International Journal of Computing and Applications | **ISSN: 0973-5704**
Volume 13, Number 2, (July-December 2018), pp 247-255

FEATURE EXTRACTION TECHNIQUES FOR HINDI SPEECH RECOGNITION SYSTEM: A STUDY

Vishal Passricha[1] and Rajesh Kumar Aggarwal[2]

[1]*Assistant Professor, Computer Engineering Department, National Institute of Technology, Kurukshetra, India. Email: vishal_pasricha@yahoo.com*
[2]*Associate Professor, Computer Engineering Department, National Institute of Technology, Kurukshetra, India. Email: rka15969@gmail.com*

Abstract: In automatic speech recognition system (ASR), features are extracted at front-end using signal parameterization techniques and classified at back-end by classifiers. Selection of feature sets is a very critical task in ASR due to its high impact on the system performance. Ever since the use of filter bank approach in the domain of speech processing, a variety of short and long-term features have been proposed by researchers for the development of robust ASR system. This paper presents a comparative study of a few well-known feature extraction techniques in the context of Hindi language. Features are analyzed in both scenarios that are similar and mismatched training, and testing conditions. Further, the combinations of alternative and main feature extraction techniques are tested empirically to get the optimal results. Deep Belief nets and hidden Markov models are used for acoustic-phonetic modeling. Strength and weaknesses of various techniques are also analyzed with the experiments for medium size vocabulary.

Keywords: MFCC, PLP, Feature Extraction, HMM, DBNN, ASR.

1. INTRODUCTION

An automatic speech recognition (ASR) system is known as speech to test system that maps the speech signal into their corresponding text. The process of conversion should be independent from recording device (i.e., microphone), the speaker's accent, and the acoustic environment. The ultimate goal of ASR is to achieve accuracy like the human listener, which has not yet been achieved [1]. ASR offers a way for man-machine interaction. Generally, human and machine interact via keyboard, mouse etc. but humans can speak faster than typing. Speech offers ease of use to the user and permits them to do another task in parallel by hands. The understandability of speech is also more than text [2].

The ASR worked in two part: feature extraction then classification. The speech signal is converted into a discrete sequence of feature vectors, which is assumed to contain only that information about given utterance which is important for its correct recognition. In the last feature, vectors are changed into corresponding words [2].

In this paper, various feature extraction techniques are reviewed with their merits and demerits, and a comparative study is also presented for similar and mismatched training, and testing conditions. All the experiments were conducted on Hindi speech corpus. The paper is organized as follows: section 2 describes the working of ASR. In section 3, the brief summary of various feature extraction techniques is given. HMM details are explained in section 4. Deep belief neural networks are covered in section 5. In section 6, an experimental comparison of ASR performance with various feature extraction methods is presented. Finally, the paper is concluded in section 7.

2. WORKING OF ASR

Speech recognition process is completed in two steps as shown in Figure 1. In the first step, the signals are preprocessed and then features are extracted from the segregated speech. Preprocessing is mainly A/D conversion, background noise filtering, pre-emphasis, blocking and windowing. The sampling rate of 8-16 KHz is used for digitization of analog speech signal [3].

Mel-frequency cepstrum coefficient (MFCC), perceptual linear prediction (PLP), RASTA-PLP, and Wavelets etc. are various techniques that are used for feature extraction [4]. Hidden Markov models, support vector machines, DBNN etc. are various classifiers that are used at the back-end.

The ASR is a machine learning problem so ASR systems are trained and tested. During training, acoustic and language models are generated. These models work as sources of knowledge in classification. The acoustic model maps the feature vectors into words. Language model provides linguistic rules of the language. In classification, the testing patterns are compared with each reference pattern and their probability of likeness is computed with each reference pattern [5].

3. FEATURE EXTRACTION

The process in which relevant properties from the raw data are extracted is called feature extraction. The dimensionality of the speech signal is reduced up to 80:1, because the characteristics of the original speech are easily maintained by feature extractor and useless information are removed The main goal of feature extractions is to eliminate the extraneous information and select the pure information that distinguishes a given sub unit from another sub-unit [6].

