Volume 5, Issue 4, 2015 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Speech Enhancement Using PCA for Speech and Emotion Recognition

Pooja R. Gaikwad, Manjushree B. Aithal, Shashikant L. Sahare
E & TC, Pune University, Pune,
Maharashtra, India

Abstract- This paper deals with application of speech and emotion recognition using distorted speech signal. When speech signal is used as input to any system some amount of background noise always gets added. To overcome this problem we transform the signal using PCA and recognition is done using the Hidden Markov Models.

Our system is able to recognize speech and emotion by extracting the MFCCs and then transformed using PCA to obtain eigenvalues. The eigenvalues with highest values contain dominant information which are retained and others are discarded as noise.

Hidden Markov Models is most efficient method used for speech and emotion recognition.

Keywords: MFCC, Kernel PCA, Hidden Markov Models, Speech Recognition, Emotion Recognition.

I. INTRODUCTION

Speech has been used in every domain of technology. It is much more natural way of interfacing system other than keyboard such as in car systems, military, telephony and other domains, people with disabilities, hand free computing, robotics, etc. The task of Speech Recognition involves mapping of speech signal to phonemes, words. It can also be called as “Speech to Text” system. It could be text dependent or independent. The problem in speech recognition is large variation in the signal characteristics. Speech recognition is strongly influencing the communication between human and machines. Hidden /Markov Models are popularly used for Speech Recognition. The other methods used are Dynamic Time Warping (DTW), Neural Networks, and Deep Neural Networks.

Emotion recognition [9] is an emerging area of research and development. The voice interactive systems can adapt as per the detected input emotion. This could lead to more realistic interactions between system and the user. From the statistics it is seen that pitch contains considerable information about emotions. Generally prosody features contain pitch, intensity, and durations. The algorithms implemented for emotion recognition are using DCT (Discrete Cosine Transform), using two-level wavelet packet decomposition, using four-level wavelet packet decomposition, K-Nearest Neighbor (KNN). Several problems arise while developing this system

Hands free systems use interactive speech as input. But it fails when noise gets added to speech signal. To overcome this number of methods has been proposed for speech enhancement which aims to improve performance of speech based systems.

Principal component analysis (PCA) is crucial method used in modern signal processing- a block that is widely used. Principal component analysis uses the applied linear algebra and is used in all forms of analysis- from neuroscience to de-noising. It is a simple, non parametric method for eliminating the redundant data from the available data (mostly noisy data in de-noising). Without any additional complication this method reduces the data resulting into lower dimensional data i.e. noise free parameter.

In this paper we proposed a method for de-noising for the speech and emotion recognition using Principal component analysis (PCA). In the real time applications, the additive (or other form of noise) is the crucial enemy for the recognition system rendering the efficiency of the system completely. Thus, the removal of this enemy has become a prime importance. The methods used by now for noise removal or speech enhancement fail for varying impulse response.

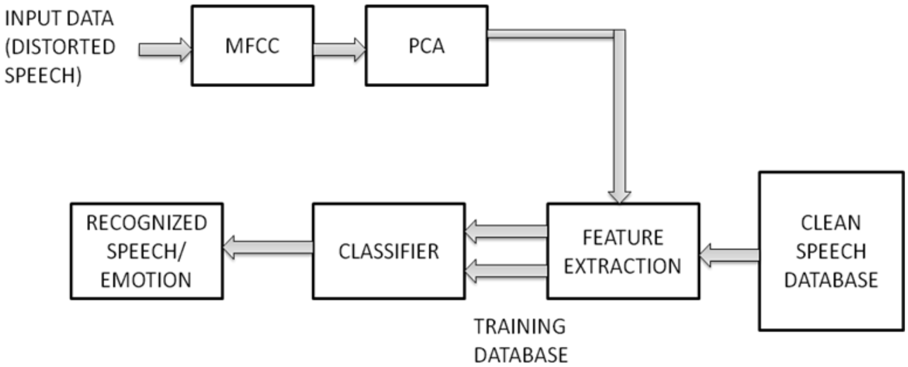

Figure 1: Block diagram of Proposed Model

To overcome this, a method is been adopted using the Principal Component Analysis. MFCC (Mel Frequency Cepstral Coefficients) is used as features since nowadays it has been widely considered; it is because the MFCCs imitate the human hearing band. Other features such as LPC (Linear Predictive Coefficients), Pitch period, first three Formants frequencies (F1, F2, and F3), first order and second order derivative of MFCCs can also be considered according to the requirements.

