and a recognition decision is made.

Speaker recognition is a difficult task. The principle source of variance is the speaker himself/herself. Speech signals in training and testing sessions can be greatly different due to many facts such as people voice changes with time, health conditions, speaking rates, and so on. There are also other factors, beyond speaker variability that present a challenge to speaker recognition technology [4].

II. PROPOSED METHODOLOGY

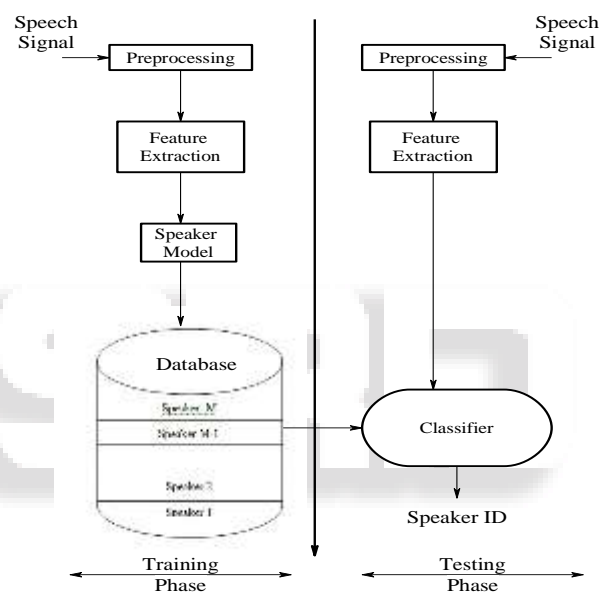


Fig 1: Block diagram of proposed methodology

A. Preprocessing

The objective in the preprocessing is to modify the speech signal, so that it will be more suitable for the feature extraction analysis. The preprocessing consists of de-noising, pre-emphasis and voice activation detection.

B. Feature Extraction

In Automatic Speaker Recognition Feature extraction is the process of retaining useful information of the signal while discarding redundant and unwanted information or we can say this process involves analysis of speech signal. However, in practice, while removing the unwanted information, one may also lose some useful information in the process. Feature extraction may also involve transforming the signal into a form appropriate for the models used for classification.

C. Feature Matching

The features of the speech signal are in the form of N dimensional feature vector. For a segmented signal that is divided into M segments, M vectors are determined

producing the $M \times N$ feature matrix. The $M \times N$ matrix is created by extracting features from the utterances of the speaker for selected words or sentences during the training phase. After extraction of the feature vectors from the speech signal, matching of the templates is required to be carried out for speaker recognition. This process could either be manual (comparison of spectrograms visually) or automatic. In automatic matching of templates, speaker models are constructed from the extracted features. There after a speaker is authenticated by comparison of the incoming speech signal with the stored model of the claimed user. The speaker models are of two types: template models and stochastic models.

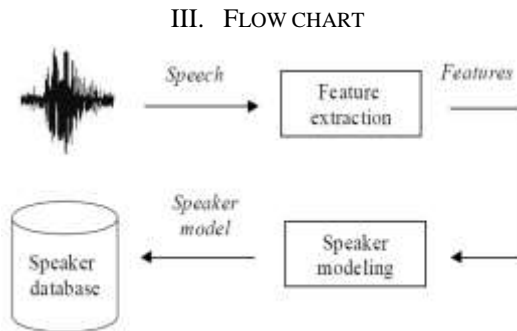


Fig 2: Flowchart for training phase

The above Fig. 2 shows the flow chart of training phase. During the period of training, 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. First step of this phase is feature extraction, which is used to extract speaker dependent characteristics from speech. The main purpose of this step is to reduce the amount of data while retaining speaker discriminative data. These features are modeled and stored in database.

