VOICE INTELLIGENCE BASED WAKE WORD DETECTION OF REGIONAL DIALECTS

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

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

Abstract— Technology is considered as a contributor to economic growth only when it reaches rural/remote areas where it can be effectively applied by farmers for their agricultural needs. Analyzing the factors affecting the adoption of technology by farmers plays a crucial role in updating the technology. 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 Even if the farmers manage to utilize mobile phones to access the information, they may lack this ability to comprehend when it is not in their native language. If Technology must reach the rural farmers, then the fact about their connection towards the voice-based apps in their own spoken language/Dialect cannot be left unnoticed. This paper presents the research work in developing the wake word detection system for 4 of the major dialects in Karnataka. The customized wake word system is designed using CNN, TensorFlow along with Keras.

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

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

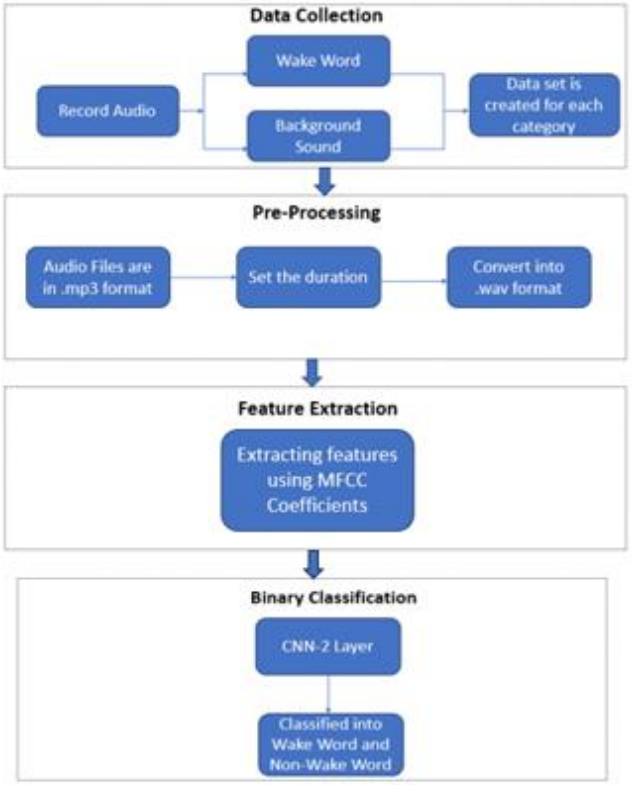
and have layer dependencies. It resulted in outperforming CNN by 25% in the rate for false rejection and sustains the linear complexity for the segment length. They have also mentioned using tensors as it may be more efficient for short range word sequences in which instead of considering the entire 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

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 for at least 100-200 times and recording the audio which do not contain the wake word for the same number. Next, we preprocess the audio file using MFCC through one hot encoding as it is a binary classification problem and deploy it to train the model using TensorFlow with Keras technique. Later, we evaluate the trained model for prediction where the system continuously listens to the audio and responds only when a wake word is detected. The result reaches to its accuracy of whether specific audio contains the wake word or not.

Figure1: Schematic Diagram of the Proposed Work



IV. METHOD

A. *DATA PREPARATION*

For wake word detection system, first we need to label the data. '1' refers to the audio file containing the wake word and '0' for the audio file that does not contain the audio file. It is considered as the binary classification problem. 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.

For samples, it includes the audio file which contains the recorded audio of wake words from 4 different regions from Karnataka namely- Chiturdurga, Coorg, Mysore and Dharwad. The audio is recorded for short 3 seconds saying "Namaskara" in their respective regional dialect. And for background sound data was collected from the same place from where the wake word data was collected having the same dialect but was randomly collected from crowded places like restaurants and local marketplaces and made sure that wake words were not present the background sounds. And it is also saved in the empty directory for background sound.