Speech is non-stationary and time varying signal. An assumption is made that the signal is stationary for short duration of time by framing the signal into short frames of 20 ms. They are then passed through Hamming window in order to avoid end effects. FFT of this signal is taken and then MFCC coefficients are calculated to obtain the features. The most commonly used feature extraction techniques are formants, pitch, Mel Frequency Cepstral Coefficients (MFCC), Linear Predictive Cepstral Coefficients (LPCC).

As the Mel scale filter bank imitates the human auditory system it is used to obtain the features. After obtaining these features they are transformed using PCA and then the ones with dominant values are selected to obtain clean speech signal. Thus, performing de-noising of the signal by discarding the eigenvalues containing noise. Then the retained eigenvalues are Vector Quantized in order to make them of fixed size. HMM are statistical models used for training and testing of the coefficients. Each word model will have sequence of codeword vectors that is states. Then maximum probability for word model is evaluated. Then the word with maximum likelihood is recognized. The maximum likelihood is calculated using Viterbi Decoding algorithm.

The performance of the system is evaluated using the SNR values. High SNR is desirable for accurate working of the system. Speech signal is recorded from 10 people and exhibition noise is added to it.

This paper is organized as follows: the section II) deals with MFCC. The section III) deals with Principal Component Analysis. The section IV) deals with Vector quantization and section V) deals with Hidden Markov Model for speech and emotion recognition. Lastly, section VI) shows experimental results and section VII) gives conclusion and future scope.

II. MEL FREQUENCY CEPSTRAL COEFFICIENTS (MFCC)[1]

The signal is then passed via Hamming window in order to avoid the end effects. If the signal is passed via rectangular window the abrupt truncations led to high frequency components in the signal which are undesirable. The equation for Hamming window is given as follows-

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{M-1}\right) \qquad (2.1)$$

Hamming window is generally selected over Hanning, Blackman, Barlett windows because they have highest stop band attenuation. Then features are extracted using MFCC. It allows better representation of frequency by approximating it to human auditory system. It takes linear cosine transform of log power spectrum on non linear scale of frequency. Mel scale is based on pitch perception. A mel is psychoacoustic unit of measure for the perceived pitch of a tone. It uses triangular windows with overlap of 50%.This scale is linear below 1000 Hz & non-linear above 1000 Hz.

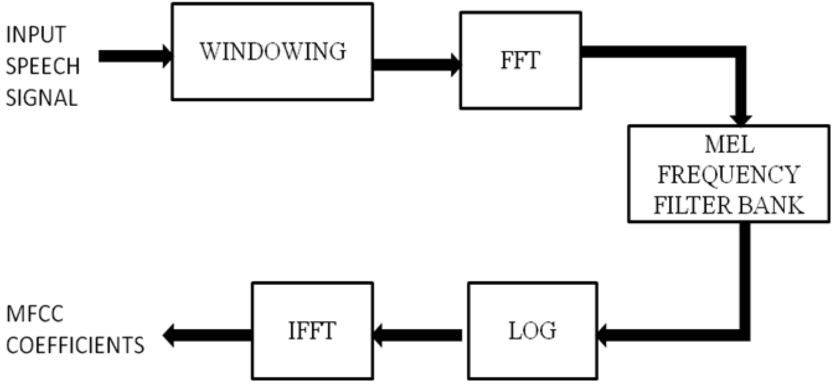

Figure 2: Block schematic for MFCC calculations

Relation between mel & linear frequencies is given as-

$$\text{mel frequency} = 2595 * \log\left(\frac{1+ linear\ frequency}{700}\right) \qquad (2.2)$$

Out of obtained MFCCs, first 20 coefficients are selected as the features. These features are then transformed using principal component analysis to remove the noise induced in the clean speech signal.

