Speaker Recognition Using MFCC and Vector Quantisation

Geeta Nijhawan [1], Dr. M.K Soni [2]
Faculty of Engineering and Technology, Manav Rachna International University, Faridabad, India
E-mail: geeta.fet@mriu.edu.in , ed.fet@mriu.edu.in

***Abstract*— Real time speaker recognition is needed for various voice controlled applications. Background noise influences the overall efficiency of speaker recognition system and is still considered as one of the most challenging issue in Speaker Recognition System (SRS). In this paper MFCC feature is used along with VQLBG (Vector Quantisation-Linde, Buzo, and Gray) algorithm for designing SRS. Voice Activity Detector (VAD) has been used which discriminates between silence and voice activity and significantly improves the performance of SRS under noisy conditions. MFCC feature is extracted from the input speech and then vector quantization of the extracted MFCC features is done using VQLBG algorithm. Speaker identification is done by comparing the features of a newly recorded voice with the database under a specific threshold using Euclidean distance approach. The entire processing is done using MATLAB tool.**

The experimental results show that the proposed method gives good performance for limited speaker database.

***Index Terms*— Hindi, Mel frequency cepstral coefficients, voice activity detector, MATLAB, Vector Quantization, LBG Algorithm**

I. INTRODUCTION

Speaker recognition is the process of recognizing the speaker from the database based on characteristics in the speech wave. Most of the speaker recognition systems contain two phases.

In the first phase feature extraction is done. The unique features from the voice signal are extracted which are used latter for identifying the speaker. The second phase is feature matching in which we compare the extracted voice features with the database of known speakers. The overall efficiency of the system depends on how efficiently the features of the voice are extracted and the procedures used to compare the real time voice sample features with the database.

A general block diagram of speaker recognition system is shown in Fig 1[1].

It is clear from the above diagram that the speaker recognition is a 1: N match where one unknown speaker's extracted features are matched to all the templates in the reference model for finding the closest match. The speaker feature with maximum similarity is selected [2].

Research and development on speaker recognition methods and techniques has been undertaken for more than five decades and it is still an active area. Classical speaker models can be categorized into template models and stochastic models, respectively. Vector quantization and dynamic time warping (DTW) are examples of template models for text-independent and text-dependent recognition, respectively. In stochastic models, each speaker is modeled as a probabilistic source with an unknown but fixed probability density function. In the training phase we estimate the parameters of the probability density function from the training data. The Gaussian mixture model (GMM) and the hidden Markov model (HMM) are the most

DOI: 01.IJRTET.11.1.1541

popular stochastic models. Speaker models can also be classified into generative and discriminative models. The generative models such as GMM and VQ estimate the feature distribution within each speaker independently[14]. The discriminative models such as artificial neural networks (ANNs) and support vector machines (SVMs) model the boundary between speakers.

