Hybrid Features for Speaker Independent Hindi Speech Recognition

Sharmila[1] , Dr.Achyuta N. Mishra[2],Dr.Neeta Awasthy[3]

Abstract— *Isolated spoken Hindi digits recognition performance has been evaluated using HTK (Hidden Markov Model Toolkit). This paper introduces and motivates the use of hybrid features for isolated Hindi digits recognition system. The speech recognizers use a parametric form of a signal to get the most important distinct features of speech signal for recognition task. In this paper Mel-frequency cepstral coefficients (MFCC), Perceptual linear prediction (PLP) coefficients along with two newly modified hybrid features are used for isolated Hindi digits recognition. Two modified hybrid features Bark frequency cepstral coefficients (BFCC) and Revised perceptual linear prediction (RPLP) coefficients were obtained from combination of MFCC and PLP. Experiments were performed for both clean as well as on noisy data. In this experiment six different noises: Car noise, F16 noise, Factory noise, Speech noise, LYNX noise and Operation room noise have been added to clean Hindi digits database at different SNR levels to get noisy database. The recognition performance with BFCC features was better than that with MFCC features. RPLP features have shown best recognition performance as compared to all other features for both noisy and clean databases.*

Keywords— *Hindi speech recognition, MFCC, PLP, Hybrid features, HTK.*

1 Introduction

Speech recognition is the key technology for effective man-machine interface. There has been a lot of research in the area of speech recognition for different languages like English, Mandarin, Arabic etc but little work has been carried out for Hindi speech recognition [1],[13]. Due to this reason only a small percentage of computer literate Indians are able to take advantage of the new advancement in the computer technology. In order to have the technology to penetrate the masses in India, Hindi speech interface is desirable. This paper is an effort in this direction where new hybrid feature extraction techniques are evaluated and have been tested for Hindi digits recognition. Although continuous speech recognition systems play an important role but isolated word recognition, particularly for the alphabet and digits, is still useful. It finds an application in areas such as recognizing telephone numbers, names and addresses, zip codes, and as a spelling mode for use with difficult words and out-of-vocabulary items in a continuous speech recognizer.

Due to these potential applications, many isolated word recognizers are optimized for the digits or alphabet or both. The digit recognition task for Hindi language is difficult due to a large number of variability in Hindi dialect. Hindi is a major Indian language belonging to the Indo-European family, which has retro flexion and germination as important feature [3]. In Hindi there are sixteen stop consonants, while English has only six stop consonants [1], [17], [14]. There is a little research in automatic speech recognition (ASR) system for Hindi and there is no system which would work well for a relatively rich dictionary. Previously the performance of isolated spoken English and Hindi digits [5], [14], [23] was evaluated using different feature extraction techniques. But in these researches the main interest was to show the recognition performance of the Hindi digits [2], [5], [8], [25] using HTK other feature extraction techniques like PLP [6], [11], [13], BFCC [8] and RPLP [8].

In this paper, the recognition performance of Hindi digits are evaluated and the performances of hybrid feature extraction techniques are compared to conventional feature extraction techniques using

HTK toolkit version 3.4. The experiments were conducted on clean as well as on noisy data for Hindi speech recognition. Matlab7.0 was used to extract PLP and hybrid features while training and testing was carried out using HTK.

A speech recognition system has three major components, database preparation, feature extraction and classification as shown in Figure 1. The recognition performance depends on the quality of database, performance of the feature extraction and classification techniques. Thus choice of features and its extraction from the speech signal should be such that it gives high recognition performance with reasonable amount of computation.

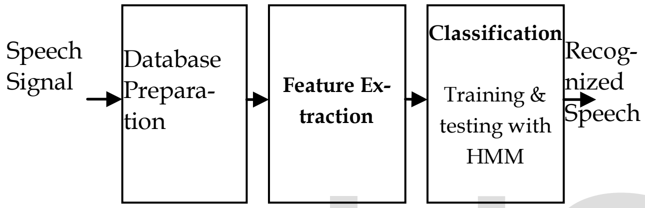


Fig . 1. Block diagram of Speech Recognition System

2 DATA BASE PREPARATION

A database of forty speakers, seventeen males and twenty three females for a total of ten Hindi digits (‘Shoonya’, ‘Ek’, ‘Do’, ‘Teen’, ‘Chaar’, ‘Paanch’, ‘Che’ ‘Saath’, ‘Aath’ and ‘Nou’) was prepared with sampling frequency 16 kHz and 16 bits per sample. Speakers were chosen from different geographical areas of India, different social classes and of different age groups (18-26 years). Cool Edit software was used for preparation of Hindi digits database. Every speaker was asked to repeat each digit ten times with short inter-digit pauses. Further, all ten repetitions of each digit were segmented manually. The database consists of 10 samples of each Hindi digit of each speaker, i.e., total 4000 speech samples (for 40 speakers), 100 samples (for each speaker). This database was prepared in two different phases at Aligarh Muslim University, Aligarh and Birla Institute of Technology, Mesra, Ranchi, India. The age group of 18-26 years was chosen as students of different dialects in this age group were easily available. A distance of 4-6 inch was maintained between microphone and the speaker at the time of database recording. Two different microphones made by Sony and i-Ball were used for recording the database in first and second phases respectively. Hindi pronunciations of digits and its corresponding English digits are shown in Table 1. Artificial noisy database was prepared for ten Hindi digits by adding different types of noises from NOISEX-92 database [21] to clean Hindi digits database. To generate noisy speech: car noise, F16 noise, factory noise, speech noise, LYNX noise and operation room noise from NOISEX-92 database were artificially added to clean speech at different signal-to-noise ratios (SNRs) in the range 30dB to 0dB