III. PRINCIPAL COMPONENT ANALYSIS([2],[3])

Principal component analysis (PCA) is often used as technique for data reduction/compression without any loss of information. It is a technique used to transform one set of variable into another smaller set, where the newly created variable is not easy to interpret. In several applications, PCA is used only to provide information on the true dimensionality of a data set. If the data set includes *M* variables, all *M* variables do not represent required information. PCA transforms a set of correlated variables into a new set of uncorrelated variables that are called *principal components.*

But if the data is already uncorrelated the PCA is of no use. Along with the uncorrelated data, the principal components are orthogonal and are ordered in terms of the variability they represent. That is, the first principal component represents, for a single dimension, the greatest amount of variability in the original data set. PCA can be applied to data sets containing any number of variables.

To decorrelate the variables, we need to rotate the variables data set until the data points are distributed symmetrically about the mean. In the decorrelated condition, the variance is maximally distributed along the orthogonal axes. It is also sometimes necessary to *center* the data by removing the mean before rotation. In statistical sense, if two variables are independent they will also be uncorrelated but reverse is not true. The rotation is so performed that the covariance (or correlation) goes to zero. A better way to achieve zero correlation is to use a technique from linear algebra that generates a rotation matrix that reduces the covariance to zero. One well known method is by pre- or post-multiplication with the orthonormal matrix:

$$\mathbf{U'CU= D} \qquad (3.1)$$

where, **C** is *m*-by-*m* covariance matrix,
D is a diagonal matrix, and
U is an orthogonal matrix that does transformation.

The covariance matrix is defined as by:

$$C = \begin{bmatrix} \sigma_{1,1} & \sigma_{1,2} \cdots & \sigma_{1,N} \\ \sigma_{2,1} & \sigma_{2,2} \cdots & \sigma_{2,N} \\ \vdots & \vdots & \vdots \\ \sigma_{N,1} & \sigma_{N,2} \cdots & \sigma_{N,N} \end{bmatrix} \qquad (3.2)$$

The diagonal elements of **D** are the variances of the new data, generally known as the characteristics roots, or eigenvalues of **C**:$\lambda_1, \lambda_2, \ldots, \lambda_n$. The eigenvalues of the new covariance matrix corresponds to the variances of the rotated variables. The eigenvalues can be obtained as:

$$det|C - \lambda I| = 0 \qquad (3.3)$$

where, **I** is the identity matrix. After obtainingλ, the eigenvectors are obtained as:

$$det|C - \lambda I|v_i = 0 \qquad (3.4)$$

where, the eigenvectors are obtained from v_i by the equation given below,

$$u_i = v_i/\sqrt{v_i' v_i} \qquad (3.5)$$

These eigenvectors are *Feature vector* which is multiplied with the input data due to which we obtain the new data set given as below:

Data adjust= Feature vector' x *Final data*

This is how PCA reduces redundant data. In case of speech signal the noise component embedded is inside the speech data due to which apparently the speech is de-noised.

IV. VECTOR QUANTIZATION[2]

Every frame of speech signal contains certain number of samples. These may vary from person to person depending upon the pronunciation and speed in which they are spoken. Either they may be very fast or very slow resulting in the variation in number of samples in input speech.

HMM has fixed number of states and to achieve this number of samples in each frame must be fixed. So vector quantization converts the MFCCs of variable length into fixed length codebook. The codebook contains coefficients of Vector Quantization.

For VQ, the LBG algorithm is used. The LBG algorithm steps are as follows:

1. Design a 1-vector codebook; this is the centroid of the entire set of training vectors.
2. Double the size of the codebook by splitting each current codebook y_n according to the formula given below:

$$y_n = y_n(1+\varepsilon) \qquad (4.1)$$

$$y_n = y_n(1-\varepsilon) \qquad (4.2)$$

Where n varies from 1 to current size of codebook and ε is the splitting parameter.

3. Nearest neighbor search: for each training vector, find the codeword in the current codebook that is closest and assign that vector to the corresponding cell.
4. Update the codeword in each cell using the centroid of the training vectors assigned to that cell.
5. Repeat steps 3&4 until the average distance falls below a present threshold.
6. Repeat steps 2, 3 &4 until a codebook size is designed.

The VQ algorithm gives a fixed size codebook. Suppose if it of size T, then it can be expressed mathematically as follows:

$$T = 2^i \quad i= 1, 2, 3,\ldots \qquad (4.3)$$

This algorithm gives codebook of fixed size.

V. HIDDEN MARKOV MODEL([1],[2],[3],[7])

HMM are been popularly used for the pattern classification. It consists of hidden states denoted by Q and observable output sequence, denoted by O. HMMs are modeled using following model parameters (A, B, π). The transition probabilities between the states are denoted by A, emission probabilities that generate the output sequence B, and the initial state probabilities π. The current state in HMM is not observable.

The parameter estimation can be defined. The total number of vectors falling in a group (cluster) representing a single state are counted. The ratio of the count to total number of vectors in a word give state probabilities. The number of times the transition from one group (cluster) to another is made gives us the transition probability. The emission probability is total number of times we get the output vector when word is in a group (cluster).

Forward and backward algorithms are used to evaluate the probability of particular output sequence O_t at time t.

A. *Forward Algorithm:*

For a particular phoneme there will be a set of output sequences appearing serially as time progresses. The output sequence is the sequence obtained from number of possible output sequences. They can be obtained by taking addition of for all the paths coming from a number different output sequences.

Let $\alpha_i(t)$ be the probability of observation sequence O_t = [o (1), o (2)… o (t)] to be produced by all observation

Initialization:

$$\alpha_1(i) = p_i b_i\big(o(1)\big) \qquad i = 1,2, \dots .N \qquad (5.1)$$

Recursion:

$$\alpha_{t+1} = [\textstyle\sum_{j=1}^{N} \alpha_t(j) a_{ji}] b_i(o(t+1)) \qquad (5.2)$$

Here i=1, 2...N t=1, 2…T-1

Termination:

$$P\big(o(1)o(2) \dots \dots ..o(T)\big) = \textstyle\sum_{j=1}^{N} \alpha_T(j) \qquad (5.3)$$

B. *Backward algorithm*

Initialization:

$$\beta_t(i) = P\left(o(t+1), o(t+2), \dots \dots ., o(T) | q(t) = q_i\right) \qquad (5.4)$$

$$\beta_t(i) = 1 \qquad i = 1,2..N \qquad (5.5)$$

Recursion:

$$\beta_t(i) = \textstyle\sum_{j=1}^{N} a_{ij} b_j(o(t+1)) \beta_{t+1}(j) \qquad (5.6)$$

Here i=1, 2 …N t=T-1, T-2…, 1

Termination:

$$\text{P (o (1) o (2)…..o (T))} = \textstyle\sum_{j=1}^{N} p_j b_j\big(o(1)\beta_1(j)\big) \qquad (5.7)$$

C. *Baum Welch Algorithms:*

To find the parameters (A, B, π) that maximize the likelihood of the observation Baum Welch Algorithm is used. The Baum Welch algorithm is an iterative expectation-maximization (EM) algorithm that converges to a locally optimal solution from the initialization values.

$\xi_t(i,j)$ can be defined as the joint probability of being in state q_i at time t and state q_j at time t+1, given the model and the observed sequence:

$$\xi_t(i,j) = P(q(t) = q_i, \text{q (t+1)} = q_j | O, \Lambda$$

$\xi_t(i,j)$ can also be expressed as follows,

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j\big(o(t+1)\big)\beta_{t+1}(j)}{P(O|\Lambda)} \qquad (5.8)$$

The probability of output sequence can be expressed as

$$\text{P (O | Λ)} = \textstyle\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j\big(o(t+1)\big)\beta_{t+1}(j) \qquad (5.9)$$

$$\text{P (O | Λ)} = \textstyle\sum_{i=1}^{N} \propto_t (i)\beta_t(j) \qquad (5.10)$$

The probability of being in state q_i at time t-

$$\gamma_t(i) = \textstyle\sum_{j=1}^{N} \xi_t(i,j) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\Lambda)} \qquad (5.11)$$

Estimates:

Initial probabilities:

$$p_i = \gamma_1(i) \qquad (5.12)$$

Transition probabilities:

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \qquad (5.13)$$

Emission probabilities:

$$b_{jk} = \frac{\sum_{t=1}^{*} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)} \qquad (5.14)$$

In the above equation * denotes the sum over t so that o (t) =o_k.

