A Comparison of Speech-based Grocery Item Classification for Visually Impaired People in Indian Native Languages

Abstract

The ubiquitous usage of online shopping over time has increased its prominence. However, the lack of user-friendly UI for visually impaired people hinders the e-commerce shopping facility. Although assistive technology has become more available for the visually impaired, some features like a smooth UI and descriptions of products are not compatible with such. We propose a total voice-based implementation of online shopping using Indian languages like Hindi and Tamil using CNN (convolutional neural network model). Our proposed model is a speech-based classification model of grocery items using CNN. A comparison of our proposed CNN models (Hindi and Tamil) are made using three different activation functions, namely relu, sigmoid, and softplus functions. We found that our proposed CNN model using relu activation function gives the highest training accuracy of 100% for both Hindi and Tamil languages. Also, it gives the highest testing accuracy of 94.5% for Hindi and 81% for Tamil languages. A comparison is done between our proposed CNN model and the open-source speech-to-text Google API merged with TF-IDF using our dataset. Our proposed CNN model outperforms the Google API approach while translating the voice based requests of the grocery items.

Keywords: Speech-to-text, Activation function, CNN, MFCC, Google API, TF-IDF, Classification, Visually Impaired

1. Introduction

The World Health Organization (WHO) in 2020 found out that around 1.3 billion people in the world suffer from some sort of visual impairment and of which 36 million are blind (WHO, 2020). Most of the online shopping websites and applications are not designed with visually impaired friendliness in mind and because of it, these people would require constant assistance from another person when shopping online. The government of India in 2016 made around 100 government websites to be friendly enough for visually impaired people to use but other than these government websites none of the other websites took the effort to convert their websites for the visually impaired (Chand et. al., 2019). This hinders the accessibility of these facilities by visually impaired people which might be considered essential in this modern era. It is very clear that in the near future almost every task can be done easily through online applications or websites, and so it is important to make those facilities accessible to everyone despite their disabilities. Most of the websites share the same set of problems, they all mostly convey the information through visuals and do not have any audio alternatives. Recently the tech giants like Google, Apple and Microsoft have made an initiative to bring this issue into light and are slowly making their services accessible to more people with disabilities, but this fell short when it comes to e-commerce websites and applications as almost all use an input method which requires visuals and the output is also made visually (Chand et. al., 2019). A way to make these existing applications friendly towards the visually impaired people would be to make the application audio driven instead of just touch driven like asking for the user's request through audio whenever the application is opened, receiving the request through voice and giving the result along with an audio clip describing the product and price. Some websites such as Amazon have

the option to use audio as input but they mostly use English as the language of choice and not all are fluent in English.

We present a novel application where all visual information has an audio played along with it and the input of the user is taken through voice requests. By allowing voice based requests the users can explore all the features even if they are visually impaired essentially improving their experience in the application. We mainly focus on the use of Hindi and Tamil language to target a wider audience who are more comfortable with their native language rather than just English. We draw conclusions that a visually impaired person who knows the languages - Hindi, Tamil and English who can place orders without hesitation using gestures The application uses a CNN model with layered neural networks for speech recognition and speech identification. The accuracy for our proposed CNN model was tested on training and testing audios. A comparison of our CNN models (Hindi and Tamil) were made using three different activation functions, namely relu, sigmoid, and softplus functions. We found that our CNN model using relu activation function gave the highest training accuracy of 100% for both Hindi and Tamil. Also, it gave the highest testing accuracy of 94.5% for Hindi and 81% for Tamil. The accuracy of the API was tested using word error rate (WER) which is useful in determining the accuracy of how the models were able to properly identify the words correctly. Term frequency–inverse document frequency (TF-IDF) is used for identification of key shopping cart items which will be linked to the database and give a satisfactory response to the user. The application of these methods on native languages and the inclusion of them in an application to help the visually impaired adds to the novelty of the paper.

