Mohit Dua*, Rajesh Kumar Aggarwal and Mantosh Biswas

Discriminative Training Using Noise Robust Integrated Features and Refined HMM Modeling

https://doi.org/10.1515/jisys-2017-0618
Received December 4, 2017; previously published online February 20, 2018.

Abstract: The classical approach to build an automatic speech recognition (ASR) system uses different feature extraction methods at the front end and various parameter classification techniques at the back end. The Mel-frequency cepstral coefficients (MFCC) and perceptual linear prediction (PLP) techniques are the conventional approaches used for many years for feature extraction, and the hidden Markov model (HMM) has been the most obvious selection for feature classification. However, the performance of MFCC-HMM and PLP-HMM-based ASR system degrades in real-time environments. The proposed work discusses the implementation of discriminatively trained Hindi ASR system using noise robust integrated features and refined HMM model. It sequentially combines MFCC with PLP and MFCC with gammatone-frequency cepstral coefficient (GFCC) to obtain MF-PLP and MF-GFCC integrated feature vectors, respectively. The HMM parameters are refined using genetic algorithm (GA) and particle swarm optimization (PSO). Discriminative training of acoustic model using maximum mutual information (MMI) and minimum phone error (MPE) is preformed to enhance the accuracy of the proposed system. The results show that discriminative training using MPE with MF-GFCC integrated feature vector and PSO-HMM parameter refinement gives significantly better results than the other implemented techniques.

Keywords: Automatic speech recognition, MFCC, GFCC, genetic algorithm, PSO, PLP, discriminative training, MMI, MPE.

1 Introduction

Current advancements in automatic speech recognition approaches have resulted in highly efficient and accurate automatic speech recognition (ASR) systems [8, 20, 38]. The prime objective in implementing a real-time accurate ASR system is to reduce the mismatch between the training and the testing phase. An efficient implementation of the training phase results in ASR systems with better accuracy rate [20]. Feature extraction from a pre-processed spoken utterance and acoustic modeling of the extracted features are the two sub-phases of the training phase of the ASR system [4, 5, 31, 35]. The development of an efficient feature extraction method and an accurate acoustic modeling approach has been an area of prime research in ASR over the last five decades [1, 2, 21, 24, 27].

The feature extraction part of an ASR system has a vital role to perform in the overall accuracy of the system. Linear predictive cepstral coefficients (LPCCs) [36], Mel-frequency cepstral coefficients (MFCC) [11], perceptual linear prediction (PLP) [16], and wavelets [39] are some of the proposed feature extraction techniques in the last few decades [6]. Out of the various proposed methods, the MFCC and PLP are the most commonly used for speech recognition systems because of their high accuracy and low computation overhead.

***Corresponding author: Mohit Dua,** Department of Computer Engineering, National Institute of Technology, Kurukshetra, India, e-mail: er.mohitdua@nitkkr.ac.in. https://orcid.org/0000-0001-7071-8323
Rajesh Kumar Aggarwal and Mantosh Biswas: Department of Computer Engineering, National Institute of Technology, Kurukshetra, India

These methods perform quite well in a noise-free environment. However, the performance of these approaches tends to degrade in the presence of additive noise. In the recent years, some researchers have shown that the gammatone-frequency cepstral coefficient (GFCC) features are robust against noise and acoustic change [2, 30, 41, 44, 45]. Researchers have also proposed sequential combinations of these approaches like MF-PLP [6] and RASTA-PLP [6, 21] to obtain better results than the traditional methods. The proposed work sequentially combines MFCC and GFCC features to obtain noise robust heterogeneous features.

In the mid-1980s, acoustic modeling using the hidden Markov model (HMM) was proposed as a substitute to template matching-based acoustic modeling [35]. Such HMM-based ASR systems used multi-word sentence-driven speaker representations [32]. However, the requirement of a large amount of training data with no intra-speaker variations has always been a challenge in the development of a robust continuous HMM-based ASR systems [14]. Many other techniques of acoustic modeling have also been suggested using the Gaussian mixture models (GMM) and support vector machine (SVM) classifiers [29]. Recently, optimization methods like particle swarm optimization (PSO) [22], differential evaluation (DE) [40], and genetic algorithm (GA) [17] have been applied to refine the HMM parameters [7, 21]. Also, discriminative training methods like MMI [7, 43] and MPE [34] have been used in the last two decades to increase the accuracy of the HMM-based ASR systems [3, 12, 15].

The proposed work mainly contributes in three ways. Initially, it integrates the MFCC-GFCC features and compares the performance of the integrated feature vector with the MFCC, PLP, GFCC, and MF-PLP feature vector. Second, it does features refinement using two different optimization techniques, GA and PSO. Last, it applies the refined heterogeneous vector to a discriminative trained acoustic model built using the MMI and MPE. The remaining part of the paper is organized as follows: Section 2 briefly describes the fundamentals of feature extraction, optimization methods, and discriminative training techniques. Section 3 gives details of the proposed architecture, Section 4 deals with details of the Hindi language speech corpus, Section 5 gives the simulation and experiment analysis, and Section 6 concludes the proposal.

2 Preliminaries

2.1 Feature Extraction

The acoustic speech input signals must be accurately and reliably represented to develop a robust ASR system. A lot of work has been done and is still being carried out on this area of speech recognition. This sub-section of the paper describes the feature extraction methods used to implement the proposed system.

2.1.1 Mel-Frequency Cepstral Coefficients (MFCC)

The Mel-frequency cepstral coefficients (MFCC) has been used by the researchers as an established and proven method to extract distinct characteristics of input speech signal [18, 35]. The process for MFCC feature extraction includes the following steps:

- Pre-emphasis of input speech signal is performed to amplify the energy at high frequencies [10]. It not only reduces the difference in power components of the signal but also distributes power across the relative frequencies. As a result, the high frequencies are more prevalent in the pre-emphasized signal.
- The samples of the pre-emphasized signal are multiplied by a Hamming window function to divide the signal into discrete portions and to minimize any signal discontinuities [10, 26].
- After windowing, the discrete Fourier transform (DFT) is applied to have magnitude and the phase representation of the windowed signal.
- Frequency wrapping using the logarithmic Mel scale is applied to convert spectrum frequencies to smaller numbers. The filter bank spacing follows the Mel-frequency scale that is mathematically expressed as:

$$\text{Mel}(f) = 259 \log_{10}(1 + f/700) \tag{1}$$

– The inverse DFT of the Mel Spectrum is performed to have the 12 MFCC coefficients and one energy coefficient. The information that provides unique characteristics of the waveform is contained by the 12 MFCC coefficients. The first and second derivatives of the MFCC coefficients are calculated and also included to capture frame to frame changes in the signal. Along with the MFCC feature extraction, the total energy of the input frame is also calculated.

2.1.2 Perceptual Linear Prediction (PLP)

The key concept behind perceptual linear prediction is to improve LPCC performance while simultaneously reducing their computational complexity. The critical band analysis, equal loudness pre-emphasis, intensity-loudness conversion, and inverse discrete Fourier transform (IDFT) in sequence are applied to the input speech signal to generate PLP coefficients from the linear prediction coefficients (LPCs). Like the MFCC, the PLP also has 39 features to represent the extracted meaningful information. However, it uses trapezoidal filters and cube root compression instead of the MFFC's triangular filter and logarithmic compression. In the PLP, the use of the LPC model and 17 infinite impulse response (IIR) band pass filters boosts the performance of the ASR system in noisy conditions [16]. It is often integrated with the relative spectral transform (RASTA) to reduce the impact of channel distortion and any type of background noise [6, 23]. The method is named as RASTA-PLP method for extracting features.

2.1.3 Gammatone-Frequency Cepstral Coefficients (GFCCs)

One of the biggest challenges for an ASR system is to perform well in real-time acoustic environments. Hence, noise sensitivity is an important parameter for a good feature extraction technique. One of the major demerits of MFCC is that it is sensitive to additive noise. The GFCC is a more comprehensive model based on the equivalent rectangular bandwidth (ERB) scale and a set of gammatone filter banks. The recent works reveal that GFCC is more noise robust and performs better than MFCC [44, 30]. To extract the GFCC feature, the following steps are performed:

– The input speech signal is multiplied with the gammatone filter bank in the frequency domain. A gammatone filter with a center frequency f can be defined as:

$$g(f, t) = at^{n-1}e^{-2\pi bt}\cos(2\pi ft + \Phi) \tag{2}$$

where a is a constant, Φ denotes the phase, and n defines the order of the filter. The value of n is usually set to less than 4, and Φ is set to the value of zero. The factor b of equation (2) is mathematically expressed as:

$$b = 25.17\left(\frac{4.37f}{1000} + 1\right) \tag{3}$$

– Like MFCC, the pre-emphasis step is executed to highlight the more prominent frequency components that carry the speech signal's vital information, and windowing is applied to minimize signal discontinuities.
– Logarithmic operation is performed, and the discrete cosine transform (DCT) is then applied to obtain the 12 uncorrelated cepstral coefficients. Finally, the first- and second-order derivatives are taken resulting in a total of 36 GFCC features.

2.1.4 Integrated Features

Koehler et al. [23] first introduced the key idea of integrated features in the year 1994. The features of the feature extraction scheme RASTA are integrated with the PLP to obtain the RASTA-PLP in their research work.

Recently, the proposed ASR systems in Refs. [6] and [21] used the integration of the MF and PLP features. This work named this MF-PLP integration as a "*heterogeneous feature vector*". However, Zhao and Wang in Ref. [44] performed an interesting analysis of the noise robustness feature of the MFCC and GFCC for speaker identification and proved that the non-linear rectification of the GFCC is the key to noise robustness. Burgos in Ref. [9] used MFCC-GFCC combination for his proposed work and proved that combination performs significantly better. The proposed work also exploits sequential integration of the MFCC features with the PLP and GFCC. However, it uses heteroscedastic linear discriminant analysis (HLDA) [25] used in Ref. [6] to reduce the number of features instead of the principal component analysis (PCA) [13] used in Ref. [9]. Earlier proposed works clearly reveal that HLDA outperforms the other feature extraction methods [19, 46]. The target of the optimal HLDA is to maximize the log-likelihood of the entire training samples denoted objective function $\breve{\boldsymbol{F}}$, where $\breve{\boldsymbol{F}}$ and log-likelihood are given by equations (4) and (5), respectively:

$$\breve{\boldsymbol{F}} = \arg\max_{F} £(\boldsymbol{F};\{\boldsymbol{t}_s\}) \tag{4}$$

$$£(\boldsymbol{F};\{\boldsymbol{t}_s\}) = \sum\nolimits_{\forall} \log p(t_s) \tag{5}$$

where $p(t_s)$ denotes the probability density of a training sample t_s, and $\boldsymbol{F}$ is the transform matrix obtained from HLDA.

Figure 1 shows the steps followed to compute the MFCC, PLP, GFCC, MF-PLP, and MF-GFCC feature extraction methods.

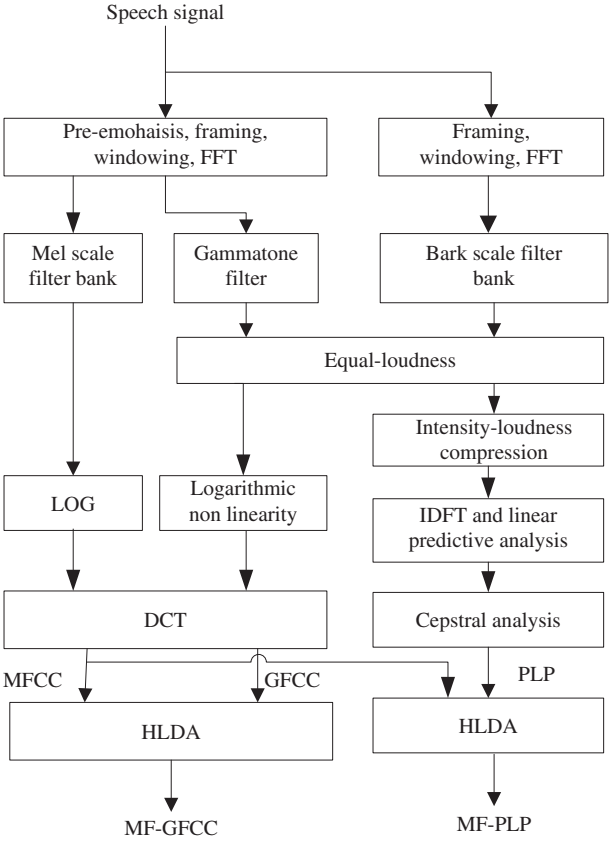

Figure 1: Proposed Feature Vector Integration.

2.2 HMM Refinement

2.2.1 Genetic Algorithms (GAs)

The genetic algorithms (GAs) are a type of evolutionary approach and were first introduced in the year 1975 by John Holland [17]. It is defined as search techniques based on the idea of the natural selection. GAs have the power to generate an elementary population of possible solutions and have a very high ability to find the best solutions among all solutions. In each iteration, the strong solution tends to acclimatize and sustain, while the weak solution tends to diminish. GA is defined as a robust search method that tries to produce the optimal results while making no assumption about the problem space.

The probability and randomness are the two basic characteristics of the Gas and, hence, make the GAs suitable for HMM refinement. The key parameters to be considered while using the GAs for HMM refinement are defined as [20]:

- Population size refers to the number of features taken into consideration in each feature vector.
- Population initialization refers to the initial feature population that is chosen randomly from the set of extracted features.
- Fitness evaluation refers to the fitness function evaluation using mean and variance variables.
- Crossover refers to the integrating of individual feature vectors to generate new feature vectors.
- Mutation refers to making alteration in the existing feature vector to generate new feature vectors.

2.2.2 Particle Swarm Optimization (PSO)

Like the GAs, the particle swarm optimization (PSO) is also a population-based optimization method and a type of evolutionary approach. It also uses random initial population of feasible solution and looks for the optimal solution in iterations. It was first introduced in the year 1995 by Kennedy and Eberhart [22]. However, it differs from the GA in the fact that it does not use any evolution functions like crossover and mutation. In the PSO, the possible solutions are named as particles. These particles follow the currently known optimum solutions in the problem space.

The PSO for the HMM refinement starts by initializing with a group of random speech features with particles X_k and velocity V_k. It then looks for the best features in the iterations. Each feature vector is updated using the P_{best} and G_{best} values in every iteration. P_{best} is the best fitness solution achieved by the algorithm so far. G_{best} is defined as the best value obtained so far by any particle in the population. After finding the P_{best} and G_{best}, the particle updates its velocity and positions using equations (6) and (7) [21].

$$V_k^{i+1} = wV_k^i + c_1 r_1 (Pbest_k^i - X_k^i) + c_2 r_2 (Gbest^i - X_k^i) \tag{6}$$

$$X_k^{i+1} = X_k^i + V_k^{i+1} \tag{7}$$

where i denotes the iteration; r_1 and r_2 denote the uniformly distributed random variables; c_1 and c_2 are the acceleration constants, and w denotes the inertia weight.

2.3 Discriminative Techniques

Discriminative training approaches are used to determine the HMM parameters in such a manner that the error rate could be reduced in the training data [7, 34, 43]. Discriminative techniques significantly enhance the recognition accuracy of the large-vocabulary ASR system. This sub-section describes the MMI and MPE discriminative techniques.

2.3.1 Maximum Mutual Information (MMI)

The MMI training is an alternative to the maximum likelihood estimation (MLE) technique that targets the optimization of mutual information between a spoken utterance and an observation sequence [12, 28, 33]. The objective function of the MMI is mathematically expressed as:

$$f_{\mathrm{MMI}}(\lambda) = \frac{1}{R}\sum_{r=1}^{R} \log \frac{P_{\lambda}(u_r|t_r)P(t_r)}{\sum_{t} P_{\lambda}(u_r|t)P(t)} \tag{8}$$

where t_r represents the correct transcription of the spoken utterance u_r, $P(t)$ is the language model probability, and f is a scalar function of the parameters λ of the HMM set.

The MMI objective function divides the probability of the correct transcription by the sum of all possible transcription probabilities. The objective function is maximized by decreasing the sum of the denominator term and increasing the numerator term. The denominator term can be decreased by reducing the sum of all possible transcription probabilities [43]. Unlike the MLE, the MMI gives a higher weight to training utterances that has low posterior probability of correct word sequence. The estimation of the model parameters is done by the extended Baum–Welch (EBW) algorithm [33]. The MMI technique has three major issues: first, it is tough to maximize the objective function; second, it is computationally expensive; and finally, it shows poor generalization to unseen data [42].

2.3.2 Minimum Phone Error (MPE)

The MPE is based on the minimum Baye's risk training. The only difference between MMI and MPE is in the computation of the probabilities of the numerator and denominator terms of the objective function [33, 42, 43]. However, it holds the merit of phone or word-level modeling over the MMI. In the MPE, the occupation probabilities are computed by an approximate error measure for every phone marked for the denominator. The objective function of the MPE is:

$$f_{\mathrm{MPE}}(\lambda) = \sum_{r=1}^{R} \frac{\sum_{s} P_{\lambda}(u_r|t)P(t)R(t, t_r)}{\sum_{o} P_{\lambda}(u_r|o)P(o)} \tag{9}$$

where $R(t, t_r)$ denotes the raw phone transcription accuracy.

The MPE performs better in comparison to the MMI discriminative technique because it supports word transcriptions with the best phone accuracy [15].

3 Proposed Architecture

An automatic speech recognition system comprises two major modules, i.e. front end and back end. The front end involves feature extraction, refinement of features, acoustic modeling, and the back end involves decoding.

In the proposed architecture, the feature vectors are generated using various feature extraction algorithms and techniques as discussed above. The feature vectors affect the parameters of the acoustic model and, in turn, optimize the various factors affecting the training phase of the ASR system. The process of speech recognition in this proposed system is accomplished by the following steps: first, feature vectors are generated using the MFCC, PLP, GFCC, MFCC + PLP, MFCC + GFCC, and these feature vectors are refined by applying the GA and PSO optimization. Second, the HMM-based acoustic model is generated using the number of Gaussian mixtures, and this HMM-GMM model is optimized using the MMI and MPE discriminative

techniques. Finally, decoding is performed using information from the language model, acoustic model, and pronunciation model.

3.1 Pre-processing and Feature Extraction

Initially, the input speech signal is parameterized using various feature extraction techniques; these are the MFCC, PLP, and GFCC. A feature vector lays emphasis on the information needed for the task and suppresses all other types of information. The MFCC is the most common used feature extraction method in the ASR. Various other methods were developed later to increase the efficiency of the system. Figure 2 gives the proposed architecture for the HMM-GMM-based ASR system using various feature extraction methods.

3.1.1 Mel-Frequency Cepstral Coefficients (MFCCs)

To extract a feature vector containing all information about the speech signal, the MFCC uses some parts of speech production and speech perception. The MFCC tries to eliminate speaker-dependent characteristics by excluding the fundamental frequency [35]. Initially, the input signal is divided into frames, which contain arbitrary number of samples. Each time frame is then distributed in a different Hamming window to eliminate discontinuities at the edges. The operation is performed using equation (10):

$$W_{fc}(c)=\begin{cases}0.54-0.46\cos\left(\dfrac{2\pi c}{N}-1\right), & 0\le c\le N-1\\ 0 & \text{, Otherwise}\end{cases} \tag{10}$$

where, $W_{fc}(c)$ is the filter coefficient of the Hamming window, N denotes the total number of samples, and c refers to the current sample.

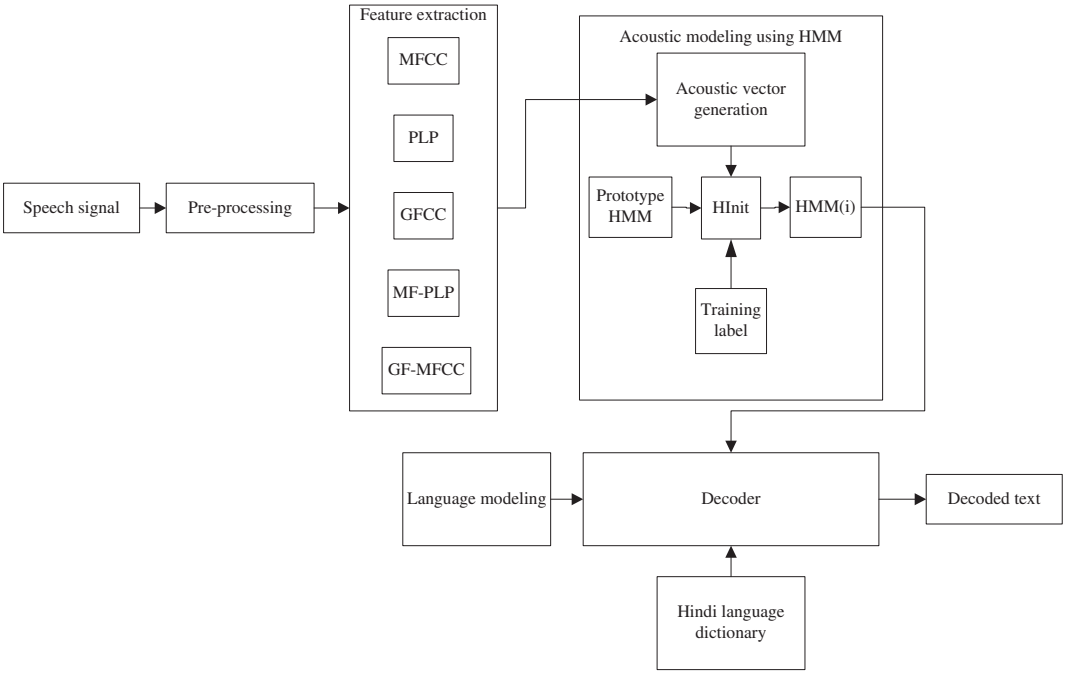

Figure 2: Proposed HMM-GMM-Based ASR System Using Different Feature Extraction Methods.

After the windowing operation, to segregate the energy comprised into each frequency band, fast Fourier transformation (FFT) is used. FFT is calculated for each frame to extract the frequency components of the input speech signal. This is achieved by reckoning the discrete Fourier transform given by equation (11).

$$v_{t,i,0} = \left| \frac{1}{N} \sum_{k=1}^{N-1} \left(e^{-j2\pi \frac{ki}{N}} \right) v_k \right| \tag{11}$$

where $i = 0, 1, 2, \ldots, (N/2) - 1$, t is the time frame, N is the number of sampling points within a time frame t, and $v_{t,i,0}$ is the vector obtained after applying the DFT.

The spectrum obtained by the DFT is filtered with a different band pass filter, and the power of the individual frequency band is enumerated. This is needed to estimate the power spectrum. The enumeration of the spectrum band is as follows:

$$v_{t,k,1} = \sum_{i=0}^{\frac{N}{2}-1} z_{k,i} v_{t,i,0} \tag{12}$$

where $k = 0, 1, 2, \ldots, N_d$ denotes the number of band pass filters, z is the amplitude of the band pass filter with the index k and frequency i, and $v_{t,k,1}$ denotes the obtained spectrum band.

The typical filter bank uses a triangular-shape band pass filter to compute the Mel frequency spectrum. The cepstral coefficients are computed using the FFT obtained using equation (12). The Fourier transformed frame is passed through the logarithmic Mel-scaled filter bank. The relation between the Mel scale and the frequency of the speech signal is given in equation (1). Using equations (1) and (12), in $v_{t,k,2}$ is obtained.

The discrete cosine transform is used for metamorphosing the Mel coefficients back to the time domain. The results obtained by the DCT generates the MFCCs. The DCT of $v_{t,k,2}$ is computed to obtain $v_{t,k,3}$ as

$$v_{t,k,3} = \sum_{k=1}^{N_d} \left(\cos \left[\frac{i(2k-1)\pi}{2N_d} \right] (v_{t,k,2}) \right) \tag{13}$$

where $k = 0, 1, 2, \ldots, N_c < N_d$, and N_c denotes the number of cepstral coefficients selected for further processing.

Generally, the first 13 coefficients are taken for the further representation of the signal. The obtained cepstral coefficients are extended using the first- and second-order derivatives. For the inclusion of the dynamic nature of the speech, first- and second-order derivatives are used. It represents the dynamic nature of speech.

The first-order derivative is obtained as follows:

$$v'_{t,k,3} = v_{t+1,k,3} - v_{t-1,k,3} \tag{14}$$

The second-order derivative is obtained as follows:

$$v''_{t,k,3} = v'_{t+1,k,3} - v'_{t-1,k,3} \tag{15}$$

A MFCC feature vector consists of 13 cepstral coefficients, 13 first- and 13 second-order derivatives. The final feature vector contains 39 coefficient values.

$$v_t = [v_{t,k,3}, v'_{t,k,3}, v''_{t,k,3}] \tag{16}$$

3.1.2 Gammatone Frequency Cepstral Coefficients (GFCC)

It is designed to simulate the process of human hearing system. The major difference between the MFCC and GFCC is its filter bank. The gammatone filter bank is group of filters that has a high impulse response similar to the magnitude characteristic of human auditory filter. The initial operations such as windowing and Fourier transform are performed similarly as the MFCC using equations (10) and (11). The produced output after the Fourier transform $v_{t,i,0}$ is passed through the gammatone filter bank. Using equations (2) and (12), the $v_{t,k,2}$ is obtained. The DCT is then applied to obtain the unrelated cepstral coefficients as in equation (13).

$$v_{t,k,3}(n,\ r)=\left(\frac{2}{P}\right)^{0.5}\sum_{j=0}^{P-1}\left\{\frac{1}{3}\log(v_{t,k,2}(n;i)\cos\left[\frac{\pi r}{2P}(2j-1)\right]\right\} \tag{17}$$

where p is the number of channels in the filter bank.

Thus, the first 12 components are then selected to obtain a GFCC feature vector that consists 12 cepstral coefficients, 12 first- and 12 second-order derivatives. The final feature vector v_t contains 36 coefficient values. The first- and second-order derivatives are computed using equations (13) and (14), respectively.

3.1.3 Perceptual Linear Prediction (PLP)

The PLP uses the first two steps similar to the MFCC and GFCC, i.e. windowing and FFT. The computed frequency value further undergoes the Bark filter bank process. The Bark Filter contains a filter bank with 27 very sharp band pass filters. The Bark frequency corresponding to a speech signal is given by:

$$v_{t,k,2}=6\ln\left[\frac{v_{t,k,1}}{1200\pi}+\left[\left(\frac{v_{t,k,1}}{1200\pi}\right)^2+1\right]^{0.5}\right] \tag{18}$$

The obtained Bark frequency component is used in the pre-emphasis process of the equal loudness emphasis step. In this method, each power spectrum coefficient is calculated and multiplied with a weight for equal loudness. In the PLP technique, the function used for equal loudness is similar to the pre-emphasis process of the MFFC feature extraction computation. The outcome of the emphasis process is used in LP (linear prediction). The relation between discrete input power spectrum $v_{t,k,2}$(m) and the LP model power spectrum $\hat{v}_{t,k,2}$(m) is given as:

$$\frac{1}{N}\sum_{m=1}^{M-1}\left(\left[\frac{v_{t,k,2}(m)}{\hat{v}_{t,k,2}(m)}\right]\right)=1 \tag{19}$$

After the LP, recursive cepstrum computation is applied to get the PLP coefficients. The first 13 coefficients are obtained, and using equations (14)–(16), the PLP feature vector v_t is obtained.

3.1.4 MF-PLP

In this system, the features of the MFCC and PLP are combined to overcome the drawbacks of both techniques MFCC and PLP [6]. All the 13 features of the MFCC from equation (13) and the four top best features of the PLP from equation (19) are combined, forming the 17 features. The first- and second-order derivatives are taken from all the 17 features, forming the 51 features. These 51 features are reduced using the HLDA technique to get the 39 final features.

3.1.5 MF-GFCC

Earlier researches reveal that the GFCC and MFCC perform better than the PLP. However, the GFCC outperforms the MFCC in the noisy environment. Hence, it is beneficial to subsume the benefits of these two approaches to reduce their individual drawbacks. All the 13 feature components extracted from equation (16) and the four top best feature components from equation (17) are combined to obtain the MF-GFCC feature vector. In this proposed approach, the static 17 features, 17 first-derivative features, and 17 second-derivative features are combined to get the 51 features. The combined 51 features are then reduced to 39 dimensions by the HLDA technique to form the standard feature vectors.

Algorithm 1 : Genetic ($GA_{population\_size}$, $GA_{population\_fraction}$, $GA_{mutation\_rate}$).

Begin
Initialization//Initialization of random population
$l = 0$
$GA_{population,l}$ = a random population generated using feature vector v_t
Evaluation//Computes fitness of each particle in population
Compute fitness (l) $\forall$ l $\in$ $GA_{population}$
do
{
Selection //selection of pairs of fittest parents
GA_p = select Parent (fitness(l), $GA_{population,l}$, $GA_{population\_size}$)
$GA_c = \varnothing$
Crossover and Mutation//generating offspring From $GA_{parents}$
For (GA_{p1}, $GA_{p2} \in GA_p$)
$GA_{offspring1}$, $GA_{offspring2}$ = Crossover (GA_{p1}, GA_{p2}, $GA_{population\_fraction}$)
GA_c = Mutate ($GA_{offspring1}$, $GA_{offspring2}$, $GA_{mutation\_rate}$)
end
Compute fitness (k) $\forall$ k $\in$ GA_c
GA_{global_best} = max_fitness (GA_c)
$GA_{population,l+1} = GA_{population,l} + GA_c$
}
while (v_t is not refined)
return GA_{global_best}
end

3.2 Genetic Algorithm (GA)

The feature vector, thus, extracted from speech by various methods is then refined through various methods GA and PSO. The GA is a population-based algorithm and used for very complex problems. Algorithm 1 gives the pseudo-code of the GAs to obtain refined features.

The algorithm uses the population size $GA_{population\_size}$ of the random feature vector, fraction of the population $GA_{population\_fraction}$, and rate of the mutation $GA_{mutation\_rate}$ as the initial input parameters. Initially, the population of the feature vector $GA_{population,l}$ is initialized, and the fitness of each feature vector in the population is computed. Then, the algorithm selects the fittest pair of parents GA_p and generates the offspring $GA_{offspring1}$, $GA_{offspring2}$. The mutation of the $GA_{offspring1}$ and $GA_{offspring2}$ generates children GA_c Finally, it mutates new offspring and their fitness is computed, and the best solution from the existing population is generated. The generated new offspring are added to the population to have a new population for the next iteration.

3.3 Particle Swarm Optimization (PSO)

Like the GA, the PSO is also a population-based optimization method and a type of evolutionary approach. It also uses random initial population of a feasible solution and looks for the optimal solution in iterations. The PSO for the HMM refinement starts by initializing with a group of random speech features with particles and their velocity. It then looks for the best features in iterations. Algorithm 2 shows how PSO is applied to obtain a refined feature vector.

Initially, the algorithm uses the population size $PSO_{population\_size}$ of a random feature vector and initializes the population of a feature vector $PSO_{population}$. Then, it computes the fitness of each feature vector in a population and assigns an individual fitness value to each feature. From the population, the feature

Algorithm 2: PSO ($PSO_{population\_size}$).

Begin
Initialization//Initialization of random population
l = 0
$PSO_{population}$ = a random population generated using feature vector v_t
$PSO_{global\_best} = \varnothing$
$PSO_{local\_best[\,PSO_{population\_size}\,]} = \varnothing$
Fitness computation//Computes fitness of each particle in population
do
{
Compute fitness (l) $\forall$ l $\in PSO_{population}$
$PSO_{local\_best[l]} = l$
If (fitness ($PSO_{local\_best[l]}$) > PSO_{local_best})
$PSO_{global\_best} = PSO_{local\_best[l]}$
l = l + 1
}
while (l < $PSO_{population\_size}$)
reinitialize l = 0
do
{
newValue = updateFeature (v_t, PSO_{global_best}, $PSO_{local\_best[l]}$)
if (fitness (newValue) > fitness ($PSO_{local\_best[l]}$)
{
$PSO_{local\_best[l]}$ = newValue
If ($fitness(PSO_{local\_best[l]}) > fitness(PSO_{global\_best})$
$PSO_{global\_best} = PSO_{local\_best[l]}$
}
l = l + 1
}
while (v_t is not refined)
return PSO_{global_best}
end

vector having the best fitness is found, and the global best PSO_{global_best} is assigned. Finally, the value of the features is updated, and the global best PSO_{global_best}, and local best PSO_{local_best} values are computed iteratively.

3.4 Generation of HMM Model

It is the step of mapping a feature vector to various HMM states. Acoustic modeling is used to generate symbols where each symbol represents a HMM state. These symbols are further used in the recognition process and matched against unknown symbols. For the HMM modeling, vector quantization is used to cluster vectors into classes [6, 35]. Each class represents a HMM state, and a symbol is defined for each class. For each class, the probability is computed by each HMM state. These states carry information from the extracted feature vector v_t that is then associated with the HMM states. A uniform number of Gaussian mixtures are applied to each HMM state. The proposed system HMM model $\lambda = (A, B, \pi)$ consists of five fundamental elements: the number of HMM states, numbers of symbols generated per HMM state, one-step HMM state transition probability for transition from state *i* to *j*, probability distribution of observation symbol in state *i*, and initial state distribution π.

Using these fundamental elements, a five-state HMM prototype of each Hindi language lexicon is created. The prototype contains two hidden states (initial and final). The prototype file exploits the refined parameters from the feature vector v_t. The HMM uses the Baum–Welch algorithm and the MLE for training

purposes. However, the development of an acoustic model with the use of the HMM only is not realistic in nature. The HMM model is based on the independent output assumption and first-order assumption. The independent output assumption assumes that the current state observation is independent of all previous observations, and the first-order assumption assumes that the next state is dependent only upon the current state. Hence, discriminative training is used to overcome these weaknesses of the HMM. Figure 3 illustrates the proposed architecture of the discriminative trained ASR system using the refined HMM parameters and optimized feature vector.

3.5 Discriminative Training

Discriminative training is a type of parameter estimation method that uses the extended Baum–Welch (EBW) algorithm [33]. It uses the language model and word transcription to create numerator and denominator lattices. These lattices are phone marked, and the EBW algorithm updates the parameters. There are various approaches for discriminative training: the minimum classification error (MCE), MPE, MMI, etc. [28]. All the approaches use an objective function, and as the scope of this objective function increases, recognition error decreases.

The discriminative techniques used in the proposed work MMI and MPE use objective functions from equations (8) and (9), respectively. The major difference between both the methods is the way of computing the numerator and denominator lattices. The discriminative training approach uses the HMM model to generalize the output of the probability distribution. A weak language model creates the lattices needed for the MMI and MPE. The numerator lattices use correct lexicons, and the denominator lattice implements confusable hypotheses [33, 42]. The training is, thus, completed using the tool HMMIRest of Hidden Markov Model Toolkit (HTK) developed by Cambridge University Engineering Department, Cambridge, USA. This tool interpolates the language model and the grammar scale factor into the acoustic models to form a numerator acoustic model and a denominator acoustic model. Various iterations of the HMM model λ are computed, and the updated parameters are used in the decoding phase.

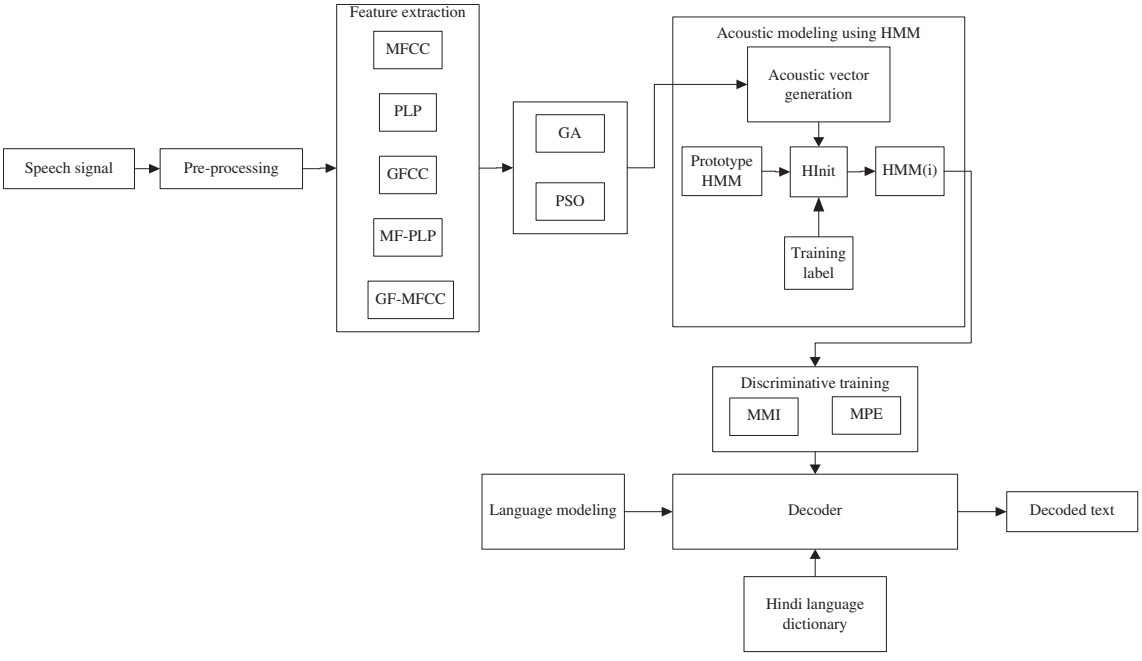

Figure 3: Proposed Architecture for Discriminatively Trained ASR System Using Optimized Features.