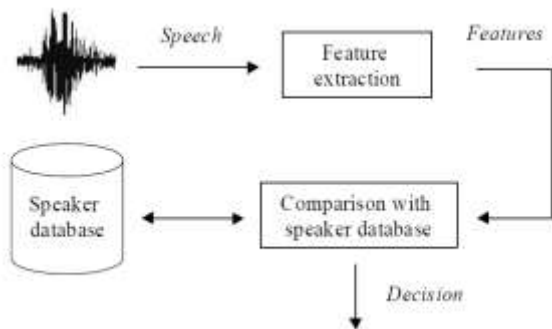


Fig. 3: Flowchart for testing phase

The above Fig.3 shows the steps involved in testing phase. During the period of testing phase, the extracted features are compared against the speaker voiceprint stored in speaker database. Based on this comparison the final decision about speaker identity is done.

IV. FEATURE EXTRACTION

MFCC is based on human hearing perceptions which cannot perceive frequencies over 1KHz. In other words, MFCC is based on known variation of the human ear's critical bandwidth with frequency. MFCC has two types of filter which are spaced linearly at low frequency below 1000 Hz and logarithmic spacing above 1000Hz. A subjective pitch is present on Mel Frequency Scale to capture important

characteristic of phonetic in speech. A block diagram of the structure of an MFCC processor is given in Fig3.

Steps involved in MFCC technique
The below Fig.4 illustrates the steps involved in calculating the MFCC.

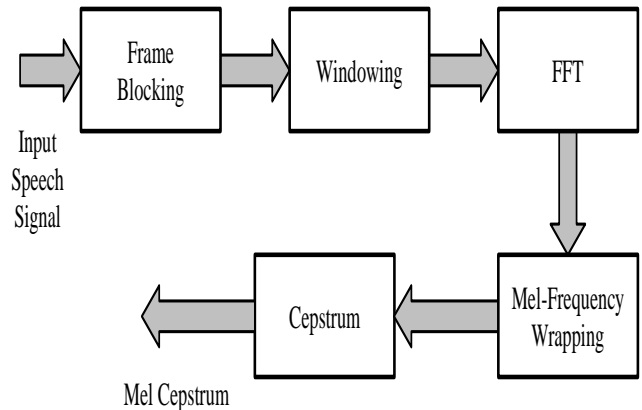


Fig 4: Steps to compute MFCC

A. Framing

In this step the continuous speech signal is blocked into frames of N samples, with adjacent frames being separated by M ($M < N$). 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.. The reason for this overlapping is that on each individual frame we will also be applying a hamming window which will get rid of some of the information at the beginning and end of each frame. Overlapping will then reincorporate this information back into our extracted features Typical values for N and M are $N = 256$ (which is equivalent to ~ 30 msec windowing) and $M = 100$.

B. Windowing

Windowing is done to provide spectral smoothing [12]. It is done on each individual frame so as to taper the signal to zero at the beginning and at the end of frame. Windowing is also essential for capturing dynamic characteristics of vocal tract system in speech production mechanism [13]. The Hamming window is used because it has a wide main lobe and small side lobes, making it a smooth low pass filter with less leakage [14].

Hamming window $w(n)$ has the form

$$w(n) = 0.54 - 0.46 \cos 2\pi \frac{n}{N} \quad 0 < n < N - 1$$

where N represents the width, in samples, of a discrete-time window function. Typically it is an integer power-of-2, such as $210 = 1024$. n is an integer, with values $0 \leq n \leq N-1$.

Hamming window is also called the raised cosine window. In a window function there is a zero valued outside of some chosen interval. The windowing is done to avoid problems due to truncation of the signal. Window function has some other applications such as spectral analysis, filter design, and audio data compression. A Hamming window is applied to each window in order to decrease the spectral distortion created by the overlap, and minimize errors

produced by FFT. The Hamming window improves the sharpness of harmonics and removes discontinuities on the edges. Hamming window with values of $N = 128, 256$ and 512 and $M = 50, 100$ and 200 , the combination of $N = 256$ and $M = 100$ gives the best performance. N is equal to the size of the window and M the overlap. The Hamming window improves the sharpness of harmonics and removes discontinuities on the edges.

C. FFT (Fast Fourier Transform)

FFT is used for doing conversion from the spatial domain to the frequency domain. Each frame having N_m samples are converted into frequency domain. Fourier transformation is a fast algorithm to apply Discrete Fourier Transform (DFT), on the given set of N_m samples shown below:

$$D_k = \sum_{m=0}^{N_m-1} D_m e^{-j2\pi km / N_m}$$

Where $k = 0, 1, 2, \dots, N_m - 1$

Basically the definition for FFT and DFT is same, which means that the output for the transformation will be the same; however they differ in their computational complexity. In case of DFT, each frame with $N-M$ samples directly will be used as a sequence for Fourier transformation. On another, in case of FFT this frame will be divided into small DFT's and then computation will be done on this divided small DFT's as individual sequence thus the computation will be more fast and easy. Thus it is in digital processing or other area instead of directly using DFT, FFT is used for applying DFT. Commonly, D_k are the combination of real and imaginary numbers thus it represents the complex numbers but, merely absolute values (frequency magnitudes) are considered to carry out further process. The obtained sequence can be interpreted as positive frequencies $0 \leq f < F_s / 2$ correspond to values $0 \leq m \leq N_m / 2 - 1$, while negative frequencies $-F_s / 2 < f < 0$ correspond to values $N_m / 2 + 1 \leq m \leq N_m - 1$, F_s is the sampling frequency. By calculating DFT we can obtain the magnitude spectrum.

D. Mel Frequency Wrapping

The frequency range in the FFT spectrum is very wide, so much data to process. So, we must use a filter bank in the Mel scale. The power spectrum of the speech that is obtained is to be integrated within overlapping critical band filter responses. Human perception of the frequency contents of sounds for speech signal does not follow a linear scale. Hence we use a frequency scale called Mel scale which is based on pitch perception and is used in the filter bank for the Mel cepstral approach.

The Mel scale is mainly based on the study of observing the pitch or frequency perceived by the human. The scale is divided into the units Mel. In this test the listener or test person started out hearing a frequency of 1000 Hz, and labelled it 1000 Mel for reference. Then the listeners were asked to change the frequency till it reaches to the frequency twice the reference frequency. Then this frequency labelled 2000 Mel. The same procedure repeated for the half the frequency, then this frequency labelled as 500 Mel, and so on.

This scale has rough linear frequency spacing below 1000Hz and a logarithmic spacing above 1000Hz. The speech consists of tones with different frequencies. For

each tone with an actual Frequency, f , measured in Hz, a subjective pitch is measured on the 'Mel' scale. As a reference point, the pitch of a 1 kHz tone, 40dB above the perceptual hearing threshold, is defined as 1000 Mel's. Therefore we can use the following formula to compute the Mels for a given frequency f in Hz.

$$\text{Mel}(f) = 2595 * \log_{10}(1 + f/700)$$

Where f is the actual frequency and $\text{Mel}(f)$ is the perceived one.

One approach to simulating the subjective spectrum is to use a filter bank, one filter for each desired Mel frequency Component. The filter bank has a triangular band pass frequency response, and the spacing as well as the bandwidth is determined by a constant Mel-frequency interval. According to the Mel frequency the width of the triangular filters varies and so the log total energy in a critical band around the centre frequency is included.

The Mel frequency warping when calculating MFCCs is accomplished by the use of a triangular Mel spaced filter bank. It consists of several triangular shaped and Mel spaced filters, and their outputs are described by

$$y(i) = \sum_{j=1}^N S_j H_{ij}$$

Where S_j is the N -point magnitude spectrum and H_{ij} the sampled magnitude response of an M -channel filter bank. The Mel frequency filter bank is applied in the frequency domain.