Table .1: Hindi digits and corresponding English digits

Hindi Digits	Hindi Pronunciations	English Digits	English Pronunciations
०	Shoonya	0	Zero
१	Ek	1	One
२	Do	2	Two
३	Teen	3	Three
४	Chaar	4	Four
५	Paanch	5	Five
६	Che	6	Six
७	Saath	7	Seven
८	Aath	8	Eight
९	Nou	9	Nine

3 FEATURE EXTRACTION TECHNIQUES

The raw speech signal has a lot of undesired information which are not required for automatic speech recognition system; hence the need for a good front-end arises. The task of this front-end is to extract all relevant acoustic information in a compact form compatible with the acoustic models. In other words, the pre-processing should remove all irrelevant information such as background noise and undesired effects of the recording devices. Features can be defined as a minimal unit, which distinguishes maximally close classes. The entire feature extraction schemes for PLP and BFCC, MFCC and RPLP are shown in Figure 2 and Figure 3 respectively.

3.1 Mel Frequency Cepstral Coefficient

Pre-emphasis filtering, normalization and mean subtraction are the three steps in pre-processing. The digitized speech is pre-emphasized using a digital filter with a transfer function. Pre-emphasis filter [12] spectrally flattens the signal and makes it less susceptible to finite precision effects later in the signal processing. Due to possible mismatch between training and testing conditions, it is considered a good practice to reduce the amount of variations in the data that does not carry important speech information as much as possible. For instance, differences in loudness between recordings are irrelevant for recognition. For reduction of such irrelevant sources of variation, normalization transforms are applied. During normalization every sample value of the speech signal is divided by the highest amplitude sample value. Mean of the speech signal is subtracted from the speech signal to remove the DC offset and some of the disturbances induced by the recording instruments. After pre-processing, language utterances were divided into equal frames of 25ms duration with an overlap of 10ms. Next each frame was multiplied by a Hamming window. The use of the window function reduces the frequency resolution, so the frames must overlap to permit tracing and continuity of the signal. The motive of utilizing the windowing function is to smooth the edges of each frame for reducing the discontinuities or abrupt changes at the endpoints. Other purpose of windowing is reduction of the spectral distortion that arises from the windowing itself. After windowing first FFT and then Mel spaced filter banks are applied to get the Mel-spectrum. The Mel scale is logarithmic scale resembling the way that the human ear perceives sound. Mel scale is given by the Equation 1.

$$Mel(f) = 2595\log_{10}(1 + f/700) \qquad (1)$$

Where f is frequency. The natural logarithm is taken to transform into the cepstral domain and the discrete cosine transform (DCT) is finally applied to get 24 MFCCs. The component due to the periodic excitation source may be removed from the signal by simply discarding the higher order coefficients. DCT de-correlates the features and arranges them in descending order of information, they contain about speech signal. Hence 13 coefficients out of 24 coefficients are used as MFCC features in our case. MFCC features are more compact since the same information can be contained in fewer coefficients

3.2 Perceptual Linear Prediction (PLP)

PLP provides a representation corresponding to a smoothed short-term spectrum that has been compressed and equalized much as done in human hearing. It can be assumed similar to Mel-cepstrum based features. In PLP technique, several well-known properties of hearing are simulated by practical engineering approximations, and the resulting auditory like spectrum of speech is approximated by an autoregressive all–pole model. PLP provides reduced resolution at high frequencies that indicates auditory filter bank based methods, yet provides the orthogonal outputs that typify cepstral analysis. The principal difference between the PLP and MFCC approaches lies in the spectral smoothing. PLP uses linear predictions for spectral smoothing; hence the name is perceptual linear prediction. The different steps of PLP analysis are as follows.

I. Power spectral estimate for the windowed speech signal is computed. This is done by windowing the analysis region with Hamming window, calculating the FFT and computing its squared magnitude.

II. The power spectrum within overlapping critical band filter responses is integrated. For PLP, trapezoidal shaped filters are applied at 1-bark intervals, where the bark axis [6] is derived from the frequency axis by using a warping function from Schroeder, given in Equation 2. The Bark scale is linear at low frequencies and logarithmic at high frequencies.

$$P(\omega)=6\ln\{\frac{\omega}{1200\pi}+[(\frac{\omega}{1200\pi})^{2}+1]^{0.5}\} \qquad (2)$$

Where ω is angular frequency in radians/second. This effectively compresses the higher frequencies into a narrow band. The critical band masking, symmetric frequency domain convolution on the Bark warped- frequency scale then allows low frequencies to mask the high frequencies while at the same time smoothing the spectrum an effect consistent with the psycho-acoustic results.