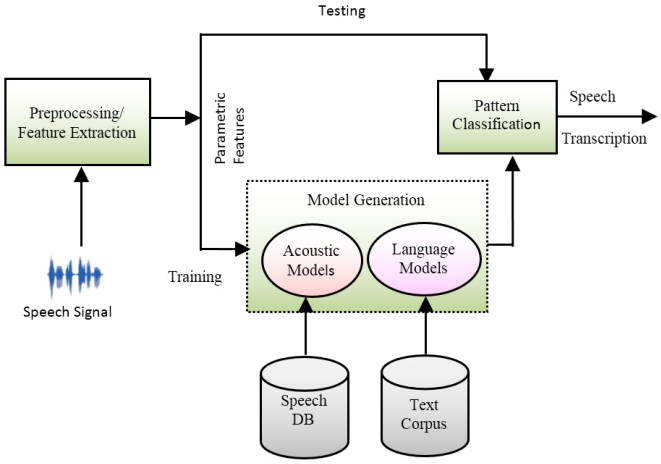

Figure 1: ASR Architecture

3.1. Mel Frequency Cepstrum Coefficient (MFCC)

The main steps to derive MFCC features are shown in Figure 2 [7]. The success rate of MFCCs is very good in speech recognition but they have the following three problems:

- It is not having any physical interpretation.
- Liftering of cepstral coefficients is not successful with continuous density hidden Markov models.
- They have insufficient sound representation capability especially at low SNR, i.e. not robust enough in noisy environments.

3.2. Perceptual Linear Prediction

The working of PLP is based on human auditory system hence speech spectrum is transformed. The steps used in the computation of PLP are shown in Figure 3 [8].

3.3. PLP Derived from Mel scale Filter Bank (MF-PLP)

In MF-PLP, both (MFCC and PLP) techniques are combined to make hybrid algorithm [9]. As a key modification is the bark filter bank is replaced by the Mel scale triangular filter bank. The first few steps of MFCC algorithm up to the output of the Mel scale triangular filter bank are taken and then the last steps that generate the cepstrum coefficients came from the PLP algorithm. It skips the copying of the outermost filter. Finally, the cepstral mean is normalized.

3.4. Gravity Centroid as Alternative Features

The sensitivity of MFCC features to additive noise and channel mismatch deteriorates the performance of ASR. To increase the accuracy of ASR, alternative features have been searched [10]. The performance of these features is not as the standard features so they are used in combination of standard features.

Spectral subband centroids and energy gravity centroids are alternative features that were proposed to combine with standard features. To compute the spectral subband centroids, the frequency band is divided among a fixed number of subbands and the centroid for each subband is computed using power spectrum of the speech signal as shown in Figure 4.

Gravity centroid features are calculated from the energy moments. First order movement gives an indication of location of the peak in a given sub-band and second order movement gives information about distribution around this peak.

