Multi Model Telugu Speech Signal Analysis Towards Controlling Home Appliances

Parabattina Bhagath, *Member, IAENG*, Matti Indu Lakshmi Prasanna, M Shanmukha, P G Nagasri, Mocherla Swetha, and Pradip K Das

***Abstract*—Speech recognition has been an intriguing research topic for over seven decades. While it has been well-developed for languages with abundant resources, it still faces limitations for languages with fewer resources, such as those spoken in Asia and Africa. The use of voice commands to operate household appliances can greatly benefit the elderly and underprivileged individuals. However, creating spoken interfaces for this purpose is challenging, as readily available pre-built models are not accessible. In this paper, we studied Telugu spoken sentence data to leverage the development of speech interfaces for the Telugu language. The study considers Hidden Markov Models and Convolutional Neural Networks and analyses the performance with various parameters of these two models. The analysis showed that the CNNs perform well for the Telugu sentence data with a recognition rate of 96.4%**

***Index Terms*—LPC, LPCC, MFCC, HMM, CNN, Low-resource speech processing**

I. Introduction

AUTOMATIC speech recognition is a wide-ranging field that focuses on interpreting spoken words. It plays a vital role in creating spoken interfaces for controlling smart appliances, particularly for people with physical disabilities and the elderly. The progress in speech processing and the Internet of Things (IoT) has made it feasible to develop interfaces for controlling appliances that benefit their everyday lives. While commercial products have been created to assist people with special needs, these advancements are limited to languages with abundant resources, like databases and techniques. There are significant shortcomings regarding low-resource languages, particularly Asian and African languages [1][2]. Telugu, a language primarily spoken in the southern part of India, is rich in literature but has minimal development in speech technology. This limitation is hindering experts from creating spoken interfaces for IoT environments. IoT opens a new way of using the things around us in our daily lives. It is greatly impacted by providing facilities to interact with smart devices. For example, we can talk to different smart appliances like televisions, air conditioners, refrigerators, etc., with voice commands [3][4]. This personalizes the usage of smart devices. As a result, new challenges are created for speech recognition. To address these challenges, new datasets and models must be created to accommodate the needs of such systems.

As it was mentioned earlier, the challenges are seen for the Telugu language as the language lacks datasets for implementing speech-enabled IoT systems. In this paper, we address this issue by providing a dataset [5] and benchmark models using Hidden Markov Models (HMMs) and Convolutional Neural Networks (CNNs). HMMs are robust statistical models for developing stochastic systems. They have been utilized to analyze speech signals in various languages, including English [6], Chinese [7], Japanese [8], and more. This modeling technique has firmly established its effectiveness in creating robust speech recognition systems. However, deep neural networks such as Convolutional Neural Networks (CNNs) [9] and Recurrent Neural Networks (RNNs) [10] have gained popularity due to their efficiency. Although the literature on sentence analysis for controlling smart appliances is limited, it holds significant use cases. The success of speech recognition systems not only depends on modeling but also on feature extraction methods. Therefore, we constructed speech recognition models with diverse features, including Linear Predictive Coefficients (LPCs) [11], Linear Predictive Cepstral Coefficients (LPCCs), and Mel-Frequency Cepstral Coefficients (MFCCs) [12].

The paper is organized as follows: In Section II, we describe the literature on speech recognition for low-resource languages. In Section III, we elaborate on the methodology used in this study. The implementation details and results observed in our work are discussed in Section IV. Finally, we discuss future directions and provide conclusive remarks in Section V.

II. Related Work

In this section, we discuss the various works reported in the literature for the Telugu language and Convolutional Neural Networks (CNNs). A speech signal contains characteristics of the speaker, language, and information. Telugu has a unique set of sounds and phonemes that differ from other languages. The language comprises 16 vowels and 36 consonants, which produce a wide range of sounds. The retroflex nature of consonants makes the language complex to process with automated systems. It also follows a critical grammar system that follows the order Subject-Object-verb,

Manuscript received March 28, 2024; revised April 18, 2025.

P. Bhagath is a professor at the Lakireddy Bali Reddy College of Engineering, Mylavaram, Andhra Pradesh, India-521230 (e-mail: bhagath.2014@alumni.iitg.ac.in).

M. I. L. Prasanna is a graduate student at Lakireddy Bali Reddy College of Engineering in Mylavaram, Andhra Pradesh, India-521230 (e-mail: indulakshmi2002@gmail.com).

M. Shanmukha is a graduate student at Lakireddy Bali Reddy College of Engineering in Mylavaram, Andhra Pradesh, India-521230 (e-mail: malempatishanmukha@gmail.com).

P. G. Nagasri is a graduate student at Lakireddy Bali Reddy College of Engineering in Mylavaram, Andhra Pradesh, India-521230 (e-mail: gnananagasri29@gmail.com).

M. Swetha is a graduate student at Lakireddy Bali Reddy College of Engineering in Mylavaram, Andhra Pradesh, India-521230 (e-mail: mocherlaswetha07@gmail.com).

P. K. Das is a Professor at the Indian Institute of Technology Guwahati, Assam, India-781038 (e-mail:pkdas@iitg.ac.in).

which is different from English [13]. Vowels play a pivotal role as phonetic units aiding in sound recognition. Studying vowels makes a significant impact on processing the spoken language. A study by Prithviraj et al. thoroughly examined Telugu vowels and changes in acoustic characteristics in vowel regions using fundamental frequency, formant frequencies, and energy [14]. The vowels also play a key role in identifying the emotions of a person [15]. Shashidhar et al. studied the role of vowels in identifying emotions using LPCC and Artificial Neural Networks [16]. Notably, the development of the IITKGP:SESC emotional speech database has been instrumental in comprehending emotions through Telugu sentences. The emotion classification problem is well addressed by Kokane and Ram Mohana Reddy for IITGKGP:SESC dataset by proposing a multi-class SVM classification system utilizing MFCCs as the primary features [17].

The phonetic characteristics of the Telugu language were studied in combination with other languages such as Kannada, Bengali, and Odia. This has opened a multilingual phone recognition problem, and become a prominent one. Manjunath et al. proposed an HMM-based recognition system that uses MFCC features to represent the phonetic characteristics [18]. The understanding of phonetic characteristics not only helps in recognising the content but also can help to identify the language. Athira and Poorna proposed a language identification system based on CNN, which is capable of distinguishing between languages such as Bengali, Hindi, Tamil, and Telugu [19]. Venkateswarlu et al. have propelled the field forward with their development of a Telugu spoken alphabet recognition system, employing multi-layer perceptron and Time Lagged Recurrent Neural Network models. Their research unveiled the superior effectiveness of MFCCs over LPCCs for classification purposes [20]. Additionally, researchers have delved into the accent recognition problem, particularly focusing on Telugu language data [21].

Speech processing is crucial in the Internet of Things (IoT) environment, revolutionizing the way we interact with our surroundings. However, despite significant advancements in speech technology, regional languages like Telugu have been overlooked. For instance, S. P. Panda proposed a voice-based authentication system that enhances security for IoT devices. In this work, an IoT system was equipped with a speaker verification system. The system verifies the existing speech signature of the user against the user input. The verification is carried out with a time-aligned Dynamic Time Warping (DTW) algorithm. DTW compares the distance between two different patterns using a cost function given in 1. In 1, C refers to the distance between any two sequences s_i , t_i, and DTW computes the final cost in a recursive fashion [22].

$$DTW[i,j] := C[s_i,t_i] + min(DTW[m-1,n], DTW[m,m-1], DTW[m-1,n-1]) \quad (1)$$

Wongsorn et al. proposed an IoT environment that controls hospital beds using voice classification, allowing patients to control their beds through speech input in the Thai language. The approach was designed for the Thai language using MFCCs as features [23].

Research in speech recognition frameworks has evolved to accommodate the needs of small-scale mobile devices. In this direction, CMU PocketSphinx was developed, which makes speech recognition feasible in mobile devices [24]. PocketSphinx uses HMM as a modeling tool and MFCC as a feature extraction method. Researchers used this to address a voice-based navigation system for mobile robots with Hindi as a primary language, and showed to be effective with a recognition accuracy of 85% [25]. Furthermore, imagine the convenience of a speech interface designed for controlling various home appliances in Hindi, all through a user-friendly mobile application. This innovative solution supports a range of appliances, including Televisions, air conditioners, and lights, making home automation more accessible than ever before [26]. In a separate study, authors proposed CNN-based Telugu digit models for a recognition task. They designed a CNN model with MFCC as features while using different activation functions at internal and output layers. The model that utilised Relu and sigmoid activation functions demonstrated the best performance, achieving a recognition rate of 97.8% [27]. These breakthroughs in voice-controlled and speech interface technologies are revolutionising the way we interact with robots and home appliances, while also enhancing recognition tasks in various languages.

