VOICE INTELLIGENCE BASED WAKE WORD DETECTION OF REGIONAL DIALECTS USING 1D CONVOLUTIONAL NEURAL NETWORK

Chaitra Gowdra Parameswarappa[1] and Shylaja Sharath[1]

[1]Affiliation not available

April 16, 2024

VOICE INTELLIGENCE BASED WAKE WORD DETECTION OF REGIONAL DIALECTS USING 1D CONVOLUTIONAL NEURAL NETWORK

CHAITRA.G.P
Computer Science Department
PES University
Bengaluru
chaitragp@pesu.pes.edu

SHYLAJA.S.S
Computer Science Department
PES University
Bengaluru
shylaja.sharath@pes.edu

Abstract— Voice-based apps can be effective among rural farmers, if it is in their own spoken language/dialect. Many voice-based apps were developed in the agricultural sector, and in each case, farmers had to either type in the queries or they had to communicate with the device which had the standard speech to deliver the solution which added to the challenge to comprehend the information. This paper presents the research work in developing the wake word detection system for major dialects based on 5 different regions in Karnataka, namely-Dharwad, Dogganal, Tulu, Kodagu and Urban Kannada.The customized wake word system is designed using 1D CNN model with 98% accuracy which showed better results over ANNs with 14.1% and RNNs with 48.1% accuracies. The diversity in regional dialects has been well identified using Conv1D model and with comparative analysis with RNNs to validate on the sequential data the predicted labels were compared and the performance of Conv1d reconciles well for the Dialect Dataset.

Keywords— TensorFlow, Keras API, Mel Frequency Cepstral Coefficients, CNN, Dialect Identification, Deep Learning Techniques, Sequential Modelling.

I. INTRODUCTION

Majority of the Indian population, around 58% -depend on agriculture as the major source of income. Information Management is the main challenge faced by rural farmers. Acquiring relevant information, comprehending it and implementing it in the right way can yield good production [19]. [11] has mentioned about how understanding the farmer's conception towards these modern facilities, can play a major role in analyzing the factors which can influence the adoption of Technology like, education, financial conditions, social constraints, dependency on Mandi (local market) for price information, and social network among the farming community. Transformation from feature/basic phones to smart phones has paved way for different approaches in the technology series [12]. In recent years, with the advancement of Technology, especially in the field of Artificial Intelligence and Data Science, various brands have come up with voice-based apps to provide a flawless experience for their consumers [19]. Mobile apps can become one of the most powerful tools for delivering relevant information to rural farmers regarding agricultural needs without any third-party influence [11].

Various agriculture related apps are released by the Government and other private companies which include both voice assistance and chatbots. Few popular apps include Krishify where farmers can search for any agricultural related topics, AgNext integrates various agriculture related services along with e-Nam platform [22]. Agri Media Video App is an online retail market. It also provides online chat services connecting farmers with field experts [12]. FarmBee app, which is available in 10 distinct Indian languages, can provide information at different stages in the life cycle of the crop. Kisan Yojana provides information regarding various government schemes and policies in the agricultural sector [23]. Kumar [14] has mentioned difficulties faced by farmers in using these applications. While using these apps, farmers had to either type in the queries or they had to communicate with the device which had the standard speech to deliver the solution. [21][14] And based on the statistics provided in the research paper around 65% of blockade are related to language since most of these apps use standard speech to deliver the solution. [23] And with respect to Field experts, human error may occur, or they may lack the expertise in providing the right solution to the farmers issues. [20] Out of 58%, only around 15 to 20% of farmers are ready to shift to online platform. This gap may affect the agriculture sector in the coming years [14]. India is a country with 120 major languages with 1600 dialects. And if Technology must reach the granular level to suite the rural farmer's need, then the fact about the language barrier cannot be left unnoticed [13]. There is a dire need in creating a bond between Technology and a farmer through which farmers can easily communicate with the device and this is only possible when the communication is through their native language and dialect [15]. Above mentioned facts only lead us to one important aspect that the farmer community may feel it is easier to speak to the device in their own native language and dialect, than typing. Vernacular voice-based apps can make the farmers feel connected to the technology through which they can leverage on getting various solutions for their agricultural

TABLE 1:SUMMATION OF DEEP LEARNING TECHNIQUES AND TENSORFLOW FOR WAKE WORD SYSTEM

Convolutional Neural networks	Recurrent Neural Networks and LSTM	Transformers	Tensors
Not suitable for sequence modelling.	Sequential processing of the data.	Attention mechanism is used to overcome the issues encountered by RNNs and LSTMs.	TensorFlow framework with Keras API provides efficient Language modelling libraries including the features of transformer models.
Uses position embedding technique	RNN's cannot access the data from faraway positions (vanishing gradient problem)	Focuses on specific parts of the data and tackles the issues with Homonyms in NLP.	These libraries do the job of word embedding (tokenization and text vectorization) effortlessly. Can be applied on raw audio data.
CNN based wake word system works on fixed sizes on inputs which can cause some errors with long durations as it may consider some non-relevant utterances as well.	LSTMs are variation of RNN and deal with vanishing gradient problem by applying gate technique which indicates what information to be kept. Best suitable to handle long sequences of input data rather than short length like wake word systems.	Transformers can handle long sequences of sequential data using attention models which can be less effective for short sequences like wake word systems.	Keras APIs provide better techniques in updating the weights by focusing on granular details of the data which yields good results in Dialect Identifications tasks. Also suitable for short word sequences like wake word systems compared to other deep learning techniques. Also, they are easier to deploy on android devices.

needs to gain potential benefits [11]. They only need a simple logical solution by speaking to the device and connecting to different platforms without any intervention from the middlemen [13].

Dialect Identification is one of the upcoming topics in the world of speech recognition tasks. It is one of the most challenging in terms of differentiating it with the spoken language in terms of complexity and overlapping phonetic systems [1]. In [5] authors have mentioned about the deficiency in the resources for Dialect Identification (DID) for modelling. There are very small variations in the parameter related to the utterance of the same word with different styles for the same language [4]. For any DID task a few parameters should be closely monitored like prosodic features including intonation, phonology, vocabulary and grammar. Using TensorFlow Lite makes the task much easier to deploy the model on Android devices.

II. LITERATURE REVIEW

In this paper, we mainly focus on wake word detection method which is the phase-1 of our research work in providing the voice-based solution to the farmers mainly focusing on dialects. Most of the work in recent years is based on using different deep learning techniques.

[8] Tsai T H proposed the wake word system in real time based on Convolutional Neural Networks (also termed as CNN). After preprocessing the data with MFCC (Mel-Frequency Cepstral Coefficient), they have used GMM (Gaussian Mixture Model) to train the speaker identification model which uses likelihood function to identify the true speaker. Next, for each GMM model posterior probability is predicted and state sequence is compared using Hidden Markov Models [HMM]. Hidden Markov Model is efficient in partitioning human speech into different syllables. And wake up action is achievable only when both state sequence and posterior probability passes the threshold. Probabilistic model, rather than assuming for the entire distribution it assumes for some moments which makes it more accurate than machine learning. The paper [2] proposed a LSTM (Long Short-term Memory) based method for trigger word or wake word detection for speech data. It is the variant of Recurrent neural Network (RNN) which supports long term dependencies among the timestep of the data (which is done on spectrogram). Here in timestep which is a backpropagation technique of using current as well as previous inputs as an input to the neuron. Authors have also explained the facts about LSTM which was developed to handle vanishing gradient and exploding issues while training RNN's. LSTM techniques are good in handling lengthy speech data.

[18] In this paper the authors have proposed a wake word detection system using Transformers, which accomplished better results over LSTM and CNN sequence modelling tasks. One of the main points highlighted in the paper is-Since wake word detection is a short-range temporal model, large sequence modelling like Transformers may not be a viable option. Transformers use attention mechanism for having long term memory. It has an attention-based encoder and decoder mechanism, where the encoder holds all the information learned for the input sequence and decoder then intakes that sequence and gives a single output also considering the previous output. The model can "attend" on all tokens which are generated previously. Authors have adopted a LF-MMI (Lattice Free-Maximum Mutation Information) system which includes gradient stopping, looking forth to the next piece of data, embedding methods based on positions in the sequence and have layer dependencies. It resulted in outperforming CNN by 25% in the rate for false rejection and sustains the linear complexity for the segment length. They have also mentioned using tensors as it may be more efficient for short range word sequences in which instead of considering the entire utterance it only focuses on the target word.

[17] proposed the system deployed with Res2Net which is the better variation for ResNet. Here it enhances the ability of detecting the wake word of different durations. It is applied on Mobvoi data which consist of two wake words. And, it has a rate of false rejection at 12% over other systems. Res2Net is a classification model with a broadened receptive field which increases the detection capability of the model. With fewer model parameters. It is done by extracting the exact features by considering the local features and then

capitulating the global feature of the same size from the given region of varied lengths.

III. PROPOSED WORK

The proposed work is based on multi-class classification problem wherein every input belongs to only one class. Wake words are used to start the conversation and wake up the device to respond to our queries. Device cannot continuously listen to the conversation it may cause the security breach and may also lead to huge load on the servers to process each audio signal. It only starts listening to our commands once the wake word is detected and device is woken up. The wake word detection system is a 5-step process.

First, we need to prepare the data set by recording the audio for few seconds containing the wake word in different dialects and recording the audio which do not contain the wake word.

Next, convert the raw audio data into waveform which is in time domain, but to better analyze and extract the important features in the audio we need to transform waveform of time domain to frequency domain mainly in our work we are implementing MFCCs. Each MFCCs are labelled and these labels and features are saved into pickle file for later use. Based on the dataset and the problem statement we need to choose the deep learning model which yields the best predictions. In this, scenario of dialect identification for wake word detection Conv1D technique suits well. We convert every MFCCs into 1 dimensional array and give it as a input to first convolution layer. Then train the model using TensorFlow with Keras technique.

Later, we evaluate the trained model for prediction where the system listens to the audio and classifies the input into one of the classes and writes the result in the csv file. The result reaches to its accuracy of whether specific audio contains the wake word or not.

Figure 1: Schematic Representation of the Proposed Work

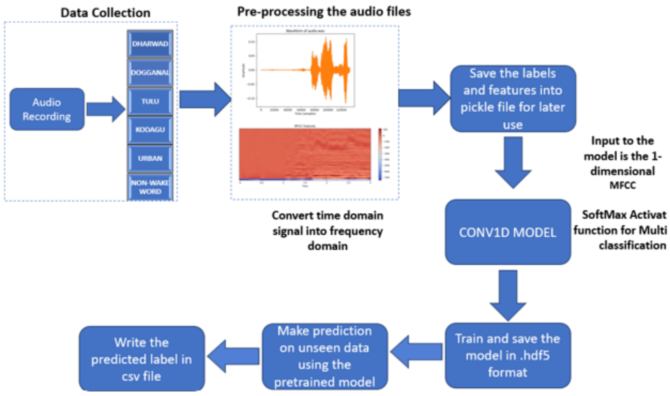

Data Collection
Audio Recording
DHARWAD
DOGGANAL
TULU
KODAGU
URBAN
NON-WAKE WORD
Pre-processing the audio files
Convert time domain signal into frequency domain
Save the labels and features into pickle file for later use
Input to the model is the 1-dimensional MFCC
CONV1D MODEL
SoftMax Activat function for Multi classification
Train and save the model in .hdf5 format
Make prediction on unseen data using the pretrained model
Write the predicted label in csv file

Figure 1 gives the step-by-step process of classification which are explained in detail in the following topics.

IV. METHOD

A. *DATA PREPARATION*

Data has been collected from 5 different regions of Karnataka, mainly from rural areas where it has its own distinct dialect of spoken Kannada. The regions include- from North Karnataka it is Dharwad and Dogganal regions. Form south it is Kodagu in Coorg, Tulu from coastal region and Urban Kannada from Bengaluru.

For samples, it includes the audio file which contains the recorded audio of wake words. The audio is recorded for short 3 seconds saying “Namaskara” in the respective regional dialects along with 2 or more words which go along with the greeting word which are specific to the region. And for non-wake word, data was collected from crowded places like restaurants and local marketplaces and made sure that wake words were not present these audio clips. Sounddevice is used for recording the sound/audio and creating a NumPy array and then Scipy.io.wav will save the NumPy array as an audio file in .wav format. Each dialect has 100 recording each with audio clips of both male and female voices with different age groups. Data Augmentation plays a crucial role in increasing the volume of the dataset by varying speeds from 0.7 to 1.4.