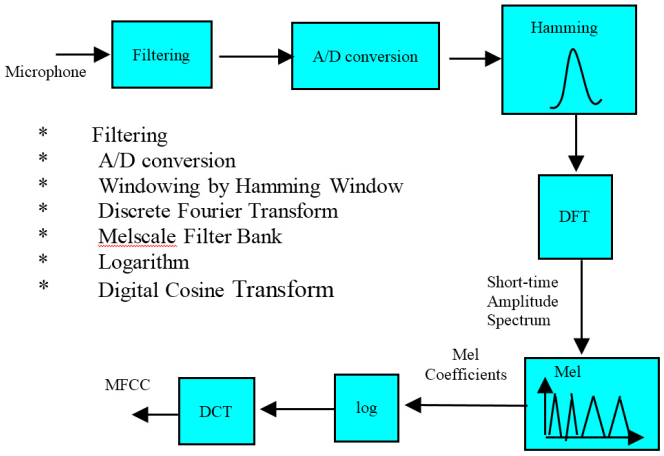

Figure 2: Steps of MFCC

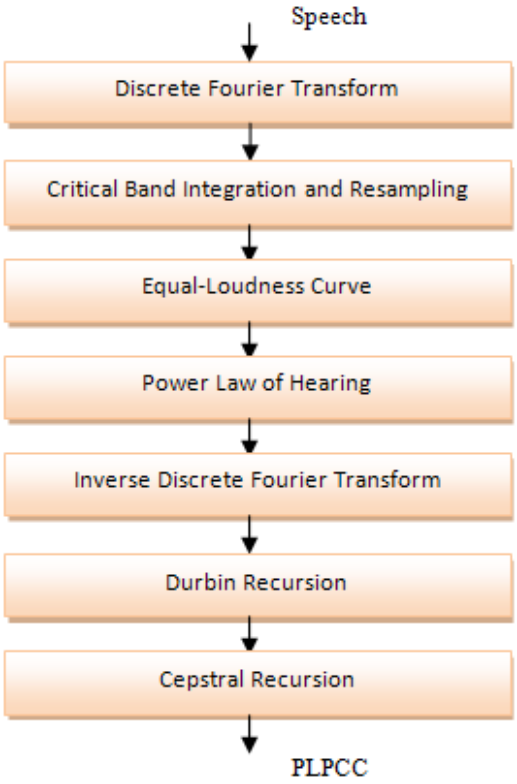

Figure 3: Phases of the PLP

3.7. TRAP

Much larger space for research is offered by the TRAP processing but its combination with the standard features is preferred to derive the complementary information of both techniques. Short term frames (about 20 to 35ms) being used for standard features are not sufficient to capture the significant discriminant information about the current phoneme [11, 12]. Second, the phonemes are not entirely separable in time, because they overlap due to the fluent working of speech producing organs from one configuration to other.

The vectors of posterior probabilities of sub-word acoustic events are obtained by segmenting the speech signals into 25ms frames having 15ms overlap. Fast Fourier transform is used to compute the spectrum of speech segment and the Bark scaled trapezoidal filters are used in filtering. By taking the logarithm of the filter's output, the vectors are converted into log-critical band spectrum. In the first half, processing is same as standard features. Second half of original speech signal is covered by 51 points TRAP vector [18]. After normalization, this vector forms an input to the most common choice for the classifier, a three-layered MLP classifier, feed-forward neural network structure.

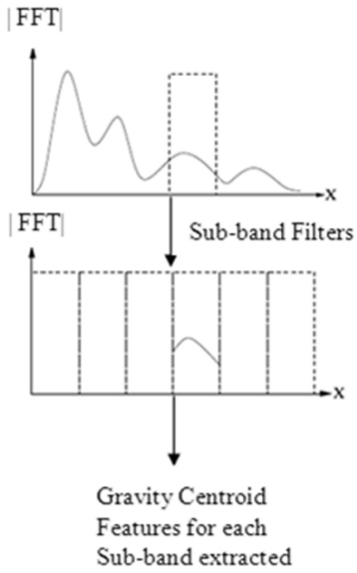

Figure 4: Gravity Centroid Alternative Features

MLP classifies the normalized TRAP vector by calculating posterior probabilities of sub word classes. The output units (posterior probabilities) of the MLP are determined by the phonetic inventory of a given language [13]. The MLP produces the vectors in a probability space which is not easily modeled

by a GMM; thus the probabilities are approximately Gaussianized by conversion to the logarithmic domain. The discrete cosine transform, principal component analysis, and neighbourhood component analysis are the classical examples, which are used to reduce the dimensionality [11]. The Karhunen-Loeve transform (KLT) is being applied to process the features which orthogonalizes the features to satisfy the typical diagonal covariance GMM assumption [14]. The technique like LDA may also be used in feature extraction. Finally, the resulting vector is concatenated with a standard feature vector, to serve as higher-dimensional observations for the acoustic models. The phases of TRAP is shown in Figure 5.

