SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V12I8P112

Volume 12 Issue 8, 138-148, August 2025
© 2025 Seventh Sense Research Group®

Original Article

Enhancing Hindi Speech Recognition with Deep CNN-LiGRU through Mel and Gammatone Frequency Cepstral Coefficient Fusion

Hetal Gaudani[1], Narendra M Patel[2]

[1]*Gujarat Technological University, Ahmedabad, India.*
[2] *Department of Computer Engineering, Birla Vishvakarma Mahavidyalaya, V V Nagar, India.*

[1]*Corresponding Author : 169999913004@gtu.edu.in*

Received: 06 June 2025 Revised: 08 July 2025 Accepted: 09 August 2025 Published: 30 August 2025

Abstract *- In India, Hindi stands as the most prevalent language. An ideal communication system in a local language would enable regular individuals to engage with machines via speech interfaces for information retrieval and daily tasks. Despite notable advancements in recent decades, achieving natural and robust human-machine speech interaction in regional languages remains challenging, especially in environments characterized by significant noise and reverberation..In such challenging, noisy environments, traditional automatic speech recognition systems employing Mel Frequency Cepstral Coefficients (MFCC) or Perceptual Linear Prediction (PLP) features have shown subpar performance. Research indicates that Gammatone Frequency Cepstral Coefficients (GFCC) perform well in noisy environments. To address this challenge, a novel approach combined features (MFGF) extracted from MFCC and GFCC to harness the strengths of both while mitigating their respective weaknesses. To bolster the robustness of speech recognition, contemporary speech recognition systems commonly leverage acoustic models based on Recurrent Neural Networks (RNNs), which inherently capture extensive temporal contexts and long-term speech modulations. The study introduces a Deep Convolutional- Light Gated Neural Network (CNN-LiGRU) model. This paper presents a refined iteration of Gated Recurrent Units (GRUs) termed Light Gated Neural Network (LiGRU), coupled with Convolution Neural Network (CNN).CNN extracts spatial features, capturing relevant patterns from the input spectrogram, while LiGRU handles the sequential nature of speech, capturing long-term dependencies and enhancing the model's proficiency in understanding and recognizing spoken language. CNN-LiGRU demonstrates a 30% improvement in recognition rates over CNN-Recurrent Neural Network (RNN). Additionally, LiGRU eliminates the reset gate, resulting in reduced per-epoch training time compared to conventional RNNs. A comparative analysis with state-of-the-art methods is conducted, and experimental results indicate that the fusion approach outperforms, achieving speech recognition accuracy with minimal loss.*

Keywords *- CNN, GFCC, LiGRU, MFCC, RNN.*

1. Introduction

Speech recognition processes an audio signal to generate a word sequence representing spoken language. The identified words have applications in various voice-controlled systems. In India, Hindi serves as the official language and is the most widely spoken, with over 425 million people using it as their primary language and another 120 million as a secondary language. Having a communication system in the local language facilitates interaction between ordinary individuals and machines for information retrieval. Over the last few years, the ASR field has seen remarkable advancement using deep learning techniques. However, Indian languages still face challenges due to linguistic complexity, dialectal variations and limited availability of annotated corpora. Traditionally, two important stages of the ASR system are feature extraction from a speech signal and acoustic modeling of the extracted features. A primary focus of ASR study over the past decades has been the building of an efficient feature extraction solution and an accurate acoustic modeling strategy. Many techniques like Mel Frequency Cepstral Coefficient (MFCC) [1], Perceptual Linear Prediction (PLP) [2], RASTA PLP [3-4], Linear Prediction Cepstral Coefficient (LPCC) [5] have been proposed to generate features from speech utterance. However, in the presence of noise such as babble noise, additive white Gaussian noise, and others, these approaches encounter performance bottlenecks [6-8]. Despite significant research advances in the field of Automatic Speech Recognition (ASR) over the past decades, a notable gap in performance between humans and machines persists. This performance gap is primarily attributed to the machines' lack of robustness in noisy environments, as pointed out in references [9, 10]. Recently, the Gammatone Frequency Cepstral Coefficient (GFCC) feature extraction approach [11] has exhibited promising performance in managing noisy voice data. The GFCC method relies on an array of Gammatone filters and an equivalent rectangular bandwidth (ERB) scale. The nonlinear design of the gammatone filter bank emulates the human auditory system's processing. It is important to note that the filter bank is the primary differentiating factor between Mel-Frequency Cepstral Coefficients (MFCC) and GFCC.

Acoustic modeling using the Hidden Markov Model (HMM) was first introduced in the mid-1980s [12]. Gaussian Mixture Models (GMM) were subsequently adopted in conjunction with HMM to model speech feature distribution. These kinds of classical models have performed acceptably, but they often fail to get the details of Hindi phonetics and contextual dependencies in continuous speech. Deep neural network-based systems have become very popular in the last few years, and they have mostly taken the place of GMM-based acoustic models. The deep learning paradigm is still changing, and new methods and strong architectures are coming out. More sophisticated architectures have been proposed by researchers, including, Convolution Neural Networks (CNN) [13-15], Recurrent Neural Networks (RNN) [16-19], RNN transducers [20], transformer networks [21], and Time Delay Neural Network (TDNN) [21, 22]. Each one has its own pros and cons. CNNs are effective at handling local spatial correlations and slight frequency shifts in audio signals, but they perform poorly in noisy environments and with variable-length inputs [23]. Since RNNs can capture extended context through long-term feedback links across speech frames, they are less sensitive to temporal distortions and deliver high recognition accuracy. This makes them especially useful in noise-resistant applications. However, RNNs suffer from the vanishing and exploding gradient problem, which necessitates the use of special structures to solve it. Despite being parameter-intensive and prone to overfitting, especially in low-resource languages, the Long Short-Term Memory (LSTM) model has been widely used in speech recognition tasks because it incorporates a gating mechanism for better information flow over successive time-steps [24, 25]. To solve the problems, the Gated Recurrent Unit (GRU) architecture was developed in 2014, which simplifies the LSTM cell architecture with two gating units [26]. Recently, a variation of GRU named LiGRU emerged by Bengio [27], which enhanced the GRU design by dropping the reset gate. The use of the ReLU activation function in conjunction with this has become more popular as it produces very high accuracy with low computational overhead among various RNN models [27]. The primary reasons for the performance decline are shallow models, suboptimal kernel and pooling strategies, and insufficient generalization due to minimal dataset-specific adjustment. Therefore, there is a need for a robust, optimized deep CNN model that can successfully extract features from Hindi speech and increase identification accuracy under different acoustic settings. In this study, the deep architecture proposed by Kumar [28] is utilized, which integrates CNN and LiGRU layers, to create a deep hybrid architecture that leverages the strengths of both models. CNN models with modified kernel sizes, strides and pooling methods are suitable for extracting position-invariant features. RNN is best suited for sequence modelling. CNN-LiGRU hybrid model architectures perform better in terms of lower WER and less training time.

