Punjabi Automatic Speech Recognition Using HTK

Mohit Dua[1], R.K.Aggarwal[2], Virender Kadyan[3],Shelza Dua[4]

[1,2]Department of Computer Engineering, NIT, Kurukshetra, India

[3]Department of Computer Engineering, DIET, Karnal, India

[4]Department of Electronics and Communication Engineering, RPIIT, Karnal, India

Abstract

This paper aims to discuss the implementation of an isolated word Automatic Speech Recognition system (ASR) for an Indian regional language Punjabi. The HTK toolkit based on Hidden Markov Model (HMM), a statistical approach, is used to develop the system. Initially the system is trained for 115 distinct Punjabi words by collecting data from eight speakers and then is tested by using samples from six speakers in real time environments. To make the system more interactive and fast a GUI has been developed using JAVA platform for implementing the testing module. The paper also describes the role of each HTK tool, used in various phases of system development, by presenting a detailed architecture of an ASR system developed using HTK library modules and tools. The experimental results show that the overall system performance is 95.63% and 94.08%.

***Keywords*- Automatic Speech Recognition system, Mel Frequency Cepstral Coefficient (MFCC), HMM, HTK, P-ASR**

1. Introduction

Speech is the vocalized form of human communication. To communicate with a machine we still require interfaces like keyboard, mouse and screen etc., operated with the help of software. A simple alternative to these hardware interfaces is a software interface i.e. an ASR (Automatic Speech Recognition) system [2]. Automatic speech recognition is the task of taking an utterance of speech signal as an input, captured by a microphone, a telephone etc., and convert it into a text sequence as close as possible to the spoken data [4]. The main difficulties in implementation of an ASR system are due to different speaking styles of human beings and environmental disturbances. So the main aim of an ASR system is to transform a speech signal into text message independent of the device, speaker or the surroundings in an accurate and efficient manner.

Some of the major application areas of Automatic speech recognition systems are dictation, controlling the programs, automatic telephone call processing and query based information system such as travel information system , weather report information system etc. Keeping all the difficulties and its wide applications into consideration the paper aims to develop a GUI (using Java) based speaker independent isolated word recognizer for limited vocabulary based on HMM [1,3] (Hidden Markov Model) using HTK open source toolkit in Linux environment for Punjabi language (Gurumukhi Script).

1.1 Motivation

Research on automatic speech recognition by machine has attracted much attention over the last five decades. The reviewed literature reveals that the agencies like AT & T Bell Labs, DARPA, IBM, and Microsoft have sponsored many programs for research in this area in the last 50 years. Still a lot of research work is being done in this area. But the main focus of research groups remained around building ASR systems for European languages especially English. The various commercially developed systems available such as Microsoft SAPI, Dragon-Naturally-Speech and IBM via voice are some examples. But the fascination of speech recognition has made various research groups curious to develop systems in their native or local languages so that the benefit of the same can be made available to people of their region. The latest examples of these can be seen through the work being done in Arabic countries in their native language i.e. Arabic [9] and in India the ASR systems developed for various languages like Hindi [11], Bengali etc. So the non availability of effective speech recognition system for Punjabi language and regional relevance has encouraged working in discovered yet unexplored area of Punjabi language speech recognition.

2. Related Work

This section of paper will represent literature review of the works that are similar to the presented work.

R. Kumar [8] implemented an experimental, speaker-dependent, real-time, isolated word recognizer for the language Punjabi and further extended its work to compare the performance of speech recognition system for small vocabulary of speaker dependent isolated spoken words using the Hidden Markov Model (HMM) and Dynamic Time Warp (DTW) technique. The presented work emphasized on template-based recognizer approach using linear predictive coding with dynamic programming computation and vector quantization with Hidden Markov Model based recognizers in isolated word recognition tasks.

A speaker independent, real time, isolated word ASR system for the Punjabi language was developed by R. Kumar et al. [7].The Vector Quantization and Dynamic Time Warping (DTW) approaches were used for the recognition system. The database of the features (LPC Coefficients or LPC derived coefficients) of the training data was created for training the system and for testing the system the test pattern (features of the test token) was compared with each reference pattern using dynamic time warp alignment. The system was developed for small isolated word vocabulary.

