INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

# Development of Speaker-Independent Automatic Speech Recognition System for Kannada Language

**Praveen Kumar**[1]*, **H S Jayanna**[2]

**1** Research Scholar, Department of ECE, Siddaganga Institute of Technology, Tumakuru, 572103, Karnataka, India
**2** Department of ISE, Siddaganga Institute of Technology, Tumakuru, 572103, Karnataka, India

OPEN ACCESS

**Received:** 11.12.2021

**Accepted:** 22.01.2022

**Published:** 02.03.2022

**Citation:** Kumar P, Jayanna HS (2022) Development of Speaker-Independent Automatic Speech Recognition System for Kannada Language. Indian Journal of Science and Technology 15(8): 333-342. https://doi.org/10.17485/IJST/v15i8.2322

* **Corresponding author**.

pravin227@gmail.com

**Funding:** None

**Competing Interests:** None



Published By Indian Society for Education and Environment (iSee)

**ISSN**
Print: 0974-6846
Electronic: 0974-5645

## Abstract

**Objectives:** The primary goal is to address attempts to establish a Continuous Speech Recognition (CSR) framework for recognising continuous speech in Kannada. It is a difficult challenge to deal with a local language such as Kannada, which lacks the resources of a single language database. **Methods**: Modelling techniques such as monophone, triphone, deep neural network (DNN)-hidden Markov model (HMM) and Gaussian Mixture Model (GMM)-HMM-based models were implemented in Kaldi toolkit and used for continuous Kannada speech recognition (CKSR). To extract feature vectors from speech data, the Mel frequency Cepstral (MFCC) coefficient technique is used. The continuous Kannada speech database consists of 2800 speakers (1680 males and 1120 females) belong to the age group 8 years to 80 years. The training and testing data are in the ratio 80:20. In this paper the hybrid modelling techniques are implemented to recognize the spoken words. **Findings:** The model efficiency is determined based on the word error rate (WER) and the obtained results are assessed with the well-known datasets such as TIMIT and Aurora-4. This study found that using Kaldi-based features ex- traction recipes for monophone, triphone, DNN-HMM and GMM-HMM acoustic models had a word error rate (WER) of 8.23%, 5.23%, 4.05% and 4.64% respectively. The experimental results suggest that the rate of recognition of Kannada speech data has increased higher than that of state-of-the-art databases. **Novelty :** We propose a novel automatic speech recognition system for Kannada language. The main reason for developing the automatic speech recognition system for Kannada language is that there are only limited sources of standard continuous Kannada speech are available. We created large vocabulary Kannada database. We implemented monophone, triphone, Subspace Gaussian mixture model (SGMM) and hybrid modelling techniques to develop the automatic speech recognition system for Kannada language.

**Keywords:** DNN; Continuous speech; HMM; Kannada dialect; Kaldi toolkit; monophone; triphone; WER

## 1 Introduction

The effective research into Kannada SR is more essentially needed. This work sets up a CSR network for the Kannada language using phoneme modelling, where each phoneme is represented by a 5-state HMM and each state is represented by a GMM. Also, this work provides a study on monophone, triphone and hybrid simulation approaches for Kannada SR. The open-source Kaldi toolkit is used to train and check the SR framework. There are two implementations of the DNN in Kaldi. The first version is the version of Dan. It does not support the pre-training of the RBM. The second version is the implementation of Kerel. It allows Restricted Boltzmann Machinery (RBM) pre-testing, stochastic gradient simulation using GPUs and differential preparation. The Kaldi tool is built using C++ and is formed on the OpenFST library which utilizes the Linear Algebra BLAS and LAPACK libraries. For this work, we have opted to incorporate the above DNN because it facilitates concurrent training of multiple CPUs. This work is an attempt to build the continuous Kannada SR system. The development of such a system would help to convert the audiobooks available in Kannada into corresponding transcripts. It can also be very useful to digitize old palm- leaf manuscript documents simply by someone reading it. Such efforts will help to contribute the research for the development of the SR system for the Kannada language.

Numerous studies have been performed with the identification different languages like Kannada, Punjabi, Tamil, Hindi, Telugu, etc.[1–3]. The SR related study in Hindi using Kaldi is documented in[4–6]. The research on continuous Hindi SR is performed by a research unit[7]. Their data set consists of 1000 unique sentences and the WER obtained was better than many of the preceding works on Hindi. Good research work is also being done on the recognition of Serbian continuous speech[8]. They have 90 hours of speech details and 21,000 utterances. The findings obtained were satisfactory, with the GMM-HMM WER being 2.19% and the DNN being 1.86%. In[9], the authors presented their work on the building of an LVCSR system for Tamil dialect using DNN. They used 8 long stretches of Tamil speech collected from 30 speakers with a lexicon size of 13,984 words, of which 5 hours of learning was used for training. The findings reveal that the SR systems produce a phone error rate (PER) of 24.21% and a WER of 4.12% respectively.

The extensive literature survey concludes that work on CKSR is not remarkable. This made us conduct some tests by developing our database of 2800 speakers gathered throughout the state of Karnataka in the real-world conditions, we would like to check the behaviour of state-of-the-art techniques for continuous Kannada speech. That database is named as continuous Kannada speech database (CKSD). The transcription and validation were performed on all speakers' wave scripts. According to the speech data the phoneme level lexicon is built.

## 2 Kannada phoneme characteristics

The Kannada language has 49 phonemic characters, categorized into 3 types:

• Swaragalu / vowels: There are thirteen vowels in the Kannada language, they are, □□□□□□□□□□□□□. These thirteen vowels are further divided into two types based on the time spent for pronouncing. One of these forms is Hrasva Swara: A freely existing independent vowel that can be pronounced in a single matra period (matra Kala) often referred to as a mantra is □□□□□□□ and the other form is Deerga Swara: A freely existing independent vowel that can be pronounced in two matras is □□□□□□.