2. Literature Survey

In the paper by Singh (Singh et. al., 2019), the authors have translated English sentences to Hindi sentences using fuzzy logic and DNN. They obtained 1000 sentences in English and used a LSTM model to perform machine translation. They got an accuracy of 85% for short sentences and 60% for longer sentences. Jiang (Jiang and Wang, 2021) focused on making use of speech recognition for the Chinese language to build a recommendation system and an intelligent search feature. They found that in comparison to a traditional system, their system can integrate multiple heterogeneous data and learn the hidden features of users. Dharmale (Dharmale et. al., 2016) proposed a method to create a speech recognition system for SMS sending in English. They used HMM (Hidden Markov Model) and developed a system that produced an accuracy of 90% with the given speech inputs. The proposed application allows the user to send hand-free SMS accurately. Rallabhandy (Rallabhandy et. al., 2020) proposed a way for keyboard-less usage of E-commerce websites for visually impaired people by using Natural Language Processing for English language. With repeated tests they were able to achieve 85% accuracy. Adi (Adi et. al., 2019) investigated the negative transfer in the phonological features of interlanguage by second language learners by using Automatic Speech Recognition systems with different models, techniques and algorithms. Through this review they were able to conclude that the writing-based system of a language might affect the speech system on a knowledge-based level, resulting in a negative transfer phenomenon if the language learner has a different writing system of the native language speaker. Miao (Miao et. al., 2020) proposed a non-autoregressive end-to-end neural TTS model based on generative flow to achieve high-quality speech generation. They found that when compared to other TTS models like Tacotron2 and Fast Speech they were able to get higher quality audio close to the ground truth. Chand M (Chand et. al., 2019) introduced a website designed to help blind or visually impaired people to access

e-commerce portals requiring no manual assistance. They also developed a recommendation system using collaborative filtering algorithms based on what customers select. Upon interpreting the results, it was clear that mean effectiveness was around 70% and mean efficiency with respect to tasks was around 75%. Kumar and Aggarwal (Kumar and Aggarwal, 2020) introduces an acoustic modelling for automatic speech recognition (ASR) system for regional languages such as Hindi using a dataset based on Time Delay Neural Network (TDNN) combined with adaptation of i-vectors. An accuracy of 89.9% was reached using the adaptation of i-vectors on TDNN networks configured with a 5-fold perturbation of speed. Manasa (Manasa et. al., 2019) describes two different approaches for developing an acoustic model for speech recognition in Hindi. The first approach is an English acoustical model that has been adapted to Hindi and the second approach is training different types of acoustical models. The paper proposes a GMM-HMM model for acoustical and language modelling. They found that the word error rate for the trained acquisition model is 16.8 percent and for the cross language adapted model is 50 percent. In the work done by Kiran (Kiran et. al., 2017) they propose a Tamil ASR system which is used for voice dialling and for sending SMS. They highlighted the different algorithms used and the highest recognition accuracy was found in using the HMM model. They found that the HMM model carried out in this paper achieved training accuracy of 100% and testing accuracy of 98%.

3. Limitations of speech recognition systems in top companies

Countless systems have existed for voice and speech classification which are used all around the world through software and mobile applications. Presently, the most popular of those systems in smartphones would be Google. Some of the limitations of these systems are as follows (Kiran et. al., 2017) -

i. There is little to no flexibility for native language only speakers as these applications support only a few selected languages.

ii. The development of speech recognition systems in local languages have slowed down and has made it difficult for Indian smartphone users to use their native languages and rather use a foreign language inorder to make use of all the technological advancement.

Table.1. Limitations of the speech recognition systems in top companies (Chand et. al., 2019)

S.no	Company name	Vision is there	Impaired vision
1	Amazon	Considered	Not considered
2	Flipkart	Considered	Not considered
3	Alibaba	Considered	Not considered
4	Audi mart	Considered	Considered

Table.1 lists out the tech giants with ecommerce websites on whether they support visually impaired UI friendliness on their desktop version website. The tech giants use a recommendation algorithm “item based collaborative filtering” which suggests products based on the products that the customers have added to their cart or have previously bought. The efficiency of this recommendation algorithm is found to be high and helps to improve sales. However, it is also to be noted that visually impair people find the constant recommendation to fill up quite a lot of pages and therefore, multiple product recommendations, occasionally irrelevant, increase the complexity of making decisions for visually impaired people to place their order through the page (Chand et. al., 2019).