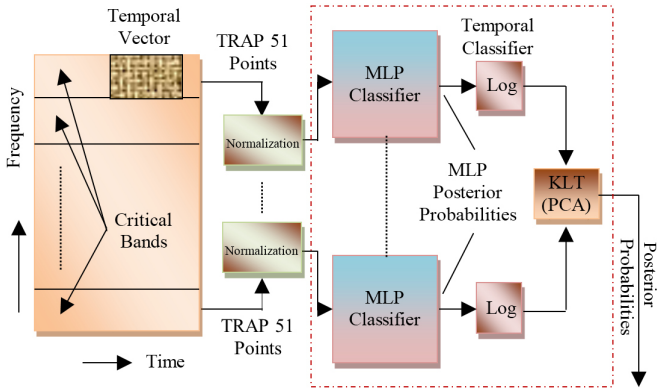

Figure 5: Phases of TRAP

3.8. RASTA

In noisy field conditions, RASTA effectively improves the accuracy of ASR. RASTA is equally effective on background noise and channel distortion [15]. The rate of change of the vocal tract shape affects the linguistic components of speech. The rate of change of nonlinguistic components in speech is independent from the rate of change of vocal tract shape. Modulation spectrum (temporal feature) describes the rate of change of short-time spectrum envelope and its maxima lies at 4Hz for a wide range of frequency bands. The spectral components that unordinary make the typical rate of change of speech are suppressed by the technique, the relative spectral (RASTA).

The RASTA filters are applied to suppress high modulation frequencies to account for the human's preference for signal change at a 4Hz rate. The frame rate is used to decide the sampling frequency of RASTA filter that is twice of the maximum modulation frequency. RASTA is usually used with PLP in combination and applies a band-pass filter to the energy in each frequency subband in order to smooth over short-term noise variations. Band pass filter's range is from 1Hz to 16Hz with sharp zero at 0Hz. The high pass filtering is used to eliminate the events changing more slowly and the low pass filtering gradually smooth parameters over about 40ms while suppressing noise. For noisy speech, the filtering effects try to do normalization to improve the results.

4. MIXTURE GAUSSIAN HMM

HMM is the most successful acoustic modeling technique. Its efficient algorithm for training and recognition make it power. HMM can efficiently model the stationary stochastic processes and the temporal relationship among the processes. This combination powers us to model dynamic speech signals using one reliable framework. Another attractive feature of HMM is very simple to train from a given set of labeled training data (one or more sequences of observations). The two training algorithms are Baum-Welch and segmental *k* means both results in well-formulated and well-behaved solutions. The main distinction between these two is that separate optimization procedure is used for them.

When the output symbols are associated with the states of the HMM then the model is known as state output HMM and when output symbols are associated with an edge then HMM model is known as edge-output HMM [16, 17]. The state output model is generally preferred over edge output model for speech recognition. A typical structure of a word based HMM is shown in Figure 6.

The role of acoustic modeling can be structured in a four-level hierarchy.

- Likelihood evaluation of spectral features at every HMM state.
- To find and manage the contextual phonetic variants (i.e. allophone, triphones, syllables) of the underline phoneme.

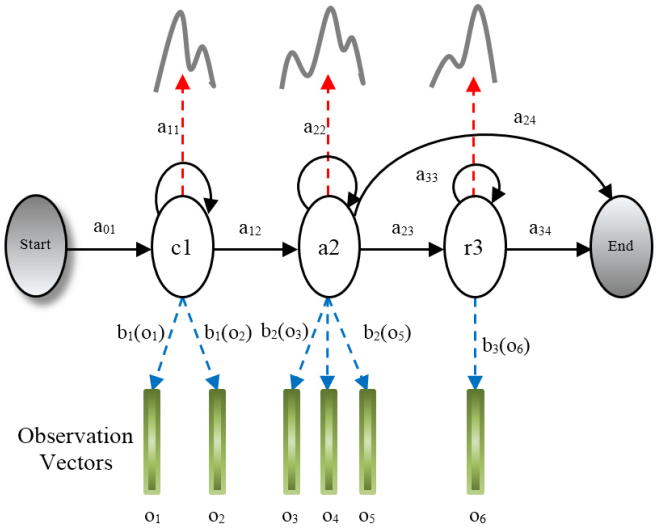

Figure 6: Block diagram of hidden Markov model

- Word composition using sub-word units (provided by HMMs) with the help of Lexicon (Pronunciation modeling).
- To generate the sequences of words or phrases up to the sentence level.