• Vyanjanagalu/ Consonants: there are thirty-four consonants, to take an individual form of the Consonant, they are dependent on vowels. These can be classified into two types of Vargeeya and Avargeeya

• Yogavaahakagalu: Anuswaras (□□) and Visarga (□□) are the two Yogavaahakagalu.

The description of the Kannada phonemes and the ITRANS (Indian Dialect transliterations) as seen in the Table 1. The ITRANS of the phonemes were seen within the brackets.

**Table 1: The Kannada character analysis and its related ITRANS**

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Vowels** | ಅ (a) | ಆ (aa) | ಇ (i) | ಈ (I) | ಉ (u) | ಊ (U) | ಋ (Ru) | ೠ (RU) | ಎ (e) | ಏ (E) | |
| | ಐ (ai) | | ಒ (o) | | | ಓ (O) | | | ಔ (ou) | | |
| **Yogavahakas** | ಅಂ (aM) | | | | | ಅಃ (aH) | | | | | |
| **Structured consonants** | ಕ (ka) | ಖ (kha) | | ಗ (ga) | | ಘ (gha) | | ಙ (nga) | | | |
| | ಚ (cha) | ಛ (Cha) | | ಜ (ja) | | ಝ (jha) | | ಞ (nja) | | | |
| | ಟ (Ta) | ಠ (Tha) | | ಡ (Da) | | ಢ (Dha) | | ಣ (Na) | | | |
| | ತ (ta) | ಥ (tha) | | ದ (da) | | ಧ (dha) | | ನ (na) | | | |
| | ಪ (pa) | ಫ (pha) | | ಬ (ba) | | ಭ (bha) | | ಮ (ma) | | | |
| **Unstructured consonants** | ಯ (ya) | ರ (ra) | ಲ (la) | ವ (va) | ಶ (sha) | ಷ (Sha) | ಸ (sa) | ಹ (ha) | ಳ (La) | | |

In Kannada, the basic language rule is when a dependent consonant combines with an independent vowel; as seen below, an Akshara is created: Consonant (Vyanjana) + Vowel (Matra) = Letter (Akshara). All the consonants (Vyanjanas) are combined with the Vowels (matra) based on this law to form Kagunitha for the Kannada alphabet. The sounds in all languages come under two classified categories: Vowels and consonants. Consonants are created with some sort of limitation or closure in the vocal tract, which blocks the flow of air from the lungs. They are categorised ac- cording to where the airflow has been narrowed in the vocal tract. This is also known as the place of articulation. In spoken language, Vowel is a sound pronounced with an open vocal tract such that there is no buildup of air pressure above the glottis at any time.

## 3 Data collection and preparation

There may be some factors that will certainly, alter the implementation of the SR system. The causes are close to session variability, intra-speaker and inter-speaker inconstancy. Bharat Sanchar Nigam Limited (BSNL) offered an Integrated Voice Response System (IVRS) call flow telephone service. The continuous voice data of 2800 speakers belong to the age group 8 years to 80 years were obtained (1680 males and 1120 females). A collection of ten phonetically rich and important Kannada sentences was pronounced by any speaker. Spoken data was obtained in the real world from different areas of the state of Karnataka. There are 30,846 words, from 30 districts, as there is a variety in Kannada-speaking languages from region to region in the state of Karnataka. In the entire process, the data collection ratio 60:40 (60% for male speakers and 40% for female speakers) is maintained. The method used for transcription is the Indic Transliteration (IT3 to UTF-8). The continuous Kannada speech information collected from the speakers shall be transcribed from the word level to the phoneme level. Tags used during the transcription of speaker data for non-lexical sounds also known as silence phones. Table 2 Gaadegalu indicates the few continuous phrases in Kannada dialects. These sentences are known as Kannada gaadegalu/nannudigalu. The collected speech data is manually transcribed and authenticated by supervisors at the word level.

**Table: 2 List of Kannada gaadegalu/naannudigalu recorded from the people across the state of Karnataka**

| English Version of Gaadagalu | Kannada Version of Gaadegalu |
|---|---|
| ati aase gatigeid:u | ಅತಿ ಆಸೆ ಗತಿಗೇಡು |
| haal:uurige ul:idavane gaud:a | ಹಾಳೂರಿಗೆ ಉಳಿದವನೆ ಗೌಡ |
| ban:daddellaa barali goovin:dana dayeyirali | ಬಂದದ್ದೆಲ್ಲಾ ಬರಲಿ ಗೋವಿಂದನ ದಯೆಯಿರಲಿ |
| hani hani seiridare hal:l:a tene tene seiridare bal:l:a | ಹನಿ ಹನಿ ಸೇರಿದರೆ ಹಳ್ಳ ತನೆ ತನೆ ಸೇರಿದರೆ ಬಳ್ಳ |
| handndele uduruvaaga chigurele nagutittu | ಹಣ್ಣೆಲೆ ಉದುರುವಾಗ ಚಿಗುರೆಲೆ ನಗುತಿತ್ತು |
| beiline eddu hola meiyitan:te | ಬೇಲಿನೆ ಎದ್ದು ಹೂಲ ಮೇಯಿತಂತೆ |
| haagalakaayige beivinakaayi saakshhi | ಹಾಗಲಕಾಯಿಗೆ ಬೇವಿನಕಾಯಿ ಸಾಕ್ಷಿ |
| haavuu saayalilla koolu muriililla | ಹಾವೂ ಸಾಯಲಿಲ್ಲ ಕೋಲು ಮುರೀಲಿಲ್ಲ |
| hiriyakkana chaal:i mane man:digella | ಹಿರಿಯಕ್ಕನ ಚಾಳಿ ಮನೆ ಮಂದಿಗೆಲ್ಲ |
| handa an:dre hendavuu baayi bid:uttade | ಹಣ ಅಂದ್ರೆ ಹೆಣವೂ ಬಾಯಿ ಬಿಡುತ್ತದೆ |
| huuvini:n:da naaru swarga seiritu | ಹೂವಿನಿಂದ ನಾರು ಸ್ವರ್ಗ ಸೇರಿತು |

## 4 Model Architecture

The CSR protocol for Kannada language involves various units, as seen in Figure 1. The detailed explanation of the proposed model architecture is explained as below:

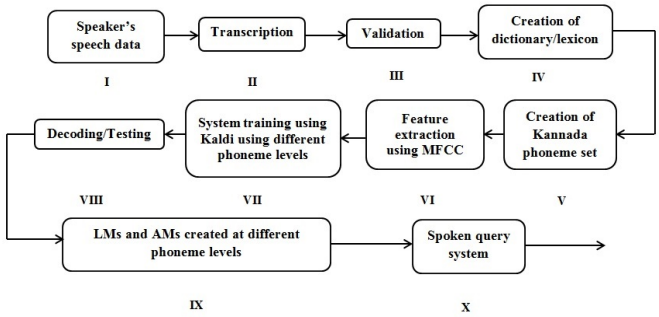


**Fig 1. Schematic view of the CSR framework for Kannada dialect**

### 4.1 Feature Extraction

MFCC stands for cepstral coefficients of Mel frequency. The abbreviation comprises 4 words: Mel, frequency, cepstral and coefficients. The MFCC concept is to translate the time-domain speech signal into a frequency domain to capture all the possible information from voice signals. The cochlea in our ear has essentially more low-frequency filters and very few higher frequencies filters. The Mel filters are used to imitate this. The concept of MFCC is, thus, to translate time-domain signals into a frequency domain signal by mimicking cochlea function using Mel filters. The coefficients of cepstral are the inverse- FFT of the logarithm of the spectrum. The MFCC features were initially proposed for the recognition of monosyllabic terms. It is basically as represents the filter (vocal tract) in the source filter speech model. The MFCC's first 13 (lower dimensions) is a bandwidth envelope. The higher dimensions that are discarded convey the spectral data. Envelopes are adequate to rep- resent the difference for many phonemes so that phonemes can be recognised through MFCC. The standard frequency scale $f_{Hz}$ was warped by logarithmic compression into one perceived pitch at a linear scale $f_{mel}$:

$$f_{mel} = 1127log\left(1 + \frac{f_{Hz}}{700}\right]$$

### 4.2 Language Model (LM)

To restrict word search, an LM is used. It determines which word should follow previously known terms and helps to limit the matching process substantially by removing words that are not feasible. The LM is a file used to recognise speech by the SR model. It contains a wide list of words and their likelihood. LMs are used to limit the number of possible words which should be considered at any point of the search in a decoder. The result is quicker performance and greater precision. The N-gram LMs are the most common LMs, with word sequence statistics and finite language patterns. To obtain a high degree of accuracy, the search space constraint must be quite successful. This means that predicting the next term must be successful. Languages constrain the search either completely or probabilistically (by listing a small subset of possible extensions) (by calculating a probability for every possible word in succession). The former typically has a grammar connected to it that is assembled into a graph. An LM typically limits the vocabulary to the words in it. This is a challenging recognition task. To cope with this, smaller chunks like sub words or even phones might be present in a language model. It is noted that in this case, the search space cap is normally worse than for a word-based language model, the accuracy of the resulting identification. The statistical LMs (SLM), were not feasible or possible for a priori defining all possible legal word sequences, are ideal for free form inputs, such as dictation or spontaneous expression. Possibly the most common trigram SLMs in ASR and a strong mix between difficulty and rigorous approximation. A model tri-gram encodes the likelihood of a phrase regardless of its immediate two-word past. In reality, tri-gram models should be "backed-offs" to bi-gram and unigram modelling so that the decoder can send out every potential word series.

### 4.3 Acoustic Model (AM)

To train ASR, training an AM and a Language Model (LM) is essential. The fundamental AM preparation includes:

- Monophone HMM instruction for a training sub-set.
- Aligned data set with the monophone model
- HMM-training using triphone.

The AM is trained through audio files and transcripts. Here are two major forms of models, one-phone (monophone) and three-phone (triphone) models. The AM is used in SR to explain the relationship between phonemes and audio signals. The monophone model tries to match the sound that a single phone has heard, whereas the acoustic triphone model relies more extensively on the background. For both model generation, the features required to be extracted from the speech signal. An AM is a file containing statistical representations of each sound which constitutes a word. A label called phoneme is given for each of these statistical representations. There are roughly 40 distinct sounds in English which are helpful to understand the voice, and we have thus 40 separate phonemes. Recognition or classification means that the maximum likelihood (ML) criterion can be used to assign any new sequence of observations to the most similar model. HMM suffer from such limits, however. Continuous HMMs with Baum-Welch or Viterbi algorithms have low descriptive potential among the models as they are based on the ML parameters. The working of AM in recognition of any word is listed below:

- The voice decoder listens to the sounds articulated by a person and scans for an HMM in the AM.
- In the AM, the decoder states the phoneme as it detects an analogous HMM.
- The decoder then records the corresponding phonemes before the user talks for a break.

• Later the decoder looks at the right collection of phonemes that is heard in its dictionary of pronunciation to decide the word spoken when the delay is hit.

• Then the decoder searches for a related word or sentence in the grammar file.

### 4.4 Monophone model generation

The HMM is used here to model phonemes. Each phoneme is represented by five HMM states, each of which is represented by one state GMM. The HMM uses a series of continuous density states to model the set of feature vectors. MFCC is used for monophone model generation. The audio signal was sampled at 8 kHz. So, we get 8000*0.020=160 observations in one window and the same is reduced to 13 static cepstral coefficients. The feature is extracted by applying a window of 20 ms shifted by 10 ms.

### 4.5 Creating triphone models

It is necessary to use some MFCC transformations to optimize recognition, in addition to the static features derived from each speech data frame. These transformations are used to create three-phone templates. Transformations shall include Delta Function Computation, Linear Discrimination Analysis (LDA) and MLLT.

• The LDA transform: is a linear transformation that helps to reduce the dimension of the data of input functions. The purpose of the LDA is to consider a linear translation of the characteristics of vectors from n- dimensional space to vectors in m-dimensional space (m<n). This makes the system faster.

• Delta function computation: the features of the MFCC have been taken into account only for phonetic frames, without taking into account the interaction between them. The phonetic signals are continuous because the signals are continuous. Acquiring a dynamic shift function across phonetic frames will enhance recognition efficiency. Delta is the Fourier representation of the temporal series of phonetic artefacts. For instance, if we have 13 MFCC coefficients, with the delta+delta-delta transform, we also have 13+ 13 delta coefficients, which merge to give a vector of 39 length features (13 + 13 + 13).

• The MLLT estimation: MLLT calculates the parameters of linear trans- formation to optimize the probability of training results given the GMMs of diagonal covariance; The rounded features depicted by the model are better compared to the original features.

## 5 GMM-HMM Modelling

The most effective and simplest classification model, Hidden Markov model, has many different applications. Speech data were taken and features were extracted from it in the context of SR. The process of GMM-HMM is shown in Figure 2.

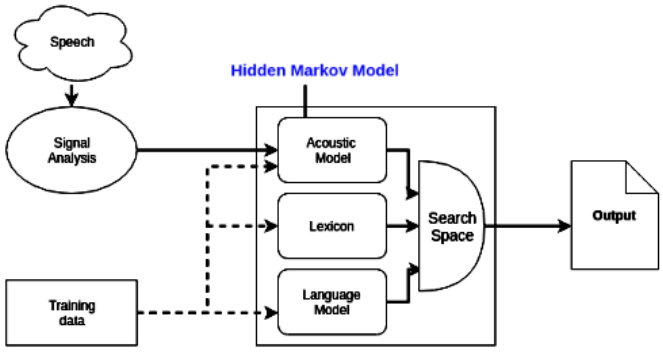


**Fig 2. Working of HMM**

To recognize speech, a classifier which will identify which (if any) phone was uttered in every frame is needed. A simple GMM can be used for that, i.e., for each phone class, fit a GMM using all the frames where that phone is found - to classify a new utterance, frame by frame checking is done to find out which phone is the most probable (i.e., which GMM gives the highest likelihood of generating the 39-feature vector). However, this kind of model doesn't exploit the temporal dependencies in the acoustic signal. The classification depends only upon the current frame, ignoring the context (i.e., previous and next frames). Moreover, this model assumes that there is no acoustic difference, in the beginning, middle and end of a phone.

The GMM- HMM model is a response to these problems. The HMM is a temporal model, which assumes that the source (observation generator) has some state which we don't know about (e.g., position of the larynx, the shape of the oral cavity, tongue placement). Most common HMM architectures for SR have phone models consisting of three states. It can be interpreted that as an assumption that a phone, when uttered, has three distinct phases - a beginning, a middle, and an ending. In each phase, it sounds a bit different. Each state is modelled by a GMM to determine the likelihood of the observation in that state, and our observations are the frames. Moreover, this model assumes that there is no acoustic difference in the beginning, middle and ending of a phone (just think of how the voice goes up and then down when 'ah' is uttered). The GMM-HMM model is a response to these problems. The HMM is a temporal model, which assumes that the source (observation generator) has some state which is not known (e.g., position of the larynx, the shape of the oral cavity, tongue placement). Most common HMM architectures for speech recognition have phone models consisting of three states. Assume that a phone, when it is uttered, has three distinct phases - a beginning, a 'middle', and an ending. In each phase, it 'sounds' a bit different. Each state is modelled by a GMM to determine the likelihood of the observation in that state, and the observations are the frames. So, wrapping it up - for the sequence of frames, each of which is being classified as a particular state, belonging to the particular phone. There may be many frames generated by one state, and there may be several states which sum up to a single phone (and going further, there may be several phones which build up a single word).

## 5.1 DNN-HMM Modelling

The Figure 3 demonstrates the general structure of the hybrid DNN-HMM frame- work. With given acoustic measurements, the DNN is trained to predict posterior probabilities of each context-dependent state. During decoding the prob- abilities of output are divided by the prior probability of each state forming "pseudo-likelihood" that is used in place of the probabilities of state emissions in the HMM [10]. The initial phase in training DNN-HMM model is to train GMM-HMM model through the data allocated for training. The standard Kaldi recipe for DNN-based acoustic model comprises of the following steps:

- Feature extraction (the features are 13 MFCCs +13 Delta+13 Delta-Delta);
- Training a triphone model with delta, delta-delta characteristics, maximum Likelihood linear transform and LDA;
- A monophonic model training;
- fMLLR functionality suited to linear regression;
- Speaker adapted training (SAT), i.e., training of the highest possibility of function space.
- DNN-HMM final model training.

DNN-HMM is training with fMLLr-adapted features; the SAT fMLLR GMM system includes the decision tree and alignments. Furthermore, the statistically inadequate modelling of GMMs in HMM is significantly insufficient where they are placed in or around a non-linear multiple in the data space. DNN-HMM is a modern paradigm of the hybrid model that has been proposed and commonly used in the recognition of speech in recent years. DNN models are better classifiers than GMMs, and with fewer parameters over complex distributions, they can generalise even better. DNN is the standard multi-layer perceptron with several layers that capture the underlying non-linear relationship between data where training is usually initialised with a pre-training algorithm. They model distributions of different classes jointly, this is called "distributed" learning, or, more properly "tied" learning. In GMM, each senone separately with a separate set of GMMs is modelled whereas in DNN the features are classified together and distribution of senone posteriors is calculated. The alignment for training is calculated for the whole utterance but the con- text for the classifier is different. DNNs can model much longer context. In GMM system it is typical to model simply 7–9 frames in a raw, GMM models does not improve if the context due to convexity of the distribution they model is increased.

ASR systems focused on the GMM-HMM framework typically require the full training of individual GMMs in each HMM condition. A modern modelling methodology is used in the SR domain called SGMM [7]. Consequently, no parameters are distributed between states. The states are described by Gaussian mixtures and these parameters transmit the normal structure between the states of the simulation technique of the SGMM. Dedicated multivariate Gaussian mixtures are used for state-level modelling of conventional GMM- HMM acoustic modelling techniques.

## 5.2 Training and Testing

The speech files are required for training with high phonetic parity and inclusion. The training data is used to create the LM and the AM along with acoustic information. In this process, the algorithm searches effectively for the best sequence in space consisting of an observation sequence, an LM and an AM. The search method is often referred to as decoding. The classical Viterbi algorithm can successfully solve HMM decoding problems. The test stage is to view in the words present in the lexicon, through the expressed word arrangement. For training and testing the Kannada ASR system [11] we used CKSD collected in real-

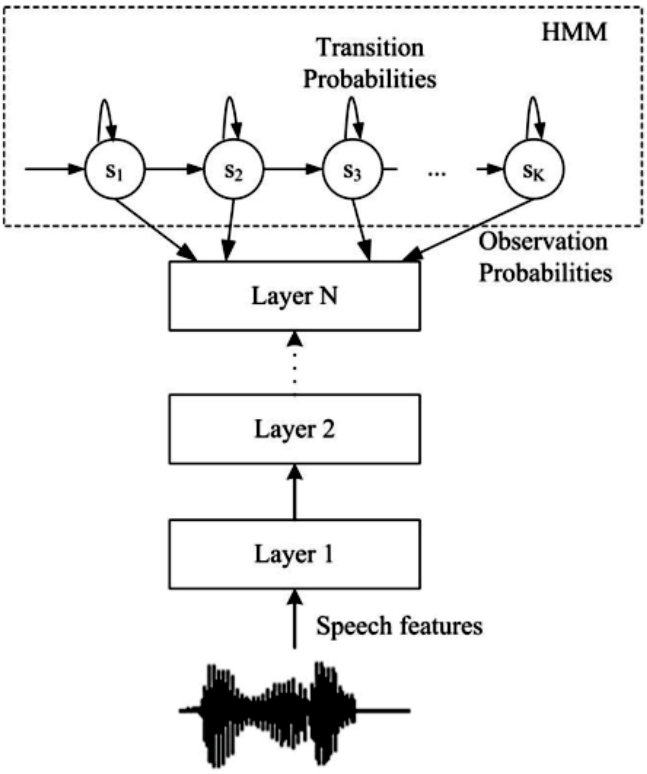


**Fig 3. DNN-HMM Hybrid Device Model** [43]

time through Bharath Sanchar Nigam Limited (BSNL) telephonic connection. At present 32 hours of information is available among which 80% of data is used for training and 20% of data is used for testing.

## 6 Experiment and Result Analysis

The results of the exploration are from a corpus of 10 Kannada sentences, which are spoken by 2800 speakers and communicated verbally. Our trials were all carried out on the Ubuntu 18.04 LTS, Intel Core i7, 3.70GHz clock- speed (64-bit work system) platform. MFCC features and their affiliates are used for the creation of models. Kaldi uses an FST architecture, which has been developed with an IRSTLM toolbox.

The table 3 displays the various WERs that have been gained at specific phoneme levels. The table indicates that the monophone has a WER of 8.23% and the WER for the triphone1, triphone2, triphone3 and SGMM are 6.22%, 5.38%, 5.12%, 4.84% respectively. The distinctive WER of the standard TIMIT database is also seen in the table. From the table, it is observed that the SGMM and the triphone modelling method offered a more distinct precision than the monophone modelling method. The CKSD has a higher degree of identification than the TIMIT database.

**Table: 3 The representation of WER at the different phoneme levels for the continuous Kannada speech database**

| Phonemes | WER_1 | | WER_2 | | WER_3 | | WER_4 | | WER_5 | |
|---|---|---|---|---|---|---|---|---|---|---|
| | CKSD | TIMIT | CKSD | TIMIT | CKSD | TIMIT | CKSD | TIMIT | CKSD | TIMIT |
| MONO | 8.23 | 8.96 | 8.54 | 9.03 | 8.36 | 8.81 | 8.64 | 8.79 | 8.66 | 8.98 |
| tri1_600_2400 | 7.52 | 7.86 | 7.48 | 7.86 | 7.26 | 7.59 | 7.81 | 8.06 | 7.65 | 8.15 |
| tri1_600_4800 | 6.57 | 6.86 | 6.61 | 6.82 | 6.75 | 7.03 | 6.81 | 7.14 | 6.69 | 7.28 |
| tri1_600_9600 | 6.24 | 6.54 | 6.37 | 6.85 | 6.48 | 6.94 | 6.22 | 6.76 | 6.35 | 6.86 |
| tri2_600_2400 | 7.45 | 7.58 | 7.24 | 7.52 | 7.14 | 7.49 | 7.35 | 7.84 | 7.27 | 7.68 |
| tri2_600_4800 | 6.52 | 6.98 | 6.27 | 6.94 | 6.29 | 6.84 | 6.35 | 6.85 | 6.33 | 6.73 |
| tri2_600_9600 | 5.59 | 6.12 | 5.54 | 6.04 | 5.38 | 5.96 | 5.84 | 6.16 | 6.01 | 6.53 |
| tri3_600_2400 | 5.79 | 6.25 | 5.61 | 6.09 | 5.54 | 6.02 | 5.58 | 6.12 | 5.88 | 6.24 |
| tri3_600_4800 | 5.45 | 5.95 | 5.62 | 6.01 | 5.38 | 5.93 | 5.48 | 5.97 | 5.41 | 5.86 |
| tri3_600_9600 | 5.12 | 5.62 | 5.34 | 5.96 | 5.27 | 5.81 | 5.23 | 5.88 | 5.32 | 5.59 |
| SGMM | 4.86 | 4.97 | 5.12 | 5.59 | 4.84 | 5.81 | 4.89 | 5.29 | 4.92 | 5.31 |

Table 4 displays the various WERs at various phoneme speeds. Itwas observed in the table that the mixture of (DNN+HMM) has a WER of 4.05% and a WER relative to (DNN+SGMM) of 4.65%. Finally, the combining of the MMI+SGMM process gives the WER 5.23%. The table reveals that the mixture of (DNN+HMM) has provided predominant precision relative to other modelling techniques.

**Table: 4 The depiction of WER for hybrid modelling techniques at different phoneme levels for continuous Kannada speech database**

| Phonemes | WER_1 | | WER_2 | | WER_3 | | WER_4 | | WER_5 | |
|---|---|---|---|---|---|---|---|---|---|---|
| | CKSD | TIMIT | CKSD | TIMIT | CKSD | TIMIT | CKSD | TIMIT | CKSD | TIMIT |
| SGMM+MMI_it1 | 5.23 | 6.45 | 5.06 | 5.89 | 4.98 | 6.02 | 5.21 | 6.66 | 4.95 | 8.98 |
| SGMM+MMI_it2 | 5.38 | 6.59 | 5.64 | 6.97 | 6.29 | 7.21 | 5.83 | 6.82 | 6.02 | 7.13 |
| SGMM+MMI_it3 | 5.88 | 6.94 | 5.64 | 7.23 | 6.02 | 7.89 | 5.98 | 6.68 | 6.23 | 7.63 |
| SGMM+MMI_it4 | 6.02 | 7.82 | 5.88 | 6.97 | 6.23 | 7.85 | 5.81 | 6.85 | 6.01 | 7.69 |
| DNN+HMM | 4.56 | 6.02 | 4.67 | 5.92 | 5.01 | 6.23 | 4.05 | 5.87 | 5.21 | 6.90 |
| DNN+SGMM_it1 | 4.87 | 6.23 | 4.65 | 5.02 | 4.94 | 5.45 | 5.10 | 5.67 | 4.86 | 5.54 |
| DNN+SGMM_it2 | 4.59 | 5.24 | 4.62 | 5.14 | 4.85 | 5.41 | 5.03 | 5.67 | 5.24 | 5.89 |
| DNN+SGMM_it3 | 5.31 | 6.12 | 4.92 | 5.61 | 5.09 | 5.84 | 4.86 | 5.29 | 4.64 | 5.77 |
| DNN+SGMM_it4 | 4.99 | 5.64 | 5.06 | 6.11 | 4.85 | 5.90 | 5.22 | 5.93 | 5.59 | 6.28 |

Similarly, the WER at different phoneme levels for CKSD and Aurora-4 database is depicted in table 5 and WER for hybrid modelling techniques is depicted in table 6. From both the tables, it is noticed that the WER for CKSD is lesser than that of Aurora-4 database.

**Table: 5 The representation of WER at the different phoneme levels for the continuous Kannada speech and Aurora-4 database**

| Phonemes | WER_1 | | WER_2 | | WER_3 | | WER_4 | | WER_5 | |
|---|---|---|---|---|---|---|---|---|---|---|
| | CKSD | Aurora-4 | CKSD | Aurora-4 | CKSD | Aurora-4 | CKSD | Aurora-4 | CKSD | Aurora-4 |
| MONO | 8.23 | 9.25 | 8.54 | 9.47 | 8.36 | 9.59 | 8.64 | 8.86 | 8.66 | 8.59 |
| tri1_600_2400 | 7.52 | 7.86 | 7.48 | 8.45 | 7.26 | 8.21 | 7.81 | 8.58 | 7.65 | 8.68 |
| tri1_600_4800 | 6.57 | 6.97 | 6.61 | 7.24 | 6.75 | 7.47 | 6.81 | 7.67 | 6.69 | 7.69 |
| tri1_600_9600 | 6.24 | 6.54 | 6.37 | 7.35 | 6.48 | 7.81 | 6.22 | 7.29 | 6.35 | 7.03 |
| tri2_600_2400 | 7.45 | 7.58 | 7.24 | 8.65 | 7.14 | 8.56 | 7.35 | 8.46 | 7.27 | 8.45 |
| tri2_600_4800 | 6.52 | 6.98 | 6.27 | 7.25 | 6.29 | 7.64 | 6.35 | 7.61 | 6.33 | 7.97 |
| tri2_600_9600 | 5.59 | 6.55 | 5.54 | 6.59 | 5.38 | 6.87 | 5.84 | 6.73 | 6.01 | 6.85 |
| tri3_600_2400 | 5.79 | 6.25 | 5.61 | 6.58 | 5.54 | 6.69 | 5.58 | 6.67 | 5.88 | 6.95 |
| tri3_600_4800 | 5.45 | 6.55 | 5.62 | 6.68 | 5.38 | 6.62 | 5.48 | 6.77 | 5.41 | 6.86 |
| tri3_600_9600 | 5.12 | 6.62 | 5.34 | 6.84 | 5.27 | 6.85 | 5.23 | 6.23 | 5.32 | 6.97 |
| SGMM | 4.86 | 5.26 | 5.12 | 5.65 | 4.84 | 5.86 | 4.89 | 5.15 | 4.92 | 5.58 |

**Table: 6 The WER representation for hybrid modelling techniques for Continuous Kannada speech database and Aurora-4 database**

| Phonemes | WER_1 | | WER_2 | | WER_3 | | WER_4 | | WER_5 | |
|---|---|---|---|---|---|---|---|---|---|---|
| | CKSD | Aurora-4 | CKSD | Aurora-4 | CKSD | Aurora-4 | CKSD | Aurora-4 | CKSD | Aurora-4 |
| SGMM+MMI_it1 | 5.23 | 7.02 | 5.06 | 6.89 | 4.98 | 6.91 | 5.21 | 7.23 | 4.95 | 7.21 |
| SGMM+MMI_it2 | 5.38 | 6.65 | 5.64 | 7.35 | 6.29 | 6.86 | 5.83 | 6.54 | 6.02 | 6.58 |
| SGMM+MMI_it3 | 5.88 | 7.28 | 5.64 | 7.86 | 6.02 | 7.61 | 5.98 | 7.54 | 6.23 | 6.98 |
| SGMM+MMI_it4 | 6.02 | 8.01 | 5.88 | 7.61 | 6.23 | 8.28 | 5.81 | 7.81 | 6.01 | 8.03 |
| DNN+HMM | 4.56 | 5.68 | 4.67 | 5.27 | 5.01 | 5.81 | 4.05 | 6.21 | 5.21 | 5.97 |
| DNN+SGMM_it1 | 4.87 | 5.94 | 4.65 | 5.68 | 4.94 | 5.94 | 5.10 | 6.01 | 4.86 | 6.24 |
| DNN+SGMM_it2 | 4.59 | 5.54 | 4.62 | 5.64 | 4.85 | 5.29 | 5.03 | 6.31 | 5.24 | 6.10 |
| DNN+SGMM_it3 | 5.31 | 6.54 | 4.92 | 5.58 | 5.09 | 5.64 | 4.86 | 5.68 | 4.64 | 5.93 |
| DNN+SGMM_it4 | 4.99 | 6.14 | 5.06 | 5.59 | 4.85 | 6.21 | 5.22 | 5.92 | 5.59 | 6.34 |

The comparison between the CKSD and the TIMIT database and the Aurora-4 repository w.r.t recognition rate is seen in Figure 4 and Figure 5 respectively. The plots reveal that the efficiency of the triphone modelling technique is greater than that of the monophone modelling technique. The efficiency of CKSD is also higher than that of the TIMIT and Aurora-4 databases.

In Figure 6 and Figure 7, a comparison of the CKSD with the TIMIT database and the Aurora-4 database for hybrid modelling techniques is shown. These plots show that the combination of DNN and HMM modelling technique is performing better than the other hybrid modelling techniques for CKSD.

Also from the above plots, it is observed that the performance of CKSD has a better edge compared to that of TIMIT and Aurora-4 database.

The results obtained from the experiments conducted as mentioned in this article were better compared to the results obtained in [1]. This is because we have implemented the hybrid modelling techniques compared to that of monophone and triphone modelling technique.