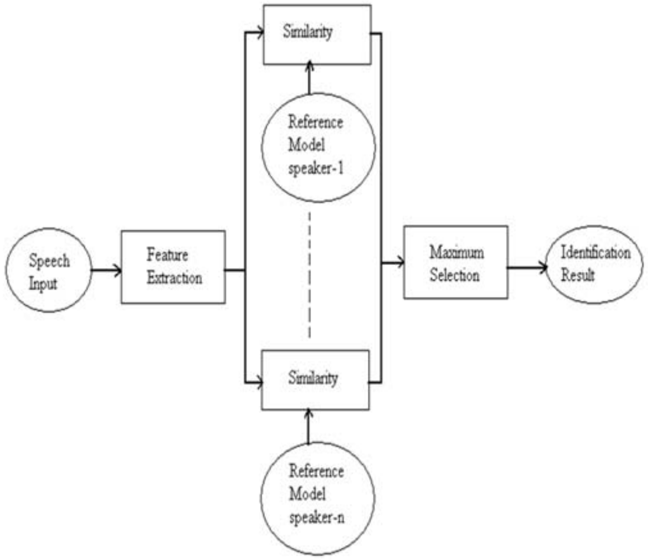


Fig.1 Speaker Recognition System

Vector quantization (VQ) is a lossy data compression method based on the principle of block coding. It is a fixed-to-fixed length algorithm. In 1980, Linde, Buzo, and Gray (LBG) proposed a VQ design algorithm based on a training sequence. In 1985, Soong et al. used the LBG algorithm for generating speaker-based vector quantization (VQ) codebooks for speaker recognition.VQ is often used for computational speed-up techniques. It also provides competitive accuracy when combined with background model adaption (Kinnuen et al.2009). Zhong-Xuan Yuan presented a new approach to vector quantization in which feature vector is represented by a binary vector called binary quantization (BQ). The results gives very good performance in terms of memory space and computation required. Also the identification system had shown strong robustness in additive white Gaussian noise. We have used VQ approach in our work as it is easy to implement and gives accurate results [3].

The remainder of this paper is organized as follows. In section II, MFCC used for feature extraction from the input voice is presented in detail. Section III gives the details about the vector quantisation method used for feature matching. Section IV gives the experimental results and finally in section V conclusions are drawn.

II. Feature Extraction

Speaker recognition is the process of automatically recognizing who is speaking on the basis of individual information included in speech waves**.** Speaker recognition system consists of two important phases. The first phase is training phase in which a database is created which acts as a reference for the second phase of testing. Testing phase consists of recognizing a particular speaker.

Speaker recognition systems contain three main modules [4]:

(1) Acoustic processing
(2) Features extraction or spectral analysis
(3)Feature matching

These processes are explained in detail in subsequent sections.

A. Acoustic Processing

Acoustic processing is sequence of processes that receives analog signal from a speaker and convert it into digital signal for digital processing. Human speech frequency usually lies in between 300Hz-8000kHz [5].Therefore 16kHz sampling size can be chosen for recording which is twice the frequency of the original signal and follows the Nyquist rule of sampling [6].The start and end detection of isolated signal is a straight forward process which detect abrupt changes in the signal through a given threshold energy. The result of acoustic processing would be discrete time voice signal which contains meaningful information. The signal is then fed into spectral analyser for feature extraction.

B. Feature Extraction

Feature Extraction module provides the acoustic feature vectors used to characterize the spectral properties of the time varying speech signal such that its output eases the work of recognition stage.

Voice Activity Detector

Voice Activity Detector (VAD) [7] has been used to primarily distinguish speech signal from silence. VAD compares the extracted features from the input speech signal with some predefined threshold. Voice activity exists if the measured feature values exceed the threshold limit, otherwise silence is assumed to be present. Block diagram of the basic voice activity detector used in this work is shown in Fig. 2.

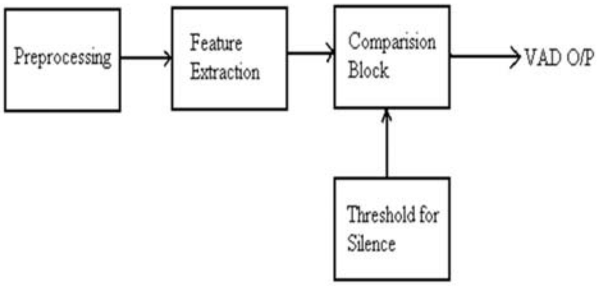


Fig.2 VAD block diagram

The performance of the VAD depends heavily on the preset values of the threshold for detection of voice activity. The VAD proposed here works well when the energy of the speech signal is higher than the background noise and the background noise is relatively stationary. The amplitude of the speech signal samples are compared with the threshold value which is being decided by analyzing the performance of the system under different noisy environments.

