# *Design and Implementation of an Automatic Speaker Recognition System using neural and fuzzy logic in Matlab*

Shivam Jain
B.Tech , ECE
VIT University, Vellore
Tamil Nadu, India
nshivamj@gmail.com

Preeti Jha
B.Tech , ECE
VIT University, Vellore
Tamil Nadu, India
preetijha25 @gmail.com

Suresh. R
Faculty, SENSE
VIT University, Vellore
Tamil Nadu, India
rsuresh@vit.ac.in

***Abstract*—Recognition of a particular speaker amongst a crowded group of people is one of the current areas of interest of researchers. This paper presents a Text Dependent Automatic Speaker recognition system developed and simulated using Matlab. For a better computational efficiency, the system is trained to store voice of the same person under various physiological conditions such as coughing, shouting, during chewing, mouth covered etc. A dictionary is created to store the signature features of each user's voice. A neural networks is then trained using back propagation and accordingly weights are obtained to recognize voice in the testing phase. The efficiency of the proposed system is then compared to the system implemented using vector quantization.**

***Keywords*—Text dependent, vector quantization, fuzzy logic, speaker recognition, Matlab.**

## I. INTRODUCTION

Speaker recognition; also known as voice recognition is the identification of the person who is speaking by characteristics of his/her voice. Automatic speaker recognition system uses intelligent algorithm equipped machine to recognize a person by a spoken phrase[1,2,3]. The system operates in two modes to recognize a particular person and to verify the person's claimed identity viz.

(i) Text dependent
(ii) Text independent.

When the same text is used for both enrollment and verification this is called text-dependent recognition[1]. In a text-dependent system, prompts can either be unique or common across all speakers (e.g.: a common pass phrase). In addition, the use of shared-secrets (e.g.: passwords and PINs).Text-independent systems are most often used for speaker identification as they require very little or no co-operation by the speakers. In this case the text during enrollment and test may be different. All the Recognition systems have two different phases viz.

(i) Training phase.
(ii)Testing phase.

In the training phase, each registered speaker has to provide a "sample" speech so as to build or train a reference model for that speaker. We recorded 1 to 2 second long code word as "open the door" at frequency of 8000Hz which gives us more than 4000 samples depending on user for each training of voice for every user.

Speaker verification system, the testing phase is the process of determining whether the speaker identity is same as who the person claims to be. Speaker verification process, in addition, computes a speaker-specific threshold from the training samples. In the testing phase the input speech is matched with stored reference model(s) and recognition decision is made.

To improve the efficiency of the recognition system, a set of five voice inputs from the same person under different conditions is stored and tested. The rest of paper is organized as followed.[1] In Section II, training and testing phase is discussed. In Section III, the results obtained from MATLAB are shown. Section IV, compares the results obtained using fuzzy logic with that from the vector quantization technique. Section V concludes the paper with the analysis of result and discussion about related future work and hardware implementation.

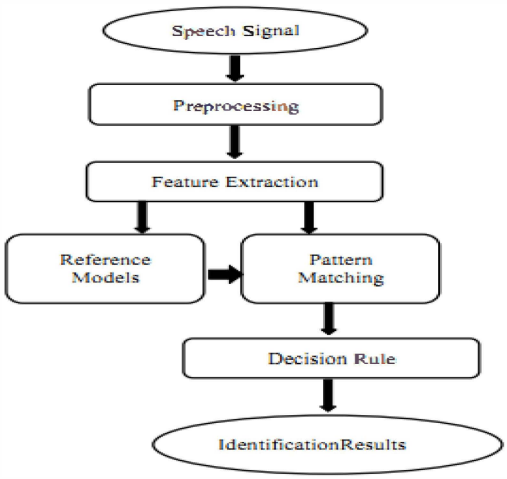



Fig 1.General Diagram for Speaker identification System[4]

## II. TRAINING AND TESTING PHASE

The speaker recognition system starts with first Feature extraction for training as well as for testing phase and then Feature matching for testing phase. In Feature extraction a small amount of data is extracted from the voice signal which is used to represent the speaker. In Feature identification of the unknown speaker is done by comparing extracted features from his/her voice input with the ones from the registered set of known speakers.

