Smart and Robust Speaker Recognition for Context-Aware Applications

M Meher Vishal, Tusar Kanti Mishra, P Sai Charan Tej, K Krishna Reddy

Abstract

The importance of robust voice recognition has rapidly increased in latest years, as the numbers of applications and devices are increasing. This effect is strongly related to the Internet of Things framework, where these concepts are widely used in smart and under developing cities.In Computer Science, Context Awareness refers to the idea that devices and computer connected can both sense and react based on their environment. The use of this system can play a vital role in distinguishing between actual user and normal user so that it can customise the services. Driven by this motivation, in this paper we present a speaker recognition system design to get efficient output in challenging and robust conditions. We proposed this system by embedding a smart pre-processing method based on the features of voice which can efficiently extract features of the voice and detect the user by feature matching by using Gaussian Mixture Model algorithm. Moreover, it can reduce the influence of noise and other factors affecting the classification. Results shown that this system can improve the classification rate for detecting the actual user even in the case of noisy environment.

Keywords - speaker recognition, smart pre-processing, Gaussian Mixture Model, noise.

1. Introduction

Speaker recognition is the process of finding identity of a speaker using his/her voice. In other words we are able to find the person who is speaking. In present day this kind of applications used for biometric and security purpose. It can provide an alternative and more secure means of permitting entry without any need of remembering a password, lock combination. The system need only voice sample of a person to grant access. The principle of this system is that every person speaks a content with different frequencies. We take advantage this variations in frequency for identifying speaker. Speaker recognition is mainly divided into speaker verification and identification. Speaker verification is the process of verifying whether a person is a valid speaker or not. Speaker identification is finding identity of speaker. In this paper we mostly discuss about speaker identification. Before we go deeper we need to remember that environment of a speaker is always not same and speaker could use different words for his identification. We designed a robust system which can work in different environment but we use same text during training and testing of the system. This makes identification process easier. To design such kind of system we need to pre-process the audio samples collected for a speaker. The number of samples per user should be a minimum of five. We split these audio signals into small frames so that we can process the signals deeper. Once we are done with framing we perform pre-processing followed by feature extraction. For feature we use Mel-frequency cepstral coefficients(MFCC). MFCCs helps in identifying the linguistic content and discarding all the other stuff which carries information like background noise, emotion etc. Finding MFCCs includes framing, power spectrum calculation, filter-banks and DCT. Using MFCC we collect up-to 20 coefficients per speaker. Now we finally use GMM model to train the system using MFCC values. During Speaker identification(testing), we use this trained model to identify speaker. (In this paper pre-processing included in MFCC).

2. Related Work

The literature survey for research was done by referring to various journal papers, conference papers, articles and internet. In this paper we improve the efficiency and accuracy of speaker recognition system in different kinds of applications in which this system is used.These early speech recognition system tried to apply a set of grammatical and syntactical rules to identify speech. These solutions usually exploit in-formation related to the users, in order to analyse the users voice they require a certain amount of time to collect features[1].

Speech recognition research has been going on for about 80 years in which there has been at-least 4 generations of approaches, and a 5th generation of approach being a present theme of research. In the current state, there are several methods for automatic classification of utterances into emotional state has be proposed. However, the reported error rate are rather high, far behind the word error rates in speech recognition. Their research has given way for performance optimisation by the use of self-adaptive genetic algorithm. This paper consists of self-adaptive genetic algorithm to increase the probability of correct classification.

In a comparative study of past work in voice recognition and reviews by modern recognition systems and humans in order to determine how far the recent dramatic advances in technology had made progress towards the goal of human-like performance is performed. Results from random study which have compared humans and machine speech recognition on similar tasks are being summarised to determine the degree to which voice recognizer must improve to match the human performance[2][3].

