International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

# Deep Learning Approaches for Speech Command Recognition in a Low Resource KUI Language

**Subrat Kumar Nayak [1], Ajit Kumar Nayak [2], Smitaprava Mishra [3], Prithviraj Mohanty [4]**

**Submitted:** 14/11/2022 **Accepted:** 15/02/2023

**Abstract:** Over the time, computers can learn to understand speech from experience, thanks to incredible recent advances in deep learning algorithms. Speech command recognition becomes necessary when it comes to assisting disabled and impaired people, and executing hands-free activities in the sector of customer service and education. Speech recognition combines multiple disciplines from computer science to identify speech patterns. Identifying speech patterns help computers differentiate between various instructions for which it has been trained to perform. This research aims to implement speech command recognition technology into gaming, assisting players to play games in their native language. Speech recognition technology can be used as a way to engage with the various situations presented in video games, enabling a greater degree of immersion than what is possible through AR (Augmented Reality) and VR (Virtual Reality) technologies on their own. This research introduces various deep learning algorithms and their comparative analysis that can be applied to process speech commands, particularly in the KUI language. An in-depth analysis of the feature extraction techniques like Mel-frequency cepstral coefficient (MFCC) and deep learning algorithms such as Artificial Neural Networks (ANN), Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN) and Attention using LSTM have been presented. Various experiments are conducted to compare the performance metrics obtained from all the models applied.

***Keywords:*** *Speech Recognition, MFCC, KUI language, Attention using LSTM, Deep Neural Network*

## 1. Introduction

The Deep Neural Networks have given better performances for ASR works in many application areas [1]. They are mostly used for classification problems. For obtaining better performance in speech processing, it requires huge dataset. Majority of the speech recognition systems are implemented using English language. In this research work, we have used 7 Kui speech commands for playing video games without physically touching. We used MFCC for feature extraction as a pre-processing step. ANN, RNN, CNN and Attention model using LSTM are trained on audio data to identify the speech commands. Several experiments have been done in different languages [3]-[5]. But very few experiments have been done for low-resource language datasets. Language is the most important medium to bring tribals from isolation into the mainstream.

[1] *Research Scholar, Department of Computer Science and Engineering, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, India*
*ORCID ID: 0000-0002-7438-9085*
[2] *Professor, Department of Computer Science and and Information Technology, Siksha 'O' Anusandhan Deemed to be University, India*
[3] *Professor, Department of Computer Science and and Information Technology, Siksha 'O' Anusandhan Deemed to be University, India*
[4] *Associate Professor, Department of Computer Science and and Information Technology, Siksha 'O' Anusandhan Deemed to be University, India*
** Corresponding Author Email: subratsilicon28@gmail.com*

Kui is spoken across eastern India, with speakers mostly concentrated in Odisha but also found in Andhra Pradesh, Bihar, Chhattisgarh, Madhya Pradesh, Maharashtra,

Jharkhand, and West Bengal. Kui is written in the Odia script, given that Kui speakers are concentrated in the state of Odisha. The majority of Kui-speaking Kondhs reside in South and Central Odisha's hilly and forested regions, particularly in the undivided districts of Kandhamal.

During covid-19 pandemic, several speech recognition applications were developed to prevent the disease by avoiding touch. Deep learning technology requires high computing capabilities. Therefore in speech processing Tensor Processing Units (TPU) and Graphics Processing Units (GPU) are required. In the next few years' speech command recognition will be an important interface which will widely be used in many applications such as smartphones, GPSs, Washing machines, mobile gaming consoles etc.

The research details included in this paper is structured into the following sections. Section 2 deliberates the related work considering the various approaches applied in the field of speech command recognition. Section 3 represents the detail overview of models used for speech command recognition. Section 4 represents the Kui speech command dataset preparation, feature extraction, experimental setup, result analysis and discussion. The conclusion and future

scope are conveyed in Sect. 5.

## 2. Related Works

Speech command recognition is done either on single words or a sentence. For single word, there may not be the prior knowledge about the context. But in case of a sentence the spectral patterns and spectral distance are matched. Most of the traditional systems use Gaussian Mixture Model and Hidden Markov Model [6]. The deep learning methods have extreme features as comparable to the traditional system, but requires large number of speech dataset. Several research works have been done to increase the accuracy of speech command recognition [7]-[8]. Researchers tried to find out a robust model for better accuracy.

In the past few years many research works have been done in the area of speech recognition. At that time very less hidden layers as well as small amount of data has been used. The researchers used both machine learning algorithms and advanced computer hardwares, which was the beginning of deep learning. The Deep Neural Network contains many hidden layers and large numbers of output layers. Warden [2] in the year 2018 collected spoken words used for training and testing, which is named as speech command. He tested the dataset by CNN models and got an accuracy of 88%. In the year 2018 Gupta, A., & Joshi published a speech recognition system using ANN and RNN. They used Restricted Boltzmann architecture and bi-directional recurrent neural networks in their work. At first, there is a rapid increase, but after a few epochs, the curve flattens, indicating that the pace of growth is declining. Guiming proposed a model that takes RNN and CNN, it gives 96.0% accuracy whereas CNN alone gives the accuracy as 85% [9]. Sushan Poudel & R Anuradha proposed a model combining CNN and RNN. In their model they got 96.66% accuracy. Xuejiao Zixuan Zhou [10] developed a project using CNN and DNN. In their model they also got accuracy of nearly 95.00%. Ghandoura et.al [7] designed a dataset in Arabic language for speech command. He applied various models using CNN, that model give an accuracy of 97%. A combination of Attention and RNN approach was adopted by S.Yang [11], which gave a very challenging accuracy for speech command recognition. Another model of CNN which is based on depth-wise separable proposed by Z hang [12] achieves maximum 95.4% of accuracy. Gupta et al. [13] proposed a method to recognise the command of low level languages to build a device. Their dataset consists of Bengali commands. The proposed method obtained an accuracy of nearly 83%. P.Phan and T.M.Giang [14] designed a model using Recurrent Neural Network using self-collected command in Vietnamese. Q.H Nguyen [15] proposed a correction system using SVM and CNN in which errors are reduced. Shuvo etal [16] achieved remarkably accuracy of 93.65 % using the CNN model in Bengali speech command. Using isolated speech Suman etal [17] obtained the accuracy of 74 % on short speech commands. An attention based speech recognition model was proposed

by Shanetal [18] which gives a character error rate of 6.49% in Mandarin Dataset. Many speech command recognition systems have been developed, with varying degrees of success depending on the methods used.

## 3. Methods

For speech recognition, a variety of neural network techniques are used [19]. However, no work has been discovered yet for speech recognition in a low-resource Kui Language. Through this study, we have compared four different neural network models namely Artificial Neural Network (ANN), Recurrent Neural Network (RNN), Convolutional Neural Network (CNN) and Attention model using LSTM [20]-[21].

### 3.1. Artificial Neural Network (ANN)

A non-linear model can be handled by artificial neural networks more skilfully. It can develop data models much more effectively. ANN is used for both regression and classification problems. More than one layer of hidden units present between its inputs and outputs. It is widely used for feature extraction in image processing and similar tasks. The basic architecture of the ANN includes both Single-Layer Artificial Neural Networks and Multi-Layer Artificial Neural Networks. ANN cannot record continuous speech with sequential information. It is one of the simplest mathematical model to increase the data analysis technologies.

### 3.2. Recurrent Neural Network (RNN)

Recurrent neural networks are a variant of neural network that are capable of recognizing patterns in sequences of data, such as text, genomes etc. The improved version of ANN is RNN. The primary distinction is that while neurons are present in ANN, they are replaced in RNN by memory blocks that store previous output. After the discovery of RNN the understanding and listening ability of the machines have been improved tremendously [14]. RNN is mainly used in situations involving time-series data or values that vary over time. RNN provides a sequence of samples that result in a single output prediction. RNN can predict an output sequence based on an input sequence. These are adaptable because input and output lengths can be adjusted. RNN computes the loss through back propagation across time. There are no limitations on the length of inputs or outputs for RNN. It works on loops to handle sequential data

### 3.3. Convolutional Neural Network (CNN)

CNN is a deep learning algorithm that can take an input, assign the significance of to the various objects and able to differentiate one from the other. Convolution is the act of multiplying two functions to produce a third function that illustrates how the form of one function is changed by the

other. The term convolution is used in CNN to refer to this mathematical activity. It is manly used in image classification. It is very useful because it minimizes the human effort. CNNs are a subclass of deep neural networks that can identify and categorise specific features [16]. It has a better accuracy as compared to other neural network models. Feature map is the output of the convolution layer. The pooling layer's primary goal is to reduce the size of the feature map. This CNN approach allows the networks recognize the features on their own by generalizing the characteristic extracted by the convolutional layer [9], [22].The convolutional methods are slow in process, more power consumption and required high band width..

### 3.4. Attention using LSTM

The LSTM is a particular kind of Recurrent Neural Network (RNN) that can solve long-term dependence. It can handle the vanishing gradient problem. Forget gate, Input gate, and Output gate make up the three gates [23] that constitute an LSTM. LSTM has a hidden state. Previous and current time stamp are recorded in this method. The cell state is also referred as long-term memory, and the hidden state as short-term memory. Due to presence of gates, it is able to solve the gradient problem in a better way. LSTM is used for classifying audio data for getting better results as compaired to others. The advantage of LSTM is that it can learn long-term dependencies. LSTM has variants like: Unidirectional, Gated Recurrent Unit and Bidirectional LSTM. LSTM networks have memory blocks instead of neurons that are linked by layers. For training, back propagation is used.

Attention mechanism was firstly introduced in the year 2015. This mechanism have altered the way of working by use of deep learning [24]. This mechanism has contributed to the disciplines like speech processing. In case of attention mechanism applied on neurons, we select the relevant things and concentrate on them. The remaining things will be discarded by this mechanism [25]. Before the attention mechanism the Encoder-Decoder architecture was very popular. But it encodes the sequence of a fixed length. Due to which, it gives not so satisfactory results for long sequences. In attention mechanism the input sequence size is large [26], as shown in figure. 1.

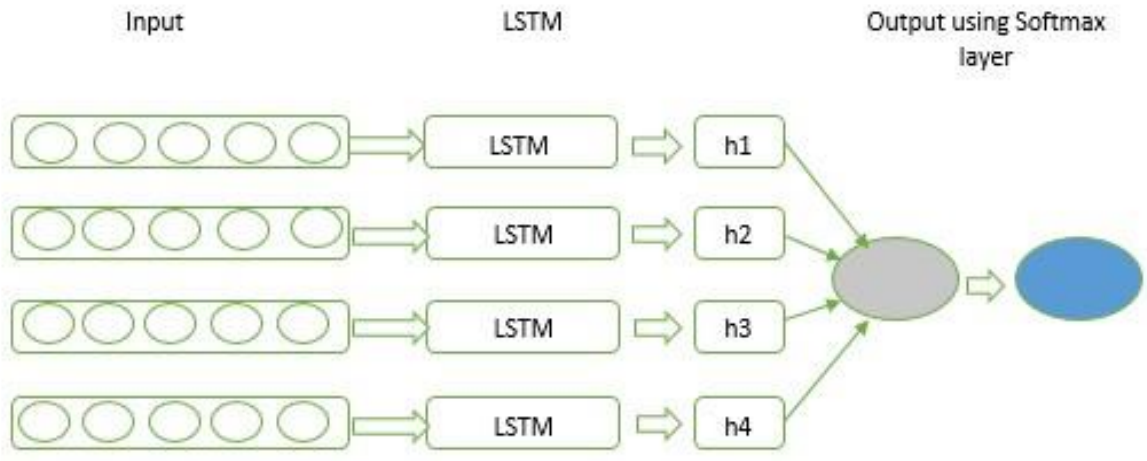


**Fig 1:** Architecture of Attention using LSTM

## 4. Experimental Analysis and Discussion

For speech recognition, a variety of neural network techniques are used [19]. However, no work has been discovered yet for speech recognition in a low-resource Kui Language. Through this study, we have compared four different neural network models namely Artificial Neural Network (ANN), Recurrent Neural Network (RNN), Convolutional Neural Network (CNN) and Attention model using LSTM [20]-[21].

### 4.1. Dataset Preparation

The dataset used for our experiments is the speech command dataset which is created in Kui language. The dataset consists of 7,090 utterances of 7 words. Each utterance is stored as one-second (or less) wave format file as linear 16-bit single channel PCM values at a 16 kHz rate. The short commands are recorded in Kui using Zoom audio recorder, mobile and laptop. It is recorded in a studio for minimizing the noise. It consists of male and female voices in 50:50 ratio. After the recording sampling rate is checked. All the audio files are saved in .wav format. The speech commands in Kui languages are shown in Table 1.

**Table 1**: Kui Speech command and its meaning

| Command (Kui) | Command (Odia) | Command (English) | Command (IPA) | Command (Roman) | #files |
|---|---|---|---|---|---|
| ସେଣ୍ଡନି | ଉପର | Up | upɔɾɔ | upara | 1012 |
| ନେ<ଡେ | ତଳ | Down | t̪ɔl̪ɔ | taḷa | 1004 |
| ଦେବାଟି | ବାମ | Left | bamɔ | bāma | 1017 |
| ଡିନେଟି | ଡାହାଣ | Right | ɖahaɳɔ | ḍāhāṇa | 1021 |
| ବନ୍ଦ | ବନ୍ଦ | Stop | bɔnɖ̪ɔ | banda | 1009 |
| ଜା>ମ୍ବା | ବିରତି | Pause | biɾɔt̪i | birati | 1021 |
| ସାଲୁମୁ | ଯାଅ | Go | dʒaɔ | jāa | 1006 |

### 4.2. Data Pre-processing

All the commands are recorded at a rate of 16 KHz. The variation in audio signal is more in lower frequencies, as a result number of features can be extracted from part of the signal. For accurate recognition of the set of commands by the neural network models, the neural network models requires a large and optimal dataset of audio samples. Therefore, some optimal features are required, which can be extracted by the method of feature extraction. The different features of the commands required for deep learning models including 1D and 2D representations. The waveforms are able to get the features in 1D. For 2D representation several feature extraction methods can be used. Out of all feature extraction methods, Mel-frequency crystal co-efficient (MFCC) gives an optimal representation of audio data [27]. Due to this reason MFCC feature extraction method is used in case of neural network. The complexity of MFCC is O (*nlogn*), where *n* is no of samples.

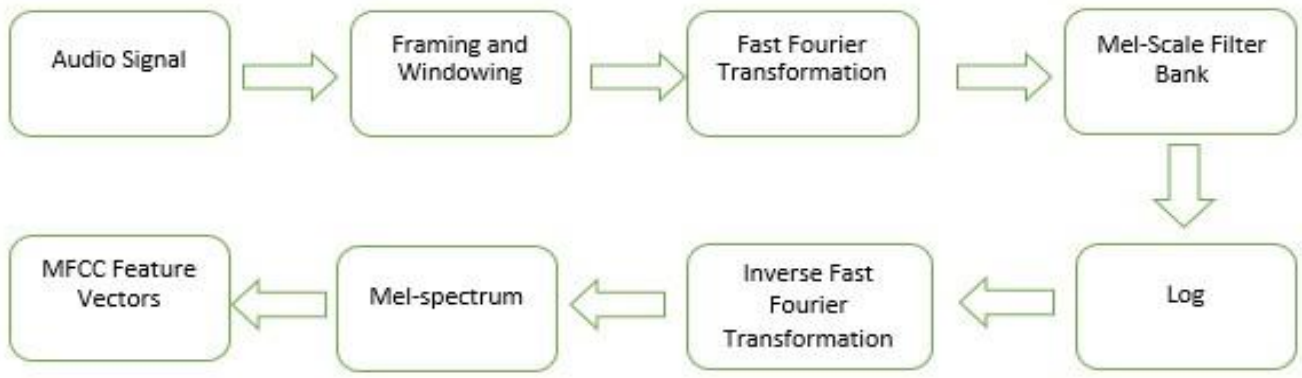


**Fig 2**: Architecture of Mel-frequency crystal co-efficient (MFCC)

Using Fast Fourier Transformation (FFT) the audio signal is converted from time domain to frequency domain. Spectrum is obtained from FFT. The signal is converted to a log scale for visible output from lower frequency. After that the signal is passed through a standard Mel scale filter bank. In the Mel-spaced filter bank the received signals overlap over one another. Discrete Cosine Transformation is required for remove the overlapping signals. The audio data which do not contain any useful data are discarded. The architecture of MFCC is shown in figure. 2.

### 4.3. Experimental Setup

As discussed above we evaluate various deep learning models on our Kui dataset. The data are divided into two parts i.e. training and testing. We take 3 different percentage of training and testing which are 70:30, 80:20 and 90:10. The batch size is set to 32 in the training process and the speech commands in a batch are configured to same length by zero padding. All the experiments are done on CPU, Core i5 processor, 16 GB RAM with python version 3.9. Four

different models were taken for implementation. The loss objective function was based on the criteria loss = "categorical_crossentropy" and the Adam optimizer whose learning rate is 0.01. In the experiment 4 number of epoch sizes were taken i.e. 50, 100, 200 and 300.

CNN is one of the most widely used methods for speech command recognition as the new technical progression in Deep Learning approaches. For CNN, filters and classification layers make up the network layer. CNN consists of RELU, MaxPooling, Dense as well as Softmax layer. To convolve the input data, kernels are used. As the dimension of the convolution layer is large, it must be reduced. For reducing the dimension we use the pooling layer. The drop out ratio taken is 30% and Initial Learning Rate Value is fixed. The flatten layer takes the input from the pooling layer and it converts into a vector. In the Softmax layer, the vector's probability is computed and used for classification. The model Attention using LSTM the accuracy is more as compare to the other models using our Kui command dataset. In a simple encoder-decoder model we may not use the hidden state of LSTM but here we passed to a dense layer. We take sampling rate as 16,000 and the normalisation in 2D.RELU is used as the activation function. LSTM_Sequence is used for the unit measurement. Softmax is used here to calculate the attention score in this model. Two Batch Normalisation layer is used here. We set return_sequence= True in LSTM layer because we need LSTM to all the hidden states. The attention layer has 3 phases. We use the mechanism which was stated by Xie etal. The Tanh or Leaky Relu activation functions are even not giving better accuracy.

### 4.4. Results Analysis

As discussed above we evaluate various deep learning models on our Kui dataset We take 3 different percentage of training and testing which are 70:30, 80:20 and 90:10. The batch size is set to 32 in the training process and the speech commands in a batch are configured to same length by zero padding.

We train the models using different parameters to get the best performance for the Kui command data set which is recognised as a low resource language. We take the learning rate as 0.01. In our experiment, first we take the epoch size of 50. In this epoch size, we take the training and testing ratio is 70-30. In ANN the training accuracy is 0.68 whereas testing accuracy vertically increases up to 10 epoch then after slowly increases and finally gives 0.60. In RNN training accuracy is 0.93 whereas testing accuracy 0.86. All the training accuracies up to 300 epoch and split ratio 80-20 graphically shown in Fig. 3. CNN also gives very good accuracy of nearly 0.87. But in attention using LSTM gives 0.92 accuracy. Similarly, for 50 epoch we take the split ratio as 80-20 and 90-10. Attention using LSTM model gives the accuracy better than the other 3 models in epoch size 50. For epoch size 100, CNN gives better result in 80-20 split ratio as compared to other two ratios. The details of all testing accuracy are given in the Table 2. In 100 epoch, attention using LSTM model also gives better accuracy then the other 3 models. The graphical representations of the accuracies are shown in figures 4-7. With the epoch size of 200 except ANN all the models have the parallel accuracy. CNN model gives better result in 90-10 split ratio as compare to other two split ratio. Attention using LSTM model also gives the better accuracy in 200 epoch. Finally, for epoch size 300, Attention using LSTM model gives the highest accuracy among all other models. In Attention using LSTM model we saw that all three splits gives the same results

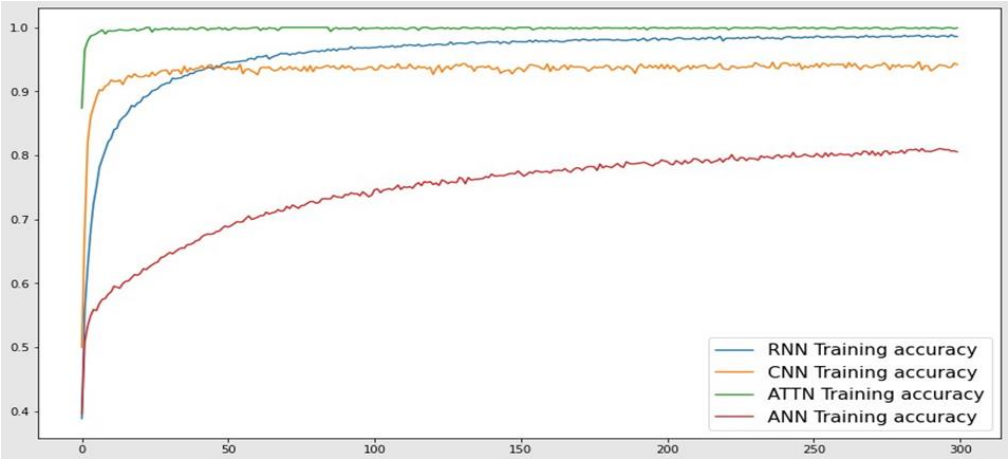


**Fig 3**: Comparison of Training accuracy with epoch size 300 and split ratio 80-20 (Training-Testing)

**Table 2**: Comparison between testing accuracy

| No of epochs | split-ratio | Accuracy | | | |
|---|---|---|---|---|---|
| | | ANN | RNN | CNN | Attention using LSTM |
| 50 | 70-30 | 0.60 | 0.86 | 0.87 | 0.92 |
| | 80-20 | 0.62 | 0.88 | 0.89 | 0.93 |
| | 90-10 | 0.62 | 0.86 | 0.87 | 0.94 |
| 100 | 70-30 | 0.61 | 0.88 | 0.89 | 0.93 |
| | 80-20 | 0.63 | 0.87 | 0.90 | 0.94 |
| | 90-10 | 0.63 | 0.88 | 0.89 | 0.95 |
| 200 | 70-30 | 0.69 | 0.75 | 0.92 | 0.94 |
| | 80-20 | 0.64 | 0.86 | 0.91 | 0.95 |
| | 90-10 | 0.72 | 0.82 | 0.94 | 0.95 |
| 300 | 70-30 | 0.72 | 0.76 | 0.93 | 0.97 |
| | 80-20 | 0.65 | 0.87 | 0.93 | 0.97 |
| | 90-10 | 0.73 | 0.79 | 0.95 | 0.97 |

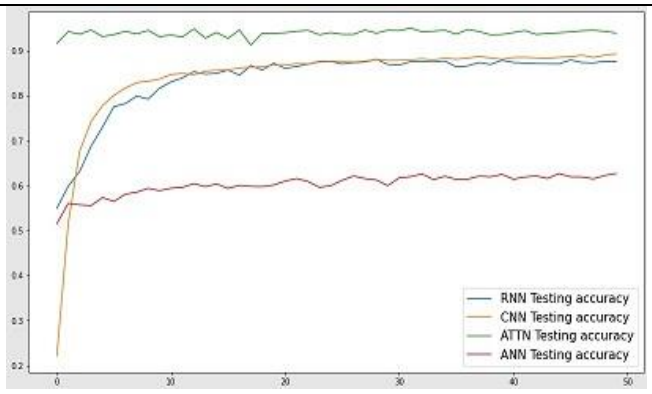


**Fig 4**: Comparison of Testing accuracy of different models with epoch size 50 and split ratio 80-20 (Training-Testing)

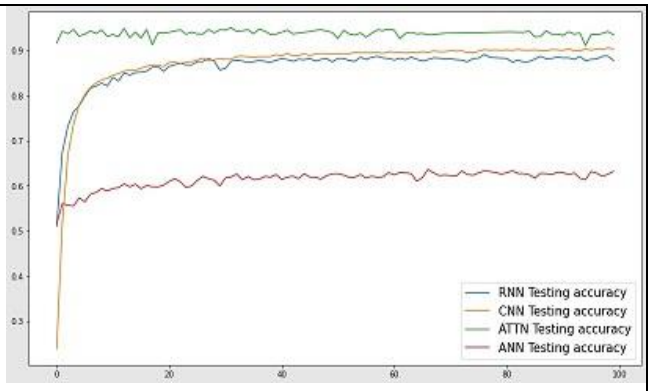


**Fig 5**: Comparison of Testing accuracy of different models with epoch size 100 and split ratio 80-20 (Training-Testing)

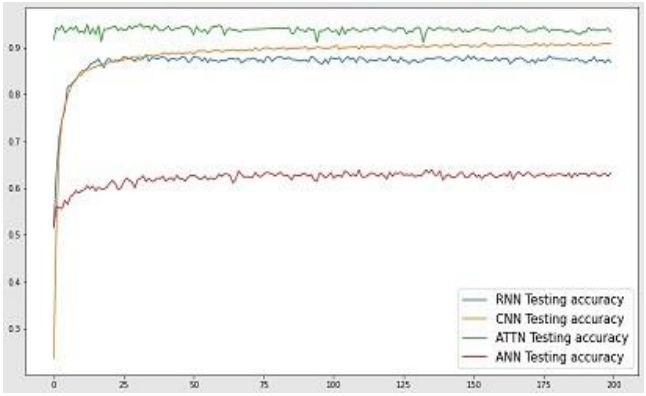


**Fig 6**: Comparison of Testing accuracy of different models with epoch size 200 and split ratio 80-20 (Training-Testing)

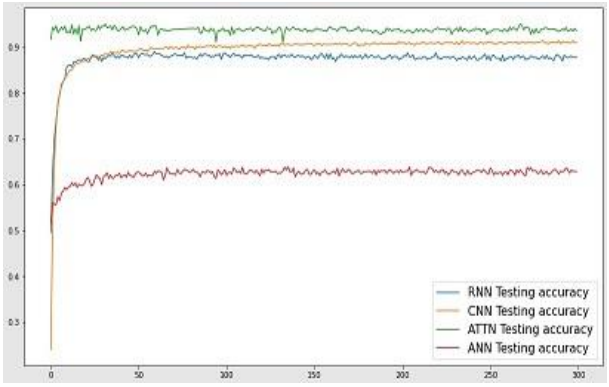


**Fig 7**: Comparison of Testing accuracy of different models with epoch size 300 and split ratio 80-20 (Training-Testing)

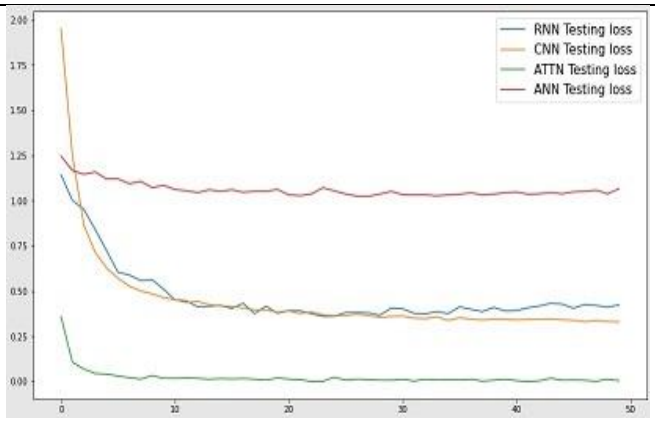


**Fig 8**: Comparison of Testing loss of different models with epoch size 50 and split ratio 80-20 (Training-Testing)

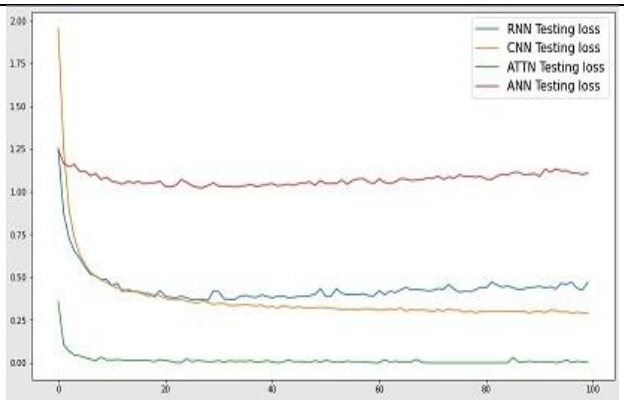


**Fig 9**: Comparison of Testing loss of different models with epoch size 100 and split ratio 80-20 (Training-Testing)

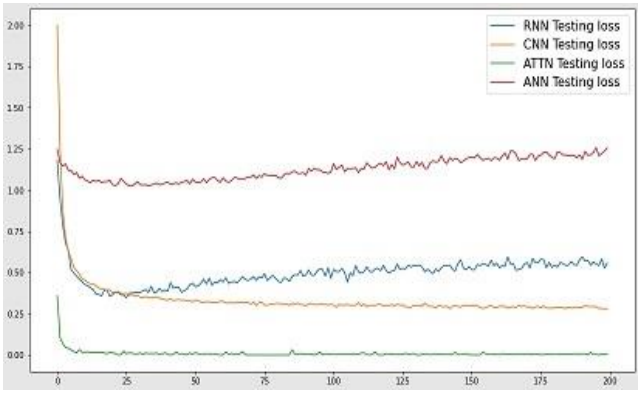


**Fig 10:** Comparison of Testing loss of different models with epoch size 200 and split ratio 80-20 (Training-Testing)

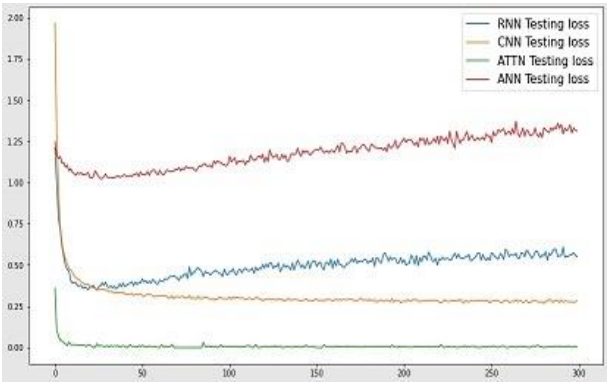


**Fig 11**: Comparison of Testing loss of different models with epoch size 300 and split ratio 80-20 (Training-Testing)

The different testing loss graphs are shown in figures 8-11. In Figure 8 we take the epoch size 50 and the spilt ratio 80-20, the loss of ANN is more as compare to the other. Attention using LSTM model have the minimal loss. For CNN model the loss consistently decreased. Attention model also has decreased loss value while increasing the epoch size. From our experiment we conclude that Attention using LSTM model have the minimal loss as compare to other 3 models.

The classification metrics that are assessed in order to determine accuracy are precision, recall, and F1-score [28]. All the metrics are found in different epoch size with different split ratio. The table 3 show the different classification metrics with split ratio 80-20 with epoch size 50,100,200 and 300. From the result analysis Attention using LSTM model is able to achieve the highest accuracy as compare to the other 3 models.

**Table 3**: Comparison of different performance metrics

| No of epochs | Performance metrics | Models | | | |
|---|---|---|---|---|---|
| | | ANN | RNN | CNN | Attention using LSTM |
| 50 | Precision | 0.55 | 0.88 | 0.93 | 0.95 |
| | Recall | 0.60 | 0.85 | 0.87 | 0.93 |
| | F1-Score | 0.57 | 0.86 | 0.90 | 0.94 |
| 100 | Precision | 0.62 | 0.92 | 0.87 | 0.97 |
| | Recall | 0.56 | 0.84 | 0.95 | 0.91 |
| | F1-Score | 0.59 | 0.88 | 0.91 | 0.94 |
| 200 | Precision | 0.61 | 0.85 | 0.89 | 0.94 |
| | Recall | 0.59 | 0.89 | 0.95 | 0.96 |
| | F1-Score | 0.60 | 0.87 | 0.92 | 0.95 |
| 300 | Precision | 0.61 | 0.84 | 0.93 | 0.97 |
| | Recall | 0.63 | 0.86 | 0.95 | 0.97 |
| | F1-Score | 0.62 | 0.85 | 0.94 | 0.97 |

## 5. Conclusion and Future works

This study mainly aims for command recognition in a self-created Kui dataset. We tried to improve the performance using the different learning rates and epochs. For models to be trained, we have used MFCC. Other feature extraction techniques may also be used to find the effect on training time. The experimental results have indicated that command recognition in this study had an accuracy 99% of training and 97% of testing.

This paper contributes in terms of experiments of CNN, ANN, RNN and attention using LSTM. Among various models, it gives better performance for our Kui command dataset. The command recognition accurate is seen more than 97%. Attention using LSTM model provides greater accuracy, decreases model complexity, and also lowers error rates. The aforementioned findings lead us to the conclusion that the Attention LSTM model can outperform other models if it is given a strong feature set to train on. As time passes and technologies evolve more amount of data are required to train the model to get more accuracy. Speech command recognition is mainly used for people who don't know how to read and write in Kui language, it also very much helpful for visually challenged persons.

The future scope of the system will be to make it capable to handle multiple voice command with speaker recognition using our Kui command dataset. It may be used for security purpose with additional speaker recognition facility. In future, we will work on the different feature extraction method and study their effect. The commands specified in our experiment are very limited. The system can be enhanced for other type of commands and gestures. In future, we will also study the application of this training strategy like speaker recognition using Kui dataset. We also intend to experiment like dialog act recognition and slot filling. In addition, we will study how to improve the robustness of these models so that it can be used in real world scenarios. Nevertheless, these methods still has to be tested on bigger and different Kui datasets and improvements are may expected. We can further design a model by substituting LSTM with Transformer model using our dataset. It is possible to increase the recognition accuracy by replenishing the audio recording. It is possible to increase the amount of words that can be recognized, further refine the model without affecting performance, and investigate the feasibility of recognizing kui words. With good generalization performance and high practical value, the outcome can increase the recognition command's accuracy in a pragmatic environment. To create a fully functional human-computer interactive speech system, we will also build additional speech tasks like speaker recognition and speaker verification and combine them with this Kui commands.

## References

[1] Zhang. Z, *et al*., "Deep learning for environmentally robust speech recognition: An overview of recent developments," *ACM Trans. Intell. Syst.Technol*.

vol.9, pp. 1–28, 2018, doi: 10.1145/3178115.

[2] Warden. P., "Speech Commands: A Dataset for Limited Vocabulary Speech Recognition," 2018, doi: 10.48550/arXiv.1804.03209.

[3] Abate .ST, Tachbelie .MY, Schultz .T, "Deep neural networks based automatic speech recognition for four Ethiopian languages," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8274- 8278, 2020 May 4, doi: 10.1109/ICASSP40776.2020.9053883.

[4] Shetty. Vishwas M., NJ.Matilda Sagaya, "Improving the performance of transformer based low resource speech recognition for Indian languages," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8279- 8283, 2020 May 4, doi: 10.1109/ICASSP40776.2020.9053808.

[5] Islam Mohammad Shakirul, Foysal Ferdouse Ahmed, Neehal Nafis, Karim Enamul, Hossain Syed Akhter, "InceptB: A CNN Based Classification Approach for Recognizing Traditional Bengali Games,"*Procedia Computer Science*, Vol. 143, pp. 595-602, 2018, doi: 10.1016/j.proc.2018.10.436.

[6] Sun Xiusong, Yang Qun, Liu Shaohan, Yuan Xin, "Improving low-resource speech recognition based on improved NN-HMM structures," *IEEE Access*, pp. 73005-14, 2020 Apr 16, doi: 10.1109/ACCESS.2020.2988365.

[7] Ghandoura A., Hjabo F., and Dakkak O. Al, "Building and benchmarking an Arabic Speech Commands dataset for small footprint keyword spotting," *Eng. Appl. Artif. Intell*. vol. 102, 2021, doi: 10.1016/j.engappai.2021.104267.

[8] Amoh Justice and Odame Kofi M., "An Optimized Recurrent Unit for Ultra-Low-Power Keyword Spotting," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*., vol. 3 of 2, pp. 1–17, June 2019, doi:10.1145/3328907.

[9] Guiming Du, Xia Wang, Guangyan Wang & Dan Li, "Speech recognition based on convolutional neural networks," *IEEE International Conference on Signal and Image Processing (ICSIP)*, 2016, doi: 10.1109/SIPROCESS.2016.7888355.

[10] Li Xuejiao, and Zhou Zixuan, "Speech Command Recognition with Convolutional Neural Network," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2020.*

[11] Yang S, Gong Z., Ye K., Wei Y., Huang Z., "Edge RNN: A Compact Speech Recognition Network with Spatio-Temporal Features for Edge Computing," *IEEE Access*, vol. 8, pp. 81468-81478, 2020, doi: 10.1109/ACCESS.2020.2990974.

[12] Zhang Yundong, Suda Naveen, Lai Liangzhen and Chandra V., "Hello Edge: Keyword Spotting on Microcontrollers," 2017, doi: 10.48550/arXiv.1711.07128.

[13] Gupta D, Hossain E, Hossain M. S, Andersson K, and Hossain S, "A digital personal assistant using bangla voice command recognition and face detection," *IEEE International Conference on Robotics, Automation, Artificial-intelligence and Internet-of-Things (RAAICON)*. , pp. 116–121, 2019, doi: 10.1109/RAAICON48939.2019.47.

[14] Hung Phan Duy, Giang T. M, Nam L. *et al*., "Vietnamese speech command recognition using recurrent neural networks," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 10, no. 7, 2019, doi: 10.14569/IJACSA.2019.0100728.

[15] Nguyen Q. H and Cao T. D, "A novel method for recognizing vietnamese voice commands on smartphones with support vector machine and convolutional neural networks," *Wireless Communications and Mobile Computing*, vol. 2020, 2020, doi: 10.1155/2020/2312908.

[16] Shuvo M, Shahriyar S. A, and Akhand M., "Bangla numeral recognition from speech signal using convolutional neural network," *International Conference on Bangla Speech and Language Processing (ICBSLP)*. , pp. 1–4, 2019, doi: 10.1109/ICBSLP47725.2019.201540.

[17] Sumon S. A, Chowdhury J, Debnath S, Mohammed N, and Momen S, "Bangla short speech commands recognition using convolutional neural networks," *International Conference on Bangla Speech and Language Processing (ICBSLP)*. , pp. 1–6, 2018, doi: 10.1109/ICBSLP.2018.8554395.

[18] Shan C, Weng C, Wang G and Xie L, "Investigating end-to-end speech recognition for mandarin-english codeswitching," *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. pp. 6056–6060, 2019, doi: 10.1109/ICASSP.2019.8682850.

[19] Nassif A. B, Shahin I, Attili I, Azzeh M, and Shaalan K, "Speech recognition using deep neural networks: A systematic review," *IEEE Access*, vol. 7, pp. 19143–19165, 2019, doi: 10.1109/ACCESS.2019.2896880.

[20] Cong Guojing, Kingsbury B, Yang C-C, Liu T, "Fast Training of Deep Neural Networks for Speech Recognition," *International Conference on Acoustics, Speech and Signal Processing (ICASSP),*pp.6884-6888, 2020, doi:

10.1109/ICASSP40776.2020.9053993.

[21] Solovyev R. A, Vakhrushev M., Radionov A., Romanova I. and Shvets A. A, "Deep learning approaches for understanding simple speech commands," *IEEE 40th International Conference on Electronics and Nanotechnology (ELNANO)*., pp.688–693,2020,doi: 0.1109/ELNANO50318.2020.9088863.

[22] Hamid O. Abdel, Mohamed A. R, Jiang H, Deng L, Penn G, and Yu D, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on audio, speech, and language processing*, vol. 22, no. 10, pp. 1533–1545, 2014, doi: 10.1109/TASLP.2014.2339736.

[23] Zazo R, Nidadavolu P. Sankar, Chen N, Rodriguez J. Gonzalez, and Dehak N, "Age Estimation in Short Speech Utterances Based on LSTM Recurrent Neural Networks," *IEEE Access*, vol.6,pp.22524–22530,2018.

[24] Shan C, Zhang J, Wang Y, and Xie L, "Attention-based end-to-end models for small-footprint keyword spotting," *Proc. 19th Annu. Conf. Int. Speech Commun. Assoc*., pp. 2037–2041, 2018, doi: 10.48550/arXiv.1803.10916.

[25] Leo Sabato, Viana Martin Loesener Da Silva, and Bernkopf Christoph, "A Neural Attention Model for Speech Command Recognition," *Engineering Applications of Artificial Intelligence*, August 2018, doi: 10.48550/arXiv.1808.08929

[26] Berg Axel, Connor Mark O, and Cruz Miguel Tairum, "Keyword transformer: A self-attention model for keyword spotting,"Proc. Interspeech 2021, pp. 4249-4253, 2021, doi:10.48550/arXiv.2104.00769.

[27] Namrata D, ''Feature extraction methods LPC, PLP and MFCC in speech recognition,'' *Int. J. for advance Res. Eng. Technol*., vol. 1, no. 6, pp. 1–4,2013

[28] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, and Duchesnay Edouard, "Scikit-learn: Machine ´ learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2012, doi: 10.48550/arXiv.1201.0490.