In audio data all the audio files are stored where each dialect has 100 recorded audio clips. Each time the audio is recorded it is saved under the unique file name which can be later helpful in conducting iteration for each of these files.

The recorded voice data collected during the data collection process can also be used for giving as the input for recording the wake word. The actual audios are recorded on field while conducting the survey about the dialect variations in the respective regions. The same audio is used for preparing the data set to get accurate results.

B. PREPROCESSING THE RAW AUDIO FILE:

While recording the audio, two parameters have been considered: -save path which is the empty directory where it must save all the audio files and n_times is the number of times the audio is recorded. Once the audio file is ready the sample rate must be initialized. [8] *Sample rate is the rate at which the sound is sampled per second. For audio signal sample rate to be considered is 44100 hertz*. For recording the wake word, minimum of 3 seconds is initialized for each dialect.

Librosa (python library) is extensively used for examining audio data. Audio preprocessing includes 3 major steps-First, raw audio file is loaded, next it must be converted into .wav file and third step is to extract the useful pattern from that spectrum. librosa. load takes the path of the NumPy array where the audio files are stored and then returns the NumPy array and sample rate of the audio file. librosa. display is an API for visualizing the spectrogram and it is built on the top of matplotlib.

Figure 2 shows the waveform plotted for one of the files in Non-Wake Word and Figure 3 shows the MFCC of the Waveform.

Figure 2: The waveform plotted for one of the files in Non-Wake Word

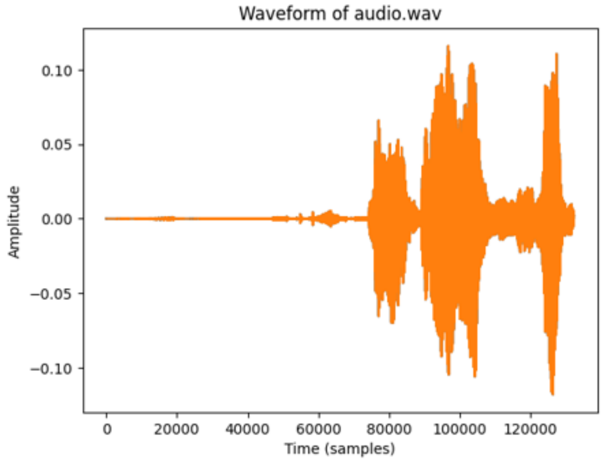
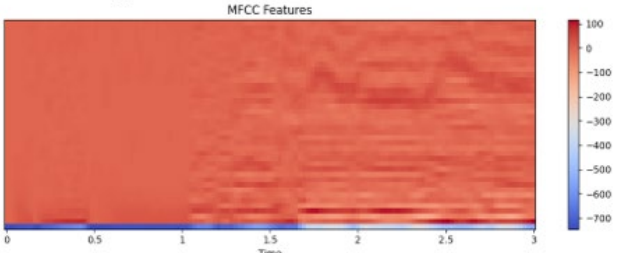


Figure 3: The MFCC of the Waveform

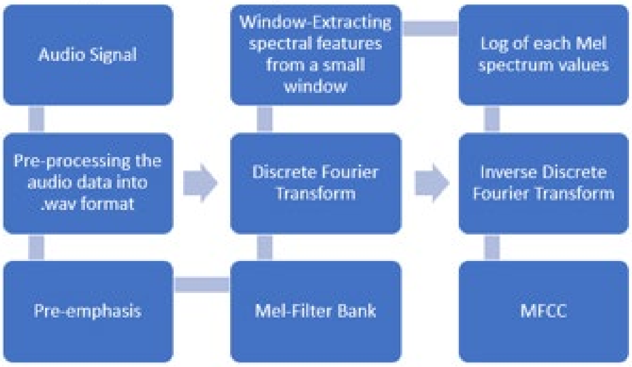


The same procedure is applied to all the audio files in the dataset. After preprocessing all the audio files, they are classified into respective labels. Signals are framed into 20-40 ms as there will be continuous variation in the audio signal. So, we need to consider a short time range where audio signal has less variation, or it is static.

After loading files using librosa, the next step of preprocessing phase is to extract useful pattern from the audio files. For this purpose, we have used MFCC-Mel Frequency Cepstral Coefficient as they yield better performance in identifying low frequency regions better than the high frequency regions. It can easily be applied to examine the patterns in lower frequencies and analyze the resonances created by the vocal tract. This leads us to spot only the linguistic position excluding the noise. Very small yet important variation in speech signal which is observed in dialects exists in the changing amplitude, pitch, speaker identity, duration and timber (which comes from the uniqueness in each speaker describing the quality of the tone). MFCC gives information about changing rates in spectral bands. Mainly the signal must shift from time domain to frequency domain which can be done using Fourier transform to examine the spectral and power components of the signal and encoding words into numbers which is a procedure for text vectorization-mapping words into vectors. It is usually applied to sentences. Figure 4 depicts the process of extracting MFCC for the Audio file.

MFCC is used to extract the pattern from the wave file. The procedure is followed for all the files in the dataset. We use the mean of MFCC to reduce the dimensionality of the data. It helps in removing the convolutional effects caused either with the recording device or with the participants vocal tract response. It may add additional features which can yield better results when given as the input to the model. Figure 4 depicts the process of extracting MFCC for the Audio file.

Figure 4: Extracting MFCC Feature Vectors


Audio Signal
Window-Extracting spectral features from a small window
Log of each Mel spectrum values
Pre-processing the audio data into .wav format
Discrete Fourier Transform
Inverse Discrete Fourier Transform
Pre-emphasis
Mel-Filter Bank
MFCC

We need to label the data. Label encoding technique is used to assign labels from 0-6 for all the audios to the respective dialects. It is considered as the Multiclass classification problem.

Moving to the next procedure of creating pandas' data frame of the final data and one more dictionary is created where this final data dataframe is saved. And this data frame can be easily accessed during the training phase. Data frame is saved in the csv format as a pickle file.

Pickle is mainly used to save the labelled dataset for applying it later for further experiments [2]. By doing this, we can transfer the pickled file to other users who are working on similar dataset, instead of transferring the entire dataset which may cause storage problems, for handling large dataset. Sometimes we may also loose the original dataset.

C.MODEL ARCHITECTURE

Based on the dataset the deep learning model which fits best is the CONV1D. Here the input shape is in the form of 1-dimensional array. MFCCs are fed as the input in the form of 1 dimensional array with 40 coefficients. The input shape is (40,1). The model has two convolution layers-the first layer has 64 filters with kernel size 3 and again the kernel is also 1 dimensional followed by ReLU activation function which converts the negative values to zero and max pooling of pool_size 2, and dropout of around 0.25 neurons. The second layer has 128 filters again followed by ReLU and max pooling and dropout of around 0.25 neurons. Next the output from max_pooling (2) is flattened and connected to dense layer with 512 neurons followed by drop out by 0.5 neurons and then with SoftMax activation function for the final dense layer with 6 neurons as per our labels (number of dialects). Each layer is also followed by the dropout layer.

Padding is mentioned in terms of 'same' which adds 0 at the two ends of the input array and it is used to get output image with the same dimension as the input shape.

The output shape is calculated as:

(n+2p-f/stride) +1- wherein

n is the input shape

p is the number of layers of zeros added at the boarder of the input data.

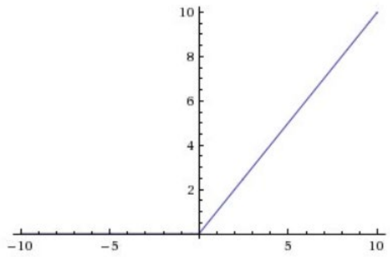
f is the filter size and

stride is taken as 1 in this customized model architecture.

ReLU-Rectified linear unit is an activation function used in convolution layers which returns 0 whenever it receives negative value in the feature map (obtained by the dot product and summation of input data and the filter). but for any positive values let it be x it returns the value back. It is represented as

f (x)= max (0, x)

It is graphically represented as:



Max_pooling layer- it is applied on convolution layer. The sliding window or Kernel slides over the feature map and takes the maximum value from the region. The size of the filter in pooling operation is smaller than the feature map. Here the stride is taken as 2 and based on this we can calculate the output shape.

Dropout layer- This can be applied booth at Convolutional layers as well as at the dense layers. But both as different effects.

1. At convolutional layer dropout are applied at a very low rate at about 0.2 which increases the performance and will not affect in extracting the important features in feature maps.
2. Next, they are applied at dense layer at the rate of 0.5 which increases the accuracy for the classification.

SoftMax Activation function-

Wherein-

$$S(y)_i = \frac{exp(y_i)}{\sum_{j=1}^{n} exp\ (y_j)}$$

$y_i = i\ th\ element\ of\ the\ input\ vctor$

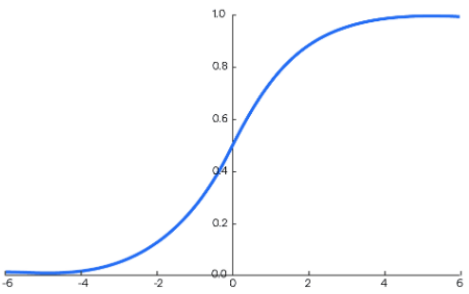
y = input vector of SoftMax function which consists of n elements with n classes.

$y_j = term\ for\ normalization\ which\ makes\ sure$ that the value of output ranges from 0 to 1.

$exp(y_i) = results\ in\ smaller\ value\ close\ to\ 0\ but\ not\ 0.$

SoftMax activation function is applied on the outputs from the dense layer mainly at the last layer of neural network for multiclass classification problem with n classes. I t returns the output vector in terms of probability scores. It highlights the maximum value and does not mention the lower values.

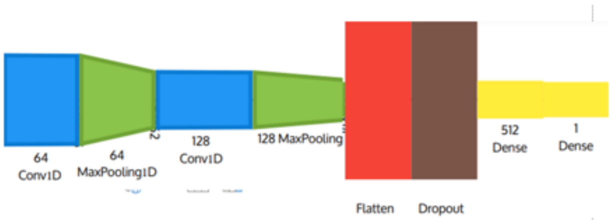
SoftMax is Graphically represented as:



The S-shaped function in the graph is in between 0 to 1 as 0.5 as its midpoint. Output is 1 for larger values and 0 for smaller or if the values are negative.

Figure 5 gives the visualization of the model using NetViz software tool.

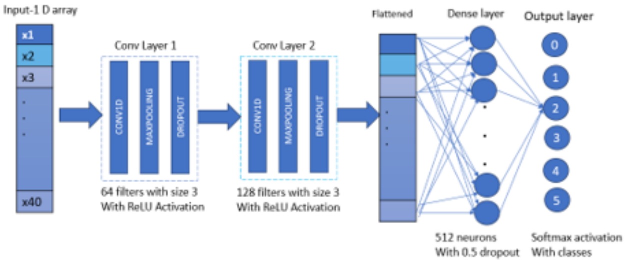
Figure 5: Visualization of the Model using NetViz



There is no such rule for the specific number of filters or layers, whichever best suits for the problem statement and gives the best predictions we can build the architecture.

Figure 6 is the architecture of the proposed model.

Figure 6: Architecture of customized Conv1d Model



In the below figure, we can see that we have around 683910 trainable parameters.

Trainable parameters are calculated only in convolution layers where filters are applied for getting the feature maps and at flatten layer.

Calculating the trainable parameters:

1.Input_shape is (X_train.shape[1],1)

2.CONV Layer: Here we have weight matrices and to calculate the learnable parameters we can use the following expression:

(Shape of the width of the filter * number of filters in previous layer + 1) * number of filters in current layer)).

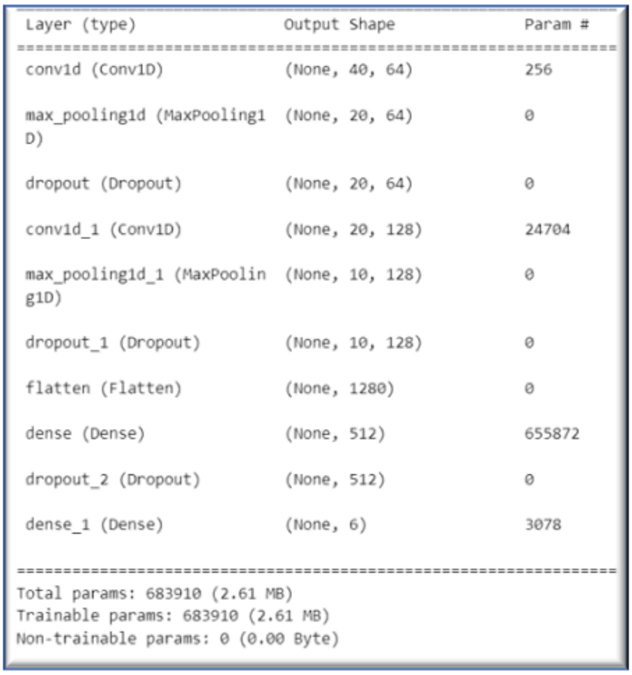
3.Pool Layer: Learning does not happen in this layer. So, learnable parameters are zero.

4.Fully connected layer:

(number of neurons in current layer * number of neurons in the previous layer) + number of neurons in current layer.

Figure 7 gives the summary of the model and figure 8 is the table depicting in detail the calculations for the trainable parameters.

Figure 7: Summary of the model



```
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 40, 64)            256

 max_pooling1d (MaxPooling1  (None, 20, 64)            0
 D)

 dropout (Dropout)           (None, 20, 64)            0

 conv1d_1 (Conv1D)           (None, 20, 128)           24704

 max_pooling1d_1 (MaxPoolin  (None, 10, 128)           0
 g1D)

 dropout_1 (Dropout)         (None, 10, 128)           0

 flatten (Flatten)           (None, 1280)              0

 dense (Dense)               (None, 512)               655872

 dropout_2 (Dropout)         (None, 512)               0

 dense_1 (Dense)             (None, 6)                 3078

=================================================================
Total params: 683910 (2.61 MB)
Trainable params: 683910 (2.61 MB)
Non-trainable params: 0 (0.00 Byte)
```

Figure 8: In detail calculations for the trainable parameters

Layers	Output shape	Calculation	Parameters
Conv1D(1st layer) with 64 filters	(None, 40, 64)	(3+1) * 64	256
MaxPooling1D	(None, 20, 64)	No learning	0
Dropout	(None, 20, 64)	No learning	0
Conv1D(2nd layer with 128 filters)	(None, 20, 128)	(3*64+1)*128	24704
MaxPooling1D	(None, 10, 128)	No learning	0
Dropout	(None, 10, 128)	No learning	0
Flatten	(None, 1280)	No learning	0
Dense	(None, 512)	(512*1280)+512	655,872
Dropout	(None, 512)	No learning	0
Dense	(None, 6)	(6*512)+6	3,078

C.MODEL COMPILATION AND TRAINING

Once we are set with the model architecture, we must compile the model. It takes two parameters: - Categorical cross-entropy loss and optimizer. 'Adam' is used as the optimizer which supports in regulating the learning rate for the entire training phase. Learning rate denotes the speed with which the ideal weights are calculated for the model.

The dataset is divided into 80:20 training and test data.

Since we are working with Multiclass classification problem, categorical cross-entropy is used to calculate the loss function.

Categorical cross-entropy formula is given by:

$$CE = -\sum_{i=1}^{i=N} y\_true_i \cdot log(y\_pred_i)$$

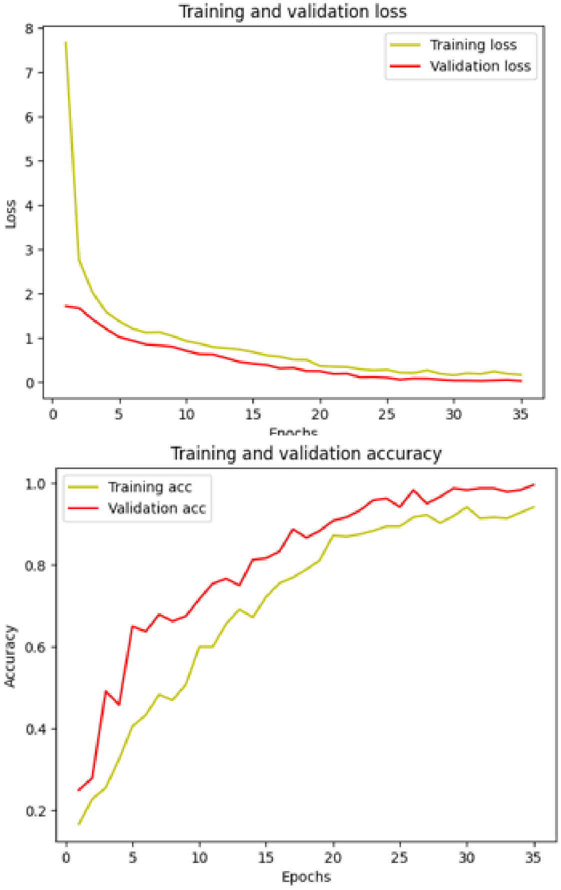
Also, we are using to_categorical() function to help the classification algorithm to understand the relationship between classes. It converts the vector of classes into binary class matrix.

Due to very small variations among the dialects of the same spoken language it is very challenging in correctly identifying the optimal weights for fine tuning it on specific points causing the variability in the data. The lower the score, the better the performance of the model. Its learning rate and convergence is faster compared to other loss functions like mean squared error. Since there are smaller variations which are difficult in rectifying weights must be adjusted respectively. Once these metrics are finalized, the model is fitted with the training data, and it is run for certain number of epochs and the trained model is saved in the. hdf5 file for later use. This way we can use this pre trained model on similar dataset.

The entire training dataset is divided into batches. The learning curves depicts the status of the training at each step during the epoch. Model performance is improved with the increase in the number of epochs .Accuracy increases with iteration over the training data. As accuracy increases the loss values gradually decrease. Accuracy had already reached 0.975 accuracy and it became stable by the end of the training(epoch

Both the graphs for loss and accuracy depicts the model is good for the dialect dataset.

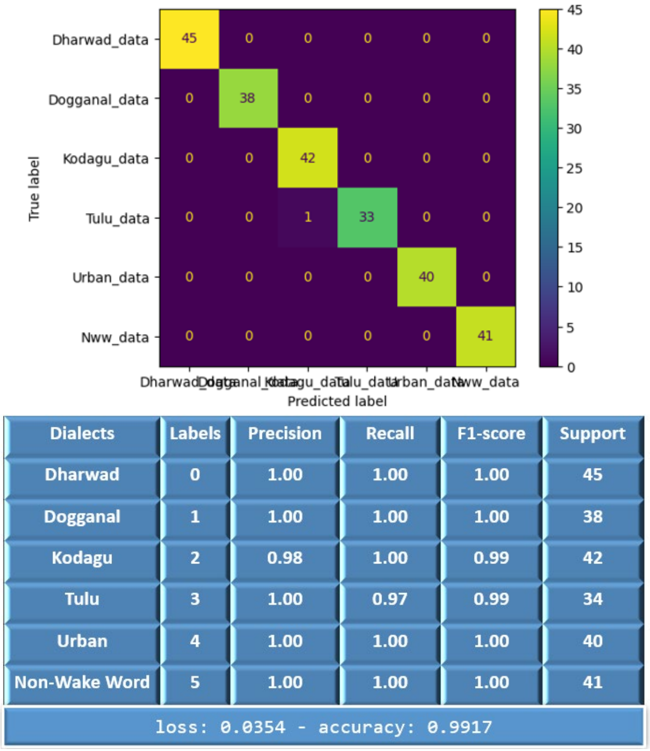
Figure 9: Depicts the loss and accuracy curves



The Performance of the model shows better results in both loss and accuracy measures. Both training and validation loss decreases and become or reach stability at much earlier step during the epoch which reflects the fact that the model fits well with the dataset.

Model Evaluation is done using classification report determining how the audio files are correctly classified into respective labels.
Figure below is the Confusion Matrix with Model Classification report with loss and accuracy checks.

Figure 10: Confusion Matrix with Model Classification report



Dialects	Labels	Precision	Recall	F1-score	Support
Dharwad	0	1.00	1.00	1.00	45
Dogganal	1	1.00	1.00	1.00	38
Kodagu	2	0.98	1.00	0.99	42
Tulu	3	1.00	0.97	0.99	34
Urban	4	1.00	1.00	1.00	40
Non-Wake Word	5	1.00	1.00	1.00	41

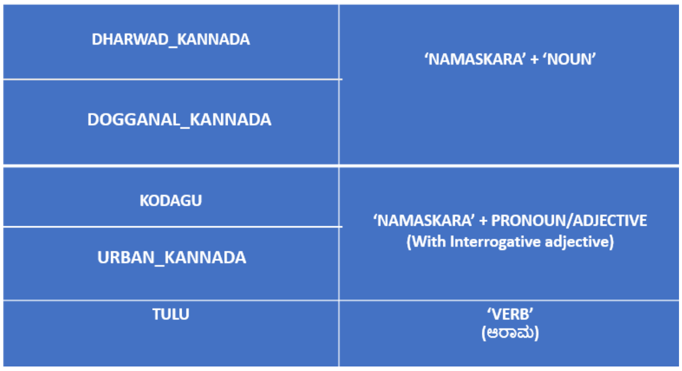
```
loss: 0.0354 - accuracy: 0.9917
```

V. Results

The output of the proposed Conv1d model is one of the dialects(class) identified. Given the input either through audio file or through saying/uttering the words for 3 seconds the model predicts one of the classes out of 6 classes of dialects which the input belongs to and writs the predicted output to the csv file. Accuracy of the prediction is 0.9917.
Another way is to record a new audio which is unseen in both training and testing data and use the pretrained model-model which is saved as .hdf5 and the prediction is the one of the classes which is saved in csv file.

The dataset for our problem statement reflects diversity among dialects. Each dialect has its own phrase for greeting. They are varied in-terms of the use of noun, verb and pronoun in their respective dialects.
Figure below shows the breakdown of Wake word/Sequence of words or phrase and their part of speech with their position. The work which proposed in [8],[18] and [2] are based on keywords of maximum length of two words which are very short and no variation is present among the multiple keywords as compared to the dataset for this paper with variation present in each category and has more than two words which can be considered as phrase but not the sentence because there is no sequence in these phrases which contradicts the use of RNNs for the purpose of our experiment.

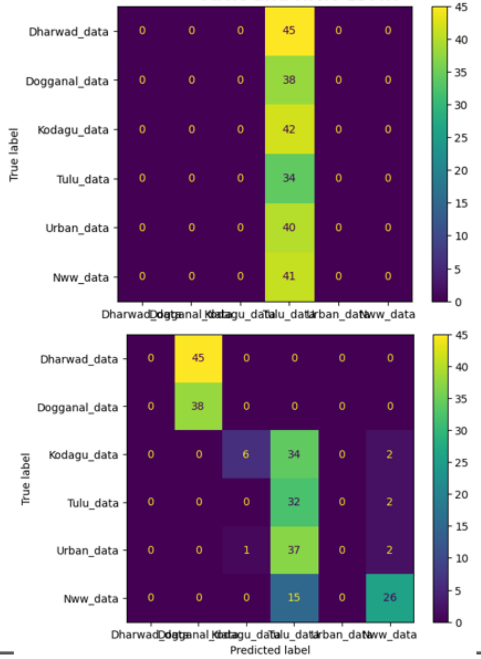
Figure 11:Breakdown of Wake word/Sequence of words or phrase and their part of speech with their position



DHARWAD_KANNADA	'NAMASKARA' + 'NOUN'
DOGGANAL_KANNADA	
KODAGU	'NAMASKARA' + PRONOUN/ADJECTIVE (With Interrogative adjective)
URBAN_KANNADA	
TULU	'VERB' (ಆರಾಮ)

Comparative analysis is carried out with ANNs and RNN model with Conv1d to validate the accuracy of the proposed Conv1d model on the dialect dataset. The ANNs performance was deficient as its classification was fragmentary which was concerning in terms of predictions.

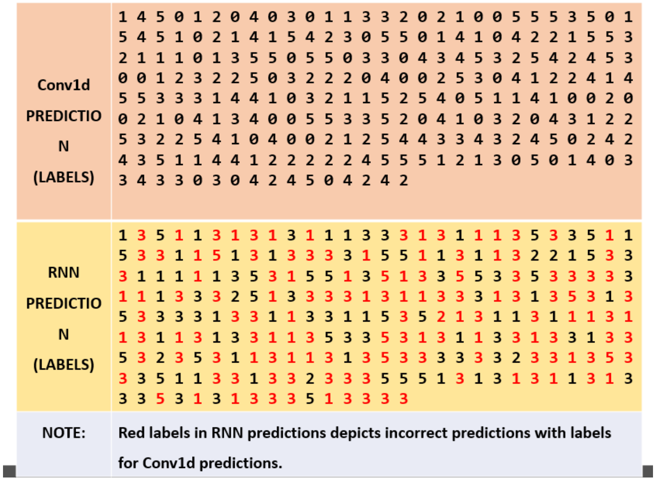
Figure 12: Confusion Matrix representing predictions on Dialect Dataset with ANN and RNN-LSTM



The confusion matrix represents the deficiency in predictions with both models. We can closely compare the RNN-LSTM model with Conv1d.
As mentioned in the breakdown words into part of speech it can be perceived that the dialects of Dharwad and Dogganal are similar but not the same. Here since the position of words in the phrase into their part of speech RNN model has misclassified or made a faulty prediction with reference to labels [0,1] and [2,4].
Comparing the Predicted labels can evidently show the progressing performance of Conv1d over RNN-LSTM model.

The below figure shows the labels which are wrongly predicted by RNN-LSTM model.

Figure 13: Comparative evaluation of predicted labels



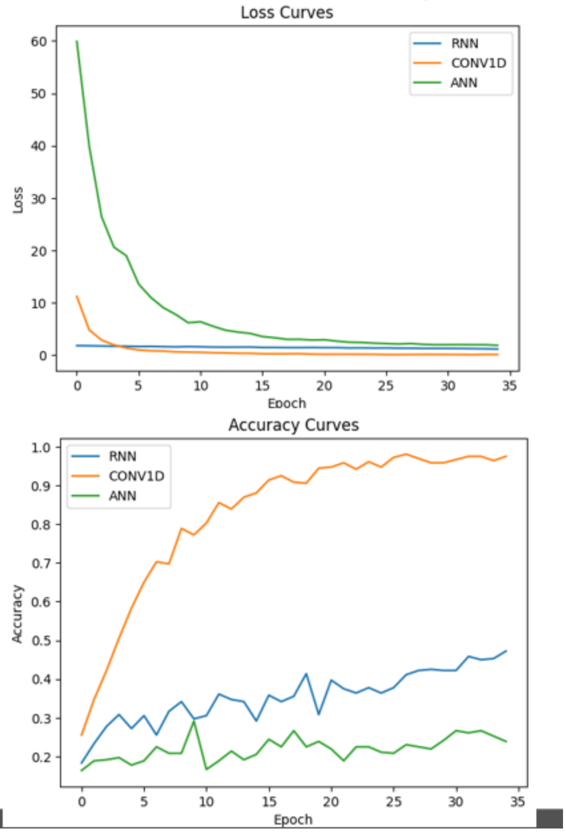
Conv1d
PREDICTIO
N
(LABELS)
	1 4 5 0 1 2 0 4 0 3 0 1 1 3 3 2 0 2 1 0 0 5 5 5 3 5 0 1
	5 4 5 1 0 2 1 4 1 5 4 2 3 0 5 5 0 1 4 1 0 4 2 2 1 5 5 3
	2 1 1 1 0 1 3 5 5 0 5 5 0 3 3 0 4 3 4 5 3 2 5 4 2 4 5 3
	0 0 1 2 3 2 2 5 0 3 2 2 2 0 4 0 0 2 5 3 0 4 1 2 2 4 1 4
	5 5 3 3 3 1 4 4 1 0 3 2 1 1 5 2 5 4 0 5 1 1 4 1 0 0 2 0
	0 2 1 0 4 1 3 4 0 0 5 5 3 3 5 2 0 4 1 0 3 2 0 4 3 1 2 2
	5 3 2 2 5 4 1 0 4 0 0 2 1 2 5 4 4 3 3 4 3 2 4 5 0 2 4 2
	4 3 5 1 1 4 4 1 2 2 2 2 2 4 5 5 5 1 2 1 3 0 5 0 1 4 0 3
	3 4 3 3 0 3 0 4 2 4 5 0 4 2 4 2

RNN
PREDICTIO
N
(LABELS)
	1 3 5 1 1 3 1 3 1 3 1 1 1 3 3 3 1 3 1 1 1 3 5 3 3 5 1 1
	5 3 3 1 1 5 1 3 1 3 3 3 3 1 5 5 1 1 3 1 1 3 2 2 1 5 3 3
	3 1 1 1 1 1 3 5 3 1 5 5 1 3 5 1 3 3 5 5 3 3 5 3 3 3 3 3
	1 1 1 3 3 3 2 5 1 3 3 3 3 1 3 1 1 3 3 3 1 3 1 3 5 3 1 3
	5 3 3 3 3 1 3 3 1 1 3 3 1 1 5 3 5 2 1 3 1 1 3 1 1 1 3 1
	1 3 1 1 3 1 3 3 1 1 3 5 3 3 5 3 1 3 1 1 3 3 1 3 3 1 3 3
	5 3 2 3 5 3 1 1 3 1 1 3 1 3 5 3 3 3 3 3 3 2 3 3 1 3 5 3
	3 3 5 1 1 3 3 1 3 3 2 3 3 3 5 5 5 1 3 1 3 1 3 1 1 3 1 3
	3 3 5 3 1 3 1 3 3 3 5 1 3 3 3 3

NOTE:	Red labels in RNN predictions depicts incorrect predictions with labels for Conv1d predictions.

The overall implementation of Conv1d model yield noticeable results over other two models which accomplishes the task.

Figure 14: Exhibits the loss and accuracy measures



The above figure clearly identifies the performance metrics of the two models over proposed Conv1d model for the dialect dataset where-in Conv1d shows satisfactory results in correctly identifying the wake word to the correct labels.

VI. CONCLUSION

Farmers in rural areas still face a huge challenge in adopting new technological trends for voice-based apps. One of the major obstacles arises with communicating and comprehending the information in the standard language used by the mobile apps. To tackle this issue apps should be more friendly and approachable and above all, farmers should feel connected to the app which magnifies the need of communication in their regional dialects. Dialect Identification task is slowly picking up the pace with advancement in AI and Data Science. In this paper for our research work, we have proposed a simple wake word detection system built on five major dialects of Karnataka using TensorFlow and Keras and CNN which works efficiently on short word ranges over other deep learning techniques Model architecture is well built to provide error free predictions. For further work TensorFlow Quantization API can be more flexible in deployment of the model. Quantization is a method which is created to make models smaller, lesser dependence on the settings in the environment where they will be deployed and are faster.

REFERENCES

[1] H. C. Das and U. Bhattacharjee, "Assamese Dialect Identification using," in IEEE World Conference on Applied Intelligence and Computing (AIC, 2022.

[2] K. Supriya, A. Divya, B. Vinodkumar and G. R. Sai, "Trigger Word Recognition using LSTM," June-2020.

[3] H. Wang, M. Cheng, Q. Fu and M. Li, "THE DKU POST-CHALLENGE AUDIO-VISUAL WAKE WORD SPOTTING SYSTEM," arXiv, 4 March 2023.

[4] M. Tzudir, M. Bhattacharjee, P. Sarmah and S. R. M. Prasanna, "Low-Resource Dialect Identification in Ao Using Noise Robust Mean Hilbert Envelope Coefficients," 2022 National Conference on Communications (NCC), 2022.

[5] J. Lee, K. Kim and M. Chung, "Korean Dialect Identification Based on Intonation Modeling," in 24th Conference of the Oriental COCOSDA International Committee for the Co-ordination and Standardisation of Speech Databases and Assessment Techniques (O-COCOSDA),, Singapore, 2021.

[6] Lokitha, Iswarya, Archana and A. Kumar, "Smart Voice Assistance for Speech disabled and Paralyzed People," in International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, 2022.

[7] Y. Wang, H. Lv, D. Povey, L. Xie and S. Khudanpur, "Wake Word Detection with Alignment-Free Lattice-Free MMI," INTERSPEECH 2020, 25-29 October 2020.

[8] T.-H. Tsai and P.-C. Hao, "Customized Wake-Up Word with Key Word Spotting using Convolutional Neural Network," in IEEE, 2019.

[9] V. Ribeiro, Y. Huang, Y. Shangguan, Z. Yang, L. Wan and M. Sun, "Handling the Alignment for Wake Word Detection:A Comparison Between Alignment-Based, Alignment-Free and Hybrid Approaches," in Accepted to Interspeech 2023, 2023.

[10] M. Tzudir, S. Baghel, P. Sarmah and S. R. M. Prasanna, "Analyzing RMFCC Feature for Dialect Identification in Ao, an Under-Resourced Language," 2022.

[11] N. C. Diaz, N. Sasaki, T. W, Tsusaka and S. Szabo, "Factors affecting farmers' willingness to adopt a mobile app in the marketing of bamboo products," Science Direct, vol. 11, 2021.

[12] R. K. Raman, D. K. Singh, U. Kumar and S. Sarkar, "Agricultural Mobile Apps for Transformation of Indian Farming," ReserachGate, vol. 07, no. 04, April 2021.

[13] S. G. Mane and K. R.V, "Design and Development of Mobile App for Farmers," International Journal of Trend in Scientific Research and Development (IJTSRD), pp. 179-182, 2019.

[14] R. Kumar, "Farmers' Use of the Mobile Phone for Accessing Agricultural Information in Haryana: An Analytical Study," Open Information Science, 7 April 2023.

[15] K. D. M, and S. K. R. M, "FARMER'S ASSISTANT using AI Voice Bot," 2021 3rd International Conference on Signal Processing and Communication (ICPSC), pp. 527-531, 2021.

[16] Z. Dan, Y. Zhao, X. Bi and Q. Ji, "Multi-Task Transformer with Adaptive Cross-Entropy Loss for Multi-Dialect Speech Recognition," MDPI, 8 OCTOBER 2022.

[17] R. Z. Qiuchen Yu, "Wake Word Detection Model Based on Res2Net," JOURNAL OF LATEX CLASS FILES, vol. 10, no. 10, 30 September 2022.

[18] Y. Wang, H. Lv, D. Povey, L. X. and S. Khudanpur, "WAKE WORD DETECTION WITH STREAMING TRANSFORMERS," in IEEE, Toronto, Canada, 2021.

[19] T.Cynthia and C. Newton, "Voice Based Answering Technique for Farmers in Mobile Cloud Computing," International Journal of Scientific Research in Computer Science Applications and Management Studies, vol. 7, no. 3, 13 JULY 2020.

[20] M.L.Dhore and M. Dhakate, "Insurance Value Chain Chatbot for Farmers," in ResearchGate, 2022.

[21] M. Ali, " Mobile Technology Used by echnology Use by Rural Farmers and Herders," Walden University, 2021.

[22] S. Sarkar, B. Kumar and S. Kumar, "Mobile Applications for Indian Agriculture and Allied Sector:An Extended Arm for Farmers," International Journal of Current Microbiology and Applied Sciences, vol. 10, no. 3, 2021.

[23] D. Landmann, C. Lagerkvist and V. Otter, "Determinants of Small Scale Farmers' Intention to Use Smartphones for Generating Agricultural Knowledge in Developing Countries: Evidence from Rural India," The European Journal of Development Research, 10 August 2020.

[24] C. R. Kinkar and Y. K. Jain, "AN OVERVIEW OF MODERN ERA SPEECH RECOGNITION MODEL," International Journal of Creative Research Thoughts (IJCRT), vol. 9, no. 9, September 2021.