3. Speaker Recognition System Architecture

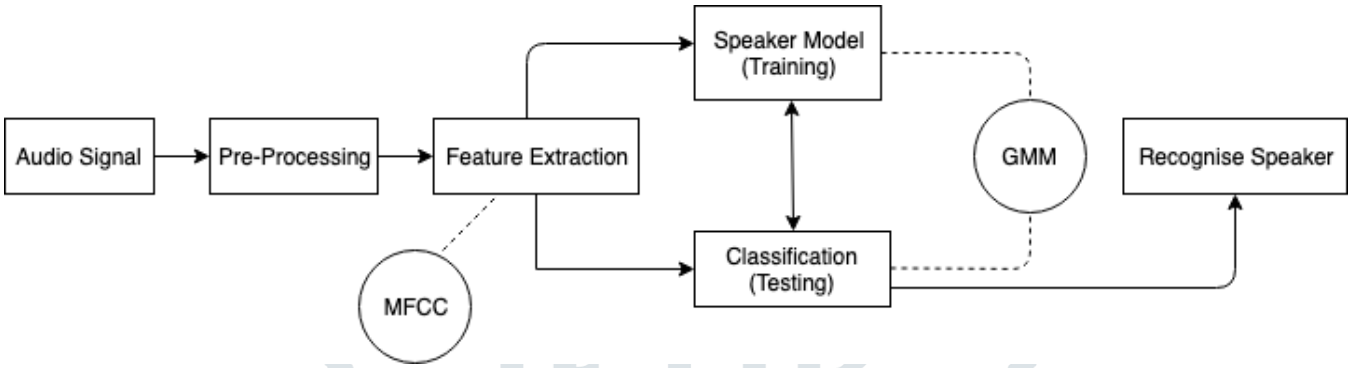


Figure 1: System Architecture

The overview of employed architecture is illustrated in the Figure 1. The proposed system is to recognise the speaker closest to feature analysis produced by the unknown user. Speech is the primary means of communication between living beings and is the interface for the humans with various developing technologies which works on voice i.e Internet Of Things. These technology was increasingly used around the world for the auto-machining and ease of living life's. Independently of this specific aspect, the general design of the proposed speaker recognition architecture is to identity the user in various environmental conditions.

The acquired audio signals fed into pre-processing stage, in which truncation, frame blocking, sampling, and noise cancellation, windowing, and short term fourier transform of the given audio input is processed. Details of this stage is discussed in Section IV.

When the audio signal is pre-processed, we proceed to feature extraction phase. In this stage speech features are computed. The best algorithm used for extraction is Mel Frequency Cepstral Coefficients (MFCC). Details of this phase are reported in Section V. After the extraction phase, the aforementioned features will be used to train a supervised classifier, in order to build proper speaker model. De-tails of this model is discussed in Section VI[3].

4. Pre - Processing

In development of this system, pre-processing is considered to be the first phase of other phases to differentiate between voiced signal and unvoiced signal and create feature vectors. In this phase the speech signal increases the amplitude of high frequency bands and decrease the amplitudes of lower bands which is implemented by FIR filter.

Sampling: The sampling rate of the signal will change based on the desired measurement being the frequency or the shape of the signal. To accurately measure the frequency of a signal, we need a sampling rate of at least twice the highest frequency in the signal. This concept is known as Nyquist's theorem. To get the shape of the signal, you will need a sampling rate of at least ten times higher than the highest frequency in the signal. The equation for frequency measurement is found below:

$$f_{max} = f_{Nyquist} = \frac{fs}{2}$$

where,
fmax is the maximum resolvable frequency
$F_{Nyquist}$ is the Nyquist frequency
fs is the sampling frequency, To measure the shape of the signal, fs will need to be divided by 10 instead of 2.

The frequency resolution (df) is dictated by the acquisition time:

$$df=\frac{1}{T}=\frac{fs}{N}$$

where T is the period of the signal
N is the number of samples acquired
fs is the sampling frequency

For example, a signal with frequency 50 Hz, there will need to be at least 0.02(1/50) seconds of data for a full period of the signal. At a sampling rate of 100 Hz for a frequency measurement, N will be 5000.

Truncation:

The default sampling frequency of wavread command is 44100 Hz. When an audio clip is recorded, say for a duration of 2 secs, the number of samples generated would be around 90000 which are too much to handle. Hence we can truncate the signal by selecting a particular threshold value. We can mark the start of the signal where the signal goes above the value while traversing the time axis in the positive direction. In the same, we can have the end of the signal by repeating the above algorithm in the negative direction.