5. DEEP BELIEF NEURAL NETWORK

The process of learning is not easy in densely-connected, directed belief nets. Hinton [18] introduced multi-layer, densely connected nets named as Deep Belief Neural Network (DBNN). In DBNNs, the hidden layers are taken form an associated memory and have binary values called feature detectors. The two most significant properties of DBNNs are:

1. The variables in one layer depending on the variables in a layer above so for the learning, the layer by layer procedure is used to update the top-down weights.
2. Once the network is trained, the bottom-up pass that starts with an observed data vector allows to reproduce proper hidden unit states. The DBNNs are self-organizing in nature and must be trained layer by layer. An unsupervised training algorithm is preferred for training the DBNN because it can efficiently handle the connection weights that is equivalent to training each adjacent pair of layers as Restricted Boltzmann Machines (RBM). The fine-tuning of all weights can be performed as the same ways as in MLPs. During this phase, a supervised objective function can also be optimized. In RBM, the visible unit represents the input features and a layer of hidden units expresses the way to represent features, and to capture higher order dependencies in the data undirected weight connections are used. The building block for DBNNs is RBMs that have an effective training procedure which makes them suitable for deep learning. The block diagram of DBNN is shown in Figure 7.

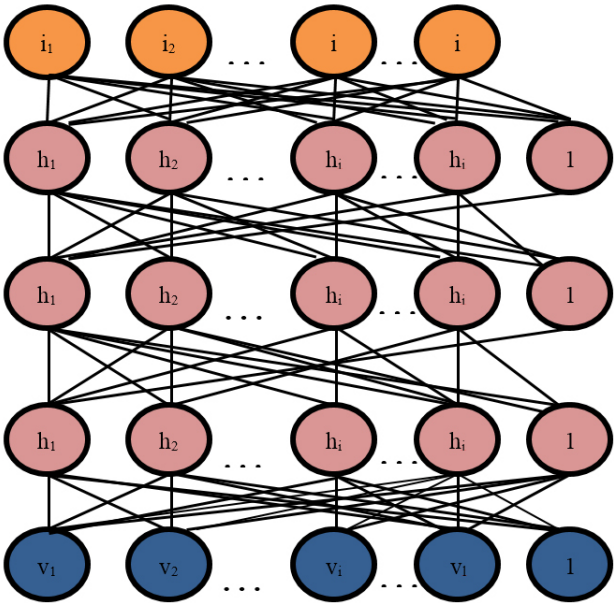

Figure 7: Block diagram of Deep Belief Network

5.1. Restricted Boltzmann Machines

An RBMs [18] is a type of bi-partite graph model (Markov Random Field) constructed from one layer of binary stochastic hidden units and one layer of stochastic visible units, that is generally Gaussian distributed conditional on the hidden units. All the visible units are connected to all the hidden units, with no visible-visible or hidden-hidden connections. The biases of the individual units and weights on the connections define a probability distribution over the binary state vector $\boldsymbol{v}$ of visible units and $\boldsymbol{h}$ of the hidden units via an energy function.

In a large RBM, exact minimum likelihood learning is infeasible because it is exponentially expensive to compute the derivative of the log probability of

training data. Nevertheless, contrastive divergence is an efficient approximate training process for RBMs which makes them suitable as building blocks for learning DBNNs [18]. The update rule for each weight w_{ij} uses the difference between two measured, pairwise correlations:

$$\Delta w_{ij} \propto \langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{reconstruction}} \quad (1)$$

where the angle brackets are used to denote expectations under the distribution specified by the subscript.

6. EXPERIMENTAL RESULTS

All the experiments were performed with MATLAB 2014b on a 3.6 GHz 4-physical core Intel® i7-4790 processor running with 8GB main memory and Windows 10 operating system. To train the acoustic model, a self-made database of 1000 Hindi words was used. Words were spoken by 20 different male and female speakers in a sound-treated room (i.e. clean environment) using 16 KHz 16-bit uncompressed audio format. The speech data was recorded by a microphone from a distance of 10-15 cm from speaker's lips. To save time, all inaudible and incomplete words were also removed.

The dataset was split into two parts 90% for training and development and 10% for testing. All the speech features were normalized. 50 target class labels (50 phonemes) are used. It is processed at 10ms frame rate having 25 milliseconds hamming window to obtain the acoustic features.