K. Kumar et al. [11] developed a connected-words speech recognition system for Hindi language. The system was developed using hidden Markov model toolkit (HTK) and the system was trained to recognize any sequence of words selected from the vocabulary of 102 words. The training data was collected from 12 speakers including both males and females and test-data collected from the five speakers was used to evaluate the performance of the recognizer.

Al-Qatab et al. [9] implemented an Arabic automatic speech recognition engine using HTK. The engine recognized both continuous speech as well as isolated words. The developed system used an Arabic dictionary built manually by the speech-sounds of 13 speakers and it used vocabulary of 33 words.

Our paper aims to discuss design and implementation of a Punjabi (Gurumukhi script) isolated word recognizer consisting of 115 word vocabulary and developed to work in both speaker dependent and speaker independent real time environments.

3. Statistical Framework of An ASR

ASR as shown in Fig. 1 mainly comprises of five parts: Acoustic Analysis for feature extraction, Acoustic model based on statistical HMM approach, Language model, Pronunciation dictionary and the decoder for recognition.

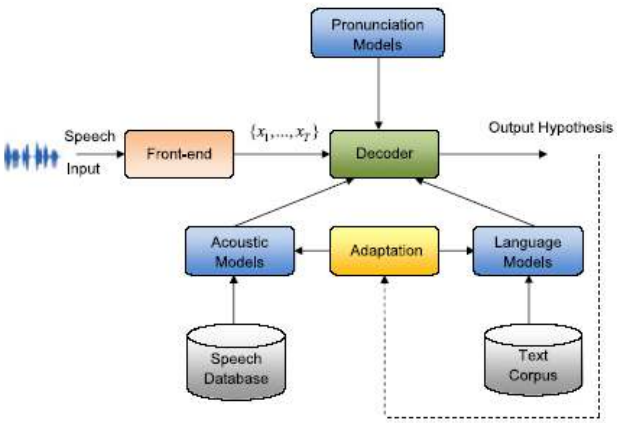

Fig. 1: Block diagram of ASR [4]

The sound waves captured by a microphone at the front end are fed to the acoustic analysis module. In this module the input speech is first converted into series of feature vectors (a.k.a observation vectors) which are then forwarded to the decoder. This decoding module with the help of acoustic, language and pronunciation models comes up with the results. Mainly, the speech recognition problem can be divided into the following four step i.e. signal parameterization using a feature extraction technique such as MFCC or PLP, acoustic scoring with Gaussian mixture models (GMMs), sequence modeling with hidden Markov models (HMMs) and generating the competitive hypotheses using the score of knowledge sources (acoustic, language and pronunciation models) and selecting the best as final output with the help of a decoder [4].

4. Punjabi Language Phonology

Punjabi is written in Gurumukhi and Shahmukhi script. Shahmukhi script is used in West Punjab which is in Pakistan. However, in East Punjab that belongs to India, Gurumukhi script is used. Gurumukhi, meaning "from the mouth of the Guru" is the most commonly used script in India for writing in Punjabi. Gurumukhi script is written from left-to-right and is spelled phonetically. Gurumukhi script is alphasyllabary in nature. An alphasyllabary system consists of two types of symbols consonants and vowels. There are 41 consonants and 9 vowels in Gurumukhi script as shown in Table-1 & Table 2. In addition to these, there are 3 auxiliary signs to add a nasal sound to a particular vowel as shown in Table 3.

Table-1: Consonants (ਵਿਅੰਜਨ)

ੳ (ura)	ਅ (aira)	ੲ (iri)	ਸ (sassa)	ਹ (haha)	ਕ (kakka)	ਖ (khkha)
ਗ (gaga)	ਘ (ghaga)	ਙ (nanna)	ਚ (chacha)	ਛ (shasha)	ਜ (jaja)	ਝ (jhaja)
ਞ (nainna)	ਠ (tainka)	ਠ (thatha)	ਡ (dadda)	ਢ (dhadda)	ਣ (naana)	ਤ (tatta)
ਥ (thattha)	ਦ (dadaa)	ਧ (dhada)	ਨ (nanna)	ਪ (pappa)	ਫ (faffa)	ਬ (baba)
ਭ (bhabha)	ਮ (mama)	ਯ (yaya)	ਰ (rara)	ਲ (lala)	ਵ (vavva)	ੜ (rarha)
ਸ਼ (sassha)	ਖ਼ (khakha)	ਗ਼ (gagha)	ਜ਼ (jajjha)	ਫ਼ (faffha)	ਲ਼ (lallha)	

Table-2: Vowels (ਲਗਾ ਮਾਤਰਾ)

ਿ (sihari)	ੀ (bihari)	ਾ (kanna)
◌ੇ (lavan)	◌ੈ (dulavan)	◌ੌ (aunkar)
◌ੁ (dulonkar)	◌ੂ (hora)	◌ੋ (kanora)

Table-3: Auxiliary Signs

◌ੰ (tippi)	◌ਃ (bindi)	◌ੱ (adak)

Alphabets of the script represent syllables. All consonants contain an inherent vowel /a/ or schwa ending, both of which

can be altered and muted by means of diacritics or matra. Vowels can also be written with separate letters when they occur at the beginning of a word or on their own. When two or more consonants occur together, special conjunct symbols are often used to add the essential parts of the first letter or letters in the sequence to the final letter.

5. Punjabi-ASR (P-ASR)

5.1 System Description

The P-ASR is implemented using Hidden Markov Model Toolkit (HTK) version 3.4[5].The Linux operating system Ubuntu version 11.10 has been used for developing the P-ASR. In addition to these Java platform is used for building a graphical user interface to make the system more interactive, fast and user friendly. The system is trained with 115 distinct Punjabi words and word model is used for the recognition.

5.2 System Architecture & Implementation

The P-ASR system architecture, as shown in Fig. 2, mainly comprises of four components, namely, Training data preparation, Acoustical analysis, Acoustic model generation and GUI based decoder.

5.2.1 Training Data Preparation

This phase consists of recording and labeling the speech signal. The implemented system is trained for 115 distinct Punjabi language words. The data is recorded with the help of a unidirectional microphone using a recording tool *audacity* in *.wav* format. The *.wav* files recorded are saved as HTK transcription.

The sampling rate used for recording is 16 kHz. Eight speakers recorded the data and each word is uttered 3 times in a data file and 3 samples of each speaker are recorded. So the 115 distinct words resulted in (115*3) samples of 8 distinct speakers files making a total of 2760 (115*3*8) files. A labeling tool *wave surfer* is used to label the speech waveforms. As each word is uttered three times in a file so labeling format is having seven successive regions: start silence, recorded word, silence, recorded word, silence, recorded word and end silence. The labeled file saved in *.lab* format is a simple text file and these are used in acoustic model generation phase of the system.

5.2.2 Acoustic Analysis

The speech recognition tools cannot process directly on speech waveforms. These have to be represented in a more compact and efficient way. This step is called acoustical analysis. The original waveform is converted into a series of acoustical vectors. Mel Frequency Cepstral Coefficient (MFCC) technique has been used for feature extraction. The computation steps of MFCC include:

Framing: The signal is segmented in successive frames (whose length is chosen between 20ms and 40ms, typically), overlapping with each other.

Windowing: Each frame is multiplied by a windowing function (e.g. Hamming function).

Extracting: A vector of acoustical coefficients (giving a compact representation of the spectral properties of the frame) is extracted from each windowed frame.

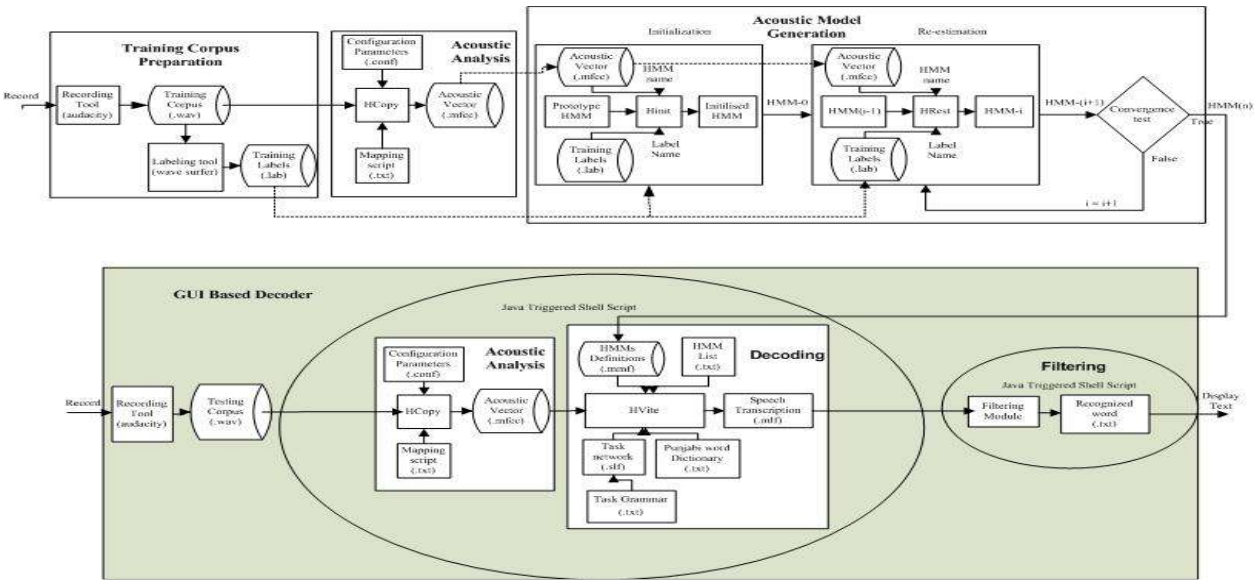

Fig. 2: System Architecture

Configuration file (*.conf*) is a text file which specifies the various configuration parameters such as format of the speech files (HTK), technique for feature extraction(MFCC), length of time frame(25msec), frame periodicity(10msec), number of MFCC coefficients(12) etc. The Acoustic Vector (*.mfcc*) files are used in both training and decoding phase of the system. The ***HCopy*** tool of HTK is used for this purpose.

5.2.3 Acoustic Model Generation

An acoustic model is defined as a reference model to which comparisons are made to recognize unknown utterances. There are two kinds of acoustic models viz. word model and phoneme model. Word model has been used as it is suitable for small vocabulary and the statistical approach Hidden Markov Modeling (HMM) for system training. In this phase of implementation, first HMM initialization is done using a prototype. This prototype has to be generated for each word in the dictionary. Same topology is used for all the HMMs and the defined topology consists of 4 active states (observation functions) and two non emitting states (initial and the last state with no observation function). Single Gaussian distributions with diagonal matrices are used as observation functions and these are described by a mean vector and variance vector in a text description file known as prototype. This pre-defined prototype along with Acoustic vector (*.mfcc* files) and Training labels (*.lab* files) is used by HTK tool ***HInit*** for initialization.

In the second step of this phase's implementation, HTK tool ***HRest*** is used for estimating the optimal values for the HMM parameters (transition probability, mean and variance vectors for each observation function). This iterative step is known as re-estimation and this is repeated several times for each HMM to train. These embedded re-estimations indicate the convergence through the change measure (convergence factor).This final step of acoustic model generation phase, known as convergence test, is repeated until absolute value of convergence factor does not decrease from one HRest iteration to another. In our system implementation re-estimation iteration are repeated for five times. So five HMMs per word in the vocabulary are generated.

5.2.4 Task Definition

Before entering the final stage of testing the developed system, the basic architecture of recognizer i.e. language model (*task grammar*) and Punjabi word dictionary i.e. Pronunciation model (*task dictionary*) are to be defined. The task grammar, specified using extended Backus-Naur form (EBNF), is written in a text file. The task grammar is compiled with HTK tool ***HParse*** to generate the task network (*.slf*). The task dictionary that is also a text file develops a correspondence between the name of the HMM and name of the task grammar variable. The names of the labels are also added in the above correspondence as these names indicate the symbols that will be output by the recognizer. These names are treated as optional, if not given, the names of grammar variables are used by default for the output purpose.

5.2.5 System Testing

This stage is responsible for generating transcription for an unknown utterance [10]. Like the training corpus preparation the testing signal is also converted into series of acoustic vectors (*.mfcc*) using HTK tool HCopy. This input observation along with HMMs definition, Punjabi word dictionary, task network and names of generated HMMs (HMM list) is taken as input by HTK tool ***HVite*** to generate the output in a transcription file (*.mlf*). The HVite tool processes the signal using Viterbi Algorithm, based on token passing algorithm, which matches it against the recognizer's Markov models. The transcription file is then processed by a filtering module which extracts the recognized word from the file and displays it in the form of text.