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. Both the audio data and background_sound is recorded 100 times. In audio data all the audio files are stored where each dialect is tested 100 times. 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:

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. 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 as '1' or '0'. When iteration is done over every audio file then the respective labels are assigned to those files. 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 2: Waveform of background_sound/1.wav

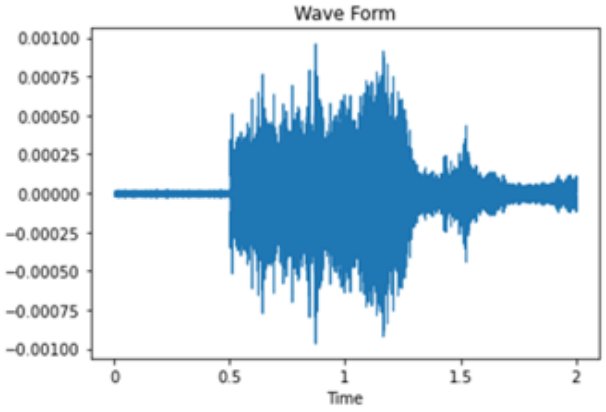


Figure 3: MFCC of the Waveform from Figure 2

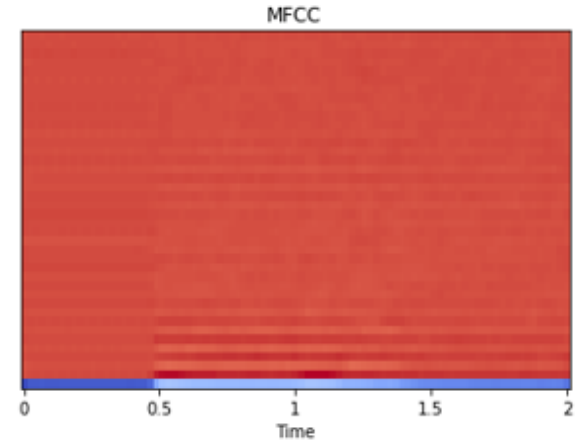


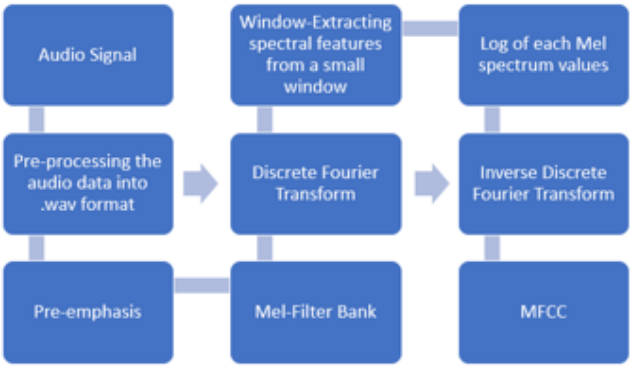
Figure 2 shows the waveform plotted for one of the files in background_sound and Figure 3 shows the MFCC of the Waveform. MFCC is used to extract the pattern from the wave file. The procedure is followed for all the files in the dataset. We use the mean of MFCC to reduce the dimensionality of the data. 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.

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. Dataframe is saved in the csv format as a pickle file.

Pickle is mainly used to save the machine learning model for applying it later for further experiments [2]. By doing this, the retraining phase is minimized by loading the pre trained model.

Figure 4: Extracting MFCC Feature Vectors



C.TRAINING THE MODEL

Here in this phase first we need to load the csv file created in the previous phase which is pickled file. And then extract the parameter which in our work it is x and y parameters.
where x is the sound data and y value checks if the sound contains wake word or not. One more procedure must be applied for y values especially for categorical data type which is one hot encoding. Many machine learning models do not work with categorical data. In order to make it functional it must be first converted into numerical data []. Issue with label encoding (which categorizes the entire data into either 0 or 1) is it may add bias to the model as the model can give high preference to those values which is, labeled as 1 than the values labelled as 0. But both the values have equal importance in the dataset. To prevail such conditions one hot encoding technique is used.