4. Proposed Method

This section is present to describe the various methods used in our proposed work. Section 4.1 gives detail on the datasets used and section 4.2 shines light on the working of our proposed CNN model that is used for Hindi and Tamil speech to text recognition. It also explains the importance of the metrics chosen as a means of comparison and describes TF-IDF in text classification to get only the important aspect from the speech. The existing works have only used HMM models and only on English language and Hindi languages. This method proposes a better performing CNN model in terms of accuracy and also targets Tamil language. Fig. 1 shows a brief description of the working of this application.

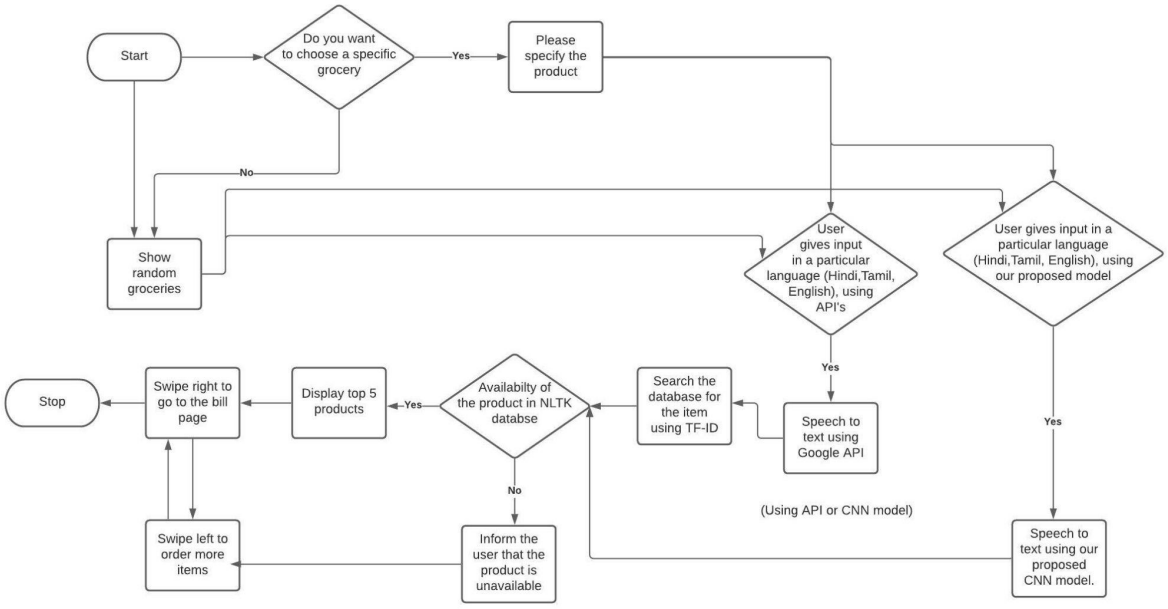

Fig.1. System design of the application

Once the user enters the application, the user is asked if he has a specific grocery request or if he is just browsing through groceries and given a choice of groceries to choose from, as soon as the user chooses. Using simple gestures the user can navigate through the pages. To swipe left or right, the user simply has to hold fingers on the screen and drag in the desired direction to navigate. Once the user swipes right, the user can use his/her voice as input. We are proposing two separate methods for processing the voice input. The first is using the Google Speech-to-Text API and classifying the recognized text using TF-IDF. The second is using a

proposed CNN model which takes the audio input and classifies it according to the grocery item. For example, if the user says “mujhe pista dal do”, which translates to “I want some pista”, the model will identify the grocery item as pista. These two cases are explained in detail below.

4.1. Dataset

A novel dataset was prepared for the purpose of this paper where five distinct speakers recorded their own voices. Each voice recording is of a person asking for a certain amount of a grocery item, for example “500-gram vasant pyaz daal do” in Hindi will correspond to the English grocery item “onion”. All the sentences cover almost all the grocery items and if two sentences share the same grocery item then they are framed differently. Hence various sentences were recorded for the purpose of the model and no two sentences were the same. A total of four hundred voice recordings were collected for Hindi (200 recordings) and Tamil (200 recordings) respectively for the purpose of building the model and for calculating the accuracy. For each grocery item, 5 audio recordings were prepared, out of which 4 were used for training and 1 was used for testing. It is similar for Tamil and this will be used for classification through our proposed CNN model.

