Speaker Adaptive Model for Hindi Speech using Kaldi Speech Recognition toolkit

Preprint · February 2018

DOI: 10.13140/RG.2.2.36716.05765

5 authors, including:

Prashant Upadhyaya
Chandigarh University
28 PUBLICATIONS **212** CITATIONS
SEE PROFILE

Sanjeev Kumar Mittal
Indian Institute of Science
6 PUBLICATIONS **22** CITATIONS
SEE PROFILE

Yash Vardhan
Indian Institute of Technology Bombay
15 PUBLICATIONS **50** CITATIONS
SEE PROFILE

Omar Farooq
Aligarh Muslim University
184 PUBLICATIONS **1,711** CITATIONS
SEE PROFILE

Speaker Adaptive Model for Hindi Speech using Kaldi Speech Recognition toolkit

Prashant Upadhyaya*, Sanjeev Kumar Mittal†, Yash Vardhan Varshney*, Omar Farooq*, Musiur Raza Abidi*
*Department of Electronics, Aligarh Muslim University - Aligarh, Uttar Pradesh, 202002, India
Email: upadhyaya.prashant@rediffmail.com, rsr.skm@gmail.com,, omarfarooq70@gmail.com
†Electrical Communication Engineering, Indian Institute of Science, Bangalore, Karnataka, 560012, India

***Abstract*—Speech communication is fast gaining market penetration as a preferable input for human computer interface (HCI) and is finding its way into the commercial applications from the academic research setup. For public applications, acceptance is determined not only by the accuracy and reliability but the ease of usage and habituation. In this work, we show that accuracy of a system can be enhanced using Speaker Adaption Technique (SAT). Kaldi speech recognition toolkit was used to evaluate the performance of our Hindi speech model. Acoustic feature were extracted using MFCC and PLP from 1000 phonetically balanced Hindi sentence from AMUAV corpus. Acoustic model was trained using Hidden Markov Model and Gaussian Mixture Models (HMM-GMM) and decoding was performed using Weight Finite State Transducers (WFSTs). Maximum improvement of 6.93% in word error rate is obtained for speaker adaptive training when used along with Linear Discriminant Analysis-Maximum Likelihood Linear Transform model over monophone model.**

***Index Terms*—Kaldi ASR, Weight Finite State Transducers, MFCC, PLP, HCI, SAT, Speech recognition.**

I. Introduction

The field of speech recognition in human computer interaction (HCI) is fast penetrating its way into commercial applications. Speech is one of the prominent interfaces, which is used in most computers and smart phone that support automatic dictation and transcription. Nevertheless there is still much scope in the field of modeling the speech signal in order to boost the system performance.

Another challenge is to test the models on various tools available for the speech recognition research, such as HTK [1],Julius [2], Sphinx [3], RWTH [4] and Kaldi [5] [6]. HTK is one of the notable speech recognition toolkit for research purpose for last few decades.

Kaldi speech recognition toolkit has become the current state-of-the-art tool that is based on theme of "Low Development cost, High Quality Speech Recognition" for prominent languages and domains [5]. The advantage of Kaldi over HTK, is the flexibility and clean structure code that is easy to understand and better weighted finite state transducer (WFST) and math support [6] [7].

Speech recognition model in Kaldi is based on finite-state transducer along with core library written in C++ to support flexible code that is easy to understand [5]. Kaldi is actively maintained under Apache License v2.0, which is suitable of wide community user-base.

Commercially, it has been observed that the success rate increase if the speech based application are designed for their native language usage. This make the device more user-friendly. In India, Hindi is the largest spoken language and in world it is the fourth most spoken language followed by Mandarin, Spanish and English [8]. Therefore, hand held devices and smart phone based applications could have more efficient interactive voice response system (IVRS) in native languages. In India it is better to use Hindi speech as an interface for controlling or accessing these applications.

However, the research conducted till now on Hindi language mainly focuses on Hindi digit recognition [9]–[11] or continuous Hind speech sentence which has small vocabulary size [12]–[16]. In [14], Hindi speech recognition using MFCC feature along with Probabilstic neural network (PNN) is reported. Experiments were performed using two Hindi speech signals which contain about twenty sponken words only.