To make the system more fast and user interactive, a Graphical User Interface is developed with two buttons *record* and *display*. The user just clicks on record button and records the sound signal using a microphone and just after clicking on display button the recognized output is displayed. As described in Fig. 2 above, this implementation has been done by using two Java platform triggered shell scripts. In the first shell script the HTK tools HCopy and HVite are triggered and then its output is filtered by the commands of second shell script and text output is displayed to the user. Hence the implemented system is more abstract and fast.

5.2.6 Performance Analysis

The system performance is analyzed by HTK tool ***HResult***. The output transcription file of the HVite tool is compared with the corresponding original reference transcription file. The following equations show the formula for evaluating performance of speech system where N is the number of words in test set, D is the number of deletions, S is number of substitutions and I is the number of insertions.

$$\text{Percentage Correct(PC)} = (\text{N} - \text{D} - \text{S})/\text{N} \times 100$$

where PC in above equation gives word correction rate.

$$\text{Percentage Accuracy(PA)} = (\text{N} - \text{D} - \text{S} - \text{I})/\text{N} \times 100$$

where PA in above equation gives word accuracy rate.

$$\text{Word } Error \text{ Rate(WER)} = 100\% - \text{Percentage Accuracy}$$

where Word Error Rate(WER) in above equation is used as one of the criterion to evaluate the performance of the system.

6. Recognition Results

The performance of the system is tested against speaker independent parameter by using two types of speakers: one who are involved in training and testing both and the other who are involved in only testing. The second parameter for checking system performance is different environments. The system is tested in a class room and in open space. A total of 6 distinct speakers are used for this and each one is asked to utter 35-50 words. The Table 4 to 7 shows the evaluation results of the P-ASR. The results shown reveal that the implemented system performs well with different speakers and in different environments. The average performance of the system lies in the range of 94 % to 96% with word error rate 4% to 6%.

Table-4: Recognition in class room environment by speakers involved both in training and testing

Speaker	Environment	No. of Spoken Words	No. of recog-nized Words	PC	PA	WER
Speaker 1	Class room	38	37	97.36	97.36	2.64
Speaker 2	Class room	42	42	100	100	0
Speaker 3	Class room	50	48	96	96	4
Total		130	127	97.78	97.78	2.12

Table-5: Recognition in class room environment by speakers involved only in testing

Speaker	Environment	No. of Spoken Words	No. of recog-nized Words	PC	PA	WER
Speaker 4	Class room	40	37	92.5	92.5	7.5
Speaker 5	Class room	37	34	91.89	91.89	8.11
Speaker 6	Class room	46	44	95.65	95.65	4.35
Total		123	115	93.49	93.49	6.51

Overall performance in a class room environment as described by Table 4 & Table 5 = **95.63%**

Table-6: Recognition in open space environment by speakers involved in training and testing both

Speaker	Environment	No. of Spoken Words	No. of recog-nized Words	PC	PA	WER
Speaker 1	Open space	39	38	97.43	97.43	2.57
Speaker 2	Open space	34	32	94.11	94.11	5.89
Speaker 3	Open space	38	36	94.73	94.73	5.27
Total		111	106	95.49	95.49	4.51

Table-7: Recognition in open space environment by speakers involved only in testing

Speaker	Environment	No. of Spoken Words	No. of recog-nized Words	PC	PA	WER
Speaker 4	Open space	40	36	90	90	10
Speaker 5	Open space	37	35	94.59	94.59	5.41
Speaker 6	Open space	46	43	93.47	93.47	6.53
Total		123	114	92.68	92.68	7.32

Overall performance in open space environment as described by Table 6 & Table 7 = **94.08%**

7. Conclusion and Future work

In conclusion, an efficient, abstract and fast ASR system for regional languages like Punjabi is need of the hour. The work implemented in the paper is a step towards the development of such type of systems. The work may further be extended to large vocabulary size and to continuous speech recognition. As shown in results, the system is sensitive to changing spoken methods and changing scenarios, so the accuracy of the system is a challenging area to work upon. Hence, various speech enhancements and noise reduction techniques may be applied for making system more efficient, accurate and fast.

Acknowledgments

The authors would like to thank to Mr. Kuldeep Kumar Garg and Mr. Gaurav Leekha for their kind cooperation and for useful discussions during implementation of the work described in the paper.