Noise Cancellation:

Noise is ubiquitous in almost all acoustic environments. The speech signal, that is recorded by a microphone is generally infected by noise originating from various sources. Such contamination can change the characteristics of the speech signals and degrade the speech quality and intelligibility, thereby causing significant harm to human-to-machine communication systems. Noise detection and reduction for speech applications is often formulated as a digital filtering problem, where the clean speech estimation is obtained by passing the noisy speech through a linear filter. With such a formulation, the core issue of noise reduction becomes how to design an optimal filter that can significantly suppress noise without noticeable speech distortion.

Frame Blocking:

In this step the continuous speech signal is divided into frames of N samples, with adjacent frames being separated by M samples with the value M less than that of N. The first frame consists of the first N samples. The second frame begins from M samples after the first frame, and overlaps it by N - M samples and so on. This process continues until all the speech is accounted for using one or more frames[3]. We have chosen the values of M and N to be N = 256 and M = 128 respectively. Figure 3 below gives the frame output of the truncated signal. The value of N is chosen to be 256 because the speech signal is assumed to be periodic over the period. Also the frame of length 256 being a power of 2 can be used for using a fast implementation of Discrete Fourier Transform (DFT) called the FFT (Fast Fourier Transform).

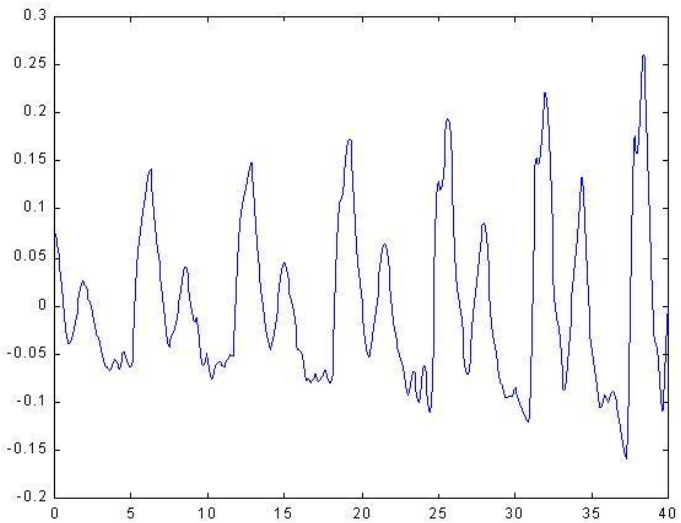


Figure 2 Frame output of truncated signal

Windowing:

The next step is to window each individual frame to minimize the signal discontinuities at the beginning and end of each frame. The concept applied here is to minimize the spectral distortion by using the window to taper the signal to zero at the beginning and end of each frame. If we define the window as $w(n)$, $0 \le n \le N-1$, where N is the frame length, then the result of windowing is the signal.

$$y(n) = x(n)w(n), \qquad 0 \le n \le N-1$$

Hamming Window:

$$w(n) = 0.54 - 0.46\, cos\left(\frac{2\Pi n}{N-1}\right) \;, 0 \le n \le N-1$$

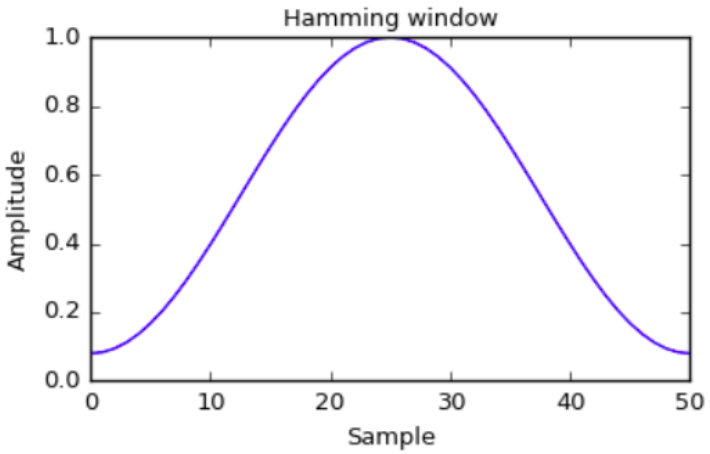


Figure 4: Hamming Window

Short Term Fourier Transform (STFT):

The next step is the application of Fast Fourier Transform (FFT), which converts each frame of N samples from the time domain into the frequency domain. The FFT which is a fast algorithm to implement the Discrete Fourier Transform (DFT) is defined on the set of N samples $\{x_n\}$, as follows:-

$$X_k = \sum_{n=0}^{N-1} \quad x_n e^{\frac{-j2\Pi kn}{N-1}} \;, \text{ k = 0,1,2,3,..,N}$$

In general X_k"s are complex numbers and we consider only their absolute values. The resulting sequence $\{X_k\}$ is interpreted as follows: positive frequencies $0 \le f < F_s / 2$ correspond to values $0 \le n \le N/2 - 1$, while negative frequencies $-F_s / 2 < f < 0$ correspond to $N/2 + 1 \le n \le N - 1$. F_s denotes the sampling frequency. The result after this step is often referred to as spectrum or periodogram[3].

5. Feature Extraction

Feature Extraction is the most important step in automated speech recognition. Since speech signals are unstable in nature, statistical representations should be generated for representation of the speech signal variability which is achieved by performing feature extraction. These features can be obtained by spectrogram of the speech signal, and we are using Mel-Frequency Cepstral Coefficients (MFCC) features in speaker identification, the advantages of perceptual frequency scale based critical bands with cepstrum analysis are combined[1][3].

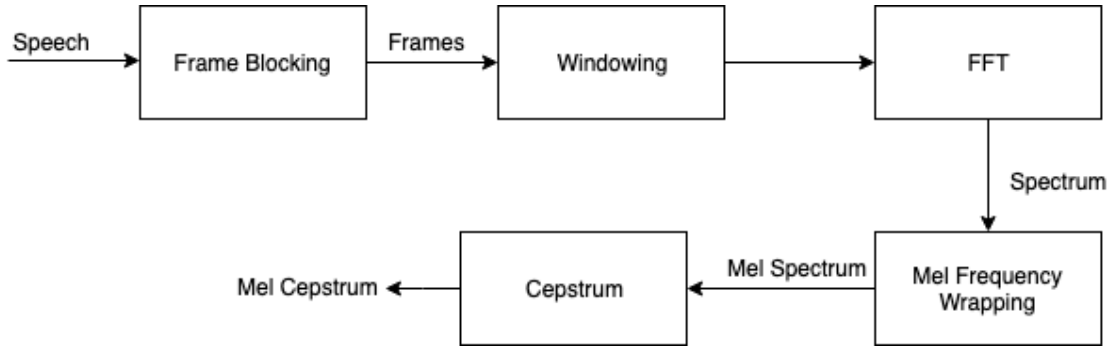
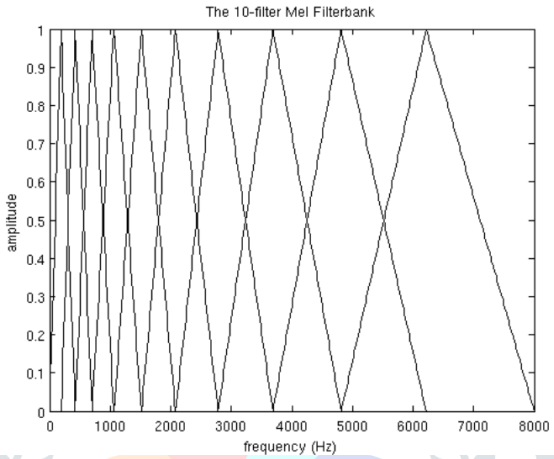


Figure 4 Mel Frequency Cepstral Coefficients



The MFCC processor involves the following steps:

Mel Frequency Wrapping:

As given in the block diagram we have already subjected the continuous speech signal to frame blocking, windowing and FFT in the pre-processing step. The result of the later step is the spectrum of the signal. Psychophysical studies have revealed that human perception of frequency content of sounds for speech signals doesn't follow a linear scale. For each tone with an actual frequency f, a subjective pitch is measured on a scale called the „mel" scale. The mel-frequency scale provides a linear frequency spacing below 1 KHz and a logarithmic spacing above 1 KHz. The Mel Frequency Scale is given by:-

$$F_{mel} = (1000/\log(2)) * \log(1+ f/1000)$$

One approach towards simulating the subjective spectrum is to use a filter bank which is spaced uniformly on the mel-scale. The filter bank has a triangular band pass frequency response. The spacing and the bandwidth is determined by a constant mel frequency interval. We choose K, the number of mel spectrum coefficients to be 20. This filter bank being applied in the frequency domain simply amounts to applying the triangle-shape windows to the spectrum. A useful way to think about this filter bank is to view each filter as a histogram bin (where bins have overlap) in the frequency domain. Figure 6 below gives an example of a mel-spaced frequency bank.

Cepstrum:

In this final step, we convert the log Mel spectrum to time domain. The result is called the MFCC (Mel Frequency Cepstral Coefficients). This representation of the speech spectrum provides a good approximation of the spectral properties of the signal for the given frame analysis. The Mel spectrum coefficients being real numbers are then converted to time domain using Discrete Cosine Transform (DCT). If we denote the Mel power spectrum coefficients that are the result of the last step as S_k, k = 1,2,...,K, we can calculate the MFCC's C_n as

$$Cn = \sum_{k=1}^{K} \quad (log\ S_k)\ Cos\left[n\left(k-\frac{1}{2}\right)\frac{\Pi}{2}\right], \text{ n = 1,2,..,K}$$

We exclude the first component from the DCT since it represents the mean value of the input signal which carries little speaker specific information[11].

6. Feature Matching

The problem of speaker recognition has always been a much wider topic in engineering field so called pattern recognition. The aim of pattern recognition lies in classifying objects of interest into a number of categories or classes. The objects of interest are called patterns and in our case are sequences of feature vectors that are extracted from an input speech using the techniques described in the previous chapter. Each class here refers to each individual speaker. Since here we are only dealing with classification procedure based upon extracted features, it can also be abbreviated as feature matching.

To add more, if there exists a set of patterns for which the corresponding classes are already known, then the problem is reduced to supervised pattern recognition. These patterns are used as training set and classification algorithm is determined for each class. The rest patterns are then used to test whether the classification algorithm works properly or not; collection of these patterns are referred as the test set. In the test set if there exists a pattern for which no classification could be derived, and then the pattern is referred as unregistered user for the speaker identification process. In real time environment the robustness of the algorithm can be determined by checking how many registered users are identified correctly and how efficiently it discards the unknown users. Feature matching problem has been sorted out with many class-of-art efficient algorithms like VQLBG, DTW and stochastic models such as GMM, HMM. In our study project we have put our focus on VQLBG, DTW and GMM algorithm. VQLBG algorithm due to its simplicity has been stressed at the beginning followed by DTW and GMM respectively.

7. Speaker Modelling

Using Cepstral coefficients and MFCC as illustrated in the previous section, a spoken syllable can be represented as a set of feature vectors. A person uttering the same word but at a different time instant will be having similar still differently arranged feature vector sequence. The purpose of voice modeling lies in building a model that can capture these variations in a set of features extracted from a given speaker. There are usually two types of models those are extensively used in speaker recognition systems:

- Stochastic models
- Template models

The stochastic model exploits the advantage of probability theory by treating the speech production process as a parametric random process. It assumes that the parameters of the underlying stochastic process can be estimated precisely, in a well-defined manner. In parametric methods usually assumption is made about generation of feature vectors but the non-parametric methods are free from any assumption about data generation. The template model (non-parametric method) attempts to generate a model for speech production process for a particular user in a non-parametric manner. It does so by using sequences of feature vectors extracted from multiple utterances of the same word by the same person. Template models used to dominate early work in speaker recognition because it works without making any assumption about how the feature vectors are being formed. Hence the template model is intuitively more reasonable. However, recent work in stochastic models has revealed them to be more flexible, thus allowing for generation of better models for speaker recognition process. The state-of-the-art in feature matching techniques used in speaker recognition includes Dynamic Time Warping (DTW), Gaussian Mixture Modeling (GMM), and Vector Quantization (VQ)[3][4].

8. Proposed Scheme using GMM

This is one of the non-parametric methods for speaker identification. When feature vectors are displayed in d-dimensional feature space after clustering, they somehow resemble Gaussian distribution. It means each corresponding cluster can be viewed as a Gaussian probability distribution and features belonging to the clusters can be best represented by their probability values. The only difficulty lies in efficient classification of feature vectors. The use of Gaussian mixture density for speaker identification is motivated by two facts[4][5]. 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[4].

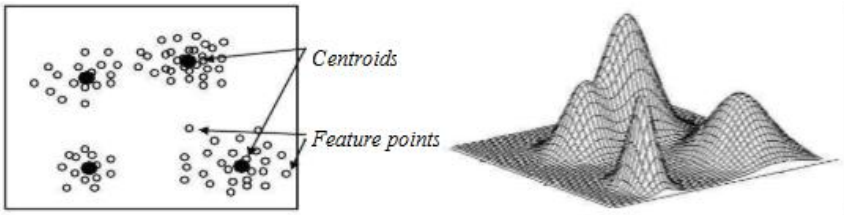


Figure 7: GMM model showing a feature space and corresponding Gaussian model

8.1 MODEL DESCRIPTION

A Gaussian mixture density is weighted sum of M component densities and given by the equation:-

$$p(\underline{x} \mid \lambda) = \sum_{i=1}^{M} p_i b_i(\underline{x})$$

where x refers to a feature vector, p_i stands for mixture weight of i^{th} component and b_i (x) is the probability distribution of the i^{th} component in the feature space. As the feature space is D-dimensional, the probability density function b_i (x) is a D-variate distribution. It is given by the expression:-

$$b_i(\underline{x}) = \frac{1}{(2\pi)^{D/2}|\sum_i|^{1/2}} \exp\left\{-1/2(\underline{x} - \underline{\mu_i})^t \Sigma_i^{-1}(\underline{x} - \underline{\mu_i})\right\}$$

where μ_i is the mean of i^{th} component and $\sum_i$ is the co-variance matrix[4].

The complete Gaussian mixture density is represented by mixture weights, mean and co-variance of corresponding component and denoted as:-

$$\lambda = \left\{p_i, \underline{\mu_i}, \Sigma_i\right\} i = 1, \ldots\ldots, M$$

Diagrammatically it can be shown as:- (Figure 7)

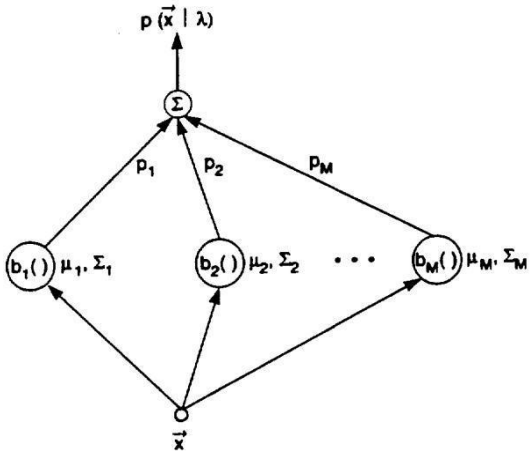


Figure 7: Description of M-component Gaussian densities

8.2 MAXIMUM LIKELIHOOD PARAMETER ESTIMATION

After obtaining the feature vectors the next task lies in classifying them to different Gaussian components. But initially we don"t know mean, co-variance of components present. Thus we can't have proper classification of the vectors. To maximize the classification process for a given set of feature vectors an algorithm is followed known as Expectation Maximization (EM)[6][7] .This algorithm works as follows:-

1. We assume initial values of μ_i, $\sum_i$ and w_i.
2. Then we calculate next values of mean, covariance and mixture weights iteratively using the following formula so that probability of classification of set of T feature vectors is maximized[7][8].

The following formulae are used:-

Mixture Weights:

$$\underline{p_i} = 1/T \sum_{t=1}^{T} p(i \mid \underline{x_t}, \lambda)$$

Means:

$$\underline{\mu_i} = \frac{\sum_{t=1}^{T} p(i \mid \underline{x_t}, \lambda)\underline{x_t}}{\sum_{t=1}^{T} p(i \mid \underline{x_t}, \lambda)}$$

Variances:

$$\underline{\sigma_i^2} = \frac{\sum_{t=1}^{T} p(i \mid \underline{x_t}, \lambda)x_t^2}{\sum_{t=1}^{T} p(i \mid \underline{x_t}, \lambda)} - \underline{\mu}_i^2$$

where p(i | [illegible] , λ) is called posteriori probability and is given by the expression:-

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

8.3 SPEAKER IDENTIFICATION

After modeling each user"s Gaussian mixture density, we have a set of models, each representing Gaussian distribution of all the components present. For K number of speakers it is denoted as $\lambda = \{\lambda_1, \lambda_2, \lambda_3, \ldots\ldots, \lambda_k\}$. The objective culminates in finding the speaker model λ having maximum posteriori probability for a given test utterance[8]. Mathematically it can be represented as:-

$$\hat{S} = arg\ max\ P_r(\lambda_k \mid X) = arg\ max\ \frac{p(X \mid \lambda_k)\ P_r(\lambda_k)}{p(X)}$$

9. RESULTS AND DISCUSSION

We evaluated the text independent speaker identification using phase information on NIT dataset. The speaker identification results by the table below:

Speed		Normal	Fast	Slow	Average
MFCC - based GMM		98.7	96.7	96.9	97.4
$\{\underline{\theta}\}$	(60 - 70 Hz)	52.6	51.6	51.7	52.0
	(300 - 1000 Hz)	61.0	57.6	56.6	58.4
	(600 - 1300 Hz)	31.6	31.7	34.7	32.7

The method phase $\{\underline{\theta}\}$ means the phase value obtained by the equation below was used as the speaker identification feature.

$$\underline{\theta}(w,t) = \theta(w,t) + \frac{\omega}{\omega_b}\left(-\theta(\omega_b,t)\right)$$

However, the phase information based method performed worse than MFCC based but it is useful for speaker recognition. The below graphs shows results using a combination of MFCC and the original phase.

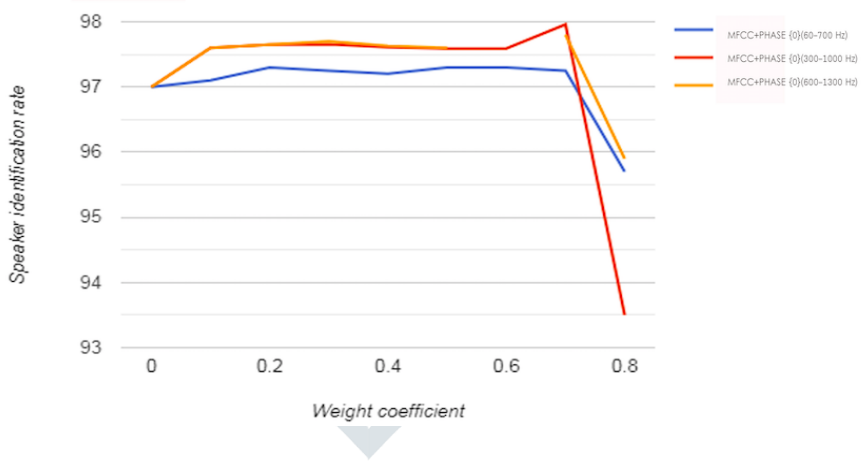


Figure 9: Results of average of 3 speaking modes.

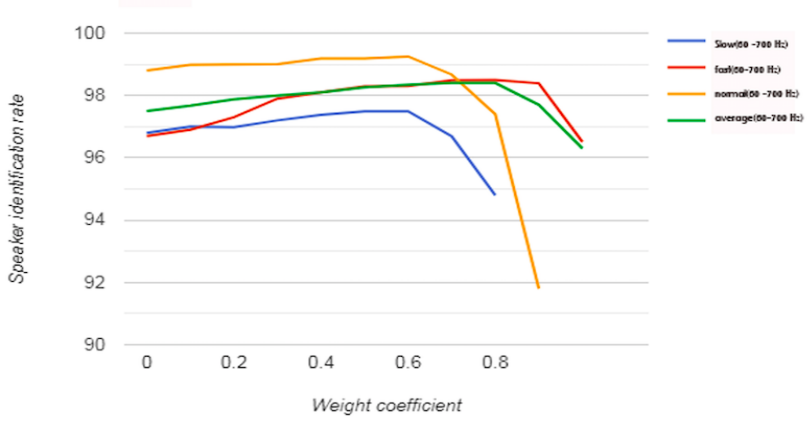


Figure 10: Results of frequency range 60-70 Hz

10. CONCLUSION

The objective of this final project was to design and implement a robust and smart speaker recognition system with user interface.This is the simulation model of the Speaker Recognition system in context aware applications.A genuine time voice recognition system is projected. It depends on MFCC for feature removal and on GMM for preparation. Firstly the voice is taken from side to side microphone and voice features are extracted by dividing voice sample into 30ms frame length with 20ms edge partly cover to the preceding edge.The speaker is recognized by comparing the log probability to the defined threshold in the system. The Speaker Recognition system needs to be evaluated on a variety of larger datasets, so that more inferences can be drawn from the results and enhancements to the Shifted MFCC can be made. Also different fusion techniques at the modeling level such as SVM Vs. GMM, HMM Vs. SVM needs to be studied, and evaluated on a variety of datasets to better understand the effect of different fusions, so that a common technique can be formulated to find the optimal fusion weights. The process of identifying human through speech is a complex one and our own human recognition system is an excellent instrument to understand this process. The human recognition system extracts several other features from a single speech signal, due to which it achieves high accuracy. The goal of a speech researcher should be to identify such missing pieces of information, in a hope to match the human recognition system someday.The emulator version of the same project could be developed to get better real time experience

11. REFERENCES

[1] PRADEEP. CH," TEXT DEPENDENT SPEAKER RECOGNITION USING MFCC AND LBG VQ", National Institute of Technology, Rourkela, 2007

[2] Seddik, H.; Rahmouni, A.; Samadhi, M.; "Text independent speaker recognition using the Mel frequency cepstral coefficients and a neural network classifier" First International Symposium on Control, Communications and Signal Processing, Proceedings of IEEE 2004 Page(s):631 – 634.

[3] International Journal of Innovative Research in Advanced Engineering (IJIRAE) : Voice Recognition Using MFCC Algorithm

[4] International Journal of Engineering Trends and Applications (IJETA) – Volume 4 Issue 2 Automated Speech Recognition System.

[5] Roucos, S. Berouti, M. Bolt, Beranek and Newman, Inc., Cambridge, MA; "The application of probability density estimation to text-independent speaker identification" IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '82. Volume: 7, On page(s): 1649- 1652. Publication Date: May 1982.

[6] Reynolds D.A.: "A Gaussian Mixture Modeling Approach to Text-Independent Speaker Identification", Ph.D. thesis, Georgia Institute of Technology, September 1992.

[7] Douglas A. Reynolds and Richard C. Rose, "Robust Text-Independent Speaker Identification using Gaussian Mixture Speaker Models", IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING, VOL. 3, NO. 1, JANUARY 1995

[8] Castellano, P.J.; Slomka, S.; Sridharan, S.; "Telephone based speaker recognition using multiple binary classifier and Gaussian mixture models" IEEE International Conference on Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 Volume 2, Page(s) :1075– 1078 April 1997