Performance for connected Hindi speech recognition was shown in [15] by varying the size of the vocalbulary words from 135 to 600 words. In their experiment,performance was evaluated using eight state GMM (Gaussian mixture model) model on HTK tool kit. Overall accuracy 93% was achieved with MFCC feature. Biswas et al. [16] perform continuous Hindi speech recognition using ANOVA fusion technique. Features are extracted by calculating harmonic energy from wavelet feature. Experiment was conducted on 91 speakers selected from Hindi speech database [17]. Experimental results reported the phoneme recognition accuracy (PRA) of 74.30% using MFCC features under clean environment.

In this paper, speaker adaptive model for continuous Hindi speech recognition using Kaldi toolkit is evaluated. MFCC and PLP features are extracted from AMUAV corpus. AMUAV corpus consists of 100 speakers and each speakers utterance the 10 short Hindi sentence from which two sentences are common to each speaker. Thus, 1000 continuous Hindi speech database is prepared which is phonetically balanced [13].

II. Acoustic and Language Modeling

Acoustic modelling (AM) and Language modelling (LM) are considered as a core of any automatic speech recognition system. The purpose is to find the most likely word sequence $w^\star$ from a given acoustic feature $\boldsymbol{X}$ as described in equation 1.

$$w^\star = \arg\max_i \{P(w_i|\boldsymbol{X})\} \tag{1}$$

where w_i is the i'th vocabulary word. Computing $P(w_i|\boldsymbol{X})$ directly is difficult so at this stage Bayes' Rule is applied and the expression is transformed to

$$P(w_i|\boldsymbol{X}) = \frac{P(\boldsymbol{X}|w_i)P(w_i)}{P(\boldsymbol{X})} \tag{2}$$

Therefore, most probable word for a give probabilities $P(w_i)$ depends only on the likelihood $P(\boldsymbol{X}|w_i)$ so equation 2 can be reduced to

$$P(w_i|\boldsymbol{X}) = \arg\max_i \{P(\boldsymbol{X}|w_i)P(w_i)\} \tag{3}$$

Finally, work of AM is to estimate the parameters θ of a

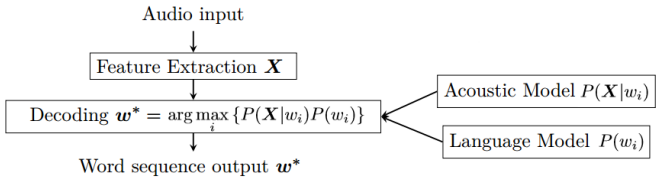

Fig. 1. The architecture of an automatic speech recognition model.

model so that the probability $P(\boldsymbol{X}|w_i;\theta)$ is as accurate as possible. Fig. 1 shows the architecture structure of automatic speech recognition model. On the other hand Language model (LM) represents the probability of $P(w_i)$ [1].

As from Fig. 1, initially, audio signal is pre-emphasized and processed frame by frame over 20-30 ms long windows for obtaining feature vectors $\boldsymbol{X}$. Finally, these features vector act as an input to the decoder where decoding is performed frame by frame using beam search.

Beam search (prune low probability hypotheses) remove nodes in time state trellis which are unlikely to succeed. Using beam search means that only hypotheses over a certain threshold (δ) are kept which result in best path only [6] [7]. Therefore, it is likely to expand the hypothesis frame wise using information from previous frames when moving forward in time to the next frame for selecting N-best path (list of N most probable hypotheses). After reaching to last frame, the best hypothesis (highest probability) will result in word sequence output $w^\star$. A hypothesis is defined as the acoustic model output, i.e., a probable word sequence. Probabilities for hypotheses are computed using the acoustic model and the language model.

A. Feature Extraction

Feature extraction is mainly used to reduce the huge amount of raw speech data into a set of useful information, aiming to solve the dimensionality problem. MFCC and PLP are the two most efficient feature extraction technique for speech recognition.