III. The spectrum is pre-emphasized to approximate the unequal sensitivity of human hearing at different frequencies. This step is implemented as an explicit weighting of the elements of critical band spectrum.

IV. Spectral amplitude is compressed. The effect of this step is to reduce amplitude variations for the spectral resonances.

V. An inverse DFT is performed. As a result of this step autocorrelation coefficients are obtained, but these coefficients are from a compressed spectrum. Since the power spectrum values are real and even, only the cosine components of inverse DFT are to be calculated.

VI. Spectral smoothing is performed. This step is done by solving the autoregressive equations constructed from the autocorrelations of the previous step.

VII. The autoregressive coefficients are converted to cepstral variables.

In this experiment two main blocks as shown in Figure 2 and Figure 3 were interchanged to develop two hybrid feature extraction techniques. The interest is to see the influence of the spectral processing on the different cepstral transformation. Figure 2 and Figure 3 show the steps of parameterization for PLP, MFCC and hybrid features. Both Hybrid features RPLP and BFCC are obtained from combinations of PLP and MFCC features as shown in Figure 2 and Figure 3.

3.3 Bark Frequency Cepstral Coefficient (BFCC)

BFCC is combination of PLP and MFCC. BFCC is the process where PLP processing of the spectra and cosine transform are combined to get the cepstral coefficients. Instead of using Mel filter bank, Bark filter bank has been applied and equal loudness pre-emphasis with intensity to loudness power law has been applied to the MFCC like features. Only first thirteen cepstral features of each windowed frame of speech utterances were taken.

3.4 Bark Revised Perceptual Linear Prediction (RPLP)

RPLP is combination of MFCC and PLP. In this approach instead of using bark filter bank, Mel filter bank has been applied to compute RPLP. The signal is pre-emphasized before segmentation and FFT spectrum is processed by Mel scale filter bank. The resulting spectrum is converted to the cepstral coefficients using LP analysis with prediction order of 13 followed by cepstral analysis.

4 HTK Toolkits

HTK toolkit uses hidden Markov models (HMM) for ASR system. It is used for research in many applications such as speech synthesis, character recognition and DNA sequencing. HTK 3.4 version