The main contributions of this work are: 1) a combination of MFCC and GFCC features for robustness, and 2) a novel lightweight CNN-LiGRU model. The proposed model, as compared to other CNN-LSTM or CNN-RNN-based models, outperforms in terms of WER with significantly less training time on two different Hindi datasets. The proposed model is systematically examined by considering various acoustic characteristics, configurations of hidden layers and neurons, and computational efficiency. The architecture combines gammaton and mel filter bank features to handle noisy environments, utilizes CNN for position-invariant feature extraction, and employs the light-gated recurrent unit model to model long-term dependencies. The performance of the proposed architecture is evaluated using a Hindi dataset. The remainder of this paper is structured as follows: Section 2 provides a brief overview of related work. Section 3 explains various feature extraction techniques. Section 4 discusses the CNN and RNN acoustic models. Section 5 details the proposed architecture, and experimental details are discussed in Section 6.

2. Related Work

Any successful speech recognition system relies on a vast amount of available training and testing data. However, state-of-the-art datasets for Indian languages such as Hindi, Bengali, Punjabi, and others are still insufficient, and only a few studies have been conducted in the field of Hindi language ASR [29]. An ASR system's feature extraction component is crucial to the system's overall accuracy. Most research studies for automatic speech recognition used the MFCC, LPC, and Dynamic Time Warping and Relative Spectra Processing features. In comparison to PLP, MFCC confirms improved feature extraction performance (Dua et al. 2018) [30]. Except for the aspirated voiced phoneme class, Farooq et al. (2010) [31] found that Wavelet packets performed better than the features obtained with MFCC and GFCC. To improve the recognition rate of Hindi ASR, Aggarwal and Dave (2013) combined traditional feature extraction approaches such as PLP, MFCC and gravity centroids, which raised the ASR recognition rate. Biswas et al. (2014) proposed wavelet packet-based ERB-like features to recognize Hindi Phonemes. Dua et al. (2018) proposed Differential Evolution (DE) based optimized GFCC features with the GMM-HMM acoustic model, and they observed that GFCC performs well in clean and noisy environments. HMM-GMM-based acoustic model was used in the mid-1980s for speech recognition. But in recent studies, deep learning based models, especially CNNs and RNNs, due to their ability to automatically learn hierarchical features, replaced the traditional model in recognition tasks. To increase the detection rate of a Hindi speech recognition system, Passricha (2019) [32] used CNN-Bi-Directional Long Short-Term Memory (BLSTM) layers. Later on, they have also investigated the effect of different pooling techniques in Hindi ASR. The impact of deep-learning systems on the Hindi dataset has not been thoroughly examined. The study also investigates a deep hybrid architecture by integrating CNN and LiGRU layers, where CNN capabilities are used to extract position-invariant features, and the LiGRU model is used to describe long-term dependencies.

3. Feature Extraction

3.1. Mel Frequency Cepstral Coefficients (MFCC)

Human spoken sounds are filtered by the vocal tract in the human body. Which sound is produced is determined by the shape of the vocal tract. Speech sounds are produced by unique forms. The envelope of the short-term power spectrum represents the curvature of the vocal tract. This envelope is accurately represented by MFCC. The MFCC feature extraction process, as illustrated in Figure 1, involves several steps.

1. The first-order high-pass filter, as shown below, is used to pre-emphasize the signal.

$$y(t)=x(t)-\alpha x(t-1) \quad \text{where } 0.95< \alpha <0.99 \qquad (1)$$

2. The speech signal is divided into short-time frames after pre-emphasis. While speech signals are time-varying, their frequency components remain stationary for brief periods, hence the need for frame segmentation.
3. A Hamming window is applied to each frame to remove edge discontinuities.
4. Performing Fast Fourier Transform (FFT) on individual frames to capture frequency components of the divided speech stream.
5. The next step involves calculating the Mel spectrum from the magnitude spectrum by utilizing a Mel filter bank, which consists of a set of bandpass filters as follows.

$$Mel(f) = 2595 \log_{10}\left(1 + \frac{f}{100}\right) \qquad (2)$$

6. Subsequent application of the Discrete Cosine Transform (DCT) to extract cepstral coefficients. The amplitudes of the resulting spectrum are the MFCCs. Typically, twelve coefficients are preserved, and an extra feature, the frame's log energy, is incorporated, resulting in a feature vector that encompasses thirteen MFCC attributes.

3.2. Gammatone Frequency Cepstral Coefficients (GFCC)

GFCC is gaining popularity due to its robustness in terms of noise compared to MFCC. It uses Gammatone filters and an ERB scale to mimic human hearing. The main difference between MFCC and MFCC is the filter bank. The process involves segmenting the signal, applying a Hamming window, a Fourier transformation, a 64-channel Gammatone Filter Bank, cubic root extraction, and DCT for cepstral coefficients. The formula for the Gammatone filter is as follows.

$$g(f,t) = t^{n-1}e^{-2\pi bt}\cos(2\pi ft) \qquad (3)$$

Here, f represents frequency, t denotes time, b is the rectangle bandwidth, and n=4 signifies the filter's order. This results in 13 GFCC coefficients, and delta and double delta are applied to the temporal information.

4. Acoustic Modal

4.1. Convolution Layer

CNNs are a prominent deep learning variant that is commonly used in vision tasks. It has also performed admirably in ASR. The primary goal of CNN is to find the local structure present in speech signals. CNN effectively minimizes spectral fluctuations and models spectral correlations. Local connectivity allows it to efficiently capture nearby spatial correlations, a task that is challenging to accomplish using a Deep Neural Network. CNNs can also handle minor changes in frequency caused by speaker variance or speaking style. Convolution Neural Networks (CNNs) are made up of three primary sub-layers: convolution, pooling, and non-linearity. The convolution layer is made up of a number of separate filters. To generate a feature map, each filter individually convolves across the input. Convolution filters are applied from the upper-left to the lower-right corner of the input feature map to capture structural locality, which is a type of local pattern. These filters are also good at filtering out background noise. After convolution, pooling decreases the feature-dimensionality and provides a compact feature representation. Pooling effectively handles small frequency shifts that are common in voice signals. For the speech-recognition problem, Passricha and Aggarwal [33] study the performance of several pooling strategies in detail. In a CNN, a nonlinear layer is made up of an activation function that accepts the feature map created by the pooling layer outputs the activation map.

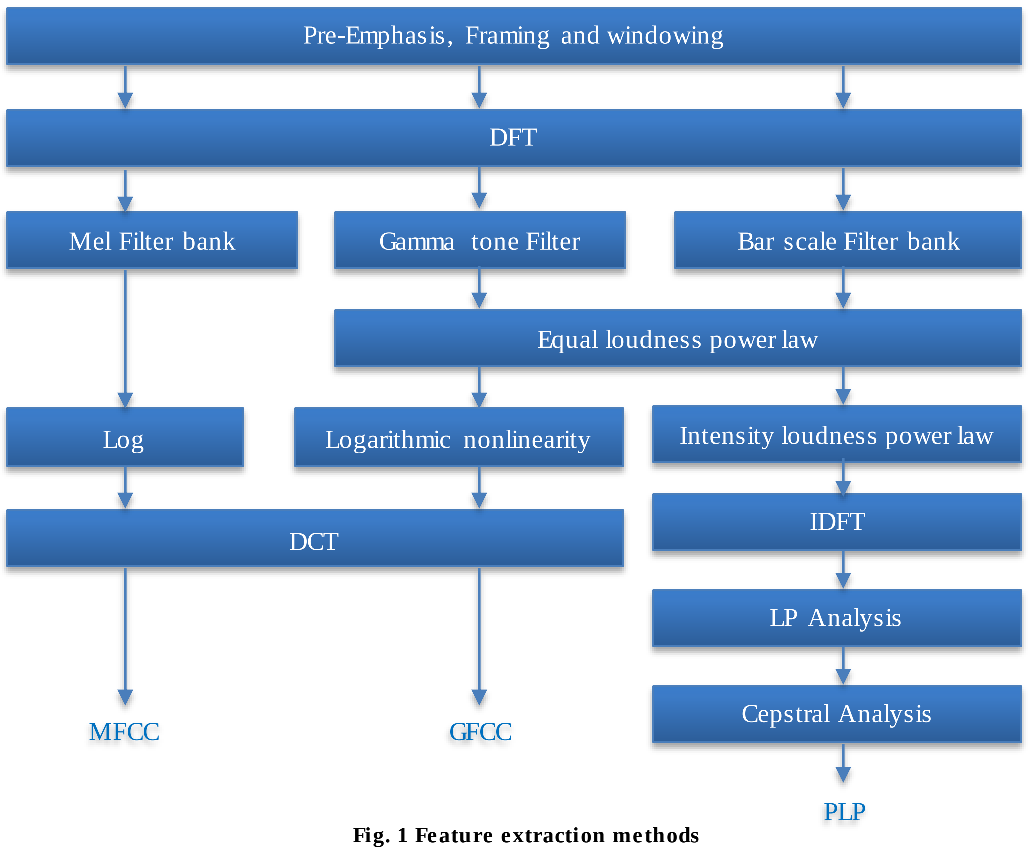

Fig. 1 Feature extraction methods

4.2. Recurrent Layer

4.2.1. Recurrent Neural Networks

An RNN is a frequently employed model for solving ordinal or temporal problems. RNNs are known for their internal memory, enabling them to retain crucial information from the input they receive. This capability makes them exceptionally precise in predicting subsequent elements, which is why they are the preferred choice for tasks involving sequential data, such as speech. Recurrent neural networks have the capacity to develop a more profound comprehension of a sequence and its contextual relevance when compared to other algorithms. Given an input sequence I = (I_1, . . . , I_N), RNN, as shown in Figure 2, computes output vector sequence O =(O_1, . . . , O_N) and the hidden vector sequence h = (h_1, . . . , h_N) through following equation from t = 1 to N :

$$h_n = H(w_{ih} I_n + w_{hh} h_{n-1} + b_{h)} \quad (4)$$

$$O_n = w_{ho} h_t + b_o \quad (5)$$

Where W denotes weight matrices, b denotes the bias vector, and H is the hidden layer activation function. RNNs are trained using the back-propagation through time training algorithm. The usual RNN, on the other hand, is better at regulating the temporal coefficient of spectral vectors. However, they suffer from vanishing gradient issues. To address this difficulty, the LSTM model was proposed.

4.2.2. Long Short-Term Memory

LSTM networks are a type of RNN that can learn long-term dependencies and were proposed by Hochreiter & Schmidhuber (1997) [32] to overcome the vanishing gradient problem that regular Recurrent Neural Networks suffer (RNN) [32]. LSTM, as shown in Figure 3, consist of memory cells and gating units. The memory cell aids in the long-term recall of information, while the gating units help with network information flow regulation [31]. The LSTM model's basic structure consists of three gates: input gates, output gates, and forget gates. LSTM is implemented by the following composite function [34]:

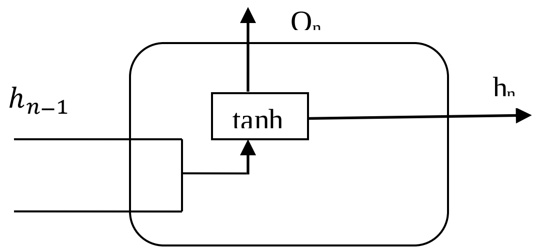

Fig. 2 RNN basic building block

$$f_n = \sigma(w_f [h_{n-1}, I_n] + b_f) \quad (6)$$

$$i_n = \sigma(w_i [h_{n-1}, I_n] + b_i) \quad (7)$$

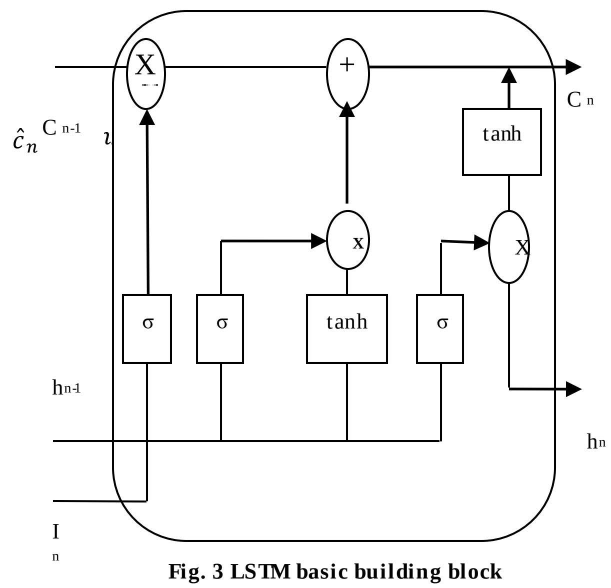

Fig. 3 LSTM basic building block

$$\hat{c}_n = tanh(w_c[h_{n-1}, I_n] + b_c) \tag{8}$$

$$c_n = (c_{t-1} \circ \mathrm{i_n}) + (\hat{c}_t \circ \mathrm{i_n}) \tag{9}$$

$$o_n = \sigma(W_o\,[i_n, h_{n-1}] + b_o) \tag{10}$$

$$h_n = \tanh(c_n) \circ o_n \tag{11}$$

Where i, f, o and c are respectively the input gate, forget gate, output gate and cell activation vector, σ is the activation function.

4.2.3. Gated Recurrent Unit

Cho and colleagues introduced the Gated Recurrent Unit (GRU) in 2014 [35]. GRU simplifies the LSTM architecture. It uses only two gates, *the update gate* and the *reset gate*, to control the memorization process.

GRU, unlike LSTM, has no output gate and consolidates the input and forget gates into a single update gate. The GRU output computations are based on the following equations:

$$r_n = \sigma(W_r[I_n, h_{n-1}] + b_r) \tag{12}$$

$$z_n = \sigma(W_z\,[I_n\ , h_{n-1}] + b_z) \tag{13}$$

$$\widetilde{h_n} = tanh(W_h[r_n \circ \mathrm{h_{n-1}}, \mathrm{I_n}] + \mathrm{b_h}) \tag{14}$$

$$\mathrm{h_n} = (1 - \mathrm{z_n}) \circ \widetilde{\mathrm{h_n}} + \mathrm{z_n} \circ \mathrm{h_{n-1}} \tag{15}$$

Where z and r are respectively the update gate, reset gate, and σ is the activation function.

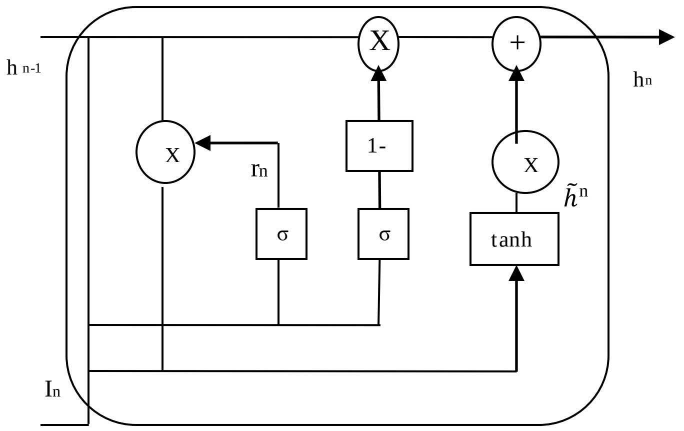

Fig. 4 GRU basic building block

4.2.4. Light Gated Recurrent Unit

Ravanelli proposed Li-GRU as a modification of the GRU architecture in 2018. Due to the removal of the reset gate r from the GRU architecture, LiGRU functions with the update gate z. The reset gate, according to [27], is critical when dealing with sequence discontinuities. In the context of speech, this is a rare occurrence because history is usually beneficial. Equation (13) is changed as a result of the removal of the reset gate r_t:

$$\widetilde{h_n} = tanh(W_h[h_{n-1}, \mathrm{I_n}] + \mathrm{b_h}) \tag{16}$$

The traditional tanh activation function has been replaced with the ELU activation function to achieve improved results. This modification is proposed based on experiments in section 6.2.3

$$\widetilde{h_n} = ELU(W_h[h_{n-1}, \mathrm{I_n}] + \mathrm{b_h} \tag{17}$$

5. Proposed ASR using deep CNN-LiGRU

The automatic speech recognition system comprises two integral components: the front end and the back end. The front end extracts relevant features from the input audio signal, undertaking tasks such as signal pre-processing, spectral analysis, and feature extraction. Meanwhile, the back end decodes these features into meaningful speech or text through processes like acoustic modeling, language modeling, and decoding algorithms. Notably, MFCC and GFCC [36] have demonstrated superior performance over PLP, LPCC, and other techniques in general speech recognition tasks, with MFCCs excelling in diverse environments and GFCCs proving effective in noisy conditions by mirroring the human auditory system. The fusion of MFCCs and GFCCs enhances discriminative power, improves modeling of speaker and speech variability, and aligns with human auditory perception, thereby enhancing overall system performance. Consequently, a proposed combined approach leveraging both MFCC and

GFCC is proposed to harness the strengths of these methods and mitigate their individual limitations.

The proposed acoustic model comprises a Convolutional Neural Network followed by a Light Gated Recurrent Unit (LiGRU), as illustrated in Figure 5. CNNs excel in capturing spatial hierarchies within data. By applying CNN layers to MFCCs) and Gammatone-Frequency Cepstral Coefficients (GFCCs), the model adeptly learns spatial features and patterns, enhancing the extraction of relevant information from the input audio. The CNN extracts spatial features, capturing pertinent patterns from the input spectrogram, while the LiGRU handles the sequential nature of speech, capturing long-term dependencies and improving the model's proficiency in understanding and recognizing spoken language. This integrated architecture is well-suited for the intricate demands of speech recognition tasks. Combined features are applied to the convolution layer to generate an output feature map. To trim dimensionality while retaining essential information and discarding irrelevant details, a max pooling layer is employed. The pooled output is then fed into the convolution layer as input. A linear layer is added after the final CNN layer to further minimize dimensionality without sacrificing vital information. The output from this linear layer is directed into stacked LiGRU layers, each consisting of 550 units. The LiGRU's ability to retain sequence history in its internal state enables it to predict the next phoneme based on previously observed properties. To derive posterior probabilities for distinct classes, Fully Connected layers (FC) with sizes of 3 and 550 nodes in each layer are utilized, respectively. FC layers are particularly apt for class-based discrimination. During training, a batch size of eight is employed, and the learning rate is initially set to 0.0001. On the development set, the learning rate is fine-tuned over 23 epochs. If the frame accuracy on the development set falls below a specified threshold, the learning rate is halved after each session. In the decoding process, the Kaldi decoder is utilized, configuring the beam to 13 and the lattice beam to 8. This intricate architecture comprises a total of 12 layers, including 3 CNN layers,2 Linear Layers, 4 LiGRU layers, and 3 Multilayer Perceptron (MLP) layers. This design optimizes the computational efficiency of the proposed model compared to existing Recurrent Neural Network (RNN) models.

6. Experimental Details

6.1. About Dataset

Two distinct and well-annotated datasets with rich phonetic information are utilized for this research, as illustrated in Table 1. The first dataset employed is called the Indic dataset [37]. This corpus comprises data from 13 significant Indian languages. Specifically for the Hindi language, it contains a collection of more than 2,318 sentences and utterances recorded by native speakers, both male and female. The second dataset utilized for the experiment is the Multilingual and Code-switching ASR Challenge Dataset - sub-task 1 [38]. This Hindi speech dataset is divided into two parts: a training set consisting of 95.05 hours of audio and a test set consisting of 5.55 hours of audio. In the training and test sets, there are 4,506 and 386 unique sentences derived from Hindi stories, respectively, with no overlap between the sentences. The training set includes utterances from 59 speakers, while the test set includes utterances from a separate group of 19 speakers. The audio files are sampled at 8 kHz and encoded in 16 bits. The combined training and test sets have a total vocabulary of 6,542 words.

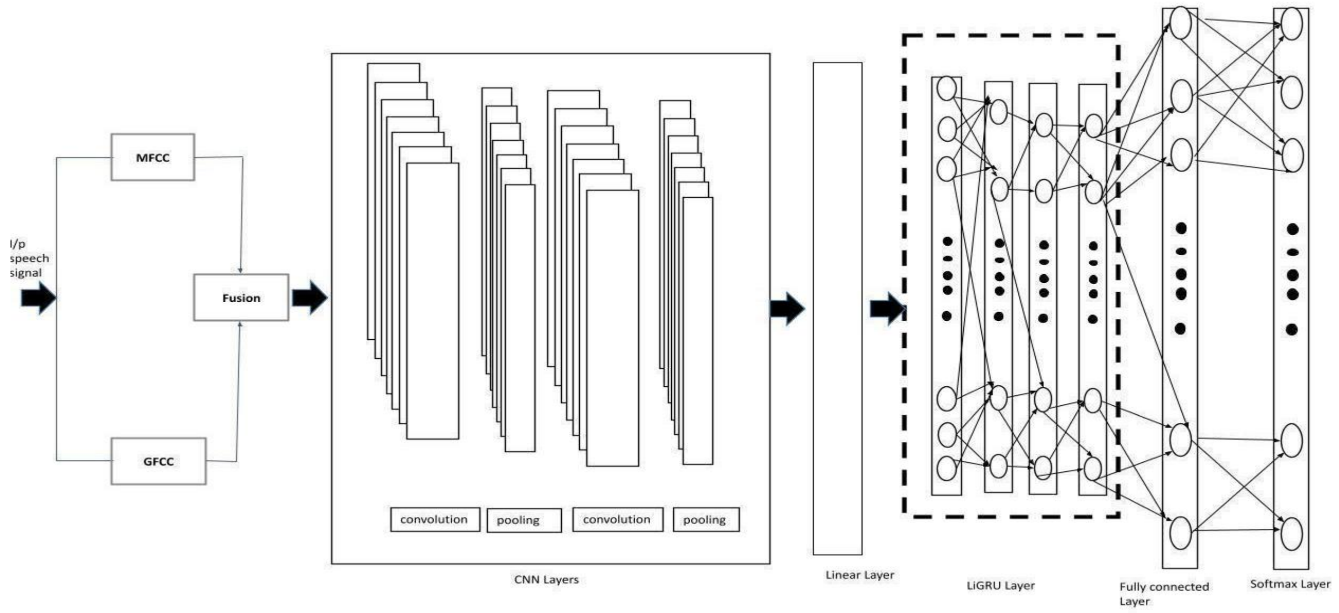

Fig. 5 Deep CNN-LiGRU architecture

Table 1. Hindi speech corpus details

SN	Dataset	#Sentences	Duration(Hours)	#Speakers
1	Indic Database	2318	10.5	2
2	Multilingual and code-switching ASR Challenge Dataset - sub-task 1	4892	101	78

6.2. Simulation Details and Experiment Result

The subsequent section of this paper provides a comprehensive explanation of the simulation and experimental outcomes of the proposed work. Initially, a Hidden Markov Model (HMM) system is trained based on the Gaussian Mixture Model (GMM) to estimate the context-dependent Phone Model, generating an alignment crucial for the subsequent training of the proposed system. The GMM-HMM system is trained using base features extracted with a Hamming window of 25 milliseconds and a 10-millisecond frame shift. The baseline GMM-HMM system is implemented using the Kaldi toolkit, and all deep neural networks are trained using PyTorch-Kaldi. To test the proposed system, separate experimental analyses were performed utilizing different combinations of feature extraction methods and proposed acoustic modeling methodologies. The accuracy study of several feature extraction techniques using different deep learning based acoustic models is discussed in experiment 6.2.1.

6.2.1. Accuracy Analysis of Feature Extraction Techniques

The proposed system, illustrated in Figure 5, undergoes evaluation using six distinct feature extraction techniques: MFCC, GFCC, PLP, MF-PLP, MF-GFCC and PLP-GFCC. Feature vectors are subjected to testing in both clean and noisy environments. For MFCC, GFCC, and PLP, the initial thirteen coefficients are extracted and expanded t by adding first and second-order derivatives to increase the number of coefficients to 39. In the case of MF-PLP, the initial features are derived by combining 13 MFCC coefficients with 13 PLP coefficients, expanded to a 78-dimensional feature vector through derivative processes. Similarly, PL-GFCC and MF-GFCC combine PLP and GFCC features, resulting in a 78-dimensional feature vector. The performance of several neural network-based acoustic modeling approaches using various combinations of acoustic feature sets is shown in Table 2. The acoustic models are trained using PyTorch-Kaldi. The Table below reports the achieved Word Error Rate (WER). Features were extracted using Kaldi and concatenated using PyTorch-Kaldi. These experiments incorporated a trigram language model. For consistency, all the Neural Network Architectures (RNN, LSTM, LiGRU and GRU) examined in this study, with the exception of CNN, comprise three hidden layers, each containing 512 hidden neurons, as well as the relu activation function. Every model was trained for 23 iterations. The truncation process for sentences longer than 1,000 characters was implemented during the training of recurrent models to address the issue of running out of memory. It was observed that MFCC outperformed GFCC in the experiments that were conducted. However, the combination of GFCC and MFCC features surpassed the performance of MFCC alone. It is important to note that these observations are based solely on clean speech signals. Additionally, it is observed that PL-GFCC yielded results nearly equivalent to MF-GFCC in this context. The GRU model shows an insignificant improvement compared to the LSTM model. Furthermore, the LiGRU model displays an additional performance enhancement over the GRU model. Notably, experiment results also reveal that PL-GFCC and MF-GFCC yield nearly equivalent results when applied to clean data in the LiGRU model. Among all the investigated neural network architectures, the MF-GFCC feature sets yield the lowest WER. This observation implies that the integration of multiple feature sets has the potential to enhance speech recognition performance within the evaluated neural network architectures.

Table 2. WER(%) of feature extraction methods with different acoustic models

Feature extraction method	Dataset-1				
	RNN	LSTM	GRU	LiGRU	CNN
MFCC	9.66	9.84	9.70	9.43	12.73
PLP	9.60	9.67	9.58	9.51	11.99
GFCC	10.08	10.53	9.94	9.79	15.74
MF-PLP	9.59	10.77	9.78	9.57	11.62
PL-GFCC	9.43	9.79	9.60	9.41	11.40
MF-GFCC	9.43	10.27	9.52	9.40	11.41

6.2.2. MFGF Deep CNN-LiGRU

The combined delta features MF-GFCC are applied to the initial layer of a Convolution Neural Network (CNN). The input data encompasses a dimension of 260, taking into account both two futures and two past frames relative to the current time step. In the analysis, the delta feature of both MFCC and GFCC was taken into account. After reviewing the delta feature of both MFCC and GFCC, minor variations are observed when comparing delta and delta-delta features, as outlined in Table 3. Therefore, the delta feature is opted for, which reduces feature dimensionality and also decreases time complexity. Following the first convolutional layer, as shown in Table 4, the CNN employs max pooling with a filter size of 5 and a stride of 3, resulting in a feature vector reduction to 86. Subsequently, the second convolutional layer further processes the feature vector, reducing it to 42 using a filter size of 3 and a stride of 2. The third convolutional layer further processes the feature vector, reducing it to 19 using a filter size of 3 and a stride of 2. The final layer operates on a feature vector of 19 dimensions with a filter size of 3, and the max pooling size is one. The final layer incorporates 60 filters, producing output vectors with a size of 1020. These vectors serve as input to a Multilayer Perceptron (MLP) layer consisting of two layers, each comprising 550 hidden nodes. The output from the MLP layer is then fed into a Light Gated Recurrent Unit (LiGRU) with three layers, each containing 550 hidden nodes. To conclude the architecture, three additional MLP layers and Fully Connected (FC) layers are incorporated. The first MLP layer receives a feature vector of size 1100 as input. The Elu activation function is used for this experiment. This comprehensive neural network design aims to optimize feature extraction and learning capabilities for enhanced performance in the specified task.

Table 3. WER(%) with delta features

Delta Features	WER%
D-0	5.06
D-1	4.76
D-2	4.75

Table 4. Parameters of MFGF Deep CNN-LiGRU

Parameter	Value
CNN Filter size in each layer	5,3,3,3
CNN strides in each layer	3,2,2,1
The number of filters in every CNN layer	80,60,60,60
The number of LiGRU's hidden layers	3
The number of hidden nodes in every layer	550
Each MLP layer's number of hidden units	550
Activation function used in CNN, MLP and LiGRU	ELU

6.2.3. The Impact of Activation Functions

Neural networks' performance and behavior are greatly influenced by activation functions. This study probably investigates how various activation functions, such as Sigmoid, Tanh, Exponential Linear Unit (ELU), Leaky ReLU, Rectified Linear Unit (ReLU), and others, affect neural network accuracy. Except for the activation function, the other configurations remained constant across all experiments conducted as mentioned previously. The results of the experiment showed that the ELU activation function yielded the best WER performance. ELU, characterized by its smooth and continuously differentiable nature, presents advantages in mitigating convergence issues and promoting stable training. Its efficacy surpasses that of ReLU, particularly in addressing the vanishing gradient problem. Comparing the ELU activation function with the sigmoid, it was observed that using ELU resulted in a relative reduction in the WER, as shown in Table 5.

Table 5. WER(%) different activation function

Activation Function	Sigmoid	Tanh	ReLU	Leaky ReLU	ELU
Dataset 1 WER [%]	9.71	9.35	9.34	9.52	9.28
Dataset 2 WER [%]	7.24	5.55	5.48	4.78	4.76

6.2.4. The Impact of Batch normalization

Batch normalization enhances training efficiency and performance of neural networks through the alleviation of internal covariate shift. Table 6 presents a comparison of the WER achieved with and without batch normalization, highlighting its importance. The results indicate that batch normalization is beneficial for improving ASR performance. This enhancement demonstrates that the LiGRU model works especially well with batch normalization, primarily as a result of the ELU activations that were used.

Table 6. WER (%) with and without batch normalization

Parameter	WER%
without batch norm	7.10
with batch norm	4.76

The proposed architecture has been subjected to training using various configurations of convolution layers and LiGRU layers, as detailed in the Table. We conducted training for 21 different architectural configurations based on Convolution Layers 3–5. The initial architecture consisted of three convolution layers, two LiGRU layers, and three FC layers, which can be found in Table 7. In the subsequent architectures, only the number of convolution layers and MLP layers was modified, while all other aspects of the configurations remained constant. The feature dimensions for all the architectures were maintained consistently as outlined in the previous Table. Through rigorous examination, we systematically increased the number of fully connected layers. This examination revealed significant observations. Initially, accuracy increases proportionately with the number of LiGRU layers; four LiGRU layers resulted in the most ideal model. Secondly, the addition of hidden layers to CNN and Fully Connected (FC) layers does not significantly improve the Deep Neural Network's ability to discriminate (DNN). In our analysis, we found that employing 3 CNN layers,4 LiGRU Layers and 3 MLP layers yielded the most favorable outcomes.

6.2.5. Performance Analysis using Different Acoustic Models

This section summarizes research on several acoustic models. Here, the combined features of the mel filter bank and gammatone filter bank are applied to the first layer of CNN. The output feature map from the CNN is given to a different recurrent neural network, followed by a fully connected layer.

Table 7. Impact of CNN and LiGRU Layer Variations on Word Error Rate (WER%)

CNN Layers	LIGRU layers	FC layers	WER [%]	CNN Layers	LIGRU layers	FC layers	WER [%]
3	2	3	4.98	4	3	5	5.04
3	2	4	5.03	5	3	3	4.94
3	2	5	5.30	5	3	4	5.17
4	2	3	5.05	5	3	5	5.36
4	2	4	5.23	3	4	3	**4.61**
4	2	5	5.36	3	4	4	4.65
3	3	3	4.76	3	4	5	4.97
3	3	4	4.82	4	4	3	4.74

The outcomes of all conducted experiments are shown in Table 8. The Word Error Rate (WER) and training computation time were measured for each experiment. We have measured the WER and training computation time for each experiment. In comparison to other RNN models, this one-gated model with the ELU activation function demonstrated remarkable accuracy while requiring minimal computing load.

Table 8. Acoustic model variants: a comparative performance analysis

Modal	WER (%)
CNN-RNN	6.65
CNN-LSTM	6.02
CNN-GRU	4.70
CNN-LiGRU	4.61

6.2.6. Comparative Analysis of Per-Epoch Training Time

To better illustrate the computational efficiency of the proposed deep CNN-LiGRU model compared to existing RNN-based models. Table 1 presents the per-epoch training time (in minutes) for the CNN-RNN, CNN-LSTM, CNN-GRU, and CNN-LiGRU models:

Table 9. A Comparative analysis of per-epoch training time

Model	Per-epoch Training Time (in minutes)
CNN-RNN	105
CNN-LSTM	186
CNN-GRU	170
CNN-LiGRU	101

As shown in Table 9, the CNN-LiGRU model achieves a reduced per-epoch training time compared to both the CNN-LSTM (186 minutes) and CNN-GRU (170 minutes), with a training time of 101 minutes. This reflects improved computational efficiency. Although the CNN-LiGRU model's training time is slightly higher than that of the CNN-RNN (105 minutes), it offers a well-rounded balance between computational efficiency and model performance, leveraging the strengths of both CNN and GRU architectures.

7. Conclusion

Experiment results suggest that incorporating various feature sets, notably MFCC and GFCC, presents an opportunity to boost the effectiveness of speech recognition within neural network structures. Moreover, the utilization of sophisticated architectures such as the Convolutional Neural Network (CNN) in conjunction with the Light Gated Recurrent Unit (LiGRU) exhibited considerable enhancements in accuracy without sacrificing computational efficiency.

Furthermore, our investigation into activation functions and batch normalization underscored their substantial influence on the performance of the system. Particularly, the Exponential Linear Unit (ELU) activation function and the implementation of batch normalization demonstrated notably positive outcomes in our experiments.

Acknowledgements

We express our genuine gratitude to System Level Solutions, Anand, India, for generously offering GPU support, which was a crucial contribution to the success of our experiments. We have run all our experiments with Quadro RTX 800. Their support has significantly contributed to the success and efficiency of our project. We deeply appreciate their valuable contribution to this study.

References

[1] Steven Davis, and Paul Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357-366, 1980. [CrossRef] [Google Scholar] [Publisher Link]

[2] Hynek Hermansky, "Perceptual Linear Predictive (PLP) Analysis of Speech," *The Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738-1752,1990. [CrossRef] [Google Scholar] [Publisher Link]

[3] Hynek Hermansky et al., "RASTA-PLP Speech Analysis," *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, San Francisco, CA, USA, vol. 1, pp. 121-124, 1992. [CrossRef] [Google Scholar] [Publisher Link]

[4] Harshita Gupta, and Divya Gupta, "LPC and LPCC Method of Feature Extraction in Speech Recognition System," *2016 6th International Conference - Cloud System and Big Data Engineering (Confluence)*, Noida, India, pp. 498-502, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[5] D.A. Reynolds, "Experimental Evaluation of Features for Robust Speaker Identification," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 4, pp. 639-643, 1994. [CrossRef] [Google Scholar] [Publisher Link]

[6] R.K. Aggarwal, and M. Dave, "Performance Evaluation of Sequentially Combined Heterogeneous Feature Streams for Hindi Speech Recognition System," *Telecommunication Systems*, vol. 52, pp. 1457-1466, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[7] Medikonda Jeevan et al., "Robust Speaker Verification using GFCC based I-Vectors," *Proceedings of the International Conference on Signal, Networks, Computing, and Systems*, vol. 1, pp. 85-89, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[8] Lisheng Fan et al., "Secure Multiuser Communications in Multiple Amplify-and-Forward Relay Networks," *IEEE Transactions on Communications*, vol. 62, no. 9, pp. 3299-3310, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[9] Alejandro Acero, *Acoustical and Environmental Robustness in Automatic Speech Recognition*, Kluwer Academic, pp. 1-186, 1993. [Google Scholar] [Publisher Link]