D.MODEL ARCHITECTURE

We need to create a dense layer of 256 neurons with Relu Activation. It is often used in deep learning models. In case of negative input, it produces zero but in case of positive.
input it returns z as the linear function. After that we use dropout layer to eliminate 50% of neurons randomly to reduce the complications of model overfitting. Since it is a binary classification problem, we have only two neurons in the output which are activated using SoftMax activation function. Output is in the form of probabilities. Prediction is made based on determining the label having highest probability.
Figure 5 shows the model architecture of a TensorFlow Keras Model built in the code for the wake word system. It specifies the following information:

1. Different layers and their order in the model.
2. The shape of the output at each layer.
3. The number of parameters considered in each layer.

Initially the parameters considered was 10496 but, in the end, it is reduced to 514. Only relevant/prominent parameters are carefully considered for the output neurons.
Both Tensorflow and CNN Architecture is depicted.

Figure 5: TensorFlow Keras Model

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 256)               10496

activation (Activation)      (None, 256)               0

dropout (Dropout)            (None, 256)               0

dense_1 (Dense)              (None, 256)               65792

activation_1 (Activation)    (None, 256)               0

dropout_1 (Dropout)          (None, 256)               0

dense_2 (Dense)              (None, 2)                 514

=================================================================
Total params: 76,802
Trainable params: 76,802
Non-trainable params: 0
```

From Figure 6-Input shape is processed by first layer CNN, using 32 filters with kernel size as (3,3) and relu as an activation function, Maxpooling1d=2,2padding=same is applied so that the input size and output size are the same. The second layer has 64 filters with the same parameters. Third layer is also of 64 filters keeping all other parameters same with dropout at 0.2. to avoid the risk of overfitting. Next step is to flatten the layer with softmax activation fuction at the output. to avoid the risk of overfitting.
Once we are set with the model architecture, we must compile the model. It takes two parameters: -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.