3.6 Decoding

The decoding module after the training phase creates an output matrix using the Viterbi algorithm. The Viterbi algorithm gives the best single state sequence from the whole observation sequence [35]. Like the training phase, the recognition phase also uses extracted feature vectors. A tri-gram language model using the extended Backus–Naur form (EBNF) has been used for the implementation. The best possible sequence of word is identified using the Viterbi algorithm. The pattern matching is done by the compilation of the trained HMM model and language model. The main focus of the Viterbi algorithm is that at every time t and for state S_i, the most probable state is determined in dependence of the most probable state sequence at time $t-1$ and the transition probability $= \{a_{ij}\}$. The Viterbi algorithm comprises the following steps:

- Initialization: This step initializes the probability vector and path vector.
- Recursion: This step is performed using dynamic programming, and a path vector is calculated.
- Termination: This step refers to finding the best probable sequence with respect to spoken utterance.
- Backtracking: In this step, the optimal state sequence, in an observation sequence with maximum probability, is calculated.

During testing, the output signal is matched according to the input speech features.

4 Hindi Language Speech Corpus

A well-annotated and time-aligned speech database developed for the Hindi language is used in this proposed work [37]. The database contains sentences that contain almost all phonemes and rich in phonemic context. The database contains 1000 sentences of 100 speakers of which 38 are those whose mother tongue is Hindi, and the rest of the others speak Hindi fluently. Out of 10 sentences uttered by every speaker, two sentences are common to all speakers, and these two "dialect" sentences contain the maximum phone of the Hindi language. The next eight sentences also try to meet the maximum phones of the Hindi language. The speech data is digitally recorded with 16-kHz sampling frequency, using two microphones in the noise-free environment. Training has been done by randomly selecting 80 speakers, of which 55 are male and 25 are female speakers. The system uses the remaining 20 speakers out of the 100 for testing purposes.

5 Simulation Details and Experiment Results

The simulations and experiments have been carried out using MATLAB R2015a and HTK 3.5 β-2 version toolkit. The feature extraction part is done using various libraries and inbuilt methods of the MATLAB toolbox. The acoustic model part and decoding algorithms are implemented using the HTK Toolkit.

Out of the 100-speaker speech database of Section 4, 80-speaker speech database has been used as a training dataset (Trainset), and the remaining 20-speaker speech database has been used as a testing dataset (Testset). The performance of the developed system is analyzed using three distinct training datasets Trainset1, Trainset2, and Trainset3 and three distinct testing datasets Testset1, Testset2, and Testset3. Trainset1 of the training dataset contains the speaker data of persons who speak Hindi frequently and are from the northern region of India. Trainset2 of the training dataset consists of the speaker data of persons who speak Hindi less frequently and are from the southern region of India. Trainset3 of the training dataset contains the speaker data from both Trainset1 and Trainset2. All three samples have different vocals (male and female).

The testing dataset is also divided into three types of test samples Testset1, Testset2, and Testset3. Testset1 of the testing dataset contains the male speaker data only. Testset2 of the training dataset contains the female speaker data only. Testset3 of the training dataset contains the speakers from both Testset1 and Testset2. Testset1 contains 12 male speakers, Testset2 contains eight female speakers, and Testset3 contains

all 20 speakers. The remaining of this section discusses the obtained results using the implemented ASR system.

This performance analysis of the developed system is done using it to transcribe some pre-recorded test words, and the recognizer output is matched with the correct reference transcriptions. Equation (20) gives the mathematical expression for evaluating the performance of the speech system.

$$\text{Accuracy rate (Ar)} = (W_n - W_D - W_S - W_I)/W_n \times 100 \tag{20}$$

where W_n denotes the number of words in the test set, W_D represents the number of words deleted, W_S refers to the number of substituted words, and W_I is the number of words inserted.

5.1 Comparative Analysis of Feature Extraction Methods

Table 1 shows the comparative analysis of all the feature extraction method using the HMM-GMM acoustic modeling. Two hundred fifty and six Gaussian mixtures per HMM state have been used to develop the baseline HMM system. The results in Table 1 clearly show that the GFCC-HMM and MF-GFCC-HMM-based systems outperform all the other feature extraction methods. It can be seen that Trainset3 with Testset3 performs better than all the other combinations, where both samples include north and south Indian male, female speech utterances.

5.2 Comparative Analysis With Different SNRs

The comparative analysis of the GFCC and MF-GFCC feature extraction methods with the HMM-GMM acoustic modeling is done using different signal-to-noise ratio (SNRs). The results in Table 2 clearly show that the MF-GFCC-HMM-based system performs better than the GFCC-HMM-based system, and increased SNR results in an increased accuracy rate.

5.3 Comparative Analysis Using Refined Features and Discriminative Methods

The HMM-based acoustic model is also discriminatively trained using the MMI and MPE methods, and the extracted features are optimized using the optimization algorithms GA and PSO. Initially, the MMI and MPE are applied with the optimized GFCC, and finally, both discriminative methods are applied with the optimized MF-GFCC to analyze the performance of the implemented system. The results in Table 3 show that the MF-GFCC-MPE-based system performs better than the MF-GFCC-MMI system.

The parameters used for the GA crossover rate, mutation rate, population size, and number of iterations are fixed at values 0.9, 0.2, 117, and 55, respectively. The performance of these optimization algorithms is

Table 1: Comparative Analysis of Feature Extraction Methods.

Training dataset	Test dataset	GFCC	MF-GFCC	MFCC	PLP	MF-PLP
Trainset1	Testset1	70.85	72.42	64.36	62.40	66.45
	Testset2	69.20	70.96	66.50	64.86	68.36
	Testset3	70.15	72.04	65.86	64.90	68.40
Trainset2	Testset1	69.66	70.65	66.87	65.10	69.25
	Testset2	71.54	72.26	68.20	66.36	68.45
	Testset3	72.36	73.80	69.56	67.15	67.05
Trainset3	Testset1	69.32	71.23	62.74	61.10	65.74
	Testset2	68.87	70.12	63.96	63.00	64.86
	Testset3	75.02	76.16	65.25	64.65	66.20

Table 2: Comparative Analysis with Different SNRs.

Training dataset	Test dataset	Feature extraction type	Different SNR (dB)				
			0 dB	5 dB	10 dB	15 dB	20 dB
Trainset1	Testset1	GFCC	43.23	52.30	63.74	68.10	70.85
		MF-GFCC	45.84	54.96	65.25	70.45	72.42
	Testset2	GFCC	43.06	52.45	62.95	68.20	69.20
		MF-GFCC	41.96	52.23	62.05	67.65	70.96
	Testset3	GFCC	41.45	51.85	62.20	67.25	69.15
		MF-GFCC	43.26	54.04	63.55	68.96	71.04
Trainset2	Testset1	GFCC	46.02	55.45	66.36	71.45	73.66
		MF-GFCC	48.85	59.05	68.85	74.04	76.65
	Testset2	GFCC	45.90	56.20	66.02	70.35	72.54
		MF-GFCC	47.60	58.15	67.20	72.02	75.26
	Testset3	GFCC	46.35	57.35	66.42	71.86	74.36
		MF-GFCC	51.20	59.96	70.45	75.32	77.80
Trainset3	Testset1	GFCC	41.86	51.68	62.25	67.65	68.32
		MF-GFCC	44.50	53.74	64.66	69.45	71.23
	Testset2	GFCC	39.45	48.85	59.35	64.86	67.87
		MF-GFCC	42.65	53.10	63.40	68.25	70.12
	Testset3	GFCC	43.54	54.25	64.10	70.54	75.02
		MF-GFCC	45.99	56.75	66.65	71.36	76.16

Table 3: Comparative Analysis Using Refined Features and Discriminative Methods.