A python package (Natural Language Toolkit WordNet database) was used for creating the database of all grocery items. We have taken 40 grocery items for training and made it into a 1-D array as shown in Fig.2 .

```
['orange', 'sweet potato', 'potato', 'pista', 'egg', 'water', 'bean', 'banana', 'lemon', 'pomegranate', 'drumstick', 'bottle go
urd', 'mushroom', 'cabbage', 'bread', 'mosambi', 'capsicum', 'berry', 'fig', 'watermelon', 'chilli', 'apple', 'rice', 'corriand
er ', 'brinjal', 'coconut', 'onion', 'beetroot', 'strawberry', 'bottle', 'cake', 'tomato', 'ball', 'juice', 'paper', 'cheese',
'mango', 'garlic', 'mango', 'pineapple']
```

Fig.2. Array of grocery items

4.2. Implementation of Proposed Model

CASE 1 (Using Google API):

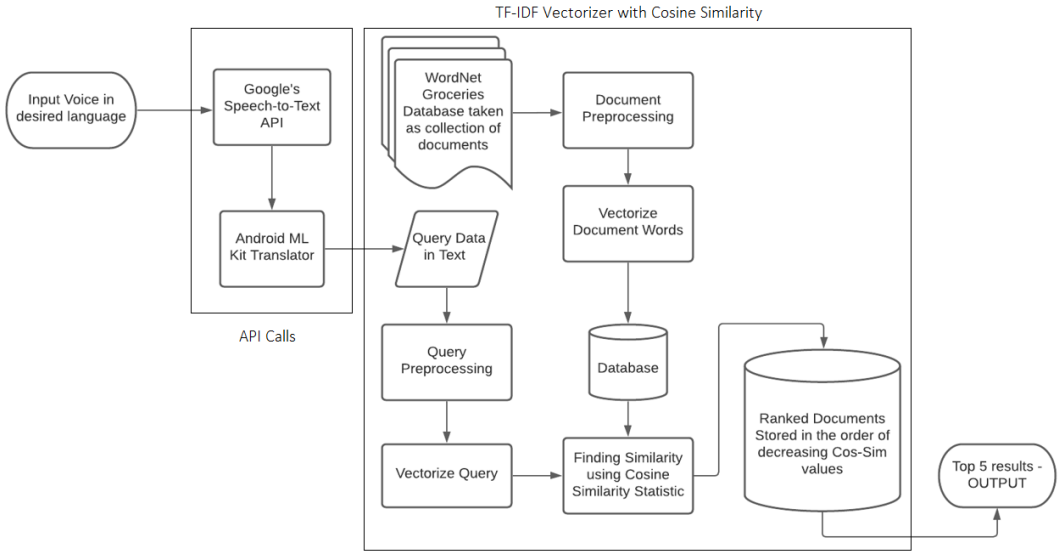

Fig.3. CASE 1 model architecture block diagram

1. Through the Google API which uses Connectionist Temporal Classification algorithm (CTC), the user specifies his grocery request through voice. For example, the user can say ‘mujhe 2-kilogram seb chahiye’ (I want 2 kilograms of apples) and the API will recognize this sentence and convert it into the text format.
2. The language - Tamil or Hindi gets converted to English. ML kit in Android is used for this purpose.
3. Using TF-IDF, the keyword (grocery item) is searched and matched.
4. TF-IDF is a technique to quantify a word in documents, it computes a weight to each word which signifies the importance of the word in the corpus.
5. When you search with a query, the database will find the relevance of the query with all of the documents, ranks them in the order of relevance and shows you the top k documents. This process is done using the vectorised form of query and documents.
6. Once the top k documents, in this proposed work, the top 5 documents are displayed and they are told out loud in the specific language.
7. The user has five options to choose from, which will be read out using Google’s Text-To-Speech API. The user will give a voice command regarding his preference among them.
8. The keyword will be matