Live Speaker Identification Using MFCC and Delta-MFCC

Bhushan Pal Singh[1] Sandeep Kumar Jha[2] Rohit Khurmi[3] Niraj Kumar Yadav[4]
[1,2,3,4]Department of Computer Science and Engineering
[1,2,3,4]HMR Institute of Technology and Management, India

Abstract— Speaker Identification is a subfield of digital signal processing, which verifies the identity of a person using features of their voice samples. Now days it is considered as a popular biometric verification technique in various fields. It finds application in forensic speaker recognition, authentication, surveillance and other related areas. In this paper we presented a system developed for text independent, live identification of speaker. Python 3 programming language is used to develop the system with simple tkinter GUI. Concept of Speaker Identification is inspired from the behavior of human ears. MFCC (Mel-Frequency Cepstral Coefficient) features are considered to be best for mimicking the voice signal processing in human ears. MFCC features represent only the power spectral envelope of single frames, but information of dynamic changes in the frames would be quit useful to include in the feature vector performing delta operation over features gives this information We used MFCC + Delta MFCC features of voice signal. Using the one more delta over the delta features may increase the accuracy but it increases redundancy also so we skipped that. [7]. The whole process of this technique involves basically two modules Feature extraction and feature matching. Feature extraction is a method of extracting a small amount of data/voice signal that can be used to represent each speaker. Feature matching is the process of comparing the voice inputs from a set of known speakers. We used GMM (Gaussian Mixture Model) for training and Loglikelihood function for feature matching.

Keywords: Speaker Identification, Text independent, Gaussican Mixture Model (GMM), Mel Frequency Cepstral Cofficient (MFCC), Delta MFCC, Loglikelyhood

I. INTRODUCTION

In our daily life we hear different type of sounds, and by using previous experience our brain indicates us, what could be the source of the sound. Similarly, our human brain has been training itself to identify the people by listening their voice. Human brain identifies the person by analyzing features included in the voice signals like loudness, pace, pitch, stress, resonance, tone, variations in pitch, variations in stress etc. Automatic Speaker identification system is an implementation of Machine learning that mimic the steps in the human ear and brain to identify the speaker. Speaker identification System can be useful in long distance communication where voice is the important factor for identification of the person. It detects weather the speaker is a familiar or known person. Speaker recognition process is categorized into text dependent or text independent. Text-dependent where a particular sentence or phrase is used to identify. Text-independent where speaker is recognized irrespective of what he is saying (voice only). It is the different characteristics of the speaker which separates the person form another. This system increases the interaction between the Human & the Computer. There are several different transformations level of a signal like acoustic, articulatory, semantic & linguistic. All the variations between these levels are taken into accounts while designing such a system.[8]. We can divide the speaker recognition model into three Classes:

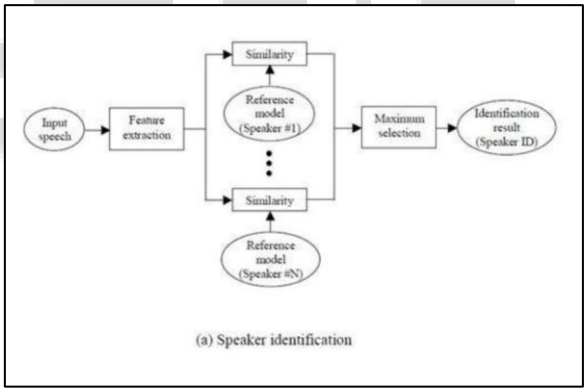
1) Text-Dependent vs Text Independent
2) Identification vs Verification
3) Open Set vs Close Set

1) Text-Dependent vs Text Independent:

This type of class is based on the text speak by the user in the identification process. Text-Dependent: In this method the voice samples which are used in the training process are identical. A user has the basic knowledge about the system.Text Independent: In this method, the voice samples which are used in the training process are not identical. A user doesn't have basic knowledge about the model i.e. he/she can speak anything.

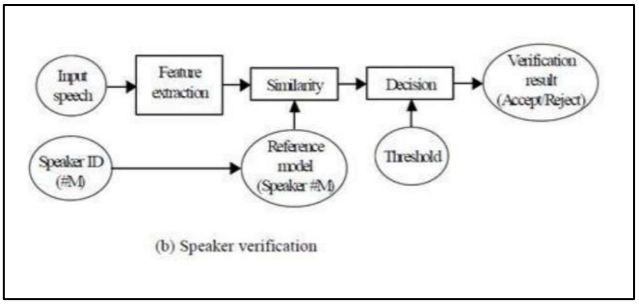
2) Identification vs Verification:

This is the most important class of the speaker recognition model. This is considered to be the most economical and natural class for preventing a system from unauthorized access. Speaker Identification: It is the process of identification of the authorized or unauthorized use by the system.[1]



(a) Speaker identification

Speaker Verification: It is the process of accepting/rejecting access. The claim of authorized/unauthorized users is known as speaker verification.



(b) Speaker verification

3) Open Set vs Close Set:

This is based on the set of trained speakers available in a speaker recognition model. Open Set: This type of system can have any number of trained Speakers. Close Set: This type of system can have a fixed number of trained speakers.

All rights reserved by *www.ijsrd.com*

The problem of speaker identification has always been a popular topic in the engineering field and is also known as pattern recognition. The various patterns are used for the training set and classification algorithms. If the pattern which is known by the system is matched with the test data of the user, then the pattern is recognized as registered otherwise the pattern is recognized as unregistered.

There are mainly two modules that are used in developing the system- Feature extraction and feature matching.

1) Feature Extraction: The aim of this module is to modify the speech waveform into a set of feature vectors for feature analysis. This module is also known as front end signal processing. For feature extraction, we use Mel Frequency Cepstral Coefficients (MFCC) and Delta MFCC.[6]
2) Feature Matching: This module is used for the pattern matching process to identify or test whether a speaker is authorized or unauthorized. We can use algorithms such as GMM (Gaussian Mixture Model) and HMM (Hidden Markov Model)

II. WORKING PROCESS

The developed system is consisting of five core processes:

A) Enrolling Voice samples in training dataset.
B) Audio Preprocessing
C) Feature extracting
D) Training
E) Matching

A. Enrolling Voice Samples in trainig dataset

The first step in any voice recognition system is to take the voice sample. When a user speaks, his voice is recorded for a duration of up to 5 seconds using a sampling rate of 8-44 KHz. And then this sample is saved as a wav file to use in further steps.

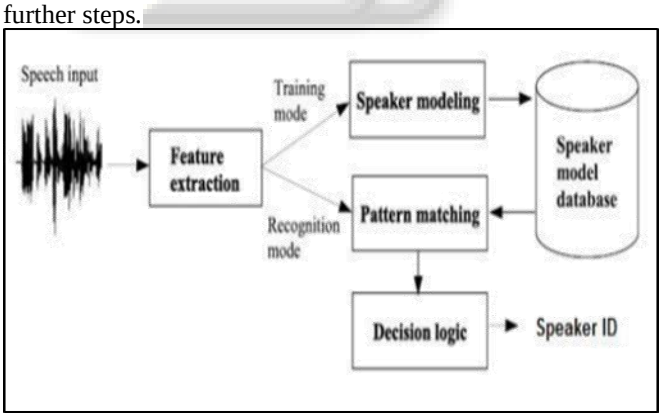


Fig. 1: Schematic diagram of closed-set speaker identification system.

B. Audio Preprocessing

The recorded audio from the system's microphone is consisting background noise that must be removed or reduced up to an optimal level. We used *noisereduce* python library that spectral gating algorithm for noise reduction.
That uses starting part of signal as noise of silence and to denoise the rest of the signal [8].

C. Feature Extraction

After taking the voice samples the next step us to extract the required features from the voice signal. Mel Frequency Cepstral Coefficients (MFCC) technique is used to extract these features. MFCC is based on human voice signal processing system. The Speech signals are considered as the convolution of source (air expelled from lungs) and filter (our vocal tract). The shape of vocal tract governs what sound is produced and MFCC best governs this shape.

MFCC and Delta MFCC

A speech signal is just a sequence of numbers which denotes the amplitude of speech spoken by the speaker. MFCC performs an audio analysis that represents the ear model which proves good result in the recognition of the speaker in the case when the high number of coefficients are used. Because of the efficiency of the ear model MFCC used to extract features as it operates in a separated mode. By using this we can recognize a person from his voice without understanding what he is speaking about. We will use MFCC or Delta MFCC or Delta Delta MFCC to find out the features of the individual speakers on the basis of classification. Speech MFCC includes preprocessing some steps of audio signals by three techniques- framing, windowing & overlapping frames.[2]

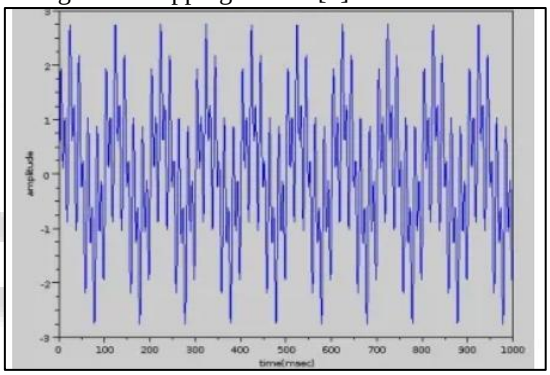


Fig. 2: Original Speech Signal

1) Framing:

Since the speech signal is non-stationary, its frequency contents continuously vary with time. Framing is generally done into two types of frames: 1. fixed-size frame 2. dynamic size frame. In the fixed-size frame, the number of the frame has different speech speed due to different lengths of voice signals. We use a dynamic size frame to acquire a fixed number of frames when we use an artificial neural network. To achieve stationarity, this speech signal must be framed. This is done by dividing the speech signal into short frames usually of 20-30 milliseconds.

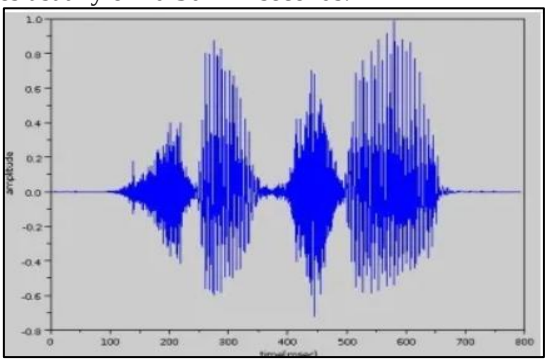


Fig. 3: Framing of Speech Signals

All rights reserved by *www.ijsrd.com*

2) Windowing:
Framing of speech signals may lead to discontinuities in the speech signal at the endpoints which may further lead to spectral leakage in the speech signal. This process minimizes the effect of spectral artifacts from the framing process. A function that has a narrow main lobe and low side-lobe levels in their transfer function is considered a good window function. The windows that are commonly used during the frequency analysis of speech sounds are Hamming and Hanning window. To prevent this framed signal is multiplied with a window function so that its amplitude falls to zero towards the endpoints.

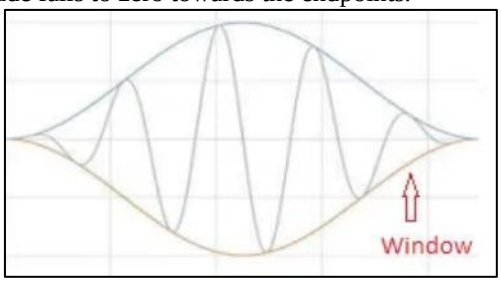


Fig. 4: Windowing Speech Signals

3) Overlapping Frames:
In this we take overlapping frames rather than disjoint frames to compensate for the lost frames at the beginning and at the end of frame in the windowing. The overlapping between frames is usually 10-15 milliseconds.

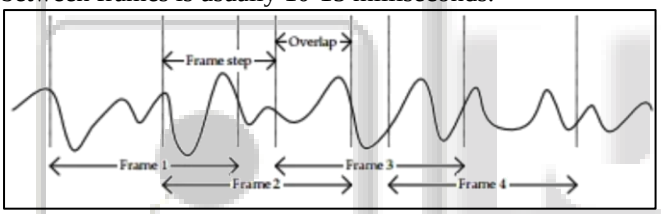


Fig. 5: Overlapping frames in Speech Signal

The purpose of MFCC is to characterize the filter part and remove the source part. MFCC focuses on series of calculation that uses cepstral with nonlinear frequency axis called Mel scale. We are focusing on two main features MFCCs and their Derivatives, say Delta-MFCC. We calculated 20 MFCCs and 20 Delta MFCCs. So totally we had 40 features in hand. Steps involved in MFCC are:

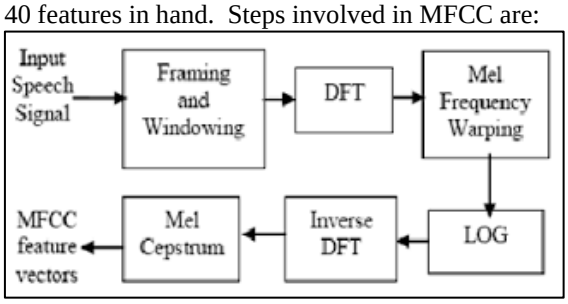


Fig. 6: Steps involved in MFCC Feature

The voice signal is Framed and windowed first using analysis window then Discrete Fourier Transform (DFT) is computed to extract information in frequency domain. Mel scale maps the measured frequency to that we perceived in the context of frequency resolution. Take log of transformed values, transform from multiplication to summation made it easy to separate source and filter using a linear filter. At the end Discrete Cosine Transform (DCT) is applied on log filter to get Mel scale cepstral coefficients.[9]

We performed MFCC +Delta MFCC feature extraction using inbuild in library python_speech_features i.e.:
mfcc(signal, samplerate=rate, winlen=0.025, winstep=0.01, numcep=20, nfilt=26, nfft=1200, lowfreq=0, highfreq=None, preemph=0.97, ceplifter=22, appendEnergy=True)
and
delta(signal, N=2) [12]

D. Model Training

For training the model extract features from the speech signal and then pass them to the statistical model, here we use Gaussian Mixture Model (GMM) as statistical model due to ease of implementation and high accuracy. Gaussian Mixture Model is used to create a unique voice print for each identity by using feature vector extracted from each speaker. These models are stored in the database by using object deserialization and are used for identification of unknown speaker during testing phase.[5]
GMM(n_components = 16, n_iter = 200, covariance_type = 'diag' , n_init = 3)

1) GAUSSIAN MIXTURE MODEL

GMM statistical speaker model is created after extracting features from the speech signals using MFCC and Delta MFCC. A GMM is a probabilistic model that assumes that all data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. One can think of mixture models as generalize k-means clustering to incorporate information about the covariance structure of data as well as the centers of the latent gaussians. Conditions when single normal distribution fail at such moment finite mixture models and their parameter estimation method can be approximated by a wide range of probability density functions. The multivariate normal distribution is one of the most useful and well-known distribution in playing predominant role in statics and in other areas of application.

A gaussian mixture density is weighted as the sum of M-component densities. In speaker recognition applications we have inculcated Gaussian classifier…

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

Where M defines the number of component densities,
pi stands for mixture weight of i^{th} component, bi(x) is a probability distribution of i^{th} component in the feature space.

As the probability density function is a D-variate distribution. It is given by the expression-

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

where μi is the mean of i^{th} component and ∑ i is the co-variance matrix

All rights reserved by *www.ijsrd.com*

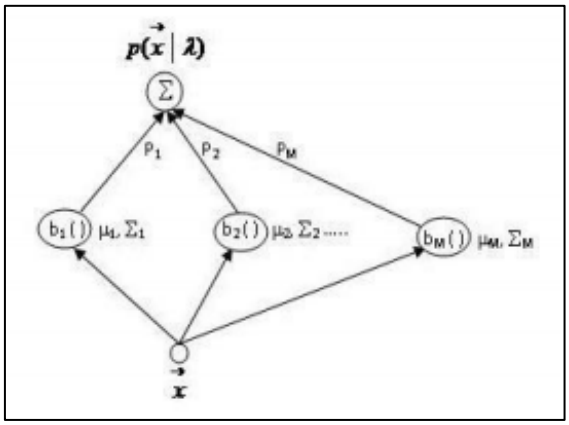


Fig. 7: Gaussian Mixture Model

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

$$\lambda = \{p_i, \vec{\mu}_i, \Sigma_i\} \; i = 1, \ldots, M.$$

Each speaker can be represented by GMM and we have to calculate a good estimation of GMM parameters to obtain optimum model representing the each speaker.

2) Maximum Likelihood Function

For testing we used maximum likelihood (ML) function for calculating GMM scores. The model with the highest GMM score will be detected as a winner. GMM likelihood function is written as-

$$P(x|\lambda) = \prod_{t=1}^{T} p(x_t|\lambda)$$

3) EM Algorithm

The EM algorithm is an efficient iterative procedure to compute the maximum likelihood estimation in the presence of missing or hidden data. Each iteration of EM algorithm consists of two steps- E-step and M-step. In the E-step missing data are estimated form the observed data. In the M step, the likelihood is maximized under the assumption that missing data are known. Algorithm is guaranteed to increase the likelihood at each step.

E. Feature Matching

In matching process, decision makes the final decision about the identity of speaker by comparing unknown speaker to all models in the database and selecting the best matching model in the database. It regulates log probability of voice vector and compares it to previously stored value. The log probability equal to the stored value provides access to the entire speaker.[11]

scores = np.array(gmm.score(vector))
loglike_lihood[i] = scores.sum()
winner = np.argmax(log_likelihood)
where: gmm =array of trained models,
vector = testing signal model
winner = detected speaker

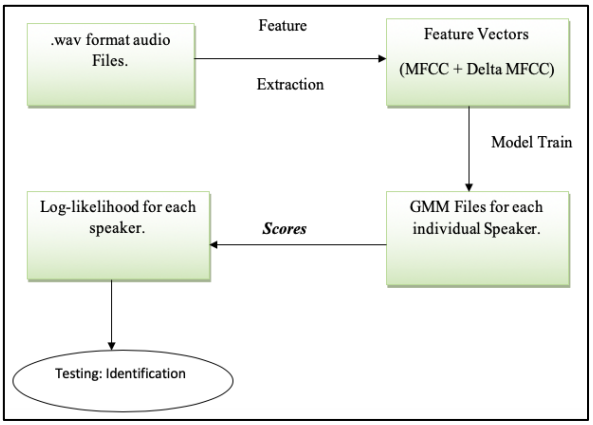


Fig. 8: State Transition Diagram

III. IMPLEMENTATIONS

1) Running the project using GUI in for different options On Vox-Forge dataset; training and testing
2) On Self-made dataset; training and testing
3) Enrolling voice samples in the dataset; record and training
4) Live Testing; record and training

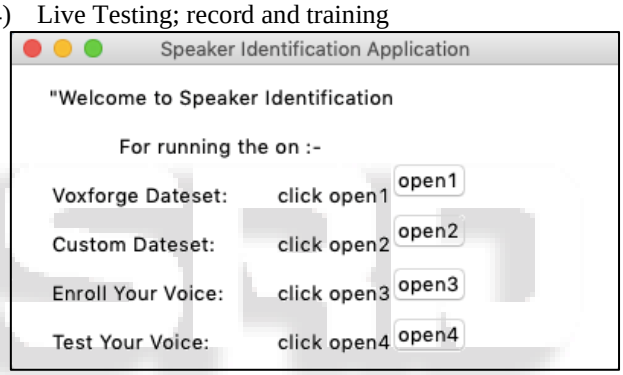


Fig. 9: Application GUI

1) Vox-Forge Model Training

We trained the model on Vox-Forge Dataset, containing 220 speakers' voice samples. Each speaker has 10 voice samples, 5 are used for training, and 5 are used for testing.

We reduced the size of the data set sequentially to reduce the training time and improve accuracy.

```
belmontguy-20110426-geu/wav/b0149.wav
belmontguy-20110426-geu/wav/b0150.wav
belmontguy-20110426-geu/wav/b0151.wav
belmontguy-20110426-geu/wav/b0152.wav
belmontguy-20110426-geu/wav/b0153.wav
('modeling completed for speaker:', 'belmontguy.gmm',
```

Fig10: Vox-Forge Dataset Training

2) Vox- Forge- Testing: Identification

Accuracy is sequentially increased by reducing the size of the dataset. We achieved 100 % accuracy on the size of 34 speakers.

```
('Testing Audio : ', 'belmontguy-20110426-geu/wav/b0156.wav')
('\tdetected as - ', 'belmontguy')
('Testing Audio : ', 'belmontguy-20110426-geu/wav/b0157.wav')
('\tdetected as - ', 'belmontguy')
('Testing Audio : ', 'belmontguy-20110426-geu/wav/b0158.wav')
('\tdetected as - ', 'belmontguy')
(1, 170.0)
```

```
testing Performance with MFCC + GMM is : ', 99.41176470588235, '%')
```

```
Speaker identified Successfully.
```

Fig. 11: Vox -Forge Dataset Identification

All rights reserved by *www.ijsrd.com*

3) Self-Made Dataset-Model Training

We have developed a customized dataset having voice samples of our friends. We have 12 speakers in our data set to check if our model performs well on a small data set also. Each of 12 speakers has 10 voice samples 5 are used for training and 5 are used for testing.

```
Enter name:     'Rohit'
* recording
* done recording
```

Fig. 12: Self-Made Dataset Model Training

4) Self-Made Dataset- Testing: Identification

On performing testing on the self-made dataset we achieved good accuracy so we conclude that our model can be used to make a live speaker identification application

```
('modeling completed for speaker:', 'Rohit.gmm',
Rohit-/Rohit11.wav
Rohit-/Rohit12.wav
Rohit-/Rohit13.wav
Rohit-/Rohit14.wav
Rohit-/Rohit15.wav
```

Fig. 13: Self-Made Dataset Model Testing

5) Enroll Voice Sample in dataset

For performing live speaker identification, first, we need to Enroll the speaker's voice samples in the dataset. In the option, speak will have to click the record button and enter the name of the speaker. After that microphone recorder will record sound for 5 sec, the speaker can speak anything in any language. Speaker can enroll more than one voice sample in the dataset.

After enrolling new voice samples, training on new voice samples will be performed by clicking the train button.

```
Enter name:     'Rohit'
* recording
* done recording
```

Fig. 14: Training the speaker's voice

```
('modeling completed for speaker:', 'Rohit.gmm', '
Rohit-/Rohit21.wav
Rohit-/Rohit22.wav
Rohit-/Rohit23.wav
```

Fig. 15: Modelling the speaker's voice

6) Live Testing recording

If a speaker's voice has been enrolled and training has been done on new his/her voice sample(s), the live identification can be performed in this option.

First speaker will have to record his/her voice and then click on test button and speaker will be detected and speaker's name will be displayed.

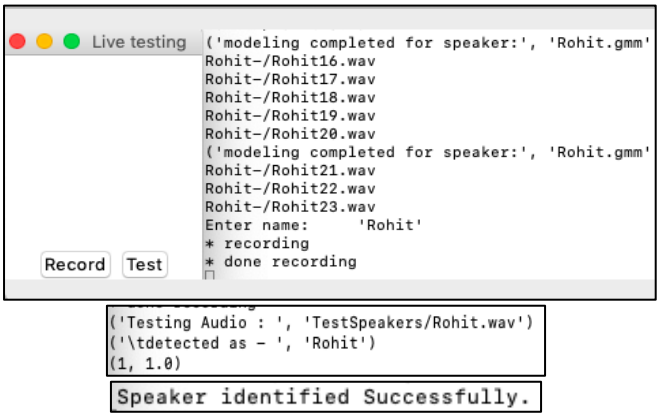


Fig. 16: Speaker Detection during Live Testing

IV. RESULTS

The Speaker identification was successful conducted with an outstanding result on both of those datasets. The accuracy was 100% in case of Vox-Forge Dataset and 95.29% in case of self- made dataset. Thus MFCC-GMM model gives satisfactory results

S.no	RESULT		
	Dataset	*Accuracy*	*Training Time(min)*
1	VoxForge(220 Speakers)	97.41 %	17 (Approx.)
2	VoxForge(100 Speakers)	98.44%	7 (Approx.)
	VoxForge(34 Speakers)	99.41%	2-3 (Approx.)
4	Self-Made(12 Speakers)	96.33%	1-2 (Approx.)

V. CONCLUSION

The simple system for live speaker identification is developed with satisfying performances. This system give outstanding performance on VoxForge dataset [10] as well as on self-made custom dataset. We used MFCC algorithm in our system because it has least false acceptance ratio. To improve system performance and also to achieve high accuracy we used 20 MFCC + 20 Delta MFCC .i.e. total 40 features of each voice signal. We used GMM model for model training and testing and achieved outstanding performance on the basis of training time and accuracy.

In future, we are planning to convert our live speaker identification system into an android application.

VI. FUTURE SCOPE

The branch of speaker identification finds immense popularity in different fields of applications like command interfaces, automated dictation and embedding recognition in a product that allows an identification level of intuitive and hand free user interaction. The various steps involved in this project will provide us a great depth knowledge of the speaker recognition community.

The speaker identification system can help in making a developed behavioral biometric system in which we can easily identify authorized persons very quickly in the future development of the biometric systems. This model can help in identifying a criminal voice by matching it with the criminal record database based on different voice samples of all the previous prisoners.

A large number of speaker recognition processes are being done in universities, national laboratories, and industries. A considerable amount of research in this field can help in building more new branches of the speaker recognition system. Speaker recognition has a great capacity of becoming a great tool of interaction between human and machine in future, Key challenges for this model in future includes well-organized methods for representation from word hypothesis, use of multiple word pronunciations. Speaker recognition is a data -driven field and involves more emphasis on realistic data. Voice recognition finds a bright future in making virtual assistance, as this assistance

All rights reserved by *www.ijsrd.com*

influenced our daily life as we can use our voice to communicate with our home appliances. So, there are many aspects of the development of a voice recognition system. Since the demand for voice recognition has been developed progressively over the last decades and applied in many new applications. Text preprocessing and pronunciation need much a more improvement in making more natural speaker recognition system.[12]

Hence, as the voice recognition system gains immense popularity, on the other hands there is still many research and improvement yet to be developed in the future to make it very accurate and generating the best branches of recognition system in the future

REFERENCES

[1] Dalei Wu, Andrew Morris, Jacques Koreman, MLP International Representation ad Discriminative Features for Improved Speaker Recognition, 2005.
[2] Unnikrishnan V M , Rajeev Rajan, "Mimicking Voice Recognition Using MFCC-GMM Framework", International Conference on Trends in Electronics and Informatics ICEI, pp.301-304, 2017
[3] Rania Chakroun, Leila BeltafaZouari, MondherFrikha, Ahmed Ben Hamida, "Improving text-independent speaker recognition with GMM", 2nd International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), pp.693-696, 2016.
[4] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740-741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
[5] ZufengWengLin Li, DonghuiGuo, "Speaker Recognition Using Weighted Dynamic MFCC Based on GMM", Anti-Counterfeiting Security and Identification in Communication (ASID), pp.285-288, 2010.
[6] Fang-YieLeu, Guan-Liang Lin, "An MFCC-based Speaker Identification System", IEEE 31st International Conference on Advanced Information Networking and Applications, pp.1055-1062, 2017.
[7] Hansen, John & Hasan, Taufiq. (2015). Speaker Recognition by Machines and Humans: A tutorial review. Signal Processing Magazine, IEEE. 32. 74-99. 10.1109/MSP.2015.2462851.
[8] Sinith, M.S., Salim, A., Gowri Sankar, K., Sandeep Narayanan, K.V. Soman, V., "A novel method for TextIndependent speaker identification using MFCC and GMM", , 2010 International Conference, Nov. 2010, pp.292-296
[9] Magre, Smita & Janse, Pooja & Deshmukh, Ratnadeep. (2014). A Review on Feature Extraction and Noise Reduction Technique.
[10] Tomi Kinnunen., and Haizhou Li., An overview of TextIndependent Speaker Recognition: from Features to Supervectors. Speech Communication, July 1, 2009.
[11] Reynolds, D. "Speaker Verification Using Adapted Gaussian Mixture Models." Digital Signal Processing 10.13 (2000): 19-41. Print.
[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, Vol. 28 No. 4, pp. 357-366

All rights reserved by *www.ijsrd.com*