MFCC Extraction

Mel frequency cepstral coefficients (MFCC) is probably the best known and most widely used for both speech and speaker recognition [8]. A mel is a unit of measure based on human ear's perceived frequency. The mel scale is approximately linear frequency spacing below 1000Hz and a logarithmic spacing above 1000Hz. The approximation of mel from frequency can be expressed as

$$mel(f) = 2595*\log(1+f/700) \qquad \text{--------} \quad (1)$$

where f denotes the real frequency and mel(f) denotes the perceived frequency. The block diagram showing the computation of MFCC is shown in Fig. 3.

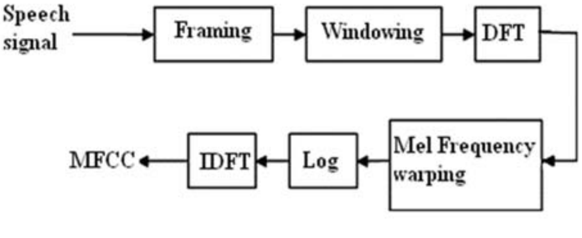


Fig.3 MFCC Extraction

In the first stage speech signal is divided into frames with the length of 20 to 40 ms and an overlap of 50% to 75%. In the second stage windowing of each frame with some window function is done to minimize the discontinuities of the signal by tapering the beginning and end of each frame to zero. In time domain window is point wise multiplication of the framed signal and the window function. A good window function has a narrow main lobe and low side lobe levels in their transfer function. In our work hamming window is used to perform windowing function. In third stage DFT block converts each frame from time domain to frequency domain [9],[10]. In the next stage mel frequency warping is done to transfer the real frequency scale to human perceived frequency scale called the mel-frequency scale. The new scale spaces linearly below 1000Hz and logarithmically above 1000Hz. The mel frequency warping is normally realized by triangular filter banks with the center frequency of the filter normally evenly spaced on the frequency axis [Fig 4]. The warped axis is implemented according to Equation 1 so as to mimic the human ears perception. The output of the ith filter is given by-

$$y(i) = \sum_{j=1}^{N} s(j)\Omega_i(j) \qquad \text{------ (2)}$$

S(j) is the N-point magnitude spectrum (j =1:N) and Ωi(j) is the sampled magnitude response of an M-channel filter bank (i =1:M). In the fifth stage Log of the filter bank output is computed and finally DCT (Discrete Cosine Transform) is computed. The MFCC may be calculated using the equation-