[10] Alejandro Acero, and Richard M. Stern, "Environmental Robustness in Automatic Speech Recognition," *International Conference on Acoustics, Speech, and Signal Processing, Albuquerque*, NM, USA, pp. 849-852, 1990. [CrossRef] [Google Scholar] [Publisher Link]

[11] Yang Shao et al., "A Computational Auditory Scene Analysis System for Speech Segregation and Robust Speech Recognition," *Computer Speech Language*, vol. 24, no. 1, pp. 77-93, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[12] Lawrence R. Rabiner, and Biing-Hwang Juang, *Fundamentals of Speech Recognition*, PTR Prentice Hall, pp. 1-507, 1993. [Google Scholar] [Publisher Link]

[13] Dimitri Palaz, Magimai-Doss, and Ronan Collobert, "Analysis of CNN-based Speech Recognition System using Raw Speech as Input," *Interspeech*, pp. 11-15, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[14] Shuo-Yiin Chang, and Nelson Morgan, "Robust CNN-based Speech Recognition with Gabor Filter Kernels," *Fifteenth Annual Conference of the International Speech Communication Association*, pp. 905-909, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[15] Vishal Passricha, and Rajesh Kumar Aggarwal, "A Comparative Analysis of Pooling Strategies for Convolutional Neural Network based Hindi ASR," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, pp. 675-691, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[16] Jeffrey L. Elman, "Finding Structure in Time," *Cognitive Science*, vol. 14, no. 2, pp. 179-211, 1990. [CrossRef] [Google Scholar] [Publisher Link]

[17] Ankit Kumar, and Rajesh Kumar Aggarwal, "Discriminatively Trained Continuous Hindi Speech Recognition using Integrated Acoustic Features and Recurrent Neural Network Language Modelling," *Journal of Intelligent Systems*, vol. 30, no. 1, pp. 165-179, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[18] Yongqiang Wang et al, "Transformer-based Acoustic Modeling for Hybrid Speech Recognition," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, pp. 6874-6878, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[19] Yajie Miao, Mohammad Gowayyed, and Florian Metze, "EESEN: End-to-End Speech Recognition using Deep RNN Models and WFST-based Decoding," *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Scottsdale, AZ, USA, pp. 167-174, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[20] Anjuli Kannan et al, "Large-scale Multilingual Speech Recognition with a Streaming End-to-End Model," *arXiv Preprint*, pp. 1-5, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[21] Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, "A Time Delay Neural Network Architecture for Efficient Modeling of Long Temporal Contexts," *Sixteenth Annual Conference of the International Speech Communication Association*, vol. 2015, pp. 3214-3218, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[22] Ankit Kumar, and Rajesh Kumar Aggarwal, "Hindi Speech Recognition using Time Delay Neural Network Acoustic Modeling with i-Vector Adaptation," *International Journal of Speech Technology*, vol. 25, pp. 67-78, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[23] Jian Kang et al., "Advanced Recurrent Network-based Hybrid Acoustic Models for Low Resource Speech Recognition," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2018, pp. 1-15, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[24] Alex Graves, Navdeep Jaitly, and Abdel-Rahman Mohamed, "Hybrid Speech Recognition with Deep Bidirectional LSTM," *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, Olomouc, Czech Republic, pp. 273-278, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[25] Y. Bengio, P. Simard, and P. Frasconi, "Learning Long-Term Dependencies with Gradient Descent is Difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157-166, 1994. [CrossRef] [Google Scholar] [Publisher Link]

[26] Kyunghyun Cho et al., "On the Properties of Neural Machine Translation: Encoder-Decoder Approaches," *arXiv Preprint*, pp. 1-9, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[27] Mirco Ravanelli et al., "Light Gated Recurrent Units for Speech Recognition," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 2, pp. 92-102, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[28] Ankit Kumar, and Rajesh Kumar Aggarwal, "A Hybrid CNN-LiGRU Acoustic Modeling using Raw Waveform Sincnet for Hindi ASR," *Computer Science*, vol. 21, no. 4, pp. 397-417, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[29] Kumud Tripathi, M. Kiran Reddy, and K. Sreenivasa Rao, "Multilingual and Multimode Phone Recognition System for Indian Languages," *Speech Communication*, vol. 119, pp. 12-23, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[30] Mohit Dua, Rajesh Kumar Aggarwal, and Mantosh Biswas, "Optimizing Integrated Features for Hindi Automatic Speech Recognition System," *Journal of Intelligent Systems*, vol. 29, no. 1, pp. 959-976, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[31] O. Farooq, S. Datta, and M.C. Shrotriya, "Wavelet Sub-Band Based Temporal Features for Robust Hindi Phoneme Recognition," *International Journal of Wavelets, Multiresolution and Information Processing*, vol. 8, no. 6, pp. 847-859, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[32] Vishal Passricha, and Rajesh Kumar Aggarwal, "A Hybrid of Deep CNN and Bidirectional LSTM for Automatic Speech Recognition," *Journal of Intelligent Systems*, vol. 29, no. 1, pp. 1261-1274, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[33] Vishal Passricha, and Rajesh Kumar Aggarwal, "A Comparative Analysis of Pooling Strategies for Convolutional Neural Network based Hindi ASR," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, pp. 675-691, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[34] Felix A. Gers, Nicol N. Schraudolph, and Jürgen Schmidhuber, "Learning Precise Timing with LSTM Recurrent Networks," *Journal of Machine Learning Research*, vol. 3, pp. 115-143, 2002. [Google Scholar] [Publisher Link]

[35] Kyunghyun Cho, Bart van Merrienboer, and Dzmitry Bahdanau, "On the Properties of Neural Machine Translation: Encoder-Decoder Approaches," *arXiv Preprint*, pp. 1-9, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[36] Hetal Gaudani, and Narendra M. Patel, "Comparative Study of Robust Feature Extraction Techniques for ASR for Limited Resource Hindi Language," *Proceedings of Second International Conference on Sustainable Expert Systems*, Singapore, pp. 763-775, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[37] Text to Speech Synthesis Systems in Indian Languages, Indic TTS, IIT Madras. [Online]. Available: https://www.iitm.ac.in/donlab/tts/index.php

[38] Diwan, Anuj, et al. "Multilingual and Code-Switching ASR Challenges for Low Resource Indian Languages," *arXiv Preprint*, pp. 1-6, 2021. [CrossRef] [Google Scholar] [Publisher Link]