D. *Viterbi Decoding:*

From HMM parameters and observation sequence Viterbi decoding finds the most likely sequence of (hidden) states. Let $\delta_t(i)$ be the maximal probability of state sequence of length t that end in state I and produce the t first observations for given model.

$$\delta_t(j) = \max\{P(q(1), \dots \dots \dots, q(t-1); o(1), \dots, o(t) | q(t) = q_i\} \ (5.15)$$

The Viterbi algorithm uses the maximization at the recursion and termination steps. It keeps track of the arguments that maximize $\delta_t(j)$ for each t and I, storing them in the N by T matrix ψ.

This matrix is used to retrieve the optimal state sequence at the backtracking steps.

Initialization:

$$\delta_1(i) = p_i b_i(o(1)) \qquad (5.16)$$

$$\psi_1 = 0$$

For i= 1, …, N

Recursion:

$$\delta_t(j) = \max[\delta_{t-1}(i) a_{ij}] b_j\,(o(t)) \qquad (5.17)$$

$$\psi_t(j) = \arg\max[\delta_{t-1}(i) a_{ij}] \qquad (5.18)$$

For j= 1, …. , N

Termination:

$$p^* = max_i[\delta_T(i)] \qquad (5.19)$$

$$q_T^* = \arg max_i[\delta_T(i)] \qquad (5.20)$$

Path (state sequence) backtracking:

$$q_T^* = \psi_{t+1}(q_{t+1}^*) \qquad (5.21)$$

For t= T-1, T-2, …, 1

VI. EXPERIMENTAL RESULTS

Speech signal is recorded by *Wavesurfer* software at sampling frequency of 8 kHz, single line channel and then saved in .wav format. Database contains 5 speech signals recorded from 10 people. The 5dB exhibition noise was then added to this recorded speech signal.

MFCC of speech signal was computed and transformed using PCA to extract the dominant part of speech signal and hence remove the noise induced in the signal. We calculated accuracy from the result obtained after classification using formula as follows:

Accuracy= percentage of match/ total percentage

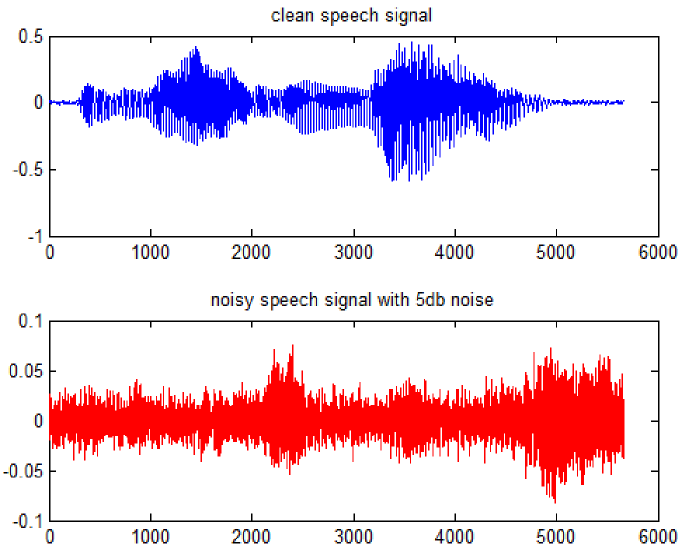

Figure 3: Plot of speech signal

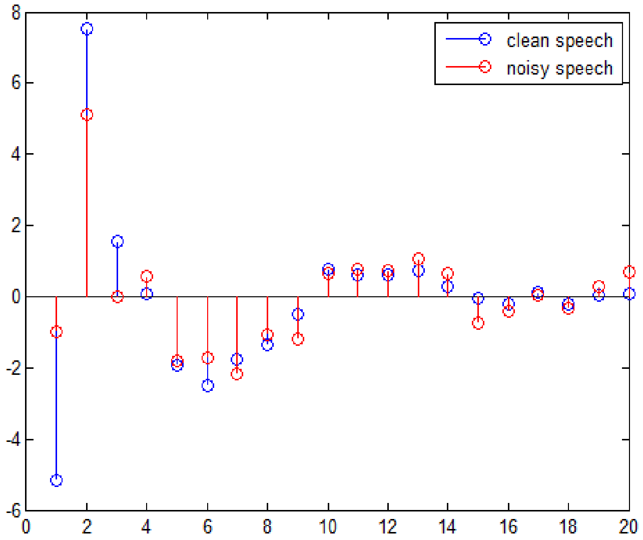

Figure 4: MFCC plot of clean and noisy speech signal

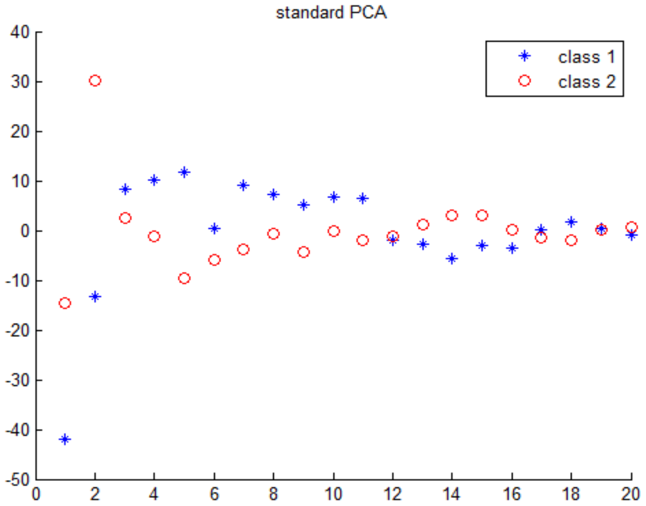

Figure 5: PCA plot

A) Application of PCA for Speech Recognition:

During the training phase all the speech samples from 10 individuals are used. Hidden Markov Model optimizes the model parameters and finds the corresponding best sequence by using the Viterbi algorithm.

Table I Without PCA DE-NOISING, ACCURACY=18%

	SENTENCE 1	SENTENCE 2	SENTENCCE 3	SENTENCE 4	SENTENCE 5
SENTENCE 1	**1**	2	2	3	2
SENTENCE 2	3	**3**	1	2	1
SENTENCE 3	4	3	**2**	1	-
SENTENCE 4	3	4	1	**1**	1
SENTENCE 5	3	4	-	1	**2**

Table II D=15, ACCURACY=49%, SNR=4.4305

	SENTENCE 1	SENTENCE 2	SENTENCCE 3	SENTENCE 4	SENTENCE 5
SENTENCE 1	**5**	-	-	2	4
SENTENCE 2	3	**4**	-	2	1
SENTENCE 3	3	-	**4**	1	2
SENTENCE 4	-	1	1	**5**	3
SENTENCE 5	2	-	-	2	**6**

Table III D=20, ACCURACY=44%, SNR=3.32133

	SENTENCE 1	SENTENCE 2	SENTENCCE 3	SENTENCE 4	SENTENCE 5
SENTENCE 1	**4**	**-**	-	3	3
SENTENCE 2	2	**4**	-	2	2
SENTENCE 3	2	1	**3**	2	2
SENTENCE 4	2	-	1	**5**	2
SENTENCE 5	-	-	-	4	**6**

Table IV D=25, ACCURACY= 40%, SNR = 1.6648

	SENTENCE 1	SENTENCE 2	SENTENCCE 3	SENTENCE 4	SENTENCE 5
SENTENCE 1	**4**	**-**	-	3	3
SENTENCE 2	2	**4**	-	2	2
SENTENCE 3	2	1	**3**	2	2
SENTENCE 4	2	-	1	**5**	2
SENTENCE 5	-	-	-	4	**6**

In the Testing phase, MFCC coefficients are obtained and then transformed using the PCA. It arranges the signal in form of descending eigen values of which the highest value is most significant and lowest is least significant.