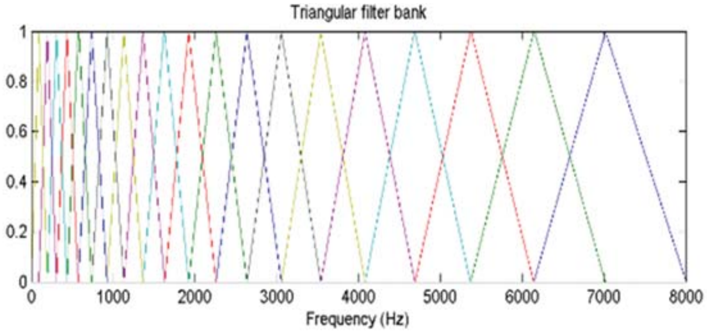


Fig.4 Triangular filter bank

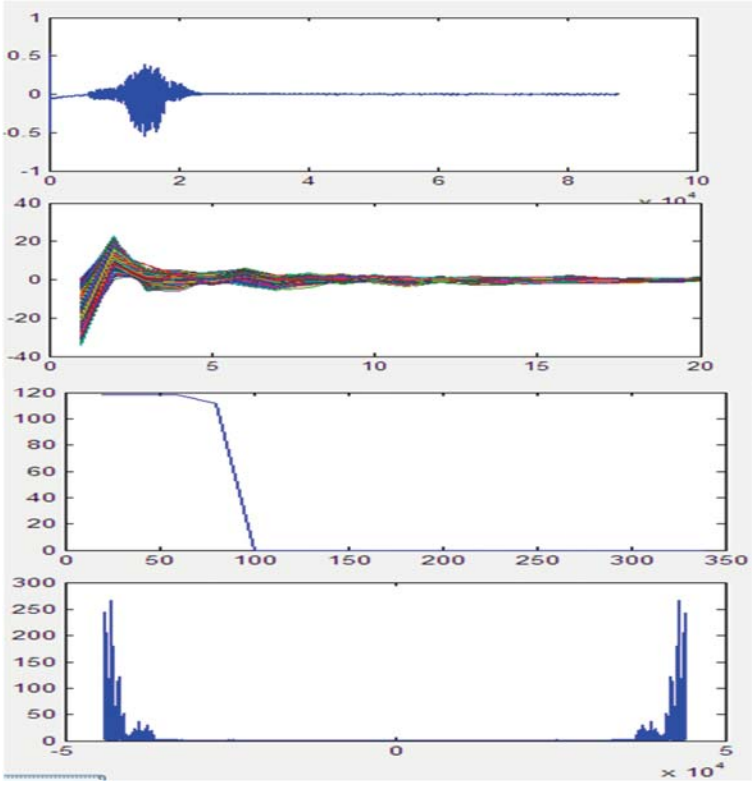


Fig. 5 GUI waveforms showing input speech, MFCC, pitch and power plots.

$$C_s(n,m) = \sum_{i=1}^{M} (\log Y(i)) \cos[i \frac{2\pi}{N'} n] \quad \text{--- (3)}$$

where N' is the number of points used to compute standard DFT.
Fig.5 shows screen shot of GUI developed using MATLAB of the input speech, MFCC, pitch and power plots.

III. Feature Matching

VQ is a process of mapping vectors from a large vector space to a finite number of regions in that space [11], [13]. Each region is called a *cluster* and can be represented by its center called a *codeword.* The collection of all code words is called a *codebook.*

Fig. 6 shows a conceptual diagram to illustrate this recognition process where only two speakers and two dimensions of the acoustic space are shown [16], [18]. The circles are the acoustic vectors from the speaker 1 whereas the triangle refers to speaker 2. In the training phase, a *speaker-specific* VQ codebook is generated for each known speaker by clustering his training acoustic vectors. The resultant codewords which are called centroids are shown in Fig. 5 by black circles and black triangles for speaker 1 and 2, respectively.The distance from a vector to the closest codeword of a codebook is called a VQ-distortion. In the recognition phase, an input utterance of an unknown voice is "vector-quantized" using each trained codebook and the *total VQ distortion* is computed [12]. A sequence of feature vectors {x_1, x_2,….,x_n}for unknown speakers are extracted.

Then each feature vector of the input is compared with all the codebooks. The codebook with the least average distance is chosen to be the best. The formula used to calculate the Euclidean distance can be defined as follows:

Let us take two points P = (p_1, p_2…p_n) and Q= (q_1, q_2...q_n). The Euclidean distance between them is given by-

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \quad \text{-------- (4)}$$

The speaker corresponding to the VQ codebook with smallest total distortion is identified as the speaker of the input speech.

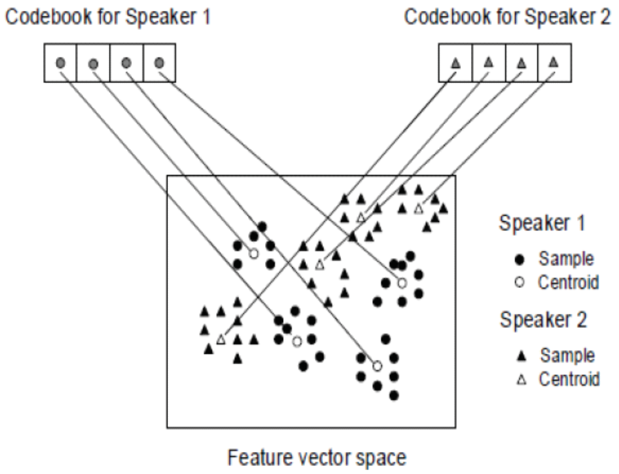


Fig. 6. Conceptual diagram showing vector quantization codebook formation.