### I.TRAINING PHASE

### *A. Preprocessing*

Before the feature extraction, preprocessing of voice signal of each speaker is done by the following techniques:

a. Silence and Noise Removal:
In recorded signals, some portions do not contain any information that is they are the silence areas. Hence it is necessary to remove these silence part[5] so as decrease the unnecessary data and automatic de-noising method of 1-D signal using wavelets[2] is used to make it more efficient.

b. Frame blocking:
The continuous speech signal is divided into frames of N samples, with overlapping frames of N-M samples. The value M should be 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. We have chosen the values of M and N to be N = 256 and M = 100 respectively. The length of frames should be in powers of 2 so that FFT and DFT is easier.

c. Windowing:
After frame formation, each frame is windowed to minimize the discontinuities of signal at the beginning and end of each frame. A function that is constant inside the interval and zero elsewhere is called a rectangular window, which describes the shape of its graphical representation. When any signal (data) is multiplied by a window function, the product is also zero-valued outside the interval; all that is left is the part where they overlap. In this case we use the hamming window for windowing. The equation for hamming window is:

$$W(n)= 0.54-0.46\cos(2\pi n/(N-1)), \qquad 0<n<N-1$$

d. Fourier Transform:
Analysis of signals is easier in frequency transform hence the signal is converted to frequency domain by DFT (discrete Fourier transforms) or by FFT (fast Fourier Transform) techniques and it's energy spectrum is viewed.

### *B. Feature Extraction*

After the preprocessing of signal, the features of voice signal are extracted using technique MFCC (Mel Frequency Cepstrum coefficients), and then stored in the codebook[6]. The code book contains the features of voice signals from all the speakers. MFCC is based on human perceptions which cannot detect the frequencies above 1 KHz. It is based on known variation of the human ear's critical bandwidth with frequency. MFCCs are computed by using a bank of triangular-shaped filters, with the center frequency of the filter spaced linearly for frequencies less than 1000 Hz and logarithmically above 1000 Hz as low frequencies containing more information compared to high frequencies. The bandwidth of each filter is determined by the center frequencies of the two adjacent filters and is dependent on the frequency range of the filter bank and number of filters chosen for design. But for the human auditory system it is estimate that the filters should have a bandwidth that is related to the center frequency of the filter.

### *C. Codebook Formation*

The extracted vocal features of speakers are stored in the form of codebook. The features are stored in form of clusters. The similar featured clusters reside nearby in the codebook. Each

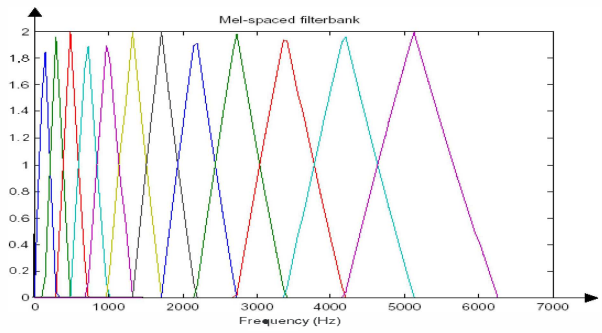



Fig 2. Filter bank of MFCC in linear frequency scale [6]

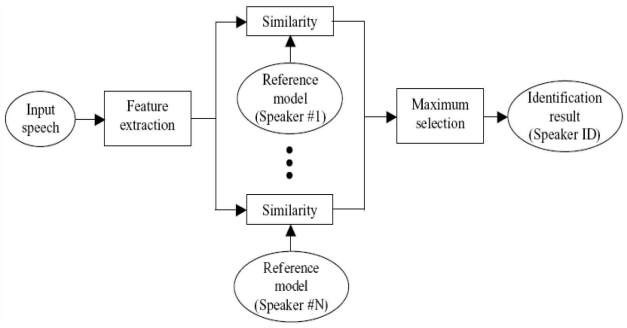



Fig 3.Block diagram of training phase[7].

## II.TESTING PHASE

For testing, same procedure is followed till the formation of codebook. Using neural logic and vector quantization technique, the detection of fed input is done as follows :

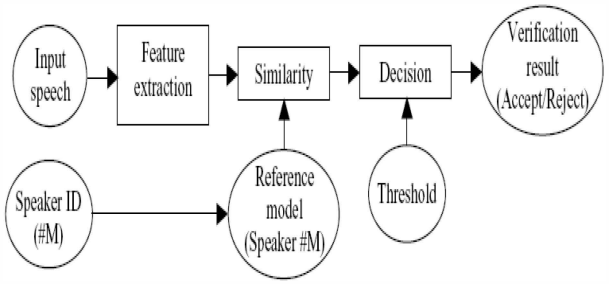



Fig 4. Block diagram of testing phase[7]

### A. *Vector Quantization*

It is a process of mapping vectors from a large vector space to a finite number of regions in that space. Each region is called a cluster and can be represented by its centroid which is stored in the codebook using either k-means[8] or Linde–Buzo–Gray algorithm[9]. Thus then both the codebooks vectors (obtained in both testing and training phase) is compared by computing euclidean distance[10] and with threshold value identity of user is known. The main task of this technique is pattern recognition of acoustic vectors that contains the identity of the user.

### B. *Neural Logic*

The features extracted after MFCC from the training phase are used in a neural network. The neural network is able to represent complex models that form the non-linear hypothesis[11,12]. The feed forward propagation algorithm is implemented to compute all the activations throughout the network, including the output value of the hypothesis using the initial random weights for prediction. Then back propagation algorithm[11] for learning the neural network parameter was applied to compute an “error term" that measures how much that node was “responsible" for any errors in output. The neural network cost function and gradient

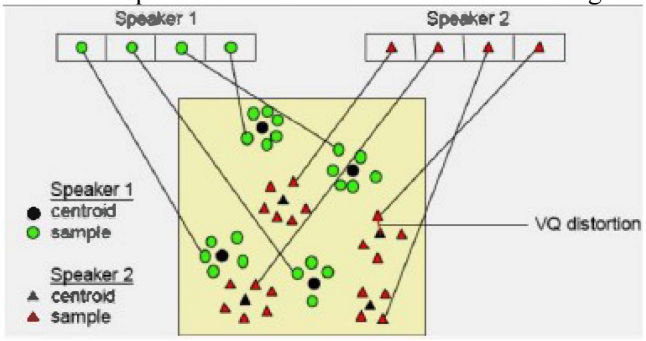



Fig 5. Vector Quantization distance calculation.

computation was implemented and were able to train the neural network by minimizing the cost function J(Θ) using an advanced optimizer.

The neural network used in this paper is shown in Figure 6. It has 3 layers namely, the input layer, the hidden layer and the output layer. The parameters have dimensions that are sized for a neural network with 40 units in the second layer.

## III. RESULTS FROM MATLAB

After the simulation of the matlab code following are the graphs obtained at training phase. In this case the system was trained using six users with each unique ids. Depicted in Fig8.

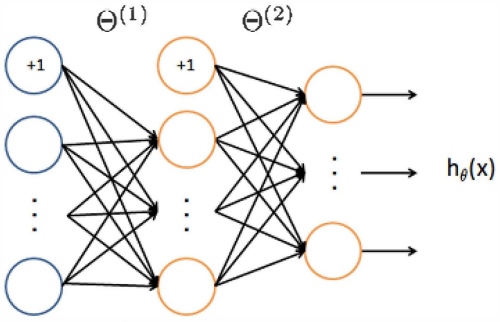
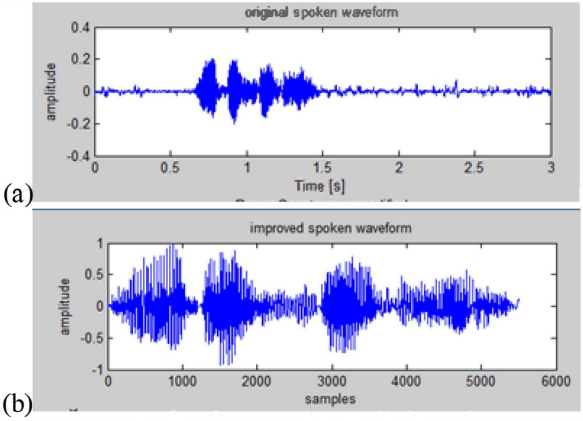
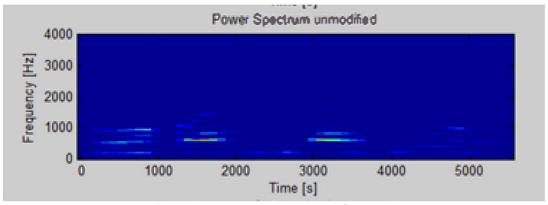


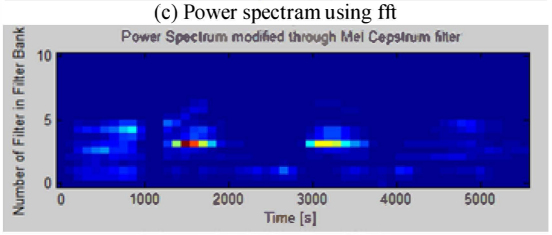

Fig 6. Neural network pattern




(a) original (b) after removing noise and silence.




(c) Power spectram using fft




(d) Power spectram after MFCC

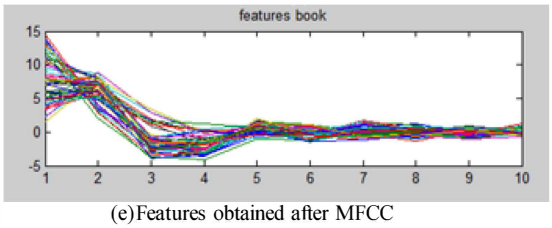
The similar graphs are obtained for all the users who stored their voices and similarly in the testing phase. The final result obtained after simulation is given in Fig 8.

## IV. COMPARISON OF RESULTS OBTAINED FROM FUZZY LOGIC WITH THAT OF VECTOR QUANTIZATION

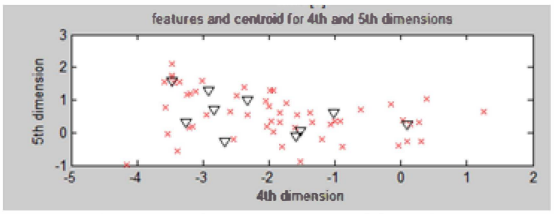
All The results obtained from Neural Logic were found to be far more accurate than that of vector quantization method. Also the time taken by the system using vector quantization is more as compared to Fuzzy Logic because dictionary size increases as number of user increases for vector quantization whereas in neural logic method weights size always remain the same. The following graphs compare the accuracy of both the techniques. Whenever the difference between the distances calculated for different users are less and similar in Vector Quantization, the probability of getting an erroneous output is more while no such problem is found when implemented using the Neural Logic.(Fig 9 and fig.10) Some other results that were obtained after simulation as –

In this simulation our system was tested two times with speaker whose voice is not stored so as to detect no match according to the threshold of minimum distance 3 in vector quantization and accuracy of more than 70% in neural logic method. In The first testing phase voice sample of 'sam1.wav' was provided and both the method vector quantization and neural logic detected 'No match' as a success but in second test phase with voice file of 'sam2.wav' was provided in which only neural logic method became a success ( fig 11, 12,13).

Similarly in other tests neural logic gave same accuracy and result with same voice sample where vector quantization method failed (Fig.14 and fig.15)




(e)Features obtained after MFCC




(f)Visualizing Features and Codebook centroids.

Fig7. Voice waveform of user id 1 (a,b,c,d,e,f)