MFCC is the most common feature for many speech recognition systems which is based on auditory filter banks using cosine transform that roughly follow the human auditory system. Fig. 2 shows the complete arrangement of equally space triangular filter bank. Mel-Frequency Cepstral Coefficients

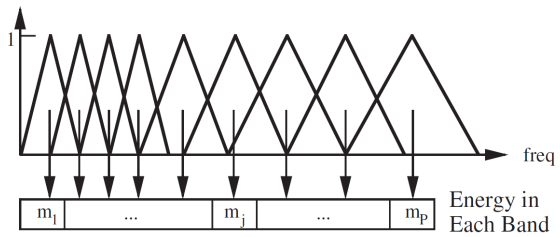

Fig. 2. Mel-Scale Filter Bank [1].

(MFCCs) are computed from the log filterbank amplitudes $\{m_j\}$ using the Discrete Cosine Transform (DCT) given as

$$c_i = \sqrt{\frac{2}{N}} \sum_{j=1}^{N} m_j \cos\left(\frac{\pi i}{N}(j-0.5)\right) \tag{4}$$

where N is the number of filterbank channels.

The log-energy of the signal is computed for speech samples $\{s_n, n = 1, N\}$

$$E = log \sum_{n=1}^{N} s_n^2 \tag{5}$$

Complete analysis of MFCC feature extraction is shown in Fig. 3. Perceptual Linear Prediction (PLP) coefficients are an alternative to MFCC. After obtaining MFCC and PLP features,

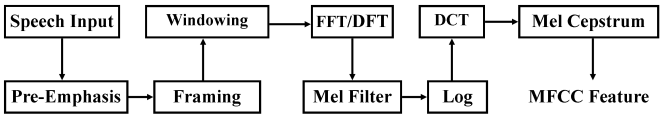

Fig. 3. MFCC feature extraction

Cepstral Mean and Variance Normalization (CMVN) [18] per speaker is computed on the extracted features. CMVN are used to minimize the effect of noise such as ambient noise and recording equipment [18]. They are calculated by subtracting mean of each and divide with the standard deviation which efficiently minimizes any stationary noise. CMVN are calculated using the equation 6 below:

$$\hat{x}_t(i) = \frac{x_t(i) - \mu_t(i)}{\sigma_t(i)} \tag{6}$$

where $x_t(i)$ is the i^{th} component of the original feature vector at time t. Mean $\mu_t(i)$ and standard deviation $\sigma_t(i)$ are calculated over some sliding finite window of length N given as

$$\mu_t(i) = \frac{1}{N} \sum_{n=t-N/2}^{n=t+N/2} x_n(i) \tag{7}$$

$$\sigma_t^2(i) = \frac{1}{N} \sum_{n=t-N/2}^{n=t+N/2} (x_t(i) - \mu_t(i))^2 \tag{8}$$

To include the temporal evolution of feature, additional feature values Δ and $\Delta\Delta$ are computed.

$$\triangle x_t = \frac{\sum_{i=1}^{n} w_i(x_{t+i} - x_{t-i})}{2\sum_{i=1}^{n} w_i^2} \tag{9}$$

where x_t is a delta coefficient at time t computed in terms of the corresponding static coefficients x_{t-i} to x_{t+i} and where n is the window width and w_i are the regression coefficients. In similar fashion, second order derivative $\Delta\Delta$ is calculated.

$$\triangle^2 x_t = \frac{\sum_{i=1}^{n} w_i(\triangle x_{t+i} - \triangle x_{t-i})}{2\sum_{i=1}^{n} w_i^2} \tag{10}$$

Finally, combined feature vectors are used for speech recognition

$$x_t = \begin{bmatrix} x_t & \triangle x_t & \triangle^2 x_t \end{bmatrix} \tag{11}$$

Also, to improve the performance of speech recognition, original feature space is projected into lower dimension space with the help of Linear Discriminant Analysis (LDA). Further, results can be improved if Maximum Likelihood Linear Transform (MLLT) is applied diagonally, on the features. It has been shown that LDA transform works best when a diagonalizing MLLT is applied afterwards.