E. Cepstrum

In the final step, the log Mel spectrum has to be converted back to time. The result is called the Mel frequency Cepstrum coefficients (MFCCs). The cepstral representation of the speech spectrum provides a good representation of the local spectral properties of the signal for the given frame analysis. Because the Mel spectrum coefficients are real numbers (and so are their logarithms), they may be converted to the time domain using the Discrete Cosine Transform (DCT). Doing this we got the Mel Frequency Cepstrum Coefficients, we called the set of coefficients acoustic vectors, consequently each input expression is transformed into a sequence of acoustic vectors. Discrete Cosine Transform (DCT) removes the correlation between the output values of the filter bank and collects features of parameters. Since we have performed FFT, DCT transforms the frequency domain into a time-like domain called frequency domain. The obtained features are similar to cepstrum, thus it is referred to as the mel-scale cepstral coefficients, or MFCC.

V. FEATURE MATCHING USING GMM

A Gaussian Mixture Model (GMM) is a parametric probability density function represented as a weighted sum of Gaussian component densities. GMMs are commonly used as a parametric model of the probability distribution of continuous measurements or features in a biometric system, such as vocal-tract related spectral features in a speaker recognition system. GMM parameters are estimated from training data using the iterative Expectation-Maximization (EM) algorithm. The use of Gaussian mixture density for speaker identification is motivated by two facts [15]. They are:-

- 1) Individual Gaussian classes are interpreted to represents set of acoustic classes. These acoustic classes represent vocal tract information.
- 2) Gaussian mixture density provides smooth approximation to distribution of feature vectors in multi-dimensional feature space [15].

Speech production is not deterministic. A particular sound is not produced by a speaker with exactly the same vocal tract shape, glottal flow, due to context, coarticulation, anatomical and fluid dynamical variations. One way to represent this variability is probabilistically through multidimensional Gaussian probability density function [20]. The use of GMMs for speaker recognition is described in [18]. A Gaussian probability density function is state dependent. A different Gaussian pdf is assigned for each acoustic class. The Gaussian probability density function of a feature vector for i th state is given by

$$b_i(\vec{x}) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma_i|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2} (\vec{x} - \vec{\mu}_i) \Sigma_i^{-1} (\vec{x} - \vec{\mu}_i)\right\}$$

where μ_i =mean vector,
 Σ_i =covariance matrix and
 D =dimension of the vector.

The probability of feature vector in any one of M acoustic class for a particular speaker model λ is represented by the union or mixture of different Gaussian pdf. This is represented as

$$p\left(\frac{\vec{x}}{\lambda}\right) = \sum_{i=1}^M p_i b_i(\vec{x})$$

Where \vec{x} is a D -dimensional random vector, $b_i\left(\frac{\vec{x}}{\lambda}\right)$, $i=1, \dots, M$ are the component densities and p_i , $i=1, \dots, M$ are the mixture weights.

The complete Gaussian mixture density is parameterized by the mean vectors, covariance matrices and mixture weights from all component densities. The parameters are collectively represented by the notation $\lambda_i = \left\{ \vec{\mu}_i, \Sigma_i, p_i \right\}$, $i=1, \dots, M$. For speaker identification, each speaker is represented by a GMM and is referred by his/her model λ .

A. Maximum Likelihood Parameter Estimation

The aim of ML estimation is to find the model parameters which maximize the likelihood of GMM. For a sequence of T training vectors the GMM likelihood can be written as

$$p\left(\frac{X}{\lambda}\right) = \prod_{t=1}^T p\left(\frac{\vec{x}_t}{\lambda}\right)$$

This expression is a nonlinear function of the parameters λ and so direct maximization is not possible. The ML parameter estimate is obtained iteratively using Expectation Maximization algorithm.