One speaker can be discriminated from another based of the location of centroids (Adapted from Song et al., 1987)[15].

A. Clustering the Training Vectors

After the enrolment session, the acoustic vectors extracted from input speech of each speaker provide a set of training vectors for that speaker. Then a speaker-specific VQ codebook is build for each speaker using those training vectors. LBG algorithm [Linde, Buzo and Gray, 1980], for clustering a set of *L* training vectors into a set of *M* codebook vectors is being used. The LBG algorithm designs an *M*-vector codebook in stages. It starts first by designing a 1-vector codebook, then uses a splitting technique on the codewords to initialize the search for a 2-vector codebook, and continues the splitting process until the desired *M*-vector codebook is obtained.

The algorithm is implemented by the following recursive procedure [13], [17]:

1. Design a 1-vector codebook; this is the centroid of the entire set of training vectors (hence, no iteration is required here).
2. Double the size of the codebook by splitting each current codebook $\mathbf{y}_n$ according to the rule

$$y_n^+ = y_n(1+ \in)$$

$$y_n^- = y_n(1- \in)$$

where *n* varies from 1 to the current size of the codebook, and ε is a splitting parameter (we choose ε =0.01).

3. Nearest-Neighbor Search: for each training vector, find the codeword in the current codebook that is closest (in terms of similarity measurement), and assign that vector to the corresponding cell (associated with the closest codeword).
4. Centroid Update:update the codeword in each cell using the centroid of the training vectors assigned to that cell.
5. Iteration 1: repeat steps 3 and 4 until the average distance falls below a preset threshold
6. Iteration 2: repeat steps 2, 3 and 4 until a codebook size of *M* is designed.

Fig. 7 [18] gives the steps of the LBG algorithm. "*Cluster vectors*" represents the nearest-neighbor search procedure which assigns each training vector to a cluster associated with the closest codeword. "*Find centroids*" is the centroid update procedure. "*Compute D (distortion)*" sums the distances of all training vectors in the nearest-neighbor search so as to determine whether the procedure has converged.

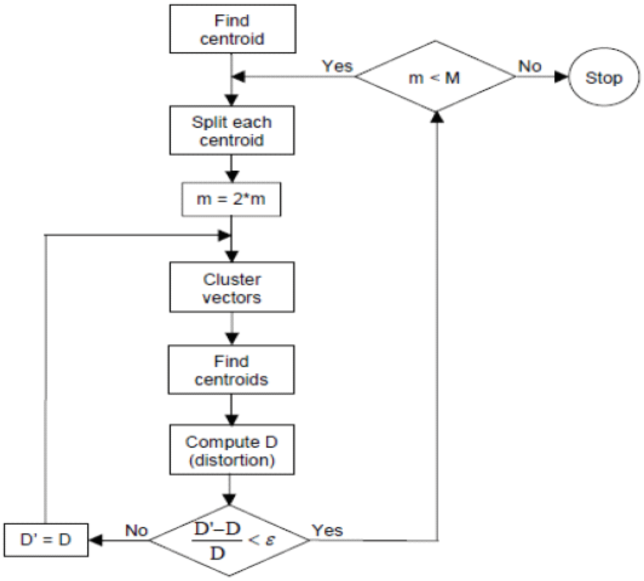
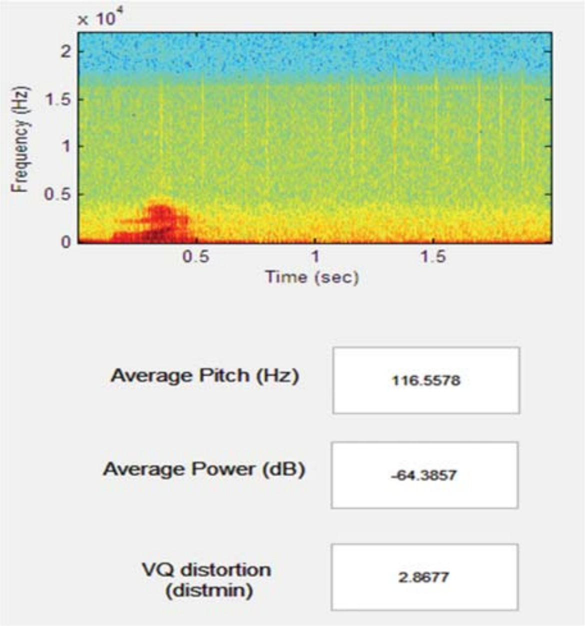