If we chose the first 25 eigen values then the accuracy obtained is 40% whereas if 15 eigen values are selected then accuracy obtained is 49%. HMM finds out the parameters and observation sequence is obtained using Viterbi decoding. If it matches with the training sequence then the speech is recognized.

B) Application of PCA for Emotion Recognition:

The emotions been taken into account are happy, angry, neutral. Often, the angry emotion is confused with the happy emotion.

Table shows the accuracy rate of 69.95 % when 15 eigen values are selected, 57.61% for 20 and 49.38% for 25 eigen values.

Table V Without PCA DE-NOISING, ACCURACY=28.804%

	EMOTION 1	EMOTION 2	EMOTION 3
EMOTION 1	**3**	1	5
EMOTION 2	7	**-**	2
EMOTION 3	3	2	**4**

Table VI D=15, ACCURACY= 69.95%, SNR= 4.4305

	EMOTION 1	EMOTION 2	EMOTION 3
EMOTION 1	**4**	2	3
EMOTION 2	1	**6**	3
EMOTION 3	-	2	**7**

Table VII D=20, ACCURACY= 57.61%, SNR= 3.3213

	EMOTION 1	EMOTION 2	EMOTION 3
EMOTION 1	**3**	2	4
EMOTION 2	1	**5**	3
EMOTION 3	-	3	**6**

Table VIII D=25, ACCURACY= 49.38%, SNR= 1.6648

	EMOTION 1	EMOTION 2	EMOTION 3
EMOTION 1	**2**	2	5
EMOTION 2	1	**4**	4
EMOTION 3	1	2	**6**

VI. CONCLUSION

This paper proposes the idea that speech and emotion recognition can be done even if it is degraded by the background noise. More improvement can be done in this system by implementing KPCA instead of PCA. HMM models use Viterbi decoding to find the most likely state sequence for the recognition.

REFERENCES

[1] Dr.Shaila Apte, *Speech and Audio Processing*, Edition 2012 by Wiley India Pvt. Ltd.

[2] Lawrence Rabiner, Biing Hwang Juang, B.Yegnanarayana, *Fundamentals of Speech Recognition*, First Impression 2009 by Dorling Kindersley (India) Pvt. Ltd.

[3] V Susheela Devi, M Narasimha Murty, *Pattern Recognition-An Introduction*, 2013,Universities Press(India) Private Limited.

[4] Tetsuya Takiguchi, Yasuo Ariki, *PCA-Based Speech Enhancement for Distorted Speech Recognition*, Journal of Multimedia, Vol.2, No.5, September 2007.

[5] Jonathon Shlens, *A Tutorial on Principal Component Analysis*, Systems Neurobiology Laboratory, Salk Institute for Biological Studies La Jolla ,CA 92037 & Institute for Nonlinear Science, University of California, San Diego La Jolla, CA 92093-0402,December 10,2005;Version 2.

[6] Lindsay I Smith, *A tutorial on Principal Components Analysis*, February 26, 2002.

[7] Lawrence R. Rabiner, *A Tutorial on Hidden Markov Model and Selected Applications in Speech Recognition*, Proceedings of the IEEE, Vol No.77, No.2, February 1989.

[8] Shashikant L. Sahare , Amruta A. Malode , *An Improved Speaker Recognition by HMM*, Proceedings of the International Conference on Advances in Electronics, Electrical and Computer Science Engineering-EEC 2012.

[9] Ankur Sapra, Nikhil Panwar, Sohan Panwar , Jaypee Institute of Information Technology, Noida , *Emotion Recognition from Speech* , International Journal of Emerging Technology and Advanced Engineering , Volume 3, Issue 2, February 2013.

[10] Mélanie Fernández Pradier, Universität Stuttgart, *Emotion Recognition from Speech Signals and Perception of Music,* Thesis.