B. Expectation Maximization Algorithm

The most popular algorithm for GMM parameters estimation is the EM algorithm. This algorithm allows iterative optimization of the mixture parameters, under non decreasing likelihood requirement. The EM algorithm begins with an initial model λ , to estimate a new model λ' . The new model then becomes the initial model and the process is repeated till convergence. The performance of this

algorithm depends on its initialization due to its tendency to converge to local extreme. A proper initialization must be done for model parameter. On each EM iteration mixture weight, mean and variance are calculated using eqn. below equations:

Mixture weight

$$\bar{p}_i = \frac{1}{T} \sum_{t=1}^T p(i/\vec{x}_t, \lambda)$$

Mean

$$\bar{\mu}_i = \frac{\sum_{t=1}^T p(i/\vec{x}_t, \lambda) \vec{x}_t}{\sum_{t=1}^T p(i/\vec{x}_t, \lambda)}$$

Variance

$$\bar{\sigma}_i^2 = \frac{\sum_{t=1}^T p(i/\vec{x}_t, \lambda) \vec{x}_t^2}{\sum_{t=1}^T p(i/\vec{x}_t, \lambda)} - \bar{\mu}_i^2$$

The a posteriori probability for acoustic class i is given by

$$p(i/\vec{x}_t, \lambda) = \frac{p_i b_i(\vec{x}_t)}{\sum_{k=1}^M p_k b_k(\vec{x}_t)}$$

VI. SIMULATION RESULTS

In order to evaluate the performance of the proposed Speaker Recognition system, speaker's voice samples are recorded for 10 seconds. From this recorded voice samples feature vectors are extracted using the Mel Frequency Cepstral Coefficients (MFCC). Finally, the extracted feature vectors are classified using the Gaussian Mixture Model (GMM) to calculate mean, variance and weights.

The speaker recognition experiment is conducted on speech database created with a collection of 30 speakers. MATLAB R2013a platform has been used experimentation. Each speaker voice samples are recorded for 10 seconds. The voice samples are recorded at a sampling rate of 8 KHz. The speaker voice is recorded using the dynamic stereo headphone. This headphone specifications consists 40mm loudhailer, 32 Ohm impedance, 105db/mw sensitivity, 1000mv power export, and 20 Hz to 20 KHz frequency response and 9.7/6.7-58db + or - 2 microphone. Database divided into two parts, those are training database and testing database. Each speaker voice is recorded two times. One is for creating training database and another is for creating testing database. Database consists of ten members of 0 to 11 years old children voice samples, ten members of 15 to 25 years old middle agers voice samples and 30 to 60 years old elders' voice samples. The features of voice samples are extracted using the MFCC and Wavelet SBC. 128 window size is choose for this method. The Mel-frequency cepstral coefficient and wavelet SBC coefficients of each signal is calculated and stored in database. Compare to MFCC, the wavelet SBC takes more time for extracting the features from speaker utterance. In this proposed framework Mel frequency cepstral coefficients gives better performance result compare to wavelet sub band coding. Classification of the speaker is done by Gaussian mixture stochastic model which use the expectation maximum algorithm.

The results which are produced during the execution of the speaker recognition system are shown in below snapshots.

A. Record the Speaker's voice



Fig 6: Sound Editor

The above snapshot shows the initial stage that is recording the speaker voice for the duration of 10 seconds with natural occurring noise like fan sound or sound of rain and so on. The input speech signal is recorded at a sampling rate of 8000Hz.

B. Input Voice Signal

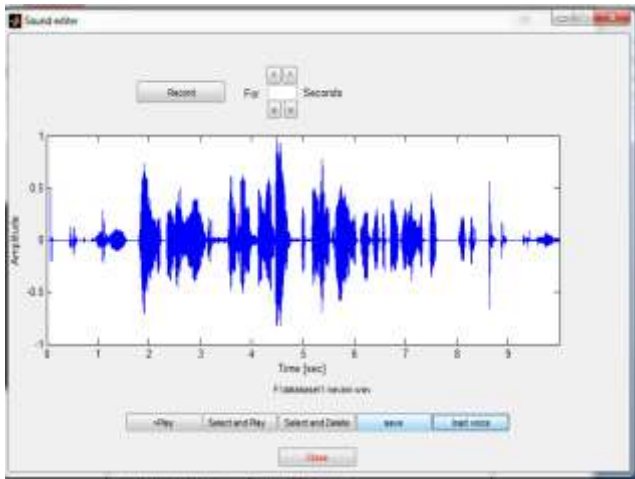


Fig. 6: Input voice signal

The above snapshot shows the one of the input voice signal which is stored in database. The recorded speech is saved with an extension of .WAV format. The voice signal is of length 10 seconds.