ISSN 2229-5518

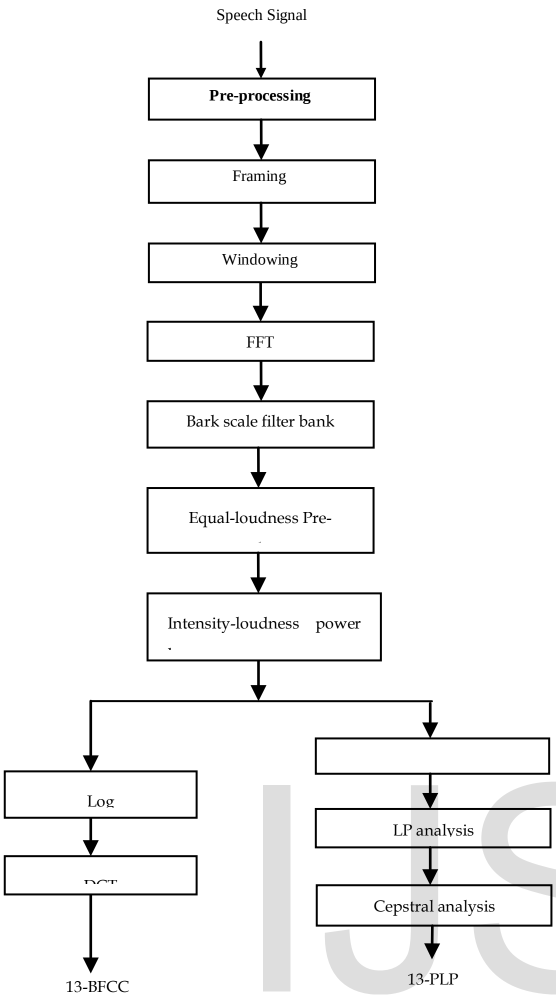


Fig. 2. BFCC and PLP Feature Extraction Meth-

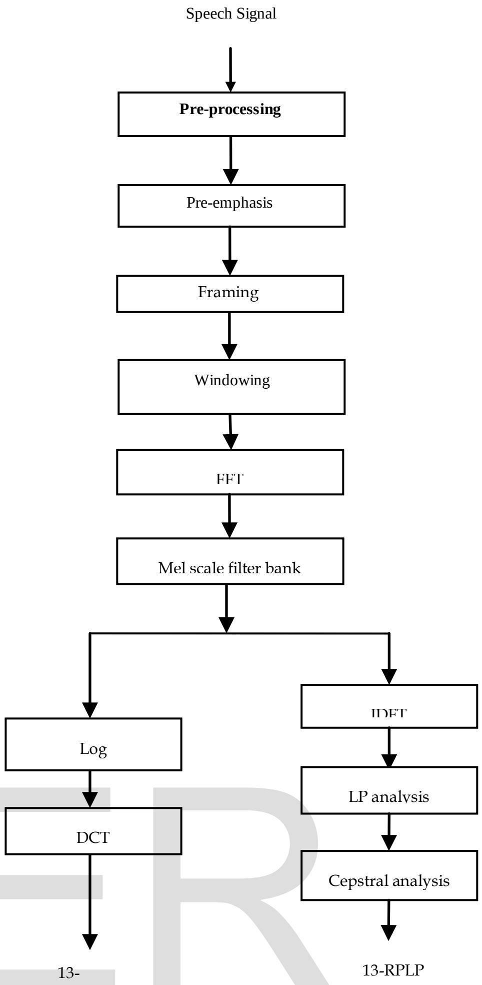


Fig. 3. MFCC and RPLP Feature Extraction Methods

is used in this research work for isolated spoken Hindi digits recognition. This toolkit consists of many modules and tools. All of them are available in C source form. The HTK provides facilities for speech analysis, HMM training, testing and results analysis. It compares possible phonetic transcriptions of words. The toolkit supports HMMs using both continuous density mixture Gaussians and discrete distributions. The choice of the HTK, as the recognition engine for the simulations was to get a standard benchmark for the performance of various feature extraction methods and for easy migration and reproduction of the simulations by other researchers. The feature extraction block supports various feature extraction technique such as Linear Prediction Coefficients (LPC) [12], Reflection Coefficients (RC) [12] Mel Frequency Cepstral Coefficients (MFCC) and more. This block can also estimate the dynamics of the features in time i.e. the derivative and acceleration. However this block does not support the PLP and hybrid features like BFCC & PLP.

HTK Parameter	Value
WINDOW SIZE	25 ms
TARGET RATE	10ms
USE HAMMING	True
PREECOEF	.97
NUMCEPS	12
CELIFTER	24

Table. 2. Feature extraction parameters setting

The extraction of these features was performed using Matlab7.0. The simulation parameters used in this block are given in Table 2. The HMM [3], [10], [15], [19] training blocksupports both continuous and discrete density modeling and uses the Baum-Welch algorithm to estimate the HMMs. The HMM was a simple Left-Right with no skip model, which was trained for each isolated digit. The way that these tools are used to build an HMM based system is illustrated in Figure 4. The speech data is parameterized and transcription label files are created starting with a very simple prototype system

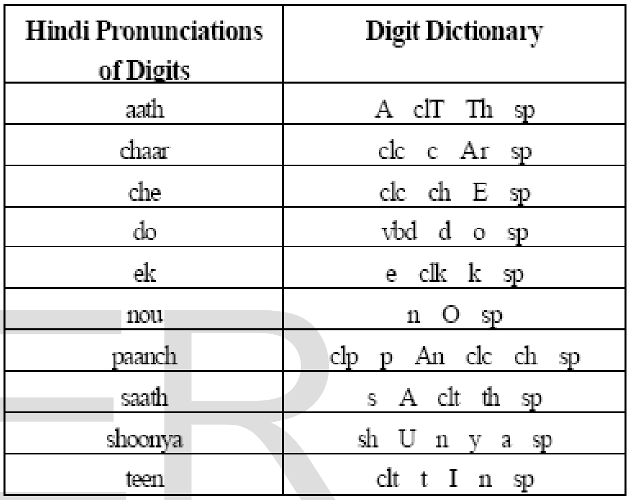
I. HMM's are repeatedly edited and re estimated until the required level of model complexity and performance is reached. The recognizer block calculates the likelihood of a given feature set to be generated from each HMM or HMM network using the Viterbi algorithm [12]. One HMM model was prepared for each Hindi digit. The simulations were performed for isolated digits recognition, which is a special case for HTK since it is designed for continuous speech; therefore the HVite tool performed hypothesis testing between the 10 trained HMMs.

5 Experimental Setup and Results

Initially the database was divided into two parts: training data and testing data. The 40 speakers were divided into a training corpus with 35 persons and a testing corpus with 5 persons. In the testing corpus there were three female and two male speakers. Six different types of noises (car noise, F16 noise, factory noise, speech noise, LYNX noise and operation room noise) were injected into the clean signal to achieve 0 dB, 5 dB, 10dB, 20dB, and 30dB SNRs samples of Hindi digits.

Training and testing was done without language model in the following manner. To build digit recognition system at first acoustic models of digits [25] has been prepared. The proper dictionary was needed to develop good recognition system. The dictionary [16], [18] shown in Table 3 is used in this experiment. For all experiments presented in this paper, the HTK toolkit was used to implement a HMM recognizer. In each experiment, there were 10 HMM models trained to recognize 10 Hindi digits of the Hindi language. Proper initialization of the HMM parameters with the training data is must before starting the training process to get precise and fast convergence. At first the samples of digit shoonya of all 35 speakers were taken for feature extraction. HCopy is used to get thirteen MFCC features of each frame of a sample. MFCC features of all frames of all samples of digit shoonya for 35 speakers were extracted and stored for HMM training. Then two tools, HInit and HCompv are used to create HMM model of the Hindi digit shoonya. HInit makes use of the set of observation sequences to get the initial estimates of parameters. It uses Viterbi alignment for segmenting training observations and then pools the vectors in each segment to re-compute the parameters [25]. By counting the number of times each state is visited during the alignment process it start to estimate the transition probabilities. HCompv tools forms the first stage of flat start training scheme and initializes the parameters such that global data variances and mean are equal to the component variances and