Since we are working with classification model, cross-entropy is used to calculate the loss function. It calculates the accuracy of the model. 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 2000 epochs and the trained model is saved for later use. Each epoch consists of many weight updates. To get optimal results in the learning rate Gradient descent step is followed which is an iterative process. Here, the dataset is passed multiple times so that the weights get updated in different steps for better learning. 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. While

running through epochs from 2 to 9 the loss value was very high, and accuracy was just at 0.5. When it came to 619 the loss value had dropped to 0.05 and accuracy had already reached 0.975 accuracy and it became stable and slowly reached 100% by the end of the training.

Figure 6 shows the CNN Model Architecture of the system.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 42, 32)	128
conv1d_1 (Conv1D)	(None, 42, 64)	6208
dropout (Dropout)	(None, 42, 64)	0
max_pooling1d (MaxPooling1D)	(None, 21, 64)	0
conv1d_2 (Conv1D)	(None, 21, 64)	12352
activation (Activation)	(None, 21, 64)	0
conv1d_3 (Conv1D)	(None, 21, 64)	12352
activation_1 (Activation)	(None, 21, 64)	0
flatten (Flatten)	(None, 1344)	0
dense (Dense)	(None, 0)	0
activation_2 (Activation)	(None, 0)	0

Total params: 31,040
Trainable params: 31,040
Non-trainable params: 0

The role of epoch is very significant in Deep learning models as it controls the accuracy and avoids overfitting.

Model Evaluation is done using classification report determining how the audio files are correctly classified into respective labels.

Figure below is the confusion matrix for predicted label.

Figure 7: Confusion Matrix

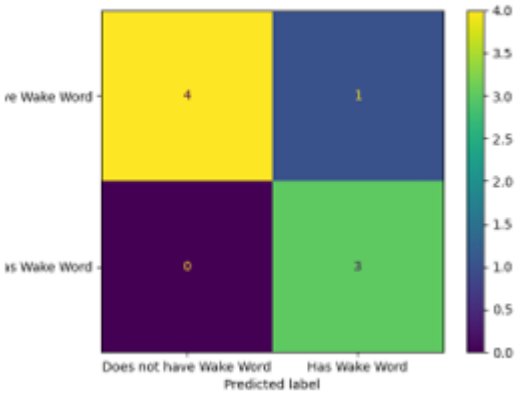
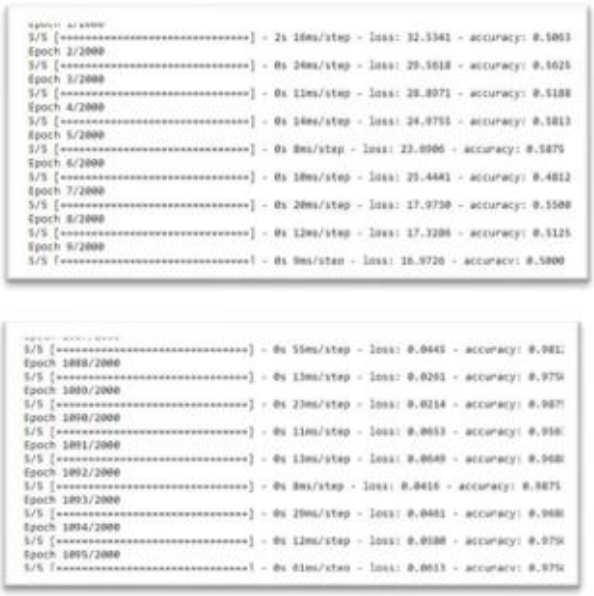


Figure 8 below depicts the loss and accuracy measures as the iteration progresses.

Figure 8: Running through Epochs at Different Stages



V. RESULTS

Now we are going to test our model's performance. Here the trained model is used to make predictions where it takes the audio as input to the loaded model for detecting the wake word. Here we have specified the confidence interval to be 0.99. It means the system will detect the wake word only when it is 99% confident.

Figure 7 indicates an important observation about the confidense rate for wake word not detected, which decreases when similar word close to the WakeWord is uttered. But, the tensorFlow model is very robust as it will not consider it has the Wake Word and avoids false predictions.

Figure 9: Wake Word Detection