Training dataset	Test dataset	Feature extraction type	Acoustic modeling unit					
			MMI	MPE	GA+MMI	PSO+MMI	GA+MPE	PSO+MPE
Trainset1	Testset1	GFCC	75.35	76.67	76.95	77.95	77.67	78.36
		MF-GFCC	77.62	78.64	78.85	79.65	79.64	80.45
	Testset2	GFCC	74.80	77.42	76.65	78.89	78.42	80.20
		MF-GFCC	75.36	77.33	77.14	79.36	78.33	81.12
	Testset3	GFCC	74.25	76.67	75.74	77.10	76.97	78.65
		MF-GFCC	76.14	77.64	77.86	79.65	78.64	80.36
Trainset2	Testset1	GFCC	78.96	80.28	80.45	82.36	81.28	83.74
		MF-GFCC	81.25	83.27	83.36	85.47	84.27	86.64
	Testset2	GFCC	77.54	79.54	79.65	81.45	80.54	82.75
		MF-GFCC	80.66	82.56	82.38	84.74	83.56	86.10
	Testset3	GFCC	78.46	80.77	79.96	81.96	81.77	83.20
		MF-GFCC	83.10	85.32	85.20	86.60	86.10	87.96
Trainset3	Testset1	GFCC	74.72	76.63	76.40	78.74	77.63	79.65
		MF-GFCC	76.93	78.95	78.74	80.10	79.95	80.45
	Testset2	GFCC	72.27	74.23	73.95	76.23	75.23	76.85
		MF-GFCC	76.32	78.87	78.45	80.32	79.87	81.25
	Testset3	GFCC	81.42	83.26	82.78	84.40	80.46	85.15
		MF-GFCC	82.15	85.55	86.32	86.65	81.95	87.35

evaluated in combination with both discriminative techniques. The results in Table 3 reveal that the MPE-based system with the PSO optimization algorithm performs best among all the methods.

5.4 Comparative Analysis Using Different Speaker Variations

In this section, the performance analysis is done using the dataset sample of the various speakers in different environments such as clean and noisy. The datasets created have speaker-dependent (SD), speaker-inde-

Table 4: Comparative Analysis Using Different Speaker Variations.

Training dataset	Test dataset	Feature extraction type	Acoustic modeling unit											
			PSO+MMI						PSO+MPE					
			SD		SI		SA		SD		SI		SA	
			Clean	Noisy	Clean	Noisy	Clean	Noisy	Clean	Noisy	Clean	Noisy	Clean	Noisy
Trainset1	Testset1	GFCC	77.95	74.75	73.58	71.95	75.50	72.75	78.36	76.21	75.45	73.64	77.55	75.85
		MF-GFCC	79.65	77.45	76.63	74.87	78.65	76.49	80.45	77.45	76.27	74.59	78.62	76.49
	Testset2	GFCC	78.89	76.64	75.43	73.69	77.48	74.55	80.20	77.36	76.58	74.35	78.56	75.55
		MF-GFCC	79.36	77.46	76.79	74.46	78.26	76.55	81.12	78.20	77.52	74.90	79.50	76.70
	Testset3	GFCC	77.10	75.25	76.14	73.42	76.55	74.60	78.65	76.63	75.85	73.62	77.80	74.53
		MF-GFCC	79.65	77.32	76.55	74.38	78.56	75.50	80.36	77.29	76.47	74.25	78.49	75.55
Trainset2	Testset1	GFCC	82.36	80.10	79.43	77.64	81.30	79.44	83.74	80.47	78.19	76.30	81.67	77.50
		MF-GFCC	85.47	82.74	81.97	79.74	83.84	81.54	86.64	84.20	82.62	80.85	85.45	82.95
	Testset2	GFCC	81.45	78.25	77.53	75.27	79.53	76.52	82.75	79.81	78.75	77.53	80.61	78.75
		MF-GFCC	84.74	81.36	80.17	78.42	82.56	79.25	86.10	83.65	82.28	80.50	84.55	82.60
	Testset3	GFCC	81.96	78.45	79.23	77.57	79.25	79.75	83.20	81.23	80.55	78.38	82.50	77.55
		MF-GFCC	86.60	82.95	81.66	78.48	83.78	80.20	87.96	85.45	84.67	82.89	86.78	83.46
Trainset3	Testset1	GFCC	78.74	76.20	75.43	73.35	77.40	75.55	79.65	77.15	76.34	74.55	78.45	75.75
		MF-GFCC	80.10	77.65	76.52	75.79	78.55	76.56	80.45	78.10	77.42	75.70	79.30	76.50
	Testset2	GFCC	76.23	74.64	73.85	71.64	75.86	73.85	76.85	74.20	72.63	70.40	75.55	73.57
		MF-GFCC	80.32	78.21	79.43	77.25	79.51	79.09	81.25	79.74	77.96	75.78	80.64	77.55
	Testset3	GFCC	84.40	80.74	80.57	78.75	82.54	80.97	82.15	79.85	81.62	70.45	79.35	77.35
		MF-GFCC	86.65	81.10	81.42	79.68	84.30	81.46	83.35	79.10	83.43	75.65	80.40	78.50

pendent (SI), and speaker-adaptive (SA) variations. The system combines the speaker adaptation technique with the discussed models. It uses the maximum likelihood linear regression (MLLR) adaptation technique to measure the performance. The PSO-optimized extracted features from the feature extraction techniques GFCC and MF-GFCC are implemented with the MMI and MPE discriminative methods. The results of these algorithms according to different conditions are analyzed in Table 4.

6 Conclusion

A novel combination of the MF-GFCC features using the optimization algorithms and discriminative training has been proposed for the noise robust Hindi ASR system. For the optimization of the features, the GA and PSO algorithm are evaluated, and for discriminative training, the MMI and MPE are tested. Also, various experiments have been carried out to check the performance of the proposed system in clean and noisy conditions. The robustness of the proposed system has been evaluated using different parameters. The results conclude that the PSO-optimized MF-GFCC features have significant improvements with the MPE discriminative training. This work can further be extended by various other feature transformation and optimization techniques applied on the front end and discriminative training of those features at the back end of the ASR system.

Bibliography

[1] A. Acero, *Acoustical and environmental robustness in automatic speech recognition*, vol. 201, Springer Science & Business Media, New York, USA, 2012.

[2] A. Adiga, M. Magimai and C. S. Seelamantula, Gammatone wavelet cepstral coefficients for robust speech recognition, in: *IEEE TENCON 2013-2013 IEEE Region 10 Conference (31194)*, Xi'an, China, 2013.

[3] R. K. Aggarwal and M. Dave, Discriminative techniques for Hindi speech recognition system, *Inf. Sys. Indian Lang.* **139** (2011), 261–266.