Table. 3. Hindi Digits Dictionary



Hindi Pronunciations of Digits	Digit Dictionary
aath	A clT Th sp
chaar	clc c Ar sp
che	clc ch E sp
do	vbd d o sp
ek	e clk k sp
nou	n O sp
paanch	clp p An clc ch sp
saath	s A clt th sp
shoonya	sh U n y a sp
teen	clt t I n sp

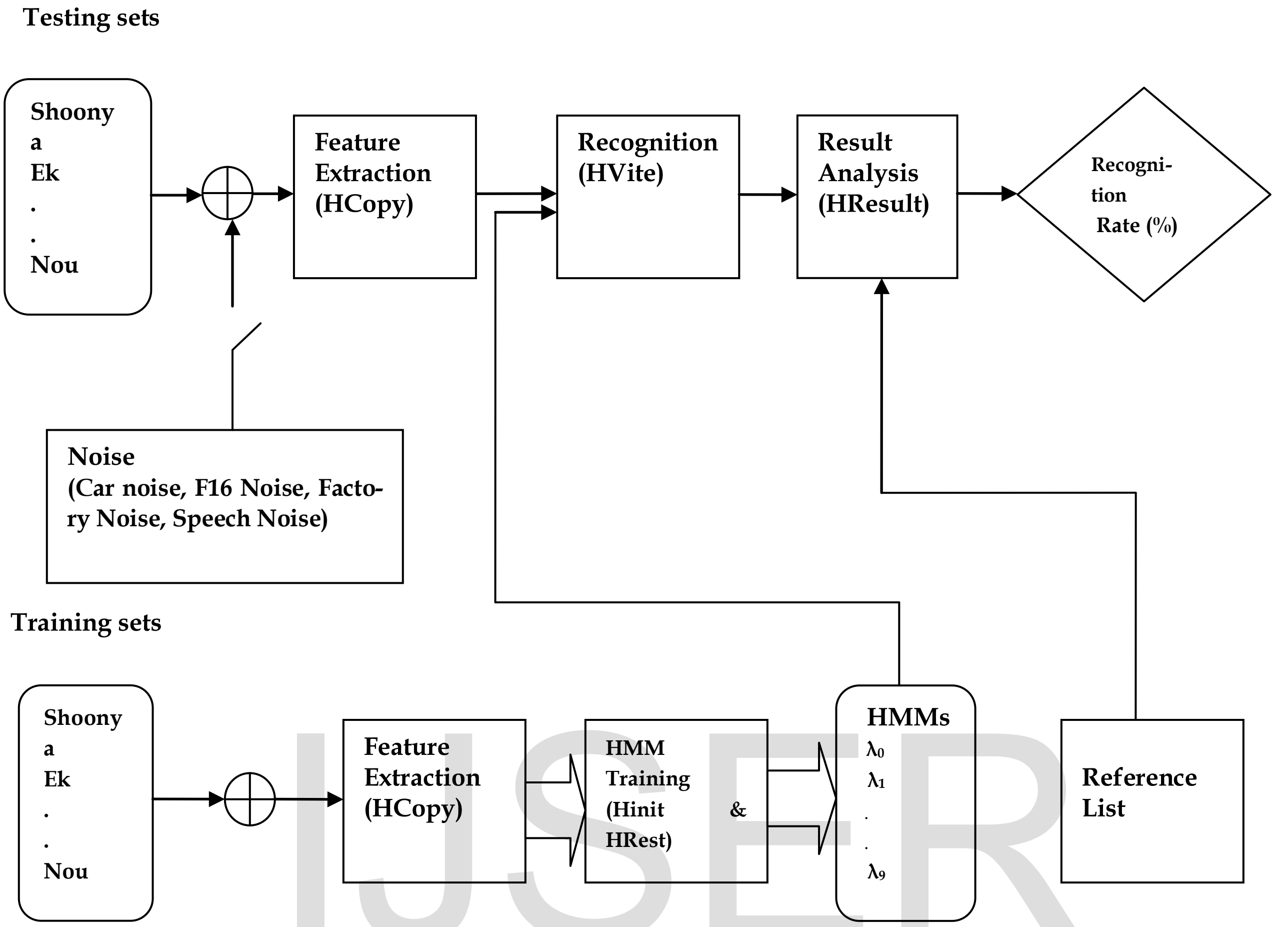


Fig. 4. HTK simulation block diagram

means. The next step uses the tool HRest which re-estimates the optimal values of parameters of a single HMM like transition probabilities, mean and variance vectors of each observation function using the Baum- Welch algorithm. Re-estimation is done several times till measures do not change and a convergence is reached to get the HMM model of digit shoonya. In the similar fashion as for Hindi digit shoonya, HMM models of all other Hindi digits were also prepared. In HTK [4], [7], [23] the task grammar is written in a text file, containing a set of rewrite rules based on Extended Backus-Naur Form (EBNF). A task dictionary is also defined which informs the system as to which HMM does each of the grammar variables correspond to. The HParse tool compiles the task grammar to give us the task network [25]. Since HTK feature extraction block does not support PLP, BFCC & RPLP feature extraction. These features are extracted using Matlab7.0 functions [20]. Then the features were saved in HTK format using functions from the Voice-box toolkit. The same procedure was followed to create HMM models for all Hindi digits as explained for MFCC. Hindi digits database of total five speakers is used for testing. For recognition of unknown digit the feature vectors of unknown digit are taken as the observation sequence. The probability of occurrence of that observation sequence is computed for each digit model. The digit whose model gave the highest probability was taken as the recognized digit.

The following experiments were performed (in all combinations):
Feature type: - MFCC, PLP, RPLP & BFCC
Feature vector length: 13
HMM states: 1...5
HMM mixes: 1...9 Testing was performed with clean data and noise injected data corpus. For each speaker there were ten repetitions of each Hindi digit. Thus a total of 10*10=100 samples of ten Hindi digits were tested for each speaker. The comparative average recognition efficiencies of all feature extraction techniques for clean data corpus have been shown in Figure 5. For clean data BFCC has shown better recognition performance compared to MFCC. During testing of clean data, RPLP has shown best recognition performance among all types of features. Comparative recognition performance of all feature extraction methods for noisy data corpus with car noise, F16 noise, factory noise, speech noise, LYNX noise and operation room noise are shown in Figure 6, Figure 7, Figure 8, Figure 9, Figure 10 and Figure 11 respectively. It is observed from the results that hybrid features perform better than conventional features at lower SNR levels. There was considerable degradation in the recognition results for SNR values below 10dB for all feature extraction techniques. The impact of different types of noise addition shows the importance of the feature extraction methods for robustness in HMM-based recognition.

Using Bark-frequency scaling, equal loudness pre-emphasis, intensity-loudness power law and DCT seems to have reasonable influence on the recognition accuracy. The auditory based approach shows the advantage in being inspired by human hearing. While on average most of the times RPLP features along with Pre-emphasis, Mel–frequency scaling and LP based analysis have shown best recognition result. Because RPLP with Mel- Scale are robust and also take into account the psychoacoustic properties of the human auditory system.

6 Conclusion

In this paper, the recognition performance of isolated Hindi digits has been evaluated using MFCC, BFCC, PLP and RPLP features in clean and noisy environments. Recognition efficiency for clean data as well as noisy data at different SNRs has been tested with different features. From all the experiments, it was concluded that BFCC has shown better recognition performance compared to MFCC because it is more invariant to fixed spectral distortion and channel noise compared to MFCC. PLP features have shown further improvement in recognition performance which was due to the fact that PLP is combination of both MFCC and LP based features. PLP features performed better because the signal was pre-emphasized by a simulated equal-loudness curve to match the frequency magnitude response of the ear as well as all signal components were perceptually equally weighted. RPLP features performed best among all feature extraction techniques. This is due to the fact that it takes advantage of pre-emphasis filter, Mel scale filter bank along with linear prediction and cepstral analysis.

For all kind of feature extraction method the system did not provide satisfactory result at 0 dB and 5 dB SNR levels. Future work will be directed towards the investigation of low SNR Hindi digits (0dB & 5dB), by taking more contexts into consideration during the feature extraction and optimizing the primary time-frequency analysis.

7 FIGURES AND TABLES

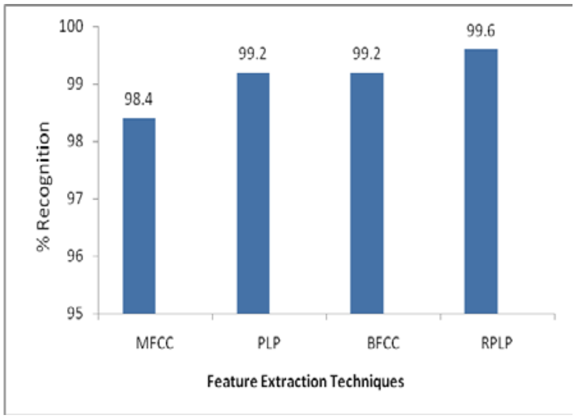


Fig. 5. Average recognition efficiency of isolated spoken clean Hindi digits database

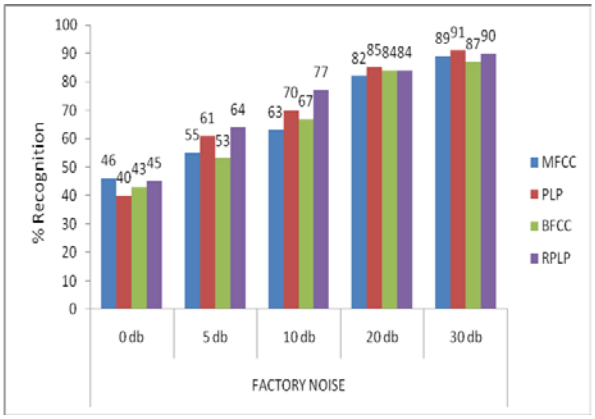


Fig .8. Average recognition efficiency of spoken Hindi digits with Factory noise

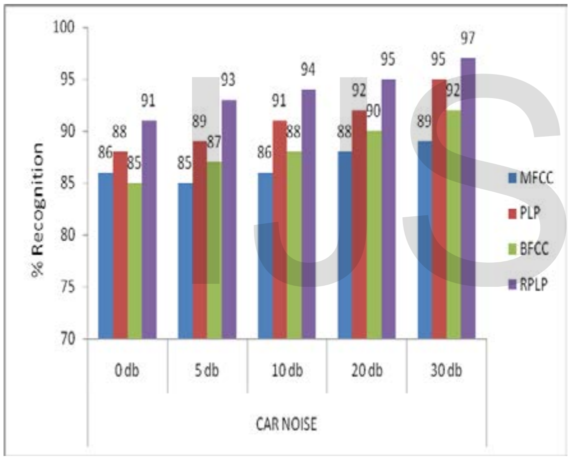


Fig. 6. Average recognition efficiency of noisy spoken Hindi digit with Car noise

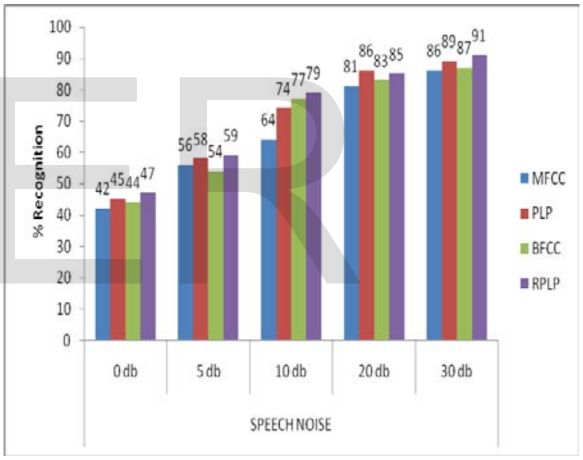


Fig. 9. Average recognition efficiency of noisy spoken Hindi digit with Speech noise

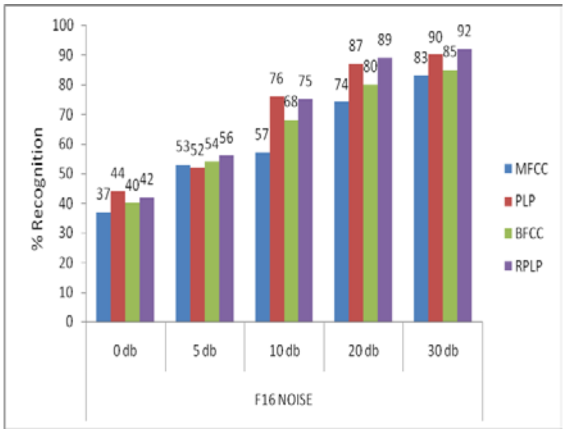


Fig. 7. Average recognition efficiency of noisy spoken Hindi digit with F16 noise

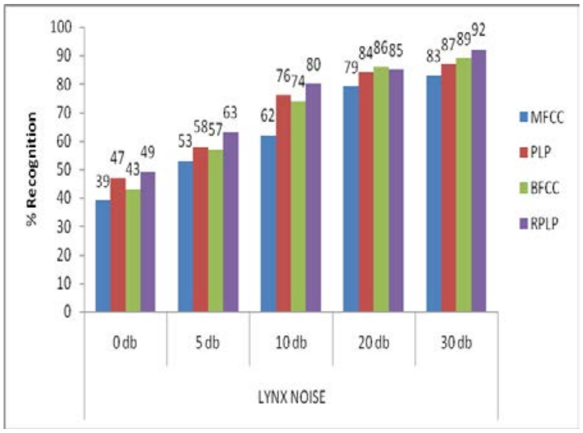


Fig. 10. Average recognition efficiency of noisy spoken Hindi digit with LYNX noise

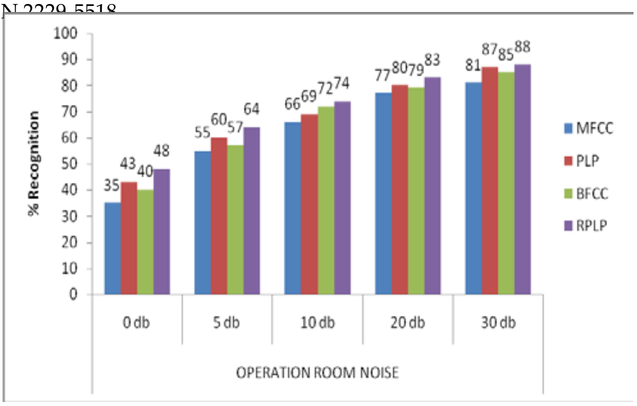


Fig. 11. Average recognition efficiency of noisy spoken Hindi digit with Operation room noise

Acknowledgment

The authors are grateful to the students of Raj kumar goel institute of technology, Birla Institute of Technology, Mesra, Jharkhand, India and Aligarh Muslim University, Aligarh, U.P, India who have actively co-operated during the collection of data for the present work. The authors would like to thank Mr. Sudhamoy Maiti of Indian Institute of Technology, Kharagpur, India for his kind support & concerned advice during the present work. The authors would also like to thank Birla Institute of Technology, Mesra, India for providing the required research facilities and opportunities for the present work.