C. Find the Speaker

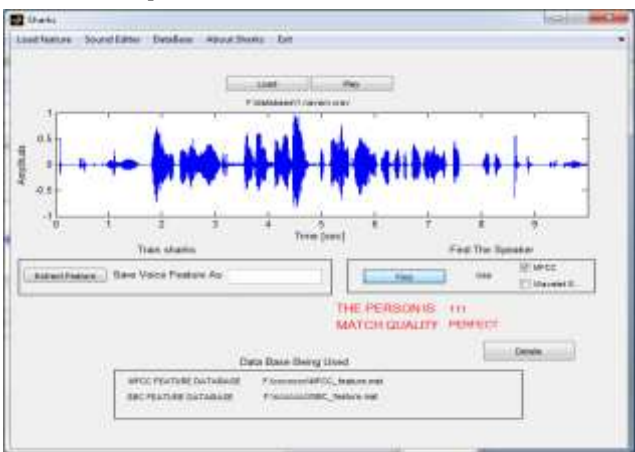


Fig 7: Find Speaker

The above snapshot shows the final step i.e. finding the speaker using Mel Frequency Cepstral Coefficients. In GUI window its shows the person id and matching quality. While recording the speaker's voice samples if background noise is not present then the match quality will be almost perfect.

D. Mean, Variance and Weights

The below tables indicate the mean, variance and weights which are calculated for the feature vectors, extracted using the MFCC of one speaker. Each column refers to a feature vector. The elements of each column are corresponding MFCCs. This mean, variance and weights are calculated using the Gaussian Mixture Model. In this work 12 Gaussian mixture components are used. Similarly, mean, variance and weights are calculated for all the speaker's voice samples stored in database.

Variables - fea[1, 1]												
fea × fea[1, 1] × fea[1, 2] × fea[1, 3] ×												
fea[1, 1] <8x12 double>												
	1	2	3	4	5	6	7	8	9	10	11	12
1	-3.5921	-3.4323	-3.2594	-2.9327	-2.5999	0.1336	2.1495	-1.3785	0.6450	-1.7165	-4.8589	-1.7350
2	-3.2129	-3.4301	-2.0778	-2.8157	-2.2907	0.0925	1.6443	-0.7293	0.9322	-0.3036	-2.2071	-2.2236
3	-1.6705	-1.5983	-0.1768	-1.8269	-1.4614	0.0678	1.2429	-0.4160	-0.1170	1.6736	-1.2982	-2.0727
4	-0.6864	0.4393	-1.6993	-0.6830	-0.9186	0.0802	0.8229	0.1485	-1.2323	-0.3349	-0.5040	-1.8030
5	-0.7647	0.5542	-1.6428	-0.2909	-0.9268	0.0539	0.4833	0.1152	-0.5004	-1.1007	-0.5694	-0.8611
6	-1.1356	-0.5245	-0.2337	-0.5319	-0.9450	0.0675	0.2004	-0.2226	0.8501	0.7921	-1.5021	0.3562
7	-1.4468	-0.8265	-0.5369	-1.0871	-0.8620	0.0448	-0.0277	-0.4632	0.7669	0.7853	-1.4247	0.3040
8	-1.2642	-1.1733	-1.1568	-1.5610	-0.4957	0.0228	-0.1557	-0.4275	0.1216	-0.7579	-0.3285	-0.3756