Fig 7. Flow diagram of the LBG algorithm (Adapted from Rabiner and Juang, 1993)

The experimental results are shown in Table 1 from which it can be seen that the diagonal element has the minimum VQ distance value in their respective row. It indicates that S1 matches with S1, S2 matches with S2 and so on. The designed system identifies the speaker on the basis of the smallest Euclidean distance compared to the codebooks in the database.

Table 2 shows the effect of changing the number of centroids on the identification rate of the system. It can be concluded from Fig.9 that increasing the number of centroids definitely improves the identification rate but it comes at the expense of increasing computation time.



Average Pitch (Hz) 116.5578

Average Power (dB) -64.3857

VQ distortion (distmin) 2.8677

Fig 8. Spectrogram, average pitch, average power and VQ distortion

TABLE I EXPERIMENTAL RESULT FOR SPEAKER RECOGNITION SYSTEM

	S1	S2	S3	S4	S5
S1	5.6223	5.8883	5.9480	5.8332	9.1132
S2	5.7665	5.4645	6.0609	5.5782	8.8568
S3	6.2807	6.1911	5.3604	5.8881	9.7971
S4	6.4374	6.1138	6.1304	5.4638	9.8151
S5	5.4051	5.3719	4.8583	4.9942	4.7008

TABLE II SYSTEM IDENTIFICATION RATE WITH NUMBER OF CENTROIDS

Number of centroids	Identification rate (%)
8	92
16	94
64	95
256	95. 5

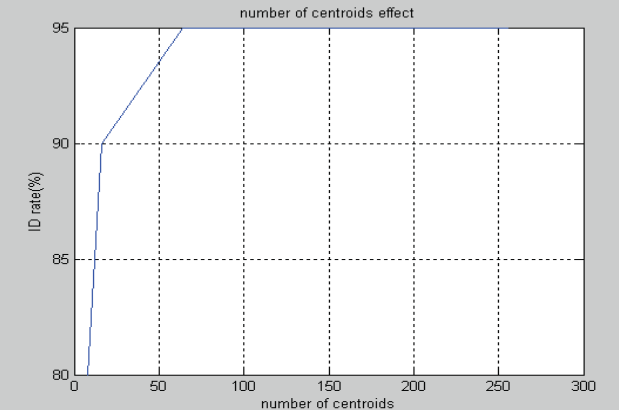


Fig 9. Number of centroids vs. Identification rate

V. CONCLUSIONS

The recognition rate obtained in this work using VAD, MFCC and LBG-VQ is good. Experiments shows that the VAD approach on an average gives a 5% error rate reduction compared to simply using a speech-free segment from the beginning of the utterance for noise modeling.

The VQ distortion between the resultant codebook and MFCCs of an unknown speaker is used for the speaker recognition. MFCCs are used because they mimic the human ear's response to the sound signals. Experimental results presented shows that the % accuracy of recognition is around 95% and there is no false recognition which shows the robust performance of the proposed design approach. Analysis of the % accuracy for recognition with codebook size shows that the performance of the proposed system increases with increase in number of centroids. However VQ has certain limitations and its efficiency deteriorates when the database size is large and hence HMM techniques or Neural Network technique can be used to improve the performance and to increase the accuracy.