Speaker Adaptive Training (SAT) is also used to further enhance the accuracy of the speech recognition model. In SAT, for each individual speakers, training small acoustic model from the training data is performed as shown in Fig. 4. Thus, accuracy of system can be further increased using

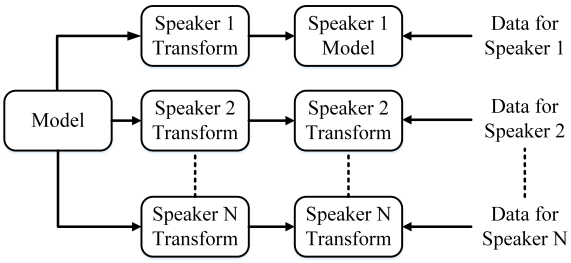

Fig. 4. Speaker Adaption model for Hindi Speech.

(LDA+MLLT+SAT) [6]. Fig. 5 shows the complete work flow of feature extraction using Kaldi ASR model.

B. Acoustic Modelling

The Acoustic Model (AM) estimates the likelihood $P(\boldsymbol{X}|w_i;\theta)$. The model parameters θ is found through training. As it is difficult to find the exact word-time alignment of the utterance, level of uncertainty in the training increases. This can be resolved using Hidden Markov Model (HMM), which reduce the uncertainty between acoustic features and corresponding transcription. HMM provides a statistical representation of the sound of the words.

On the other hand when performing speech recognition tasks for large dataset, word is not a feasible choice of training as an increased vocabulary would result in problem of dimensionality too large to handle. Solution to this problem can be solved

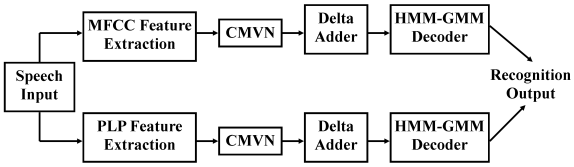

Fig. 5. Complete workflow for recognition perform using Kaldi ASR model.

by training the model using the phone. A phone is defined as the smallest unit of speech.

Further, accuracy of the system can be increased by training the model with triphones instead of monophones. A triphone is a sequence of three phones and it captures the context of the single middle phone very efficiently. For given N monophone there are N^3 potential triphones. Similar triphones are tied together using state-tying thereby reducing dimensionality. In Kaldi the state-tying is performed using decision trees. Each triphone is modelled by a Hidden Markov Model.

HMM have chain of state in which first and last state are called the non emitting state. Each state in the HMM corresponds to one frame in the acoustic input. The model parameters that are estimated in the acoustic training are $\theta = [a_{ij}, b_j(\boldsymbol{o}_t)]$, where a_{ij} corresponds to transition probabilities from i_{th}state to j_{th}state and $b_j(\boldsymbol{o}_t)$ to output observation distributions. Below is the formula for computing $b_j(\boldsymbol{o}_t)$

$$b_j(\boldsymbol{o}_t) = \prod_{s=1}^{S}\left[\sum_{m=1}^{M_s} c_{jsm}\mathcal{N}(x;\boldsymbol{\mu}_{jsm},\boldsymbol{\Sigma}_{jsm})\right]^{\gamma_s} \tag{12}$$

where M_s is the number of mixture components in stream s. c_{jsm} weight of the m'th component and $\mathcal{N}(\cdot;\boldsymbol{\mu},\boldsymbol{\Sigma})$ is the mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ of multivariate Gaussian given as

$$\mathcal{N}(\boldsymbol{o};\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^n|\boldsymbol{\Sigma_j}|}} e^{-\frac{1}{2}(\boldsymbol{o_t}-\boldsymbol{\mu_j})'\boldsymbol{\Sigma_j}^{-1}(\boldsymbol{o_t}-\boldsymbol{\mu_j})} \tag{13}$$

where n is the dimensionality of $\boldsymbol{o}$.