Table 1: Mean

Variables - fea[1, 2]												
fea × fea[1, 1] × fea[1, 2] × fea[1, 3] ×												
fea[1, 2] <8x12 double>												
	1	2	3	4	5	6	7	8	9	10	11	12
1	0.3714	0.1306	1.5653	0.2896	1.1479	0.1023	0.3266	3.0032	1.9812	0.7149	0.7588	0.4315
2	0.1612	0.9289	1.6913	0.1673	0.3382	0.0563	0.2103	0.7635	1.5583	1.2748	0.7924	0.4732
3	0.1900	0.2844	0.6936	0.2071	0.1833	0.0527	0.0963	0.5904	1.3051	0.4400	0.2271	0.4170
4	0.1507	0.3957	0.9069	0.1751	0.2638	0.0287	0.0630	0.6285	2.5567	0.3509	0.4491	0.7056
5	0.1330	0.4001	0.4276	0.1155	0.1374	0.0241	0.0291	0.3879	1.1563	0.4412	0.8845	0.7044
6	0.0491	0.4291	0.3187	0.0603	0.0758	0.0262	0.0299	0.3063	0.3653	0.4157	0.1736	0.7903
7	0.0505	0.1906	0.4251	0.0583	0.0680	0.0124	0.0296	0.1450	0.2938	0.2182	0.1775	0.8140
8	0.0687	0.0985	0.2991	0.0886	0.3357	0.0120	0.0420	0.1792	0.4679	0.1289	0.4311	1.3318

Table 2: Variance

Variables - fea[1, 3]				
fea × fea[1, 1] × fea[1, 2] × fea[1, 3] ×				
fea[1, 3] <12x1 double>				
	1	2	3	4
1	0.0504			
2	0.0176			
3	0.1394			
4	0.0714			
5	0.0518			
6	0.1712			
7	0.0977			
8	0.1205			
9	0.1363			
10	0.0305			
11	0.0422			
12	0.0710			

Table 3: Weights

VII. CONCLUSION

The performance of the speaker identification system using various feature extraction and feature matching techniques is studied and compared. MFCC algorithm is selected for our system because it has a very less false acceptance and zero false rejection rate. GMM model. GMM performs the function of feature matching with highest accuracy rate .

REFERENCES

- [1] D. A. Reynolds, T. F. Quatieri, and R. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, Vol. 10, No. 1–3, pp. 19–41, 2000.
- [2] J. P. Campbell, "Speaker recognition: A tutorial," *Proc. IEEE*, Vol. 85, No. 9, pp. 1437–1462, Sep. 1997.
- [3] Md.Rashidul Hasan, Mustafa Jamil, Md. Golam Rabbani Md. Saifur Rahman, "Speaker Identification Using Mel Frequency Cepstral Coefficients", 3rd International Conference on Electrical & Computer Engineering ICECE, Dhaka, Bangladesh , 28-30 December 2004.
- [4] Fatma zohra Chelali, Amar.Djeradi, Rachida.Djeradi, "Speaker Identification System based on PLP Coefficients and Artificial Neural Network", *Proceedings of the World Congress on Engineering*, London, U.K., Vol II WCE, July 6 - 8, 2011.
- [5] Janne Pytkkönen and Mikko Kurimo, *Analysis of Extended Baum–Welch and Constrained Optimization for Discriminative Training of HMMs*, *IEEE Transactions On Audio, Speech, And Language Processing*, Vol. 20, No. 9, November 2012.
- [6] Pradeep. CH, "Text Dependent Speaker Recognition Using Mfcc and Lbg Vq", *National Institute of Technology, Rourkela*, 2007.
- [7] F. Phan, M. T. Evangelia, and S. Sideman, *Speaker identification using neural networks and wavelets*, *IEEE Engineering in Medicine and Biology Magazine*, Vol. 19, 2000, pp. 92- 101.
- [8] Phadke, S.; Limaye, R.; Verma, S.; Subramanian, K., "On design and implementation of an embedded automatic speech recognition system", *VLSI Design*, 2004. *Proceedings. 17th International Conference on*, 2004 Page(s):127 – 132.
- [9] Steven B. Davis and Paul Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-28, No. 4, August 1980.
- [10] Jun Xu, Aladdin Ariyaeinia, Reza Sotudeh , Zaki Ahmad, *Preprocessing Speech Signals in FPGAs*. School of Electronic Communication and Electrical Engineering, IEEE 2005.