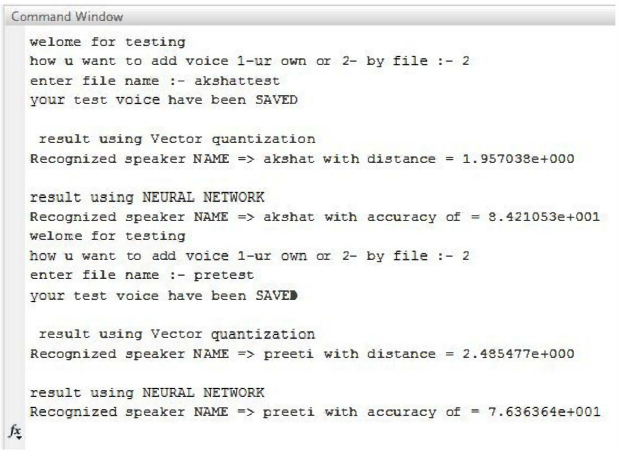



Fig 8. Final Result

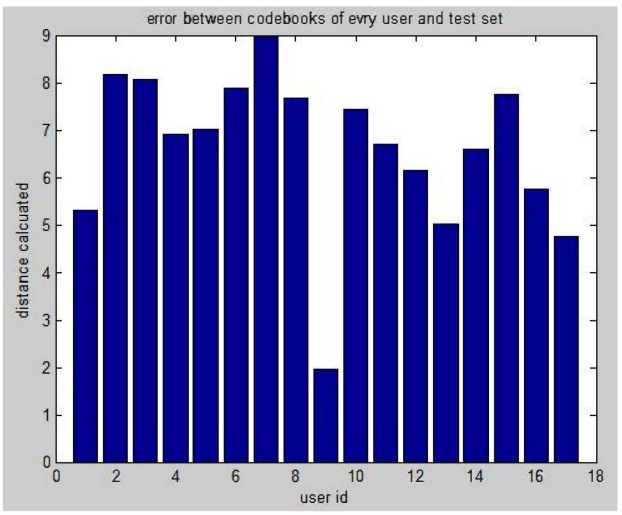



Fig 9.Output obtained from Vector Quantization of Akshat

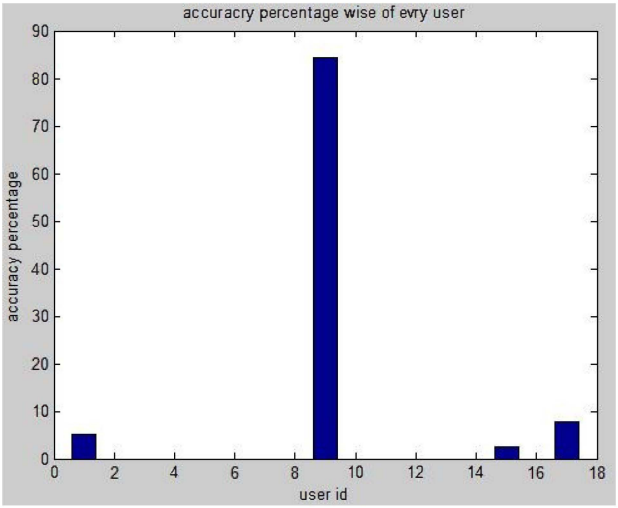



Fig10. Output obtained from Neural Logic

## V. CONCLUSION AND FUTURE WORK

Although many recent advances and successes in speaker recognition have been achieved, there are still many problems for which proper solutions remain to be found. Most of these problems arise due to variability, including speaker-generated variability and variability in channel and recording conditions. It is very important to investigate feature parameters that are stable over time, insensitive to the variation of speaking manner, including the speaking rate and level and robust against variations in voice quality due to causes such as voice disguise or colds. The speaker recognition system developed using Neural Logic has overcome the above problem of variability in speech of speaker. Hence, the algorithm along with the required hardware can be used for security purposes in high security areas.

```
Command Window
welome for testing
how u want to add voice 1-ur own or 2- by file :- 2
enter file name :- sam1
your test voice have been SAVED

 result using Vector quantization
!! NO MATCH !!

result using NEURAL NETWORK
!! NO MATCH !!
welome for testing
how u want to add voice 1-ur own or 2- by file :- 2
enter file name :- sam2
your test voice have been SAVED

 result using Vector quantization
Recognized speaker NAME => shivam id no.- 1 with distance = 2.952856e+000

result using NEURAL NETWORK
!! NO MATCH !!
```