REFERENCES

[1] Davis, K. “*Stop Voicing in Hindi*”, Journal of phonetics, Vol. 22, pp. 177–193, 1994.

[2] Davis, S. B., Mermelstein, P. “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences”, *IEEE Transactions on acoustics, speech and signal Process*, 28(4), pp. 357–366, 1980.

[3] Ephraim, Y., Merhav, N. “ Hidden Markov Processes”, *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1518-1569, January 2002.

[4] Evermann, G. *HTK History* [Online], 2002. Available from World Wide Web: <http://htk.eng.cam.ac.uk/history.shtml>

[5] Gazi, H.M., Farooq, O., Khan, Y. U., Datta, S. “Wavelet-based, speaker-independent isolated Hindi digit recognition”, *International Journal of Information and Communication Technology*, 1(2), pp. 185- 198, 2008.

[6] Hermansky, H. “Perceptual linear predictive analysis of speech”, *Journal of Acoust. Soc. America.*,87(4), pp.1738–1752,1990.

[7] HTK, “*Hidden Markov Model toolkit*”, Cambridge University Engineering Department, Cambridge, UK. http://htk.eng.cam.ac.uk.

[8] Josef, R., Pollak, P. “Modified Feature Extraction Methods in Robust Speech Recognition”, *Radioelektronika, 17th IEEE International Conference*, pp.1-4, 2007.

[9] Kumar, M., Rajput, N., Verma A. “A large vocabulary continuous speech recognition system for Hindi”, *IBM Journal of Research & Development*, Vol.48, No. 5/6, pp. 703-714, 2004.

[10] Ma, G., Zhou, W., Zheng, J., You, X., Ye, W. “A Comparison between HTK and SPHINX on Chinese Mandarin”, *IEEE International Joint Conference on Artificial Intelligence*, pp. 394-397, 2009.

[11] Makhoul, J. “Linear prediction: A tutorial review”, *Proc. of IEEE*, 63(4), pp.561-580, 1975.

[12] Rabiner, L. R., Juang, B. H. “*Fundamentals of speech recognition*”, Englewood Cliffs: Prentice-Hall, 1993.

[13] Samudravijaya, K. “Durational Characteristics of Hindi Stop Consonants”, *Proc. of Eurospeech03*, Geneva, Switzerland, pp. 81-84, 2003.

[14] Samudravijaya, K., Barot, M. “A Comparison of Public-Domain Software Tools for Speech Recognition”, *In WSLP*, pp.125-131, 2003.

[15] Seifu, Z. “*Hidden Markov Model based large vocabulary speaker independent continuous Amharic speech recognition*”, Thesis report of Master of Science, Department of Information Science, Addis Ababa University, 2003.

[16] Sharma. “*Hindi word-accent*”, Indian Linguistics, p.30, 1969

[17] Shrotriya, N., Verma, R., Gupta S. K., Agrawal , S. S. “Durational Characteristics of

Hindi Consonant Clusters”, *Proc. of Int. Conf. on Spoken Language Processing*, pp. 2427–2430, 1996.

[18] Shukla , S. “Syllable structure and word stress in Hindi”, *The Georgetown Journal of Languages & Linguistics*, pp.1.2, 1990.

[19] Torre, T. D., Gomez, V.J., Hemandez, G.L. “Initialization, Training, and Context-Dependency in HMM-Based Formant Tracking”, *IEEE transactions on Audio, Speech and Language Processing*, 14(2), pp. 511-524, 2006

[20] V. Young, S.J. “Large Vocabulary Continuous Speech Recognition Using HTK”, in *Proceedings of ICASSP*, 1994.

[21] Varga, A., Steeneken, H.J.M., Jones, D. “The noisex-92 study on the effect of additive noise on automatic speech recognition system”, *Reports of NATO Research Study Group (RSG.10)*, 1992.

[22] Verma, R., Chawla, P. “Comparative analysis of Hindi retroflex and dental CV syllables and their synthesis”, *Proc. of Workshop on Spoken Language Processing*, pp. 163-167, 2003.

[23] Vertanen, K. “Baseline WJS acoustic models for HTK and Sphinx: Training recipes and recognition experiments: ”,http://www.inference.phy.cam.ac.uk/kv227/papers, 2007.

[24] Woodland,VOICEBOX: “Speech Processing Toolbox for MATLAB”,http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html

[25] Young, S., Ollason, D., Valtchev, V., Woodland, P. “*HTK Book (for HTK Version 3.4)*”, Cambridge: Cambridge University Engineering Department, 2006.