Here, the observation vectors at time t gets split into a number of S independent data streams o_{st}. The exponent γ_s is a stream weight. The prior probabilities satisfy the probability mass function constraints

$$\sum_{m=1}^{M} c_{jsm} = 1, \quad c_{jsm} \geq 0 \tag{14}$$

If M is set to one, a single Gaussian distribution is obtained. In the training of a GMM, the aim is to update the mean μ_{jsm} and covariance Σ_{jsm}.

C. Language Modelling

Generally, language models are used to estimate the probability of given word sequence, $\hat{P}(w_1, w_2, \ldots, w_m)$. Thus evaluating $P(w_i)$ as defined in Equation 3. Probability estimation of n-gram model is based on, counting events in context for

given training set using maximum likelihood estimation given as

$$\hat{P}(w_i|w_{i-n+1},\ldots,w_{i-1}) = \frac{C(w_{i-n+1},\ldots,w_i)}{C(w_{i-n+1},\ldots,w_{i-1})} \quad (15)$$

here $C(.)$ is count of given word sequences in the training text.

An n-gram referred to as n words, symbol, etc present in given corpus. Given its $n-1$ predecessors, the n-gram language model (LM) is used to predict each symbol in the sequence.

D. Speech Decoder

Here the objective is to find the most likely word sequence $w^{\star}$ from a given acoustic feature $\boldsymbol{X}$ as described in equation 1. Finally, the decoder search for the most appropriate phone sequence which corresponds to word using monophone and triphones model. Using language model, the final word hypothesis obtained using acoustic model are verified using Language Model Weight, w_{lm}. Then the equation becomes

$$w^{\star} = P(w_i|\boldsymbol{X}) = \arg\max_i \left\{P(\boldsymbol{X}|w_i)P(w_i)^{lm}\right\} \quad (16)$$

Finding the corresponding word sequence that maximizes the above equation 16 is the task of the decoder. This problem can be tackled and solved using search algorithms. Kaldi solves the search task using word lattices.

In Kaldi, Weight Finite State Transducers (WFSTs) is used for training and decoding algorithm [6], which provide the well structure graph operation to improve the acoustic model. Thus, providing "pdf-ids" that assign a numeric value to a decoding graph that corresponds to context-dependent states. Decoding can be improved by combining he "transistion-id" with the arc (transition) in a topology specified for that phone [6], [7]. Then the decoding is performed on so called decoding graph HCLG which is constructed from simple FST graphs as given in equation 17 [5]–[7]. .

$$HCLG = H \circ C \circ L \circ G \quad (17)$$

The symbol $\circ$ represents an associative binary operation of composition on WFST. Here, G is an acceptor that encodes the grammar or language model, L represents the lexicon (its input symbols are phones and output symbols are words), C represents the relationship between context-dependent phones on input and phones on output and H contains the HMM definitions, that take as input id number of PDF and return context-dependent phones.

Finally, ASR performance is measured in term of Word Error Rate (WER) which is computed at word level using Levenshtein distance [20]. It allows the minimal number of substitutions, deletions and insertions to make two strings equal. The performance of the system is measured in term of word error rate (WER) defined as follows:

$$WER(\%) = \frac{(D+S+I)}{N} * 100(\%) \quad (18)$$

where *N* is the number of word used in the test, *D* is the number of deletions, *S* is the number of substitutions and *I* is the number of insertion error.

III. Data Preparation for Kaldi ASR

Data preparation for AMUAV Hindi speech [13] for Kaldi ASR was reported in [21], along with all the meta-data of each speakers that were used for training and testing the acoustic and language models. Complete Kaldi directories structure for AMUAV data preparation is created in Kaldi-trunk (main Kaldi directory) as shown in Fig. 6 which was trained using 900 sentences.

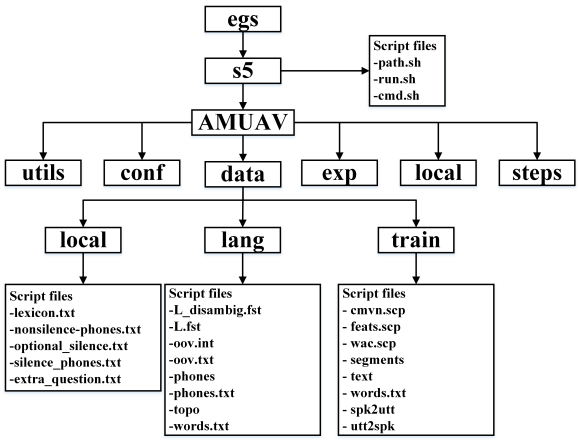

Fig. 6. Kaldi directories structure for AMUAV corpus

IV. Experimental Approach

The machine configuration on which experiment was conducted runs Ubuntu 16.04 LTS (64-bit operating system) on Intel Core 7 Duo Processor at 2.20 GHz. Experimental results on AMUAV corpus is reported which consists of 1000 phonetically balanced Hindi sentences that are spoken by 100 speakers. Context-dependent triphone system with simple GMM-HMM model was developed. For experimental purpose N-gram model (i.e., N = 2, 3 and 4) was used for the recognition. Table I shows the performance of WER for

TABLE I
WER PERFORMANCE FOR MONOPHONE, TRIPHONE, LDA+MLTT AND LDA + MLLT + SAT USING MFCC FEATURE

Feature	2-gram	3-gram	4-gram
Monophone	18.44	18.32	17.57
Triphone	14.85	14.73	15.59
LDA + MLLT	13.37	12.62	12.62
LDA + MLLT + SAT	11.63	11.39	11.51

MFCC feature using monophone, triphone, LDA+MLTT and LDA+MLTT+SAT training model using 2-gram, 3-gram and 4-gram LM model. As seen form the Table I, for MFCC feature, best performance is shown for speaker adaptive model (SAT) for all n-gram language model. Best recognition rate was achieved at 3-gram language model except monophone trained model (best result was achieved at 4-gram model). Increasing of the LM model to 4-gram value, degrades the performance of speech recognition model.

Similarly, results for PLP feature are shown in Table II.

Again from Table II speaker adaptive training have shown best performance than other feature transform model. However, best result for triphone and LDA + MLLT + SAT have shown better performance at 4-gram language model for PLP features.

TABLE II
WER PERFORMANCE FOR MONOPHONE, TRIPHONE, LDA+MLTT AND LDA + MLLT + SAT USING PLP FEATURE

Feature	2-gram	3-gram	4-gram
Monophone	20.67	19.80	19.80
Triphone	14.48	14.36	13.74
LDA + MLLT	14.85	14.36	14.60
LDA + MLLT + SAT	11.63	10.89	10.77

V. CONCLUSION

In this paper, speaker adaptive training model for continuous Hindi speech recognition using AMUAV corpus is reported using Kaldi toolkit. It is found that SAT model enhances the performance of speech recognition system. For fair comparison, two well known acoustic features MFCC and PLP are reported. Further, it is shown that for MFCC feature improvement in word error rate of 6.93% was obtained at 3-gram language model and 8.91% for PLP feature. However, best recognition for PLP feature was obtained at 4-gram language model. This is due to the fact that, when SAT model are trained with PLP features, due to which less sensitive frequency components comes on the same equal loudness curve which make features independent and may result in improvement. In future, our task is to train the model using Deep Neural Network (DNN) and compare the performance of the automatic speech recognition model with our existing results.

VI. ACKNOWLEDGMENT

The authors would like to acknowledge Institution of Electronics and Telecommunication Engineers (IETE) for sponsoring the research fellowship during this period of research.

REFERENCES

[1] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, "*The HTK Book (for version 3.4)*". Cambridge University Engineering Department, 2009.

[2] A. Lee, T. Kawahara, and K. Shikano, "Julius - an open source realtime large vocabulary recognition engine," in *EUROSPEECH*, 2001, pp.1691-1694.

[3] W. Walker, P. Lamere, P. Kwok, B. Raj, R. Singh, E. Gouvea, P. Wolf, and J. Woelfel, "Sphinx-4: A flexible open source framework for speech recognition," Sun Microsystems Inc., Technical Report SML1 TR2004-0811, 2004.

[4] D. Rybach, C. Gollan, G. Heigold, B. Hoffmeister, J. Loof, R. Schluter, and H. Ney, "The RWTH Aachen University Open Source Speech Recognition System," in *INTERSPEECH*, 2009, pp. 2111-2114.

[5] Kaldi Home Page (kaldi-asr.org)

[6] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, and J. Silovsky, "The Kaldi speech recognition toolkit", In IEEE 2011 workshop on automatic speech recognition and understanding (No. EPFL-CONF-192584), IEEE Signal Processing Society, 2011.

[7] D. Povey, M. Hannemann, G. Boulianne, L. Burget, A. Ghoshal, M. Janda, M. Karafit, S. Kombrink, P. Motlek, Y. Qian, and K. Riedhammer, "Generating exact lattices in the WFST framework", In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4213-4216, 2012.

[8] http://www.internationalphoneticalphabet.org.

[9] T.Pruthi, S. Saksena and P. K. Das, "Swaranjali: Isolated word recognition for Hindi language using VQ and HMM", In International Conference on Multimedia Processing and Systems (ICMPS), IIT Madras. 2000.

[10] O. Farooq, S. Datta and A. Vyas, "Robust isolated Hindi digit recognition using wavelet based de-noising for speech enhancement", Journal of Acoustical Society of India, 33(1-4) pp. 386-389, 2005.

[11] A. Mishra, M. Chandra, M. Biswas and S. Sharan, "Robust Features for Connected Hindi Digits Recognition", International Journal of Signal Processing, Image Processing and Pattern Recognition, 4(2), pp. 79-90, 2011.

[12] V. Chourasia, K. Samudravijaya, M. Ingle and M. Chandwani, "Hindi Speech Recognition under Noisy Conditions", International Journal of Acoustic Society India, pp. 41-46, 2007.

[13] P. Upadhyaya, O. Farooq, M. R. Abidi, and P. Varshney, "Comparative Study of Visual Feature for Bimodal Hindi Speech Recognition",In Archives of Acoustics, 40(4), pp. 609-619, 2015.

[14] P. Sharma, A. Singh, and K. K. Singh, "Probabilistic neural networks for Hindi speech recognition", In Communication and Computing Systems: Proceedings of the International Conference on Communication and Computing Systems (ICCCS 2016), Gurgaon, India, 9-11 September, 2016, p. 463. CRC Press,

[15] S. Sinha, S. S. Agrawal, and A. Jain, "Continuous density hidden morkov model for context dependent Hindi speech recognition, Int. Conference on Advances in Computing, Communication and Informatics (ICACCI), pp. 1953-1958, IEEE, 2013.

[16] A. Biswas, P. K. Sahu and M. Chandra, "Admissible wavelet packet sub-band based harmonic energy features using ANOVA fusion techniques for Hindi phoneme recognition," In IET Signal Processing 10, no. 8 (2016): 902-911.

[17] K. Samudravijaya, P.V.S. Rao, and , S.S. Agrawal,"Hindi speech database." Proc. Int. Conf. on Spoken Language Processing (ICSLP00), Beijing, China, October 2002, pp. 456459.

[18] O. Viikki and K. Laurila, "Cepstral domain segmental feature vector normalization for noise robust speech recognition", In Speech Communication, 25(1):133147, 1998.

[19] S. Molau, F. Hilger and H. Ney, "Feature space normalization in adverse acoustic conditions", In Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP03). 2003 IEEE International Conference on, volume 1, pages I656. IEEE, 2003.

[20] G. Navarro, "A guided tour to approximate string matching",In ACM computing surveys (CSUR), 33(1):3188, 2001.

[21] P. Upadhyaya, O. Farooq, M. R. Abidi and Y. V. Varshney, "Continuous Hindi Speech Recognition Model Based on KALDI ASR Toolkit", In International Conference on Wirelesss Communication, Signal Processing and Networking, IEEE WiSPNET, pp. 812-815, 2017.