The role of Convolutional Neural Networks (CNNs) in advancing speech processing models is undeniable. For instance, Chen and Jian introduced a multi-featured emotion recognition system, leveraging Siamese networks and 3-D CNNs to extract crucial features [28]. In a similar vein, Akinpelu et al. proposed a lightweight deep neural network framework integrating Random Forest, Multi-Layer Perceptron (MLP), and the VGGNet architecture for emotion recognition [29]. Furthermore, Wubet et al. put forth a compelling approach utilizing shared features to enhance the performance of neural networks in accent classification. This involved feature extraction from similar accent utterances and the training of CNN and LSTM architectures, with the integration of outcomes from both models yielding remarkable results [30]. Moreover, Abdel-Hamid et al. introduced a groundbreaking hybrid NN-HMM framework for speech recognition, demonstrating its effectiveness in minimizing spectral variations of speech signals through local filtering and max-pooling layers, as proven on the TIMIT dataset [31]. In addition, Sreeram and Sinha's innovative Target-to-Word (T2W) transduction model offers a promising solution for code-switching speech recognition, thereby paving the way for recognizing multilingual utterances, such as the Hindi-English sentences in the HingCos corpus [32]. Additionally, the effective utilization of CNNs in speaker verification and keyword spotting for low-resource languages further underscores their versatility and impact in the field of speech processing [33]. Nadimpalli et al. developed a keyword search method based on spoken audio for low-resource languages. This procedure uses a metric called as keyword confusability to distinguish the words. This is defined using Levenshtein distance represented in 2 [34].

$$d_t = round\left(\frac{1}{N}\sum_{N=1}^{N} min\left(dist_{\forall i, t \neq w_{ni}}(t, w_{ni})\right)\right) \quad (2)$$

Pereira et al. studied different deep-learning models such as LeNet-5, SqueezeNet, and EfficientNet for keyword spotting.

In this work, the authors used Short-Time Fourier Transform (STFT) as features to represent the speech signals [35]. Our detailed methodology, outlined in the upcoming section, promises to make a substantial impact in the realm of speech processing for the Telugu language within an IoT framework.

III. Methodology

The initial stage in the recognition process involves the preparation of datasets that follow the preprocessing of the input sentences to normalize the data. The normalized signal is then transformed into a set of MFCC features, enabling us to decipher the patterns within the input speech signal. The recognition algorithm is designed to pinpoint the optimal model that can produce the given observation sequence with the highest probability among the available models. For the specified word sequence and observation sequence, the recognition algorithm evaluates how effectively the models can generate the word sequence. In this process, the prior probability $P(W)$ and the conditional probability of the observation sequence $P(O|W)$ are utilized. This crucial process equips the front end to accurately recognize the input word. In the following sections, we will delve deeper into these essential steps. First, we present the dataset preparation in the next subsection.

A. Dataset Preparation

The current dataset is specifically designed for developing recognition algorithms for controlling home appliances. The chosen keywords are tailored to meet the requirements of recognition systems in home automation, which need to control devices such as bulbs, fans, televisions, and air conditioners. Telugu speakers commonly use English equivalent words to refer to these appliances. Controlling these devices involves two activities: switching on and switching off, and the corresponding words for these actions vary for each device. Therefore, a set of spoken commands listed in Table I is included in the dataset. Each sentence used as a command consists of 2 words, in which each word is a combination of different phonetic units, including vowels and consonants.

TABLE I
List of Sentences used in the dataset

S.No	Telugu Sentence	English Equivalent Sentence
1.	Fan Tippandi	Switch on the fan
2.	Fan Aapandi	Switch off the fan
3.	Bulb Veliginchandi	Bulb on
4.	Bulb Aapandi	Bulb off
5.	Light Veliginchandi	Light on
6.	Light Aapandi	Light off
7.	TV Pettandi	Tv on
8.	TV Katteyandi	Tv off
9.	AC Veyandi	Ac on
10.	AC Aapandi	Ac off

The spoken sentences were recorded using a mobile application with specific settings. The first parameter, known as the channel type, determines how many channels are used in the recording process. For speech experiments, a single channel is usually preferred, and the dataset was recorded using a single channel. The second parameter is the sampling rate, which is determined based on the frequency range of human hearing, typically 16Hz to 20kHz. The Nyquist rate recommends a sampling rate of 16 kHz, meaning that the sampling rate must be higher than twice the maximum required frequency. Studies have shown that the frequency at which speech can be perceived is 8 kHz. Consequently, many speech databases use a sampling rate of 16 kHz. Figure 1 illustrates different recording sessions during the dataset collection. The dataset used for this research work is published in [36]. The second phase, also known as feature extraction, primarily gets the important clues from the raw speech signals. The detailed procedures used for extracting the features are elaborated in the next subsection.

Fig. 1. Recording sessions

B. Feature Extraction

One crucial step in the approach is to extract features that reveal essential patterns from the raw speech signals. This step starts with normalizing the speech signal by removing the DC component from the original signal. This is followed by a segmentation process that makes the source signal into equal-sized frames with a pre-specified overlap window length. Once the features are computed for a speech segment, the window moves towards the right side, and the computation procedure proceeds further until the complete signal is processed. This process is known as windowing and is commonly used in major feature extraction procedures. In the present study, we considered three different feature extraction methods known as Linear Predictive Coding (LPC), Linear Predictive Cepstral Coefficients (LPCCs), and Mel-frequency cepstral coefficients (MFCCs) [37].

1) LPC feature extraction: LPC was designed to model a human voice production system that comprises two essential components vocal tract and the vocal cords. In speech production, a speech signal $x[n]$ is uttered with the vibrations take place in these two components and is represented by 3 where $h(n)$ represents vocal tract and vocal cords are represented by $e(n)$.

$$x[n] = h[n] * e[n] \tag{3}$$

The primary aim of LPC is to estimate speech production parameters using the speech signal $x[n]$. In deriving these parameters, the samples of the speech signal are estimated based on the previous 'p' samples as shown in 4 where β_i is the autocorrelation coefficient.

$$x[n] = \beta_1 x[n-1] + \beta_2 x[n-2] + ... + \beta_k x[n-p] \tag{4}$$

For a source speech signal $x[n]$ with 'n' samples and a window size given by W_s, the LPC method requires the