References:

[1] L.R. Rabiner , "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", Proc. of the IEEE Vol. 77, Issue 2,pp. 257–286,1989.

[2] R. Rabiner, and B. H. Juang, Fundamentals of Speech Recognition, Prentice-Hall International, New Jersey, 1993.

[3] R. Rabiner, and B. H. Huang, "An introduction to hidden markov models," IEEE Acoust. Speech Signal Processing Mag., pp. 4-16, 1986.

[4] R. K. Aggarwal, and M. Dave "Acoustic modeling problem for automatic speech recognition system: conventional methods (Part I)" International journal Speech Technology, Springer, Vol.14, issue 2, 2011.

[5] HTK "Hidden Markov Model Toolkit", available at http://htk.eng.cam.ac.uk,2012.

[6] S. Young, "Hidden Markov Model Toolkit: Design and Philosophy," CUED/F-INENG/TR.152, Cambridge University Engineering Department, Sept. 1994.

[7] R. Kumar and M. Singh, "Spoken isolated Word Recognition of Punjabi Language Using dynamic time Warping Technique" Demo in Proceedings of Information System for Indian Languages, Punjabi University, Patiala, India, March 9 - 11, 2011. Vol. 139 of Communication in Computer and Information Science (CCIS), Page 301, Springer Verlag.

[8] R. Kumar "Comparison of HMM and DTW for Isolated Word Recognition of Punjabi Language" In Proceedings of Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, Sao Paulo, Brazil. Vol. 6419 of Lecture Notes in Computer Science (LNCS), pp. 244– 252, Springer Verlag, November 8-11, 2010.

[9] B. A. Q. Al-Qatab and R. N. Ainon, "Arabic Speech Recognition Using Hidden Markov Model Toolkit (HTK)", Paper presented at International Symposium in Information Technology (ITSim). Kuala Lumpur, June 15-17, 2010..

[10] K. Kumar and R. K. Aggarwal "Hindi Speech Recognition System using HTK" International journal of Computing and Business Research ISSN Vol. 2 issue 2 May 2011

[11] K. Kumar, R. K. Aggarwal, and A. Jain "A Hindi speech recognition system for connected words using HTK" International Journal Computational Systems Engineering, Vol. 1, No. 1, 2012.

[12] R. Reddy, Spoken Language Processing: A guide to Theory Algorithm, and System Development, Prentice-Hall, New Jersey, 2001.

[13] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, The HTK Book, Microsoft Corporation and Cambridge University Engineering Department, 2009.

[14] SPHINX,Sphinx, available at http://cmusphinx.sourceforge.net/html/cmusphinx.php, 2011.

[15] S. Young , N.H Russell, and J.H.S Thornton, Token Passing: A Conceptual Model for Connected Speech Recognition Systems, Technical Report, Department of Engineering, Cambridge University, Cambridge, UK, 1989.

[16] Anusuya, M. A., & Katti, S. K.. Front end analysis of speech recognition: A review. International Journal of Speech Technology,Springer, Vol.14, pp. 99–145, 2011.

Mohit Dua did his B.Tech. degree in Computer Science and Engineering from Kurukshetra University, Kurukshetra, INDIA in 2004 and M.Tech degree in Computer Engineering from National Institute of Technology, Kurukshetra, INDIA in 2012. He is presently working as Assistant Professor in Department of Computer Engineering at NIT Kurukshetra, INDIA with more than 7 years of academic experience. He is a life member of Computer Society of India (CSI) and Indian Society for Technical Education (ISTE). His research interests include Speech processing, Theory of Formal languages and Statistical modeling.

R.K. Aggarwal received his M.Tech. degree in 2006 and is pursuing Ph.D. from National Institute of Technology, Kurukshetra, INDIA. Currently, He is working as an Associate Professor in the Department of Computer Engineering of the same Institute. He has published more than 30 research papers in various International/National journals and conferences and also worked as an active reviewer in many of them. He has delivered several invited talks, keynote addresses and also chaired the sessions in reputed conferences. His research interests include speech processing, soft computing, statistical modeling and science and spirituality. He is a life member of Computer Society of India (CSI) and Indian Society for Technical Education (ISTE). He has been involved in various academic, administrative and social affairs of many organizations having more than 20 years of experience in this field.