```
1/1 [==============================] - 0s 31ms/step
Wake Word NOT Detected
Confidence: [0.2913584]
Say Now:
1/1 [==============================] - 0s 62ms/step
Wake Word NOT Detected
Confidence: [0.48000938]
Say Now:
1/1 [==============================] - 0s 47ms/step
Wake Word Detected for (0)
Confidence: [1.]
```

Figure 10: Wake Word Detected for the Input Audio

```
Say Now:
1/1 [==============================] - 0s 31ms/step
Wake Word Detected for (2)
Confidence: [1.]
Say Now:
1/1 [==============================] - 0s 31ms/step
Wake Word NOT Detected
Confidence: [0.682611]
```

Figure 11: Confidence rate decreases with familiar pronunciation of the word close to the Wake Word.

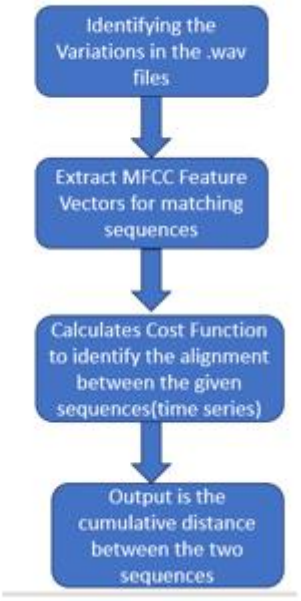
```
Confidence: [0.7978476]
Say Now:
1/1 [==============================] - 0s 67ms/step
Wake Word NOT Detected
Confidence: [0.70921725]
Say Now:
1/1 [==============================] - 0s 33ms/step
Wake Word NOT Detected
Confidence: [0.9918122]
Say Now:
1/1 [==============================] - 0s 46ms/step
Wake Word NOT Detected
Confidence: [0.9679671]
```

VI. IDENTIFYING THE VARIATION IN DIALECTS USING DTW METHOD

To identify the variations in the dialects, DTW method is used. DTW algorithm is applied to calculate the similarity between two sequences (temporal time series). Here the algorithm calculates the similar elements in the two sequences. Dialects have smaller variations in terms of utterance and pitch. To identify these variations, we need to calculate the similarity/cost matrix. If the sequences have higher similarity, then the values will be closer to zero.

For the experiment we have taken around 6 dialects for Kannada language spoken in 6 different regions and compared with the Normal spoken Kannada. To understand how the spoken language changes in every region, these 6 dialects are compared with one another. Figure 12 shows the process of applying DTW algorithm on audio files with MFCC.

Figure 12: Process Chart for DTW for Audio files



DTW yields better results over Euclidean distance, wherein eucledian distance fails to identify the smaller variations in two time series in which both series look very similar visually. Because in eucledian distance the amplitude is compared between the time series at the given time T.

Whereas DTW algorithm works by comparing the amplitude of the first sequence(signal) at time T with the second sequence(signal) at T+1 and T-1 to avoid giving lower score for sequences which have similar shapes and altered phases.

DTW Algorithm works in the following steps.

1.An empty matrix C is created for cost calculation. x and y are depicted with respective amplitudes of the two sequences compared.

2.Cost is calculated beginning from left and base at the corners.

$$C(i,j)=\text{mod}((M(i)-N(j)) + \min(C(i-1,j-1),C(i,j-1),C(i-1,j))$$

C is the Cost Matrix,

i is the iterate state of series M.

j is the iterate state of series N.

3.Identification of the Wrapping pathway. It begins from the upper right corner and travels to the base over to the left. Identification of the track is done with the neighbor whose value is minimum. The process continues till the base at the left and all the these values or travel series is denoted by d.

4.Calculation of the final distance. Normalized Distance(time) is given by:

$D=\sum d(i) \div \sum k$ where $\sum$ ranges from i=1 to k.

K denotes the length of d.

The below table gives the comparative reading or the similarity values of 6 different dialects against the Normal Language.

Normal Kannada is the spoken language which is communicated among greater audience, and it is standarised and not specific to any region. Table 2 identifies how the variation is in regard to normal spoken language and the dialects specific to the regional boundaries.

.

Table 2: DTW scores to Identify the similarity among the two speakers

SPEAKER 1	SPEAKER 2	Normalized distance between two sounds
Normal Kannada	Tulu	3575.68
Normal Kannada	North Karnataka	3835.55
Normal Kannada	Dogganal	3555.41
Normal Kannada	Mangaluru	3054.59
Normal Kannada	Davanagere	3458.83
Normal Kannada	Coorg	7987.05

In Table 3 we can clearly see the similarities among the dialects. Lowest similarity score or higher similarity is found in the dilaects among the regions of Dogganal and Davanagere which are closer w.r.t geographical location which is approximately 70.9km.

Table 3: Depicts the similarity among the Dialects.

SPEAKER 1	SPEAKER 2	Normalized distance between two sounds
Tulu	Mangaluru	1862.67
Dogganal	Davanagere	851.83
Dogganal	North Karnataka	1006.19
Coorg	Tulu	7072.97
North Karnataka	Davanagere	454.50
Davanagere	Mangaluru	1684.81

VII. 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 four major dialects of Karnataka using TensorFlow and Keras and CNN which works efficiently on short word ranges over other deep learning techniques and works well for post deployment on other devices. Model architecture is well built to provide error free predictions. To identify the variations in the dialects, DTW method is used. 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.