HMM having 8-states was trained for each of the 1000 words. The emission distribution of each state was modeled by a mixture of 8 Gaussians each with a diagonal covariance matrix. The DBNN had 4 layers with 512 units in each layer. Because of computations and time limitations, only one DBNN topology was tested in research. DBNN pre-training was performed for 5 epochs. The stochastic gradient descent training was used to optimize the network parameters based on the cross-entropy criteria. DBNN was trained for up to 20 epochs. For the first epochs, the momentum values were 0.0 and 0.9 for rest of the epochs.

A single pass over the entire training set during pre-training took about 3-5 minutes. An epoch of fine tuning with back-propagation to around 12-15 minutes. The DBNN was trained with an initial learning rate of 0.01. The decoder used was HVite, which is part of HTK package. The testing of the model was made by hundred random words and the performance of the model was measured by recognition rate.

$$\text{Accuracy (\%)} = 100\text{-WER (\%)}$$

Word error rate runs three types of errors: insertion, deletion and substitution errors. If there are N words in the reference transcript, and the ASR output has S substitutions, D deletions and I insertions, then

$$\text{WER} = 100 \times \frac{\text{S} + \text{D} + \text{I}}{\text{N}} \%$$

The following results were analyzed:

- Variation in accuracy with a different type of feature extraction techniques for a number of training samples in a clean and typical field environment.
- Variation in accuracy with alternative features in the noisy environment.

6.1. Comparison of Features in Clean Environment

In these experiments, HMM model was implemented and trained using a different number of epoch of training data (i.e. 5, 10, 15, 20). Similarly, DBNN model was also trained by using a different number of the epoch of training data. Word accuracy was measured in case of the feature extraction techniques as shown in Table 1. Experiment results showed that for a given training samples, MF-PLP showed one to two percent improvement in comparison to MFCC in case of HMM whereas MFCC was showing best result for DBNN. However, as the number of epochs of training increases, word accuracy also increases.

Table 1
Comparison of feature in clean environment

Model	*HMM*				*DBNN*			
Technique./Epoch	*5*	*10*	*15*	*20*	*5*	*10*	*15*	*20*
MFCC	75	79	85	91	78	85	91	95
PLP	74	79	85	89	77	83	89	92
MF-PLP	77	82	87	93	77	83	90	94
4GC+MFCC	72	77	80	85	75	78	82	88
TRAP	72	76	81	84	75	79	83	86
RASTA-PLP	74	78	83	89	76	81	85	90

6.2. Comparison of Features in Typical Field Conditions

In these experiments, both acoustic models (HMM and DBNN) were tested in field conditions (i.e. typical office environment). Rest of the things were same as mentioned above. Experimental results showed that for a given training samples, PLP showed the best results in case of HMM whereas for DBNN the MFCC features were again best among others. The results are shown in Table 2.

Table 2
Comparison of feature in typical field environment

Model	*HMM*				*DBNN*			
Technique/Epoch	*5*	*10*	*15*	*20*	*5*	*10*	*15*	*20*
MFCC	72	75	80	85	76	81	86	92
PLP	73	77	82	87	75	80	85	91
MF-PLP	68	73	77	83	74	78	81	86
4GC+MFCC	71	75	79	83	75	80	84	87
TRAP	71	75	80	82	73	76	81	83
RASTA-PLP	72	74	81	86	75	78	83	89

6.3. Experiment with White Noise

In these experiments, both the acoustic models were tested by mixing the white Gaussian noise in clean samples with the NOISEX92 [19]. The four number of gravity centroids features were added with MFCC without increasing the dimensionality of the feature vector. It was possible by replacing the last four MFCCs features by the 4GC. Experimental results showed a noteworthy improvement in the case of 4GC+MFCC, specifically at low SNR. The results are shown in Table 3 for white noise.

6.4. Experiment with Factory Noise

In these experiments, both the models were tested by mixing the factory noise source with the help of NOISEX92. In this type of noise source, 4GC+MFCC combination could not achieve a remarkable success. The reason is that spectral peaks were available which cannot be handled properly by the spectral subband filters. RASTA-PLP showed better result when the SNR ratio was low. The result was best with PLP at SNR level 15dB and MFCC at SNR level 20dB or 25dB for HMM. RASTA-PLP also showed the significant improvement in results for DBNN also at low SNR ratio and at medium and good level MFCC showed best results. The results are given in table 4 for factory noise.

Table 3
Comparison of feature with different level of white noise

												HMM								
Noise	*5dB*				*10dB*				*15dB*				*20dB*				*25dB*			
Epoch	*5*	*10*	*15*	*20*	*5*	*10*	*15*	*20*	*5*	*10*	*15*	*20*	*5*	*10*	*15*	*20*	*5*	*10*	*15*	*20*
MFCC	24	25	27	28	29	33	37	42	51	57	62	67	70	76	80	84	76	79	82	88
PLP	28	29	31	32	33	36	39	43	53	58	63	67	71	76	80	84	74	76	80	85
MF-PLP	33	34	35	37	36	38	43	47	53	57	63	68	71	75	79	83	72	76	80	84
4GC+MFCC	39	40	42	44	50	54	58	63	65	71	76	79	71	76	81	85	73	77	82	86
TRAP	31	32	33	34	32	34	36	38	44	48	52	56	64	69	75	78	72	76	78	81
RASTA-PLP	35	36	37	38	36	38	41	44	48	52	56	61	68	72	78	81	71	75	80	84
												DBNN								
MFCC	25	26	28	29	29	34	38	44	53	59	63	68	73	78	83	86	78	82	85	90
PLP	29	30	32	33	34	38	41	45	55	59	65	70	72	78	82	86	76	79	83	88
MF-PLP	35	36	38	40	38	40	46	50	55	60	66	71	73	77	82	84	76	79	84	88
4GC+MFCC	41	42	44	46	53	57	62	66	68	73	78	82	74	79	84	87	76	80	85	88
TRAP	33	34	35	36	34	37	39	41	46	51	55	59	66	71	77	81	74	79	81	85
RASTA-PLP	36	37	38	39	38	40	43	46	51	55	59	64	70	75	81	83	73	77	83	86

Table 4
Comparison of feature with different level of factory noise

											HMM									
Noise	*5dB*				*10dB*				*15dB*				*20dB*				*25dB*			
Epoch	*5*	*10*	*15*	*20*	*5*	*10*	*15*	*20*	*5*	*10*	*15*	*20*	*5*	*10*	*15*	*20*	*5*	*10*	*15*	*20*
MFCC	24	25	27	29	31	34	39	43	52	58	63	67	70	77	81	85	77	79	83	89
PLP	28	29	31	32	34	37	41	44	53	59	63	67	69	76	80	84	75	77	81	86
MF-PLP	29	30	33	34	35	38	41	44	54	59	63	66	68	75	78	82	73	74	79	83
4GC+MFCC	22	24	26	28	29	32	35	38	44	48	53	60	65	70	74	79	67	72	78	82
TRAP	24	26	27	29	26	28	31	33	39	44	48	52	60	66	72	75	70	73	75	79
RASTA-PLP	34	36	37	38	36	39	42	45	51	56	62	65	68	75	77	82	72	76	80	84
											DBNN									
MFCC	25	26	28	30	32	36	41	45	54	60	65	69	72	78	82	86	79	80	85	91
PLP	30	31	32	34	35	38	42	46	54	60	64	69	71	78	82	86	76	78	82	87
MF-PLP	31	32	35	36	37	39	42	45	56	61	65	68	69	76	79	83	75	76	81	85
4GC+MFCC	23	25	27	28	30	33	36	39	45	49	54	62	66	71	76	81	69	73	80	84
TRAP	25	27	29	31	28	30	33	35	41	46	50	55	62	67	73	76	71	75	77	81
RASTA-PLP	36	38	39	40	37	40	44	47	52	57	64	67	70	78	83	85	73	77	83	86

7. CONCLUSION

This paper attempted to cover a review of the feature extraction techniques for ASR system in the context of Hindi language. An insight into the strengths and weaknesses of current techniques was also provided theoretically and empirically. Experimental results showed that MF-PLP outperforms other feature extraction techniques in ideal lab conditions, MF-PLP superseded PLP for HMM as classifier whereas MFCC performed better when DBNN was a classifier. However, in typical field conditions when training (in sound treated lab) and testing conditions were mismatched, the performance of PLP was better in comparison to MFCC for HMM as a classifier. When DBNN was used as classifier the performance of the MFCC was better than PLP. Addition of alternative features like gravity centroids to MFCC showed a significant improvement at low SNR compared to standard MFCC in case of white Gaussian noise for both the classifiers but as the signal becomes strengthen the techniques like MFCC and PLP showed good results. However, little improvement was gained by this combination in case of factory noise since these sources contain strong peaks. RASTA-PLP showed good results when the signal strength was weak and MFCC showed good results when the signal strength was near to clean.