Fig.11 Result obtained from non user voice samples

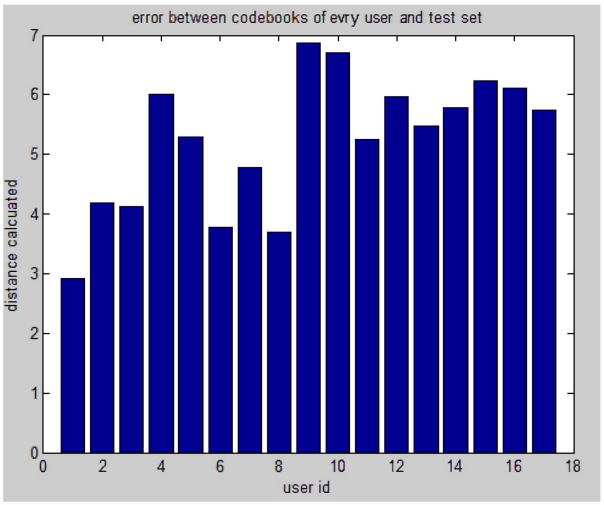



Fig.12 Output from Vector Quantization

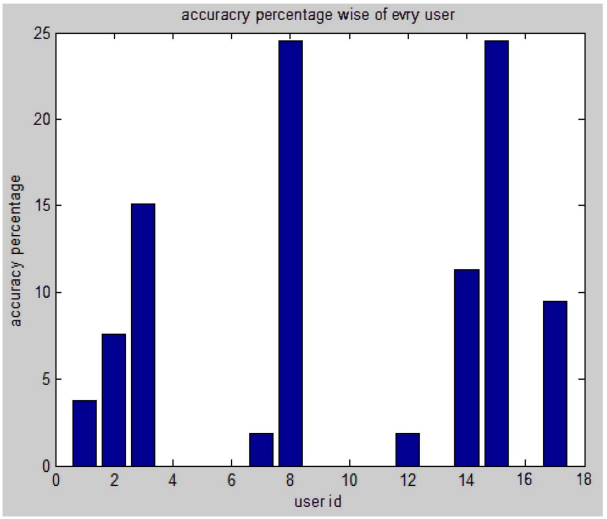



Fig 13. output from neural showing accuracy of less than 25 % hence no match.

```
welome for testing
how u want to add voice 1-ur own or 2- by file :- 2
enter file name :- shivamtest
your test voice have been SAVED

 result using Vector quantization
Recognized speaker NAME => anuj id no.- 6 with distance = 2.736005e+000

result using NEURAL NETWORK
Recognized speaker NAME => shivam id no.- 1 with accuracy of = 9.545455e+001
welome for testing
how u want to add voice 1-ur own or 2- by file :- 2
enter file name :- shivamtest
your test voice have been SAVED

 result using Vector quantization
Recognized speaker NAME => shivam id no.- 1 with distance = 2.972259e+000

result using NEURAL NETWORK
Recognized speaker NAME => shiva  id no.- 1 with accuracy of = 9.545455e+001
```

Fig.14 Different Result obtained from same user voice samples

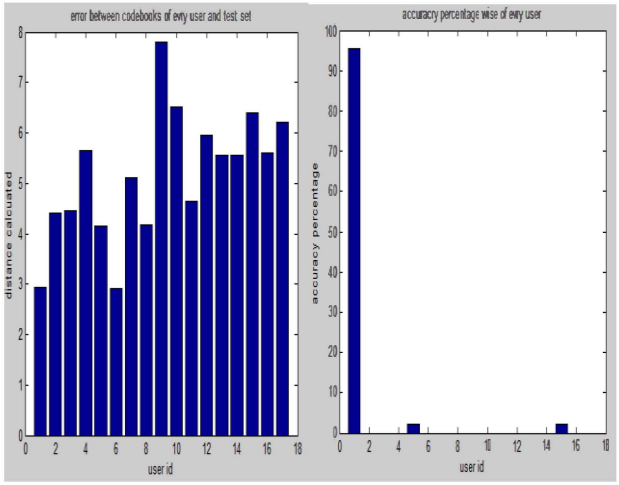



Fig.15 Output from Vector Quantization and neural


## ACKNOWLEDGMENT

We would like to thank Dr. G. Ramachandra Reddy, Dean, SENSE, Prof. P. Arulmozhivarman, Program Chair, B.Tech, ECE and Prof Suresh R for their kind guidance and support in compilation of the paper.


## REFERENCES


[1] Joseph P. Campbell, "Speaker recognition: A tutorial", Proc. IEEE, vol 85, September 1997.

[2] Adrian E. Villanueva- Luna, Alberto Jaramillo -Nuñez, Daniel Sanchez-Lucero, Carlos M. Ortiz-Lima, J. Gabriel, Aguilar-Soto, Aaron Flores-Gil and Manuel May-Alarcon (2011), "De-Noising Audio Signals Using MATLAB Wavelets Toolbox", Engineering Education and Research Using MATLAB, Dr. Ali Assi (Ed.), ISBN: 978-953-307-656-0, InTech.

[3] D. A. Reynolds, "An Overview of Automatic Speaker Recognition Technologyî", Proc. IEEE, pp. 4072-4075, 2002.

[4] Mahdi Keshavarz Bahaghighat, Farshid Sahba, Ehsan Tehrani, "Text-dependent Speaker Recognition by combination of LBG VQ and DTW for persian language", International journal of Computer applications (0975 – 8887)volume 51– no.16, August 2012.

[5] G. Saha, Sandipan Chakroborty and Suman Senapati, "A New Silence Removal and Endpoint Detection Algortihm for Speech and Speaker Recognition Applications," January 2005.

[6] Darshan Mandalia, Pravin Gareta ,Rachna Sharma, "Speaker Recognition Using MFCC, Vector Quantization Model", India, 2011.

[7] Adarsh K.P., A. R. Deepak, Diwakar R., Karthik R. **"**Implementation of a Voice-Based Biometric System",2007.

[8] Adam Coates, AndrewY. Ng, "Learning Feature Representations with K-means".

[9] Linde, Y.; Buzo, A.; Gray, R. (1980). "An Algorithm for Vector Quantizer Design". IEEE Transactions on Communications **28**: 84.

[10] B. Planerer ,"An Introduction to speech recognition", March 28,2005.

[11] Ryszard Tadeusiewcz, "Sieci neuronowe", Kraków , 1992

[12] R. Rojas, "Neural Networks", Berlin , 1996