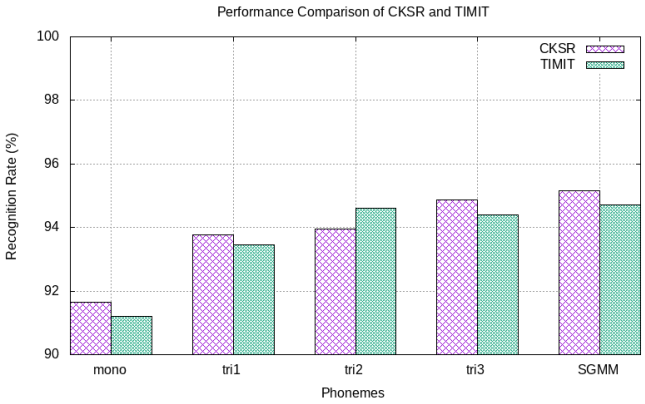


**Fig 4. The performance comparison of CKSD and TIMIT database**

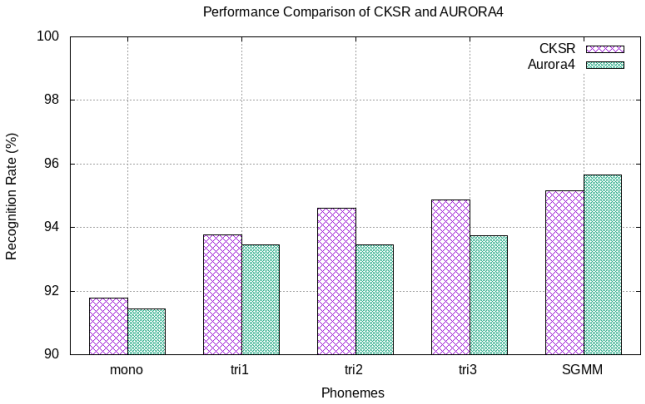


**Fig 5. The performance comparison of CKSD and Aurora4 database**

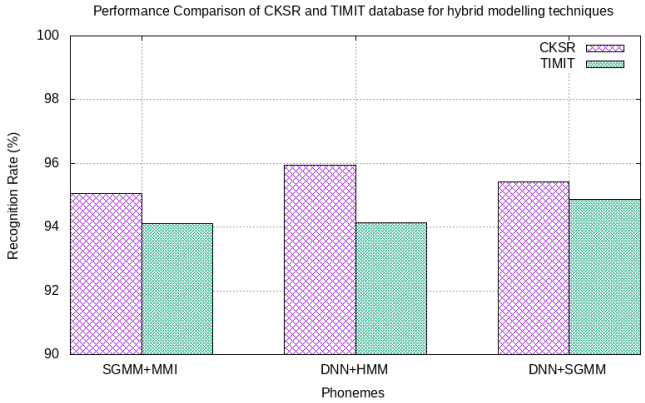


**Fig 6. The performance comparison of CKSD and TIMIT database for hybrid modelling techniques**

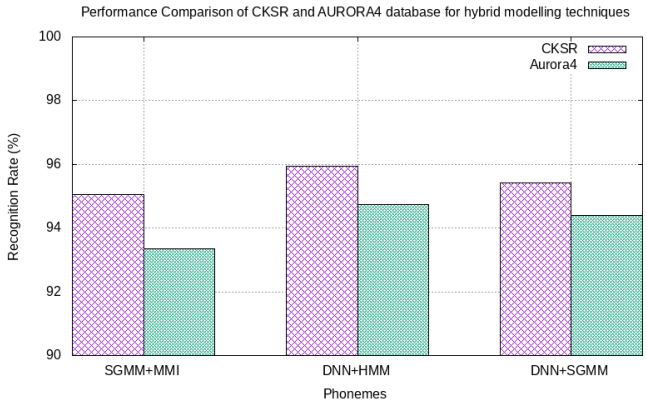


**Fig 7. The performance comparison of CKSD and Aurora4 database for hybrid modelling techniques**

## 7 Conclusions

The continuous Kannada speech recognition system has been demonstrated in this study. The speech data have been collected, transcribed and checked with the transcription system. Using an SR toolkit, the ASR models were developed. In the lexicon were included all alternate pronouncements for the Kannada speaking sentence. The WERs performed for the monophonic, triphone1, triphone2, triphone3, combination of SGMM and MMI, combination of DNN and HMM, combination of DNN and Subspace Gaussian mixture model are 8.36%, 6.22%, 5.38%, 5.12%, 4.84%, 4.98%, 4.05%, 4.59%, respectively. The recognition rate of CKSR system is higher than that the WER for TIMIT and Aurora-4 database. The SGMM and hybrid DNN-based modelling techniques have achieved the least WER for continuous Kannada speech data. These least WER models (SGMM and DNN-based models) could be used to build a stable ASR framework. The developed ASR system has been tested under deteriorated conditions by different speakers. Also, the precision of the ASR system can be expanded to allow us to implement noise reduction methods more effectively. The expected challenge is to further improve the system efficiency by increasing the number of speakers and also by increasing the number of phonemes appropriately.

## References

1) Kumar P, Jayanna PS, S H. Creation and Instigation of Triphone based Big-Lexicon Speaker-Independent Continuous Speech Recognition Framework for Kannada Language. *International Journal of Innovative Technology and Exploring Engineering*. 2019;9(2S):152–158. doi:10.35940/ijitee.b1090.1292s19.
2) Guglani J, Mishra AN. Continuous Punjabi speech recognition model based on Kaldi ASR toolkit. *International Journal of Speech Technology*. 2018;21(2):211–216. Available from: https://dx.doi.org/10.1007/s10772-018-9497-6.
3) Kalamani M, Krishnamoorthi M, Valarmathi RS. Continuous Tamil Speech Recognition technique under non stationary noisy environments. *International Journal of Speech Technology*. 2019;22(1):47–58. Available from: https://dx.doi.org/10.1007/s10772-018-09580-8.
4) Upadhyaya P, Farooq O, Abidi MR, Varshney YV. Continuous hindi speech recognition model based on Kaldi ASR toolkit. *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. 2017;p. 786–789. doi:10.1109/WiSPNET.2017.8299868.
5) Sharma RS, Paladugu SH, Priya KJ, Gupta D. Speech Recognition in Kannada using HTK and Julius: A Comparative Study. *2019 International Conference on Communication and Signal Processing (ICCSP)*. 2019;p. 68–0072. doi:10.1109/ICCSP.2019.8698039.
6) Amin MAA, Islam MT, Kibria ST, Rahman MS. Continuous Bengali Speech Recognition Based On Deep Neural Network. *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*. 2019;p. 1–6. doi:10.1109/ECACE.2019.8679341.
7) Upadhyaya P, Mittal SK, Farooq O, Varshney YV, Abidi MR. Continuous Hindi speech recognition using Kaldi ASR based on deep neural network". *Machine Intelligence and Signal Analysis*. 2019;p. 303–311.
8) Kipyatkova I, Karpov A. DNN-Based Acoustic Modeling for Russian Speech Recognition Using Kaldi. *Speech and Computer*. 2016;p. 246–253.
9) Madhavaraj A, Ramakrishnan AG. Design and development of a large vocabulary, continuous speech recognition system for Tamil. *2017 14th IEEE India Council International Conference (INDICON)*. 2017;p. 1–5.
10) Speech and Language Processing. In: Jurafsky D, Martin JH, editors. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. 2020. Available from: https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf.
11) Kumar PSP, Yadava GT, Jayanna HS. Continuous Kannada Speech Recognition System Under Degraded Condition. *Circuits, Systems, and Signal Processing*. 2020;39:391–419. Available from: https://dx.doi.org/10.1007/s00034-019-01189-9.