References

[1] T. Takiguchi, A. Sako, T. Yamagata, and Y. Ariki, "System Request Utterance Detection Based on Acoustic and Linguistic Features, Speech Recognition", France Mihelic and Janez Zibert (Ed.), InTech 2008.

[2] D. O'Shaughnessy, "Interacting with Computers by Voice: Automatic Speech Recognition and Synthesis", IEEE Proceedings. Vol. 91 No. 9 pp. 1272-1305, Sep 2003.

[3] C. Becchetti, and K.P. Ricotti, "Speech Recognition Theory and C++ Implementation", ISBN: 978-0-471-97730-8. New York, NY:John Wiley, March 1999.

[4] M.A. Anusuya, and S.K. Katti, "Front-end analysis of speech recognition: a review", International Journal of Speech Technology. Springer, Vol 14 No. 2, pp. 99-145, June 2011.

[5] F. Jelinek, "Statistical Methods for Speech Recognition". MIT press Cambridge ISBN: 0-262-10066-5, 1997.

[6] L. Rabiner, and B.H. Juang, "Fundamental of Speech Recognition", Pearson Education, ISBN: 0-13-015157-2, 1993.

[7] S. Davis, and P. Mermelstein, "Comparison of parametric representations for monosyllabic word

recognition in continuously spoken sentences", IEEE Transactions on Acoustics, Speech and Signal Processing. Vol. 28, No. 4. pp. 357-366, Aug 1980.

[8] H. Hermansky, "Perceptually linear predictive (PLP) analysis of speech". Journal of Acoustic Society of America. Vol. 87, pp.1738-1752, Nov 1989.

[9] D.S. Kim, S.Y. Lee, and R.M. Kil, "Auditory processing of speech signals for robust speech recognition in real-world noisy environment". IEEE Transactions on Speech and Audio Processing. Vol 7, No. 1, pp. 55-69, 1999.

[10] J. Chen, Y. Huang, Q. Li, and K.K. Paliwal, "Recognition of noisy speech using dynamic spectral subband centroids", IEEE Signal Processing Letters, Vol. 11, No. 2, pp. 258-261, 2004.

[11] A. Faria, "An investigation of Tandem MLP features for ASR". International Computer Science Institute, Tech. Rep, 2007.

[12] H. Hermansky, and S. Sharma, "TRAPS- classifiers of temporal patterns". In ISCA ICSLP. Vol. 3 pp. 1003-1006, 1998.

[13] N.S. Miller, M. Collins, and T.J. Hazen, "Dimensionality reduction for speech recognition using neighborhood components analysis". In Interspeech, Vol. 2, pp. 1397-1400, 2007.

[14] Y.A. Ghassabeh, F. Rudzicz, and H.A. Moghaddam, "Fast incremental LDA feature extraction". Pattern recognition. Vol. 48, No. 6, pp: 1999-2012, 2015.

[15] H. Hermansky, and N. Morgan, "RASTA processing of speech". IEEE Transaction on Speech and Audio Processing. Vol. 2, No. 4, pp. 578-589, 1994.

[16] R. Bakis, "Continuous speech word recognition via centisecond acoustic states". *Proc. ASA Meeting*, Washington DC, USA, 1976.

[17] X. He, L. Deng, "A new look at discriminative training for hidden Markov models". Pattern Recognition Letters. Vol. 28, pp. 1285-1294, 2007.

[18] G.E. Hinton, and S. Osindero, Y The, "A fast learning algorithm for deep belief nets". Neural Computation. Vol. 18, pp. 1527-1554, 2006.

[19] A. Varga, and H.J.M. Steeneken, "Assessment for automatic recognition: II, NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems. ESCA Journal of Speech Communication. Vol. 12, No. 3, pp. 247-251, 1993.