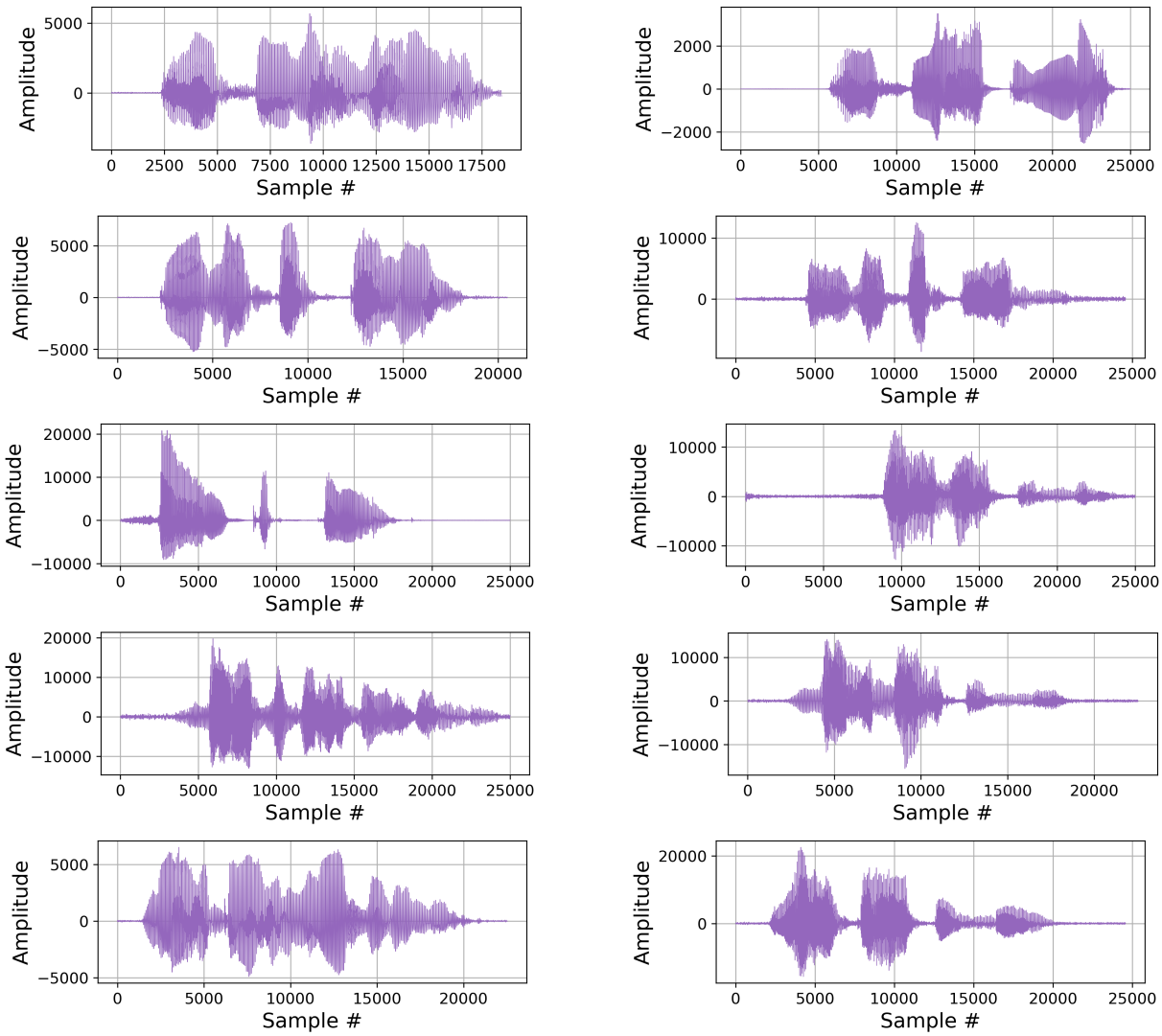

Fig. 2. Speech utterances for the sentence data

five different steps to compute LPC feature vectors, each consisting of 13 values. At first, a low-pass filter is used to flatten the speech signal and this step is called as pre-emphasis. This step is succeeded by frame blocking, where a spoken utterance is divided into an equal number of chunks with a pre-defined size given by W_s with an overlapping factor. The third step, windowing is used to minimize the effect of the signal at both ends. For a speech signal $x(n)$ and a Hamming window $h[n]$, each windowed sample can be obtained using 5 where K is the length of the window. The process of windowing is illustrated in Figure 4.

$$x_i(k) = \sum_{n=1}^{N} x_i(n)h(n)e^{j2\pi kn/N}, 1 \leq k \leq K \qquad (5)$$

The LPC cepstral coefficients are computed with an additional step after obtaining the LPC coefficients. The values are obtained from the Fourier transform of the logarithm of the magnitude spectrum.

2) MFCC Features Computation: The first step in computing MFCCs is pre-emphasis, where a high-pass filter is applied to modify the energy distribution across frequency components and adjust the energy levels. After pre-emphasis, the signal is segmented into equal-sized chunks using a windowing technique. Typically, speech segments are divided into 20-ms to 30-ms windows, employing a Hamming window to smoothly taper the edges of each segment. The hamming window used is represented by 5. This approach is necessary because speech signals are processed as short-term windows, as they are considered stationary over brief periods. While segmenting the signal, it is common practice to overlap these speech segments to preserve the temporal characteristics of the signal. Figure 4 illustrates the windowing process, where a short segment of speech is selected and a Hamming window is applied before calculating the MFCC features. Following segmentation, the power spectrum is derived using the Discrete Fourier Transform (DFT). The distribution of the frequency components is illustrated with DFT [38].

Next, the mel-filter bank is calculated, and the most significant coefficients are selected through the Discrete Cosine Transform (DCT) applied to the mel-filter bank. Each selected frame is used to compute 13 MFCC features, meaning that a speech frame is represented by an MFCC vector consisting of 13 values. The extraction procedure involves the equations from 6 through 8. The equations are used iteratively to extract MFCC features from the entire speech signal, resulting in a comprehensive set of MFCC

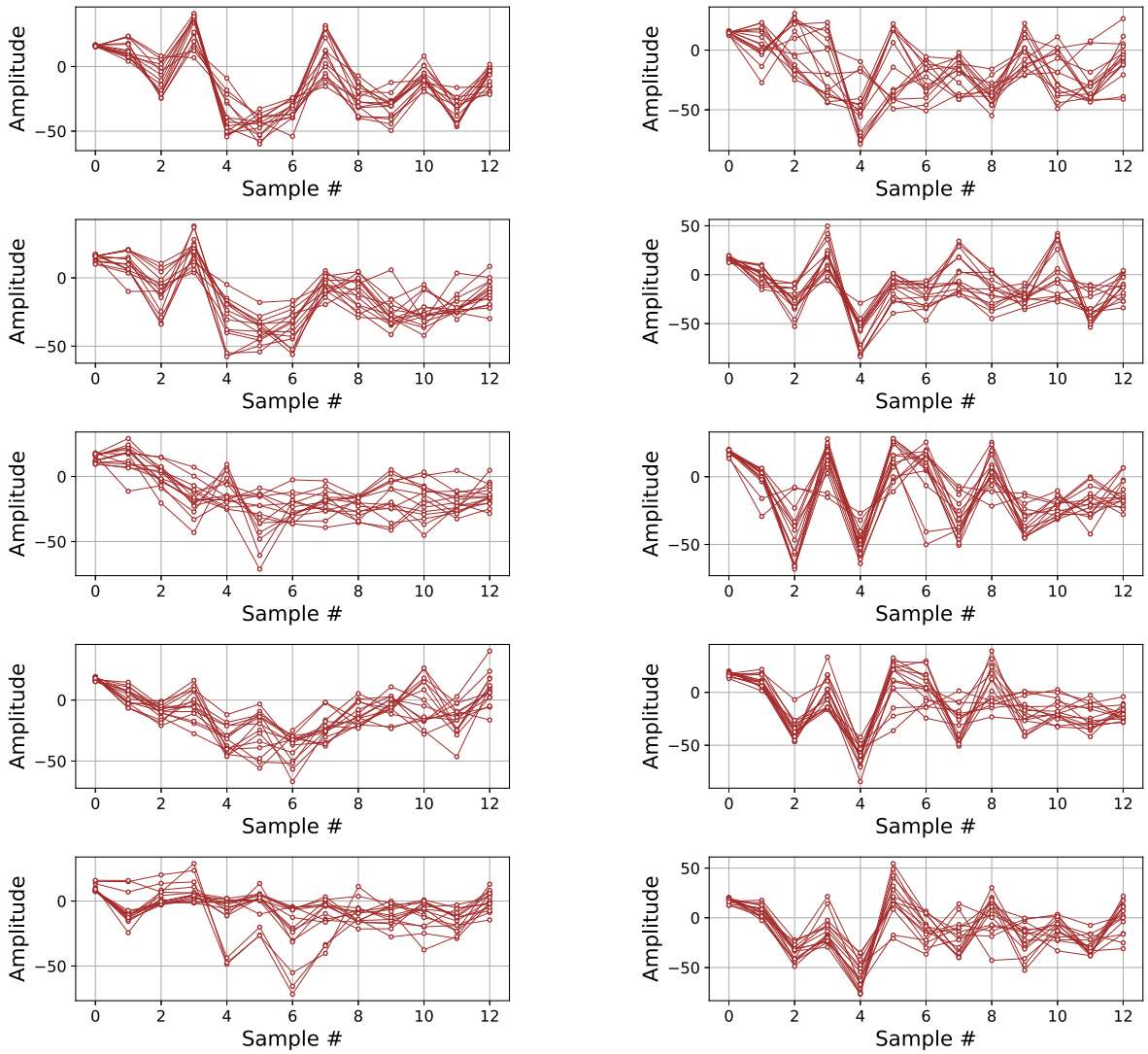

Fig. 3. 13 MFCC Features for the sentence data

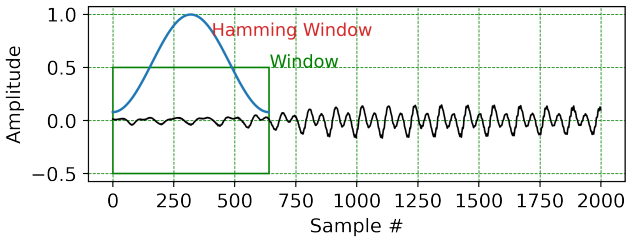

Fig. 4. Illustration of the windowing process for feature computation

features [39].

$$S_i(k) = \sum_{n=1}^{N} s_i(n)h(n)e^{j2\pi kn/N}, 1 \leq k \leq K \quad (6)$$

$$P_i(k) = 1/N|S_i(k)|^2 \quad (7)$$

$$H_m(k) = \begin{cases} 0 & k < f(m-1) \\ \frac{k-f(m-1)}{f(m)-f(m-1)} & f(m-1) \leq k \leq f(m) \\ \frac{f(m+1)-k}{f(m+1)-f(m)} & f(m) \leq k \leq f(m+1) \\ 0 & k > f(m+1) \end{cases} \quad (8)$$

After extracting features from the source speech signals, the next step is to construct the model. In this study, we employed two different models to improve our analysis. In the following subsection, we will discuss the application of the Hidden Markov Model (HMM) for sentence recognition, highlighting its significance and effectiveness in our research.

C. Sentence Modeling using HMM

HMM is known to be a stochastic process by which a process can be estimated via a model analogous to the actual one. HMM assumes that any signal is probably described as a random process with well-defined parameters that can be assessed. This is competent for speech recognition in which the characteristics of a speech signal are used for estimating the properties of system that produces the speech signals.

HMM is often described with a set of components represented by 9. The total number of states in HMM is denoted by $\mathbb{N}$, while $\mathbb{O}_{\aleph}$ indicates the number of distinct observation symbols associated with each state. The other three parameters are as follows: A is the state transition probability matrix, B is the symbol probability distribution (also known as the emission probability distribution), and π is the initial state probability distribution. The Hidden Markov Model (HMM)

utilizes several hyperparameters that can be adjusted across different iterations, with predefined initial values assigned prior to the training process. The current approach also incorporates additional hyperparameters, which are detailed in Table II.

A conventional HMM sentence recognition system can be built by answering the three important problems. For a given model and observation sequence, the first problem computes the probability that the given model can generate the input sentence. The second problem deals with choosing a corresponding optimal sequence of states in a model efficiently. The adjustment of model parameters such that the observation sequence generation by a model can be optimal is dealt by the third problem.

$$\lambda = (N, \mathbb{O}_{\aleph}, A, B, \pi) \tag{9}$$

TABLE II
DESCRIPTION OF THE PROPOSED HMM MODEL

S. No.	Parameter	Description
1.	Number of states	5
2.	Type of the Co-variance Matrix	Full
3.	Number of iterations for convergence	150
4.	Emission Probability	Gaussian Mixture

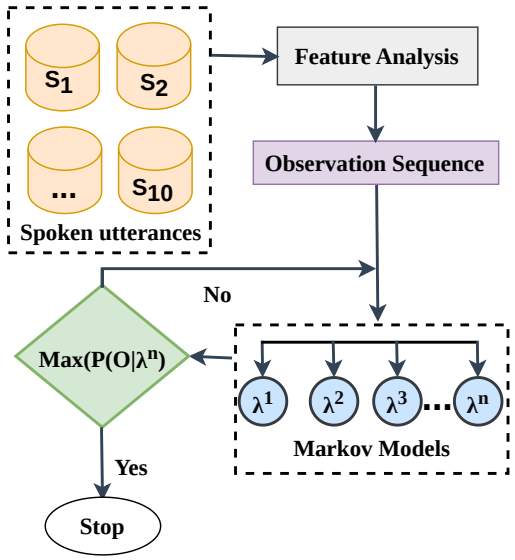

Fig. 5. HMM Based Sentence Modeling

To address these challenges, the overall procedure is organized into two primary steps: model building and recognition. Initially, the process involves utilizing ten distinct spoken sentences in Telugu, which are stored in .wav format, to construct sentence models. These spoken sentences facilitate the extraction of essential features and observation sequences required for developing the models. Upon the presentation of a new observation sequence, the Hidden Markov Model (HMM) adjusts its parameters to achieve optimal performance. In essence, it is necessary to develop a sequence generator that can accurately produce the designated observation sequence. The parameters of the model will be refined through iterative adjustments in response to new observation sequences encountered during this process. Figure 5 depicts the model-building procedure.

One of the key considerations in designing an HMM for sentence modeling is determining the appropriate number of states. The speech production mechanism is generally represented as a state machine, and the transitions are in HMM. HMM assumes that each state is responsible for emitting the observation symbols. The observation symbols can be phonetic units or a sequence of phonemes. The sequence of the phonemes is represented as transitions between the states in HMM. As it was discussed earlier, the states and transitions are associated with probabilities. Given a set of models $\psi = \lambda_1, \lambda_2, \lambda_3, ..., \lambda_n$ and an observation sequence $\mathbb{O}$, the recognition problem aims to find the most suitable model λ_k that can generate the input observation sequence $\mathbb{O}$.

A tri-state HMM model is commonly utilized in many speech systems. However, research has indicated that the number of states in an HMM can influence the performance, and this varies depending on language and applications. Consequently, we examined three different models as follows:

1) Three States Model
2) Four States Model
3) Five States Model

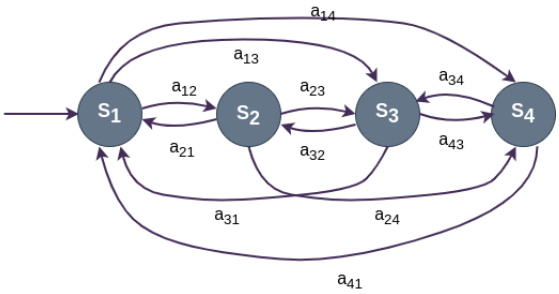

Fig. 6. 4-State HMM for the sentence modeling

Decoding involves deciphering the sentence through the hidden states of the HMM. To recognize a given sentence, the sentence recognition system first converts the input sentence into an observation sequence $\mathbb{O}$, which is then decoded by calculating the probability $P(\mathbb{O}|\lambda)$. This probability represents the likelihood of the model generating the observation sequence provided. Subsequently, the model with the highest probability is chosen, and the sentence label associated with the model is assigned to the input observation sequence. The process of sentence recognition is usually handled by the model evaluation process of HMM. It states the probability of a model generating a given observation sequence. For a given observation sequence O and a model λ, the evaluation process computes 10 to find the appropriate sentence model. While computing the probability, we should consider the probability of all conceivable hidden states and finally sum up those probabilities. In the recognition task shown in Figure 7, the method chooses the model that maximizes the probability computed in 10 for all the models.

$$P(O) = \sum_{r=1}^{r_max} P(O|\lambda_r)P(\lambda_r) \quad (10)$$

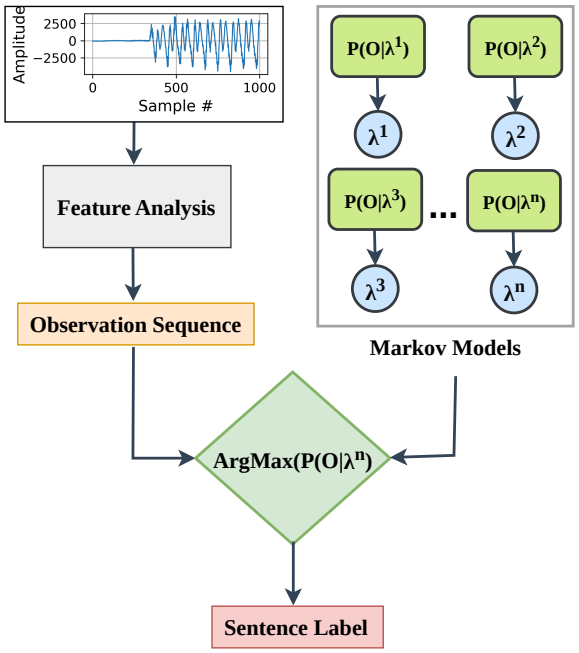

Fig. 7. HMM Based Sentence Recognition

D. Sentence Modeling using Convolutional Neural Networks

Convolutional Neural Network (CNN) is a deep learning model consisting of a sequence of layers known as convolution, pooling, fully connected, and output. Usually, a CNN model will have multiple levels of such layers, and the number of levels determines the depth of the architecture. The convolution layer at various levels is responsible for extracting meaningful patterns from the input data by applying a series of transformations. The second component of the network is the pooling operation used to reduce the dimensions of the feature matrix. Once the feature matrix is formed, a fully connected layer maps the feature matrix to a known class label. An activation function is generally used in a convolution layer to adapt the non-linearity that will help to handle complex data efficiently. The functionality of the proposed CNN architecture for sentence classification can be well understood with Figure 8.

Consider a sample space $S = \{s_1, s_2, s_3, ..., s_N\}$ with size N, and a set of class labels $\psi = \{\psi_1, \psi_2, \psi_3, ..., \psi_q\}$ with q be the number of sentence types. The primary goal of a CNN classification is to map each s_1 to ψ_1, s_2 to ψ_2 and so on. For doing this, the input matrix is required which is formed with a set of MFCC features extracted from the entire sentence data. Each input matrix consists of a set of rows and columns that reflect the number of speech segments and MFCCs, respectively. Generally, the number of features extracted from the speech signals is less than the size of the speech segment. Once the features are extracted, labels are assigned to each feature vector belonging to different sentences. This creates

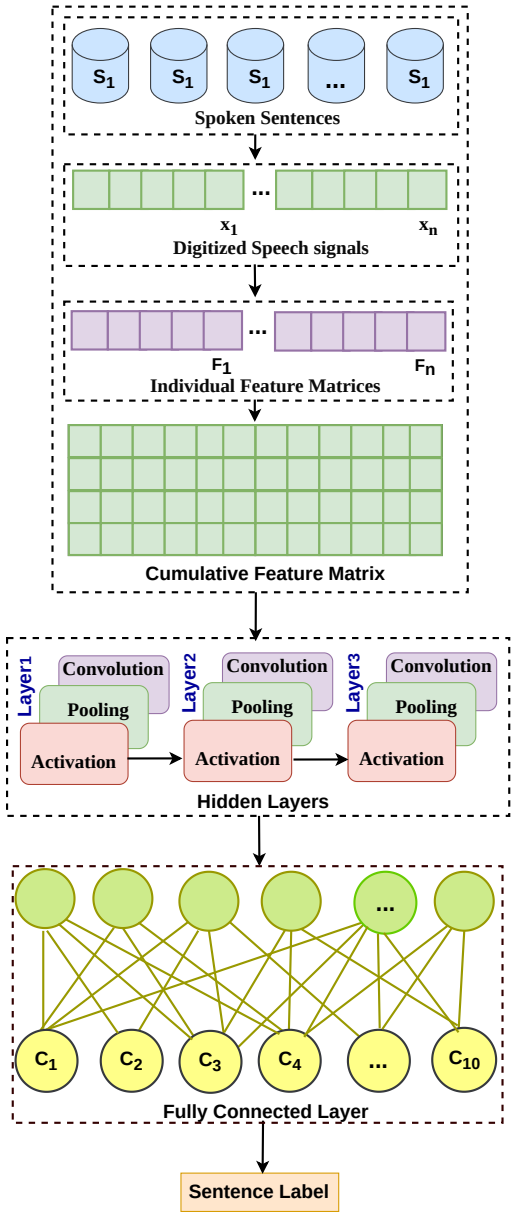

Fig. 8. **CNN-based Sentence Modeling Framework**

a feature matrix consisting of a feature vector and a class label. Subsequently, the feature matrix is processed through different layers of the CNN model to form a network with 10 different classes. The proposed CNN for sentence recognition is shown in Figure 9. The model has different components as listed below:

1) Convolution Layer
2) Pooling
3) Dense
4) Dropout
5) Optimizer
6) Activation function

The input matrix is first convoluted using a kernel in the convolution layer, performing dot products to create a feature map representing 10 different classes. The proposed

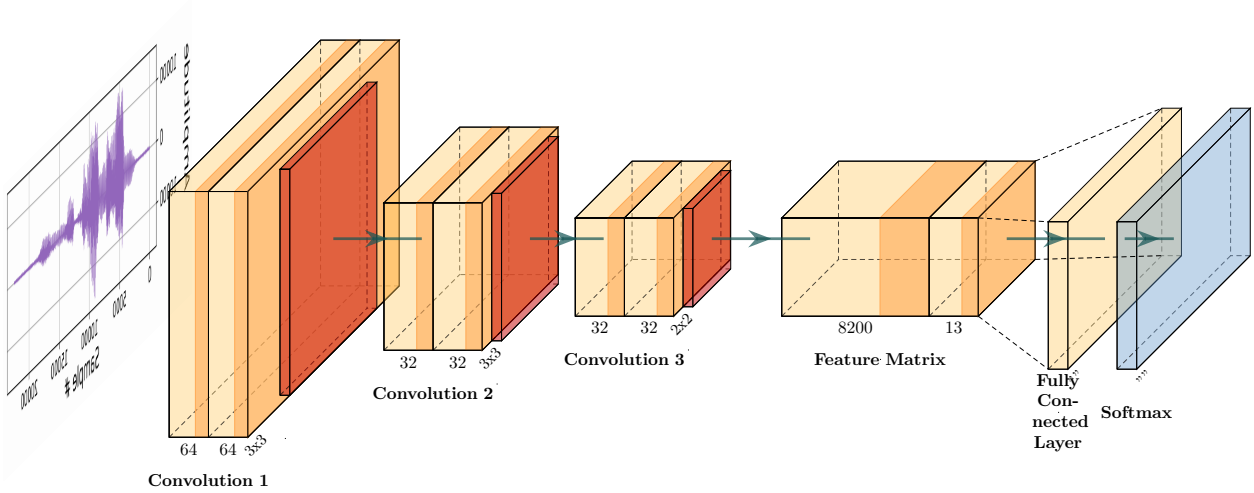

Fig. 9. Visualization of CNN Architecture

architecture has 3 convolution layers in which each layer processes a feature matrix of sizes 64×64, 32×32, and 32×32. This step requires a kernel, and it is defined as a square matrix. We used kernel sizes of 3×3, 3×3, 2×2 for three subsequent layers. The convolution layer transforms the input matrix by processing the feature matrix using a kernel. For any feature matrix designated by $F_{64\times64}$ and a kernel represented by $K_{3\times3}$, the convolution operation can be given as shown in Equation 11 which gives a 62 output matrix.

$$F * K = \sum_{a=-\infty}^{\infty} F(a)K(t-a) \tag{11}$$

The CNN model consists of an activation function to learn nonlinear patterns by adjusting weights at different layers. This can be achieved by activating a few neurons of the network and selectively choosing the output from the required neurons. Many choices for this step include ReLU (Rectified Linear Unit), Tanh (Hyper tangent), ELU (Exponential Linear Unit), SELU (Scaled Exponential Linear Unit), Softmax, etc [40]. For an input y_i, the functionality of activation functions can be well understood with 12 through 16. In 16, ω and α are the constants with approximately 1.0507 and 1.6733 respectively. In the present model, the ReLU was used as it was proven successful in most of the CNN architectures [41].

$$softplus(y_i) = log \times (1 + e^{y_i}) \tag{12}$$

$$softmax(y_i) = \frac{e^{y_i}}{\sum_{j=1}^{N} e^{y_j}} \tag{13}$$

$$tanh(y_i) = \frac{e_i^y - e^{-y_i}}{e_i^y + e^{-y_i}} \tag{14}$$

$$RELU(y_i) = max(0, y_i) \tag{15}$$

$$\begin{aligned} SELU(y_i) &= \omega \times y_i, y_i \geq 0, \\ &= \omega \times \alpha \times e^{y_i} - 1, y_i < 0 \end{aligned} \tag{16}$$

The next subsequent step after the activation function is pooling that down-samples the feature matrix with a max pooling operation. This operation replaces a subset of the feature matrix with statistics that represent the subset, reducing the overall number of parameters in the model and making it computationally efficient. In addition to that, the max pooling generates a smaller number of useful features that can be processed subsequently by the higher layers. It also ensures that the network is capable of handling slight variations in spoken utterances [42]. The flattened layer placed after three convolution layers transforms the multi-dimensional features into a vector with a single dimension. This feature vector is further taken by a fully connected layer, which essentially connects neurons from different layers. This enables the model to learn possible non-linear combinations of the features learned by the convolution layers. The final layer determines the most probable match by computing class probabilities. Before the output layer, a dropout layer is used in the network to avoid the effect of overfitting. This layer drops the connections between the neurons that are unnecessary in the network, stopping the network from learning the features that are not essential for classification. The architecture of the model and the parameters used in the proposed architecture are summarized in Table III. The proposed CNN architecture was trained and evaluated with the Telugu sentence dataset. Details of the dataset and the obtained results are discussed in Section IV.

TABLE III
DETAILS OF THE PROPOSED CNN MODEL ARCHITECTURE

	Layer	Size	Kernel Size	Stride	Activation
Input	Feature Matrix	-	-	-	-
1	Convolution	64×64	3	1	**SeLU**
2	Max Pooling	3×3	-	2	**SeLU**
3	**Convolution**	32×32	3	1	**SeLU**
4	Max Pooling	3×3	2	2	**SeLU**
5	**Convolution**	32×32	2	1	**SeLU**
6	Max Pooling	2×2	2	2	**SeLU**
7	Dense Layer	64	-	-	**Softmax**
Output	FC	10	-	-	**Softmax**

IV. IMPLEMENTATION DETAILS, RESULTS AND DISCUSSIONS

In this section, we outline the implementation details and the observations from the experiments conducted. The necessary programs for feature extraction and modeling were developed as Python modules. We utilized several libraries, including sklearn, Python_Speech_Features, librosa, and TensorFlow. Specifically, librosa and Python_Speech_Features were employed for feature extraction, while sklearn and TensorFlow provided modules for HMM and CNN, respectively. The experiments were conducted using the TELUGU VAKYALU dataset, which consists of 8,200 sentences spoken by 40 different speakers [5]. The dataset creation procedure is discussed in Section III-A. Initially, the raw speech data was transformed into useful features, a process detailed in Section III-B. In this study, we evaluated the performance of four different models and proposed an optimal model for sentence classification. The analysis included two types of models: the HMM and the CNN, along with three feature extraction methods: LPC, LPCC, and MFCCs. This led to the following models: LPC+HMM, LPCC+HMM, MFCC+HMM, and MFCC+CNN. Initially, we built HMM models using the LPC, LPCC, and MFCC features. Our findings indicated that MFCC features provided the best representation of the sentence data we analyzed. Subsequently, we used MFCC features to train the CNN models with the sentence dataset.

A. Performance of HMM Models

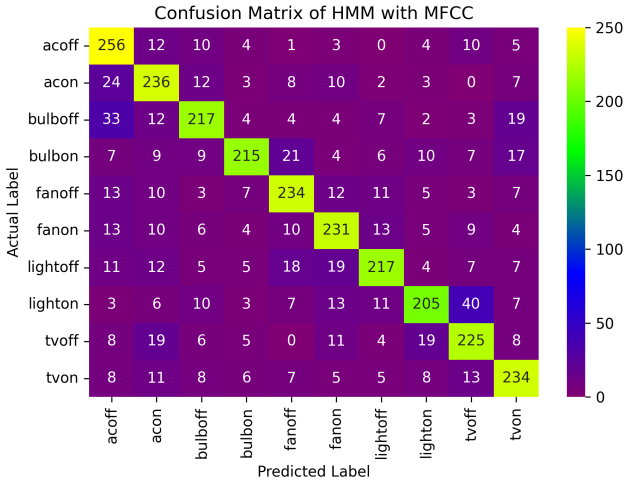

Fig. 10. Confusion Matrix for the HMM+MFCC Sentence Model

Table VI summarizes the recognition rates of 40 speakers across four different models. In the first model, LPC+HMM (Linear Predictive Coding combined with Hidden Markov Models), LPC analyzes speech signals as outputs from both the vocal cords and the vocal tract. This model treats the source of the speech as producing a sequence of sounds that pass through a resonant filter. A sentence represents a series of acoustic events occurring at the source before being filtered. The recognition rates for this model range from 40% to 80% for different speakers across all 10 sentences. Linear Predictive Cepstral Coefficients (LPCCs), which can be derived from LPC coefficients, are computed by applying the inverse Fourier transform to the log magnitude of the frequency response of an autoregressive filter. The LPCC+HMM model shows improved recognition accuracy, ranging from 45.7% to 87.56%. The model that uses Mel-Frequency Cepstral Coefficients (MFCCs) captures the frequency components of a speech signal and enhances the performance compared to the LPC and LPCC models. The recognition rates for speakers using this model vary from 50% to 97.5%.

B. Performance of CNN Model

The analysis of the performance of MFCC features led to the implementation of a CNN model, which is described in Section III-C. This model undergoes multiple iterations known as epochs, and the entire process is referred to as optimization. During optimization, the model fine-tunes its parameters by evaluating its performance against validation data. In each epoch, the model assesses its performance using various metrics, including model accuracy, model loss, validation accuracy, and validation loss. Ideally, we expect the model's accuracy to increase while its loss decreases gradually. The progress of these metrics over 60 epochs is illustrated in Figure 11. We observed fluctuations in these metrics for the first 50 epochs, after which the model began to stabilize from the 50^{th} epoch onward. To further examine the model's behavior, we extended the analysis by running an additional 10 epochs. During this period, we found that the model's performance metrics remained unchanged.

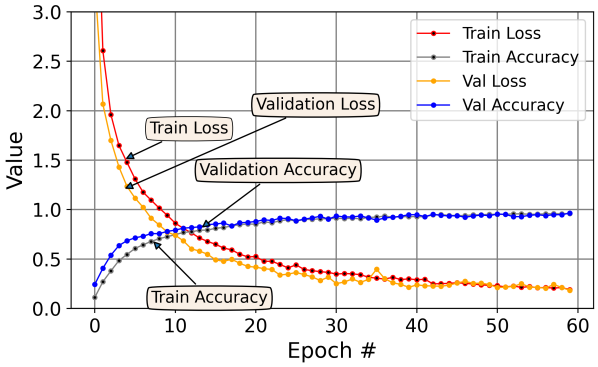

Fig. 11. CNN Performance Metrics for 60 iterations

The sentence recognition was evaluated against predicted and true values. The confusion matrix for a single speaker, consisting of 160 sentences, is presented in Figure 12. This confusion matrix indicates a high recognition rate for the sentences. A summary of the recognition rates for 10 sentences from the complete dataset is provided in Table V. The sentence "AC Veyandi," meaning "Switch on the AC," achieved the highest recognition rate, while "Bulb Veliginchandi," meaning "Switch on the bulb," recorded the lowest. The performance of the proposed MFCC+CNN model is represented by the red line, demonstrating that this model outperforms the other models based on HMM for most speakers.

The training procedure for the Convolutional Neural Network (CNN) involves multiple epochs and incorporates various activation functions across different layers. The proposed CNN model and its activation functions are summarized in

TABLE IV
PERFORMANCE OF PROPOSED CNN ARCHITECTURE WITH DIFFERENT ACTIVATION FUNCTIONS

	Activation Functions					Metrics			
S. No	Layer 1	Layer 2	Layer 3	Dense Layer	Output Layer	Train Accuracy	Train Loss	Validataion Accuracy	Validataion Loss
1	relu	relu	relu	relu	softmax	100	19.69	90.91	83.02
2	selu	selu	selu	selu	softmax	100	15.37	90.91	58.04
3	elu	elu	elu	elu	softmax	100	17.49	100	37.04
4	tanh	tanh	tanh	tanh	softmax	100	26.02	72.73	105.31
5	relu	selu	elu	tanh	softmax	100	24.52	90.91	80.11
6	relu	relu	relu	relu	softplus	83.64	39.66	82.82	115.14
7	selu	selu	selu	selu	softplus	99.81	15.57	90.91	61.26
8	elu	elu	elu	elu	softplus	88.97	20.83	81.82	43.58
9	tanh	tanh	tanh	tanh	softplus	100	47.52	100	50.66
10	relu	selu	elu	tanh	softplus	95.66	63.57	82.82	84.07

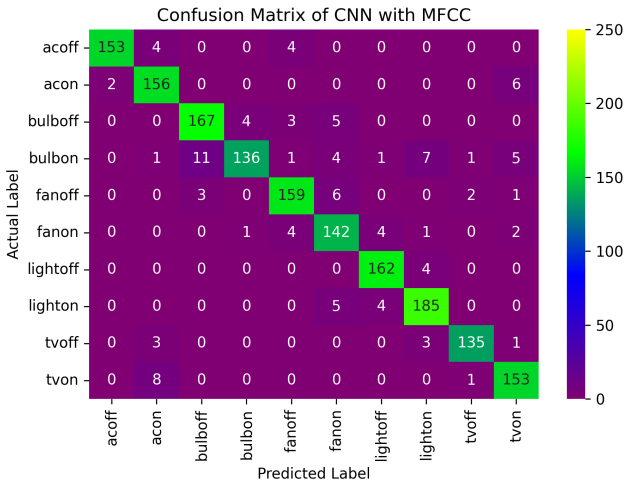

Fig. 12. Confusion Matrix for the CNN+MFCC Sentence Model

TABLE V
CLASSIFICATION REPORT FOR MFCC USING CNN

S.No	Sentence	MFCC+CNN
1	Fan Thippandi	94
2	Fan Aapandi	98
3	Bulb Veliginchandi	94
4	Bulb Aapandi	99
5	Light Veliginchandi	96
6	Light Aapandi	95
7	TV Pettandi	95
8	TV Katteyandi	99
9	AC Veyandi	99
10	AC Aapandi	95

Table IV. This model utilizes the SELU activation function in the internal layers and the softmax function in the dense and output layers. The selection of these functions was based on an empirical evaluation of the CNN architecture's performance using different combinations of activation functions. The study examined various activation function options, leveraging existing knowledge to inform these choices. The performance metrics for the CNN using these combinations are presented in Table IV. From this table, it is evident that the proposed model achieves the highest training accuracy along with the lowest loss. The model closest to optimal performance is the one using the ReLU activation function in the internal layers and softmax in the output layer, resulting in a training loss of 19.60%. Although the model utilizing the ELU activation function showed promising accuracy, it struggled with test data during the experiments. This indicates a need for further optimization to fully realize its potential. The SELU activation function has an advantage over ReLU due to its self-normalization capability. Like ReLU, SELU does not suffer from the vanishing gradient problem and can learn more quickly, which contributes to its effectiveness in the proposed model for sentence recognition. The precision, recall, and F1-score metrics for the proposed model, along with those for other models, are shown in Table VII. The computation procedure for these metrics is outlined from 17 to 19. In this context, TP indicates True Positives, FP denotes False Positives, and FN represents False Negatives. TP reflects the number of correctly predicted sentences, while FP and FN denote the counts of incorrectly predicted sentences. Finally, the performance of the models studied in this work is compared in Figure 14. It is clear that the proposed CNN model outperforms the HMM models with various features.

$$Precision = \frac{TP}{TP + FP} \tag{17}$$

$$Recall = \frac{TP}{TP + FN} \tag{18}$$

$$F1Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{19}$$

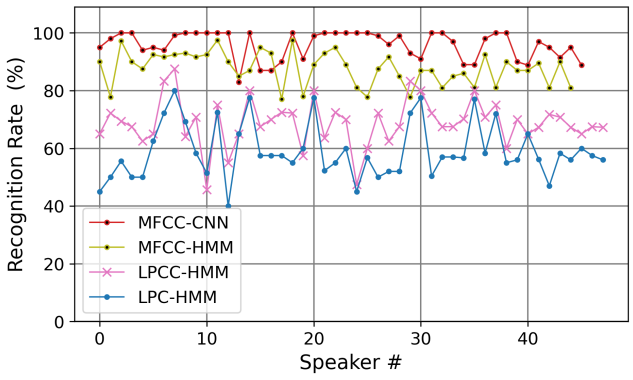

Fig. 13. Sentence Recognition Accuracy of LPC-HMM, LPCC-HMM, MFCC-HMM, MFCC-CNN Models

V. CONCLUSION AND FUTURE WORK

In this paper, we studied HMM and CNN frameworks for modeling Telugu sentences using different feature extraction methods, including LPC, LPCC, and MFCC. After analyzing

TABLE VI
RECOGNITION RATE OF DIFFERENT SPEAKERS

Speaker #	HMM			CNN
	LPC	LPCC	MFCC	MFCC
1	45	65	90	95
2	50	72.22	77.77	98
3	55.55	69.44	97.22	100
4	50	67.5	90	100
5	50	62.5	87.5	94
6	62.5	65	92.5	95
7	72.22	83.33	91.67	94
8	80	87.5	92.5	99.2
9	69.23	64.1	93	100
10	58.33	70.83	91.67	100
11	51.42	45.71	60	100
12	72.5	75	92.5	100
13	40	55	97.5	100
14	65	65	90	83
15	77.5	80	85	100
16	57.5	67.5	87	87
17	57.5	70	95	87
18	57.5	72.5	93	90
19	55	72.22	77	100
20	60	57.5	97.5	91
21	77.5	80	78	99
22	52.27	63.63	89	100
23	55	72.5	93	100
24	60	70	95	100
25	45	47.5	89	100
26	56.75	60	81.08	100
27	50	72.2	77.77	99
28	52	62.5	87.5	96
29	52	67.5	50	99
30	72.2	83.33	91.68	93
31	77.54	80	85	91
32	50.4	72.22	77.77	100
33	57	67.5	87	100
34	57	67.5	87	97
35	56.7	70.2	81	89
36	77	80	85	89
37	58.33	70.8	86	98
38	72	75	81.08	100
39	55	60	92.5	100
40	56	70	81.08	90

TABLE VII
PERFORMANCE OF DIFFERENT CLASSIFIERS

S.No	Model	Precision	Recall	F1-Score
1.	LPC + HMM	0.60	0.55	0.57
2.	LPCC + HMM	0.57	0.56	0.56
3.	MFCC + HMM	0.75	0.74	0.74
4.	**MFCC + CNN**	0.93	0.92	0.93

these models, we concluded that the MFCC+CNN model had a comparatively better recognition rate than the other models, LPC+HMM, LPCC+HMM, and MFCC+HMM. Although MFCC+HMM had a decent recognition accuracy, MFCC+CNN was the best model for sentence classification with 96.4% accuracy. The paper discusses the detailed steps taken in each approach by presenting the results obtained for different models. However, the phonetic characteristics in different contexts need to be examined to further improve the recognition rate. This can be taken up as future work.

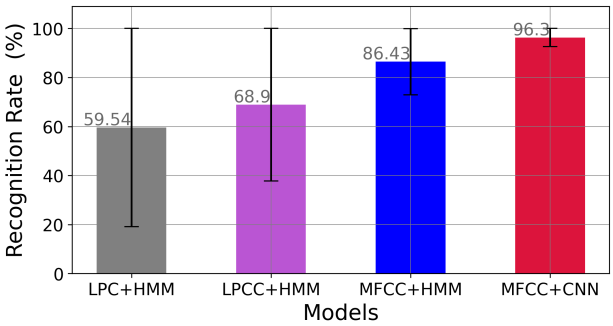

Fig. 14. Recognition rate of different models

REFERENCES

[1] S. Ritchie, Y.-C. Cheng, M. Chen, R. Mathews, D. van Esch, B. Li, and K. C. Sim, "Large vocabulary speech recognition for languages of africa: multilingual modeling and self-supervised learning," arXiv, 2022, arXiv:2208.03067.

[2] E. Gauthier, L. Besacier, S. Voisin, M. Melese, and U. P. Elingui, "Collecting resources in sub-Saharan African languages for automatic speech recognition: a case study of Wolof," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation 2016*, LREC 2016, 23-28 May 2016, Slovenia, pp. 3863–3867.

[3] "Application of iot voice devices based on artificial intelligence data mining in motion training feature recognition," *Measurement: Sensors*, vol. 34, p. 101260, 2024.

[4] M. Mehrabani, S. Bangalore, and B. Stern, "Personalized speech recognition for internet of things," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, 2015, pp. 369–374.

[5] P. D. P. K. D. Parabattina Bhagath, Vanga Lasya, "Telugu vakyalu: Spoken telugu sentences for iot applications," in *26th Conference of the Oriental COCOSDA International Committee for the Co-ordination and Standardisation of Speech Databases and Assessment Techniques 2023*, O-COCOSDA 2023, 04-06 December 2023, India, pp. 1–5.

[6] W. Ying, "English pronunciation recognition and detection based on hmm-dnn," in *11th International Conference on Measuring Technology and Mechatronics Automation 2019*, ICMTMA 2019, 28-29 April 2019, China, pp. 648–652.

[7] W. Wang, W. Xu, X. Sui, L. Wang, and X. Liu, "Investigations of low resource multi-accent mandarin speech recognition," in *IEEE International Conference on Information and Automation 2015*, ICIA 2015, 8-10 August 2015, China, pp. 62–66.

[8] S. Ando and H. Fujihara, "Construction of a large-scale japanese asr corpus on tv recordings," in *IEEE International Conference on Acoustics, Speech and Signal Processing 2021*, ICASSP 2021, 6-11 June, Canada, pp. 6948–6952.

[9] P. Bhagath, M. Pullagura, and P. K. Das, "Comparative analysis of spoken telugu digits using mfcc and lpcc via hidden markov models," in *10th International Conference on Signal Processing and Integrated Networks 2023*, SPIN 2023, 23-24 March 2023, India, pp. 615–620.

[10] X. Li and X. Wu, "Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing 2015*, ICASSP 2015, 19-24 April 2015, Australia, pp. 4520–4524.

[11] P. Hedelin and J. Skoglund, "Vector quantization based on gaussian mixture models," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 4, pp. 385–401, 2000.

[12] A. Chowdhury and A. Ross, "Fusing mfcc and lpc features using 1d triplet cnn for speaker recognition in severely degraded audio signals," *IEEE Transactions on Information Forensics and Security*, vol. 15, no. 1, pp. 1616–1629, 2020.

[13] V. K. R. Maddela and P. Bhaskararao, "Phonetic–acoustic characteristics of telugu lateral approximants," *Circuits Syst. Signal Process.*, vol. 41, no. 6, p. 3508–3546, Jun. 2022. [Online]. Available: https://doi.org/10.1007/s00034-021-01949-6

[14] P. R. Myakala, R. Nalumachu, and V. K. Mittal, "Study of telugu vowels using acoustic features," in *IEEE Region 10 Conference 2016*, TENCON 2016, 22-25 November 2016, Singapore, pp. 864–868.

[15] Shashidhar Koolagudi G, Shivakranthi B, K Sreenivasa Rao, and P. B. Ramteke, "Contribution of telugu vowels in identifying emotions," in *Eighth International Conference on Advances in Pattern Recognition 2015*, ICAPR 2015, 27-31 July 2015, Istanbul, pp. 1–6.

[16] S. Renjith and K. G. Manju, "Speech based emotion recognition in tamil and telugu using lpcc and hurst parameters — a comparitive study using knn and ann classifiers," in *International Conference on Circuit, Power and Computing Technologies 2017*, ICCPCT 2017, 20-21 April 2017, India, pp. 1–6.

[17] K. Amol T. and R. M. R. Guddeti, "Multiclass svm-based language-independent emotion recognition using selective speech features," in

International Conference on Advances in Computing, Communications and Informatics 2014, ICACCI 2014, 24-27 September 2014, India, pp. 1069–1073.

[18] K. E. Manjunath, K. S. Rao, and D. B. Jayagopi, "Development of multilingual phone recognition system for indian languages," in *IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems 2017*, SPICES 2017, 8-10 August 2017, India, pp. 1–6.

[19] A. N..P and P. S.S., "Deep learning based language identification system from speech," in *International Conference on Intelligent Computing and Control Systems 2019*, ICCS 2019, 20-24 May 2019, China, pp. 1094–1097.

[20] R. L. K. Venkateswarlu, R. R. Teja, and R. V. Kumari, "Developing efficient speech recognition system for telugu letter recognition," in *International Conference on Computing, Communication and Applications 2012*, ICCCA 2012, 22-24 February 2012, India, pp. 1–6.

[21] G. R. Krishna, R. Krishnan, and V. K. Mittal, "Foreign accent recognition with south indian spoken english," in *IEEE 17th India Council International Conference 2020*, INDICON 2020, 10-13 December 2020, India, pp. 1–5.

[22] S. P. Panda, "Intelligent voice-based authentication system," in *Third International Conference on IoT in Social, Mobile, Analytics and Cloud 2019*, I-SMAC 2019, 12-14 December 2019, Nepal, pp. 757–760.

[23] A. Wongsorn, K. Srijiranon, and N. Eiamkanitchat, "Application of iot using neuro-fuzzy based on thai speech classification to control model hospital bed with arduino," in *IEEE 8th Global Conference on Consumer Electronics 2019*, GCCE 2019, 15-18 October 2019, Japan, pp. 702–706.

[24] D. Huggins-Daines, M. Kumar, A. Chan, A. Black, M. Ravishankar, and A. Rudnicky, "Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 1, 2006, pp. I–I.

[25] B. Parabattina, P. Chandra, V. Sharma, and P. K. Das, "Voice-controlled assistance for robot navigation using android-based mobile devices," in *2021 Sixth International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 2021, pp. 21–25.

[26] P. Bhagath, S. Parihar, and P. K. Das, "Speech recognition for indian spoken languages towards automated home appliances," in *2021 2nd International Conference for Emerging Technology (INCET)*, 2021, pp. 1–5.

[27] P. Bhagath, A. U. M. Rao, B. S. Ram, and M. A. K. Reddy, "Telugu spoken digits modeling using convolutional neural networks," in *IEEE 13th International Conference on Pattern Recognition Systems 2023*, ICPRS 2023, 4-7 July 2023, Ecuador, pp. 1–7.

[28] G. Chen and S. Jiao, "Speech-visual emotion recognition by fusing shared and specific features," *IEEE Signal Processing Letters*, vol. 30, pp. 678–682, 2023.

[29] S. Akinpelu, S. Viriri, and A. Adegun, "Lightweight deep learning framework for speech emotion recognition," *IEEE Access*, vol. 11, pp. 77 086–77 098, 2023.

[30] Y. A. Wubet, D. Balram, and K.-Y. Lian, "Intra-native accent shared features for improving neural network-based accent classification and accent similarity evaluation," *IEEE Access*, vol. 11, pp. 32 176–32 186, 2023.

[31] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing 2012*, ICASSP 2012, 25-30 March 2012, Japan, pp. 4277–4280.

[32] G. Sreeram and R. Sinha, "Exploration of end-to-end framework for code-switching speech recognition task: Challenges and enhancements," *IEEE Access*, vol. 8, pp. 68 146–68 157, 2020.

[33] S.-H. Kim, H. Nam, and Y.-H. Park, "Analysis-based optimization of temporal dynamic convolutional neural network for text-independent speaker verification," *IEEE Access*, vol. 11, pp. 60 646–60 659, 2023.

[34] V. L. V. Nadimpalli, S. Kesiraju, R. Banka, R. Kethireddy, and S. V. Gangashetty, "Resources and benchmarks for keyword search in spoken audio from low-resource indian languages," *IEEE Access*, vol. 10, pp. 34 789–34 799, 2022.

[35] P. H. Pereira, W. Beccaro, and M. A. Ramírez, "Evaluating robustness to noise and compression of deep neural networks for keyword spotting," *IEEE Access*, vol. 11, pp. 53 224–53 236, 2023.

[36] P. Bhagath, V. Lasya, P. Dhyeya, and P. K. Das, "Telugu vakyalu: Spoken telugu sentences for iot applications," in *2023 26th Conference of the Oriental COCOSDA International Committee for the Coordination and Standardisation of Speech Databases and Assessment Techniques (O-COCOSDA)*, 2023, pp. 1–5.

[37] S. Davis and P. 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.

[38] P. Bhagath, M. Jain, and P. K. Das, "Dynamic speech trajectory based parameters for low resource languages," in *Machine Learning, Image Processing, Network Security and Data Sciences*, A. Bhattacharjee, S. K. Borgohain, B. Soni, G. Verma, and X.-Z. Gao, Eds. Singapore: Springer Singapore, 2020, pp. 256–270.

[39] W. Jiang, P. Liu, and F. Wen, "Speech magnitude spectrum reconstruction from mfccs using deep neural network," *Chinese Journal of Electronics*, vol. 27, no. 2, pp. 393–398, 2018.

[40] S. R. Dubey, S. K. Singh, and B. B. Chaudhuri, "Activation functions in deep learning: A comprehensive survey and benchmark," *Neurocomputing*, vol. 503, pp. 92–108, 2022.

[41] H. Li, Y. Liang, H. Gao, L. Liu, Y. Wang, D. Chen, Z. Luo, and G. Li, "Silent speech interface with vocal speaker assistance based on convolution-augmented transformer," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–11, 2023.

[42] A. Dutt and P. Gader, "Wavelet multiresolution analysis based speech emotion recognition system using 1d cnn lstm networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 2043–2054, 2023.

Authors Biographies

Dr Parabattina Bhagath is a Professor of Artificial Intelligence & Data Science at Lakireddy Bali Reddy College of Engineering, Mylavaram, Andhra Pradesh, India. Before joining LBRCE, he worked as a Project Engineer in the Department of Civil Engineering at IIT Guwahati, India. He has been teaching for 15 years for the undergraduate students of CSE.

Dr P. Bhagath became a professional member of IAENG in 2020. He completed a Master of Technology in CSE from the Indian Institute of Technology, India in 2010 and a Doctor of Philosophy from the Indian Institute of Technology, India in 2020. His research interests include speech processing, graph signal processing, and IoT.

Dr Bhagath is a Senior member of IEEE since 2022 and a professional member of ACM since 2020.

M Indu Lakshmi Prasanna completed her Bachelor of Technology in Computer Science and Engineering at Lakireddy Bali Reddy College of Engineering, Mylavaram, Andhra Pradesh, India. Her research pursuits focus on speech processing, an interdisciplinary field combining computational linguistics and artificial intelligence. Specifically, she explores speech recognition, language modeling, and natural language processing for low-resource languages. Ms. Prasanna has developed language models for spoken units in multiple languages, including Telugu and English. Her work involves analyzing spoken language structures, creating algorithms for speech processing, and enhancing human-machine interaction. As a student member of IEEE (2023), Ms. Prasanna remains committed to advancing her expertise in speech technology and contributing to the growth of language technologies.

M Shanmukha has successfully completed his Bachelor of Technology in Computer Science and Engineering at Lakireddy Bali Reddy College of Engineering in Mylavaram, Andhra Pradesh, India. Currently, his research focuses on the dynamic field of speech processing—an exciting intersection of computational linguistics and artificial intelligence that transforms the way computers engage with human languages through spoken communication.

Shanmukha has made significant strides in developing advanced language models for various spoken units across multiple languages, including Telugu, English, and Hindi. These innovative models are crafted to improve the understanding and generation of human dialogue, paving the way for seamless and intuitive interactions between people and machines. His research meticulously analyzes the intricacies of spoken language, resulting in powerful algorithms designed to process and interpret speech with remarkable effectiveness.

Gnana Nagasri P completed her B.Tech in Computer Science and Engineering at Lakireddy Bali Reddy College of Engineering, located in Mylavaram, India. Her research focuses on Telugu spoken language modeling for classifying gender and age groups using Hidden Markov Models(HMM). By analyzing speech patterns and leveraging techniques like Graph Eigenvalues. She has contributed to developing effective solutions for speech-based applications in regional languages. She can be contacted at gnananagasri29@gmail.com .

Swetha Mocherla completed her Bachelor of Technology in Computer Science and Engineering from Lakireddy Bali Reddy College of Engineering, Mylavaram, India. Her primary area of interest is speech processing, a specialized field at the intersection of computational linguistics and artificial intelligence that focuses on enabling seamless communication between humans and computers through spoken language.

Swetha has worked on developing language models for spoken units in the Telugu language. The model is aimed to improve the ability of machines to comprehend and generate human speech, fostering the relevant and natural interactions between humans and the machines. Her research involves the detailed analysis of language structures and the creation of algorithms capable of interpreting and processing speech signals effectively.

Prof. Pradip K Das completed B.Sc (Hons) from Gauhati University in the year 1989. He completed his M.Sc (Mathematical Statistics) and PhD (Computer Science) from the University of Delhi in the years 1991 and 1999 respectively.

Dr. P. K. Das is currently holding the position of Professor in the Department of CSE, IIT Guwahati, India. He worked as Sr. Lecturer in the Department of CSE, IIT Guwahati from 1998 to 2008. He worked as an Associate Professor in the same institute from 2008 to 2015.

Dr. Das is a professional member of IEEE. He published 10 journals and conference proceedings in reputed conferences. He completed 20 research funding projects funded by various agencies.