[4] R. K. Aggarwal and M. Dave, Acoustic modeling problem for automatic speech recognition system: advances and refinements (Part II), *Int. J. Speech Technol.* **14.4** (2011), 309–320.
[5] R. K. Aggarwal and M. Dave, Acoustic modeling problem for automatic speech recognition system: conventional methods (Part I), *Int. J. Speech Technol.* **14.4** (2011), 297.
[6] R. K. Aggarwal and M. Dave, Performance evaluation of sequentially combined heterogeneous feature streams for Hindi speech recognition system, *Telecommun. Syst.* **52** (2013), 1–10.
[7] L. Bahl, P. Brown, P. de Souza and R. Mercer, Maximum mutual information estimation of hidden Markov model parameters for speech recognition, in: *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'86*, Tokyo, Japan, vol. 11, IEEE, 1986.
[8] J. M. Baker, L. Deng, J. Glass, S. Khudanpur, C.-H. Lee, N. Morgan and D. O'Shaughnessy, Developments and directions in speech recognition and understanding, Part 1 [DSP Education], *IEEE Signal Process. Mag.* **26.3** (2009), 75–80.
[9] W. Burgos, *Gammatone and MFCC Features in Speaker Recognition*, Dissertation, 2014.
[10] H. P. Combrinck and E. C. Botha, *On the Mel-scaled cepstrum*, Department of Electrical and Electronic Engineering, University of Pretoria, Pretoria, South Africa, 1996.
[11] S. Davis and P. Mermelstein, Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences, *IEEE Trans. Acoust. Speech Signal Process* **28.4** (1980), 357–366.
[12] M. Dua, R. K. Aggarwal and M. Biswas, Discriminative training using heterogeneous feature vector for Hindi automatic speech recognition system, in: *2017 International Conference on Computer and Applications (ICCA)*, Dubai, United Arab Emirates, IEEE, 2017.
[13] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, San Diego, CA, USA, 2013.
[14] S. Furui, 40 years of progress in automatic speaker recognition, *Advances in Biometrics* **5558** (2009), 1050–1059.
[15] D. Gillick, S. Wegmann and L. Gillick, Discriminative training for speech recognition is compensating for statistical dependence in the HMM framework, in: *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, Japan, IEEE, 2012.
[16] H. Hermansky, Perceptual linear predictive (PLP) analysis of speech, *J. Acoust. Soc. Am.* **87.4** (1990), 1738–1752.
[17] J. H. Holland, *Adaptation in natural and artificial systems. 1975*, University of Michigan Press, Ann Arbor, MI, 1992.
[18] X. Huang, A. Acero and H.-W. Hon, *Spoken Language Processing: a Guide to Theory, Algorithm, and System Development*, Prentice Hall PTR, NJ, USA, 2001.
[19] N. Jakovljevic, D. Miskovic, M. Janev, M. Secujski and V. Delic, Comparison of linear discriminant analysis approaches in automatic speech recognition, *Elektron. Elektrotech.* **19.7** (2013), 76–79.
[20] V. Kadyan, A. Mantri and R. K. Aggarwal, Refinement of HMM model parameters for Punjabi automatic speech recognition (PASR) System, *IETE J. Res.* (2017), 1–16.
[21] V. Kadyan, A. Mantri and R. K. Aggarwal, A heterogeneous speech feature vectors generation approach with hybrid hmm classifiers, *Int. J. Speech Technol.* **20** (2017), 1–9.
[22] J. Kennedy and R. Eberhart, Particle swarm optimization, in: *IEEE Int. Conf. Neural Networks*, Perth, WA, Australia, vol. 4, 1995.
[23] J. Koehler, N. Morgan, H. Hermansky, H. G. Hirsch and G. Tong, Integrating RASTA-PLP into Speech Recognition, in: *1994 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Adelaide, SA, Australia, 1994, ICASSP-94, vol. 1. IEEE, 1994.
[24] T.-W. Kuan, A.-C. Tsai, P.-H. Sung, J.-F. Wang and H.-S. Kuo, A robust BFCC feature extraction for ASR system, *Artif. Intell. Res.* **5.2** (2016), 14.
[25] N. Kumar and A. G. Andreou, Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition, *Speech Commun.* **26.4** (1998), 283–297.
[26] G. Kunkle and A. Gerald, *Sequence scoring experiments using the TIMIT corpus and the HTK recognition framework*, Dissertation, Florida Institute of Technology, Florida, USA, 2010.
[27] J. Li, L. Deng, J. Glass, S. Khudanpur, C.-H. Lee, N. Morgan and D. O'Shaughnessy, An overview of noise-robust automatic speech recognition, *IEEE/ACM Trans. Audio Speech Lang. Process.* **22.4** (2014), 745–777.
[28] E. McDermott, T. J. Hazen, J. L. Roux, A. Nakamura and S. Katagiri, Discriminative training for large-vocabulary speech recognition using minimum classification error, *IEEE Trans. Audio Speech Lang. Process.* **15.1** (2007), 203–223.
[29] M. McLaren, R. Vogt, B. Baker and S. Sridharan, A comparison of session variability compensation techniques for SVM-based speaker recognition, in: *Eighth Annual Conference of the International Speech Communication Association* Antwerp, Belgium, pp. 790–793, 2007.
[30] F. Meriem, H. Farid, B. Messaoud and A. Abderrahmene, New front end based on multitaper and gammatone filters for robust speaker verification, in: *Recent Advances in Electrical Engineering and Control Applications*, Springer International Publishing, Cham(ZG), Switzerland, pp. 344–354, 2017.
[31] T. Mittal and R. K. Sharma, Speech recognition using ANN and predator-influenced civilized swarm optimization algorithm, *Turk. J. Electr. Eng. Comput. Sci.* **24.6** (2016), 4790–4803.
[32] J. M. Naik, L. P. Netsch and G. R. Doddington, Speaker verification over long distance telephone lines, in: *1989 International Conference on Acoustics, Speech, and Signal Processing, 1989, ICASSP-89*, Glasgow, UK, IEEE, 1989.
[33] D. Povey, *Discriminative training for large vocabulary speech recognition*, Dissertation, University of Cambridge, Cambridge, United Kingdom, 2005.

[34] D. Povey and P. C. Woodland, Minimum phone error and I-smoothing for improved discriminative training, in: *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Orlando, FL, USA, vol. 1, IEEE, 2002.
[35] L. R. Rabiner and B. H. Juang, *Fundamentals of speech recognition* (Vol. 14), PTR Prentice Hall, Englewood Cliffs, 1993.
[36] D. A. Reynolds, Experimental evaluation of features for robust speaker identification, *IEEE Trans. Speech Audio Process.* **2.4** (1994), 639–643.
[37] K. Samudravijaya, P. V. S. Rao and S. S. Agrawal, Hindi speech database, in: *International Conference on spoken Language Processing*, Beijing, China, 2002, pp. 456–464.
[38] G. Saon and J.-T. Chien, Large-vocabulary continuous speech recognition systems: a look at some recent advances, *IEEE Signal Process. Mag.* **29.6** (2012), 18–33.
[39] A. Sharma, M. C. Shrotriya, O. Farooq and Z. A. Abbasi, Hybrid wavelet based LPC features for Hindi speech recognition, *Int. J. Inf. Commun. Technol.* **1.3–4** (2008), 373–381.
[40] R. Storn and K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, *J. Global Optim.* **11.4** (1997), 341–359.
[41] X. Valero and F. Alias, Gammatone cepstral coefficients: biologically inspired features for non-speech audio classification, *IEEE Trans. Multimedia* **14.6** (2012), 1684–1689.
[42] K. Vertanen, *An Overview of Discriminative Training for Speech Recognition*, University of Cambridge, Cambridge, UK, 2004.
[43] C. P. Woodland and D. Povey, Large scale discriminative training of hidden Markov models for speech recognition, *Comput. Speech Lang.* **16.1** (2002), 25–47.
[44] X. Zhao and D. L. Wang, Analyzing noise robustness of MFCC and GFCC features in speaker identification, in: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2013.
[45] X. Zhao, Y. Shao and D. L. Wang, CASA-based robust speaker identification, *IEEE Transactions on Audio, Speech, and Language Processing* **20.5** (2012), 1608–1616.
[46] H. Zhou, D. Karakos, S. Khudanpur, A. G. Andreou and C. E. Priebe, On projections of Gaussian distributions using maximum likelihood criteria, in: *Information Theory and Applications Workshop, 2009*, IEEE, 2009.