REFERENCES

[1] Ch.Srinivasa Kumar, Dr. P. Mallikarjuna Rao, 2011, "*Design of an Automatic Speaker Recognition System using MFCC, Vector Quantization and LBG Algorithm''*, International Journal on Computer Science and Engineering,Vol. 3 No. 8 ,pp:2942-2954.

[2] Amruta Anantrao Malode,Shashikant Sahare,2012 **,** "*Advanced Speaker Recognition*", International Journal of Advances in Engineering & Technology ,Vol. 4, Issue 1, pp. 443-455.

[3] A.Srinivasan, "*Speaker Identification and verification using Vector Quantization and Mel frequency Cepstral Coefficients*",Research Journal of Applied Sciences,Engineering and Technology 4(I):33-40,2012.

[4] Vibha Tiwari, "MFCC and its applications in speaker recognition",International Journal on Emerging Technologies1(I):19-22(2010)

[5] Md. Rashidul Hasan,Mustafa Jamil,Md. Golam Rabbani Md Saifur Rahman, "*Speaker Identification using Mel Frequency Cepstral coefficients",3*rd International Conference on Electrical & Computer Engineering,ICECE 2004,28-30 December 2004,Dhaka ,Bangladesh

[6] Fu Zhonghua; Zhao Rongchun; "*An overview of modeling technology of speaker recognition*", IEEE Proceedings of the International Conference on Neural Networks and Signal Processing Volume 2, Page(s):887 – 891, Dec. 2003.

[7] Dr. Mahesh S. Chavan ,Mrs. Sharada V. Chougule, " *Speaker Features And Recognition Techniques: A Review*" ,International Journal Of Computational Engineering,May-June 2012 , Vol. 2 , Issue No.3 ,720-728, ISSN: 2250–3005

[8] S. Furui, "*Speaker-independent isolated word recognition using dynamic features of speech spectrum,"* IEEE Trans. Acoust., Speech, Signal Process., vol. ASSP-34, pp. 52-9, Feb. 1986.

[9] Sasaoki Furui, "*Cepstral analysis technique for automatic speaker verification*," IEEE Trans. Acoust., Speech, Signal Process., vol. 29(2), pp. 254-72, Apr. 1981.

[10] D.A. Reynolds, "*Experimental evaluation of features for robust speaker identification*," IEEE Trans. Speech Audio Process., vol. 2(4), pp. 639-43, Oct. 1994.

[11] B. Yegnanarayana, K. Sharat Reddy, and S.P. Kishore, "*Source and system features for speaker recognition using AANN models,*" in proc. Int. Conf. Acoust., Speech, Signal Process., Utah, USA, Apr. 2001.

[12] C.S. Gupta, "*Significance of source features for speaker recognition*," Master's thesis, Indian Institute of Technology Madras, Dept. of Computer Science and Engg., Chennai, India, 2003

[13] Y. Linde, A. Buzo, and R.M. Gray, "*An algorithm for vector quantizer design*," IEEE Trans. Communications, vol. COM-28(1), pp. 84-96, Jan. 1980.

[14] R. Gray, "*Vector quantization*," IEEE Acoust., Speech, Signal Process. Mag., vol. 1, pp. 4-29, Apr. 1984.

[15] F.K. Soong, A.E. Rosenberg, L.R. Rabiner, and B.H. Juang, "*A Vector quantization approach to speaker recognition*," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., vol. 10, Detroit, Michingon, Apr. 1985, pp. 387-90.

[16] T. Matsui, and S. Furui, "*Comparison of text-independent speaker recognition methods using VQ-distortion and Discrete/continuous HMMs,*" IEEE Trans. Speech Audio Process., vol. 2(3), pp. 456-9, July 1994.

[17] DSP Mini-Project: An Automatic Speaker Recognition System ,http://www.ifp.uiuc.edu/~minhdo/teaching/ speaker _recognition

[18] Voice Recognition using DSP:http://azhar paperpresentation.blogspot.in/2010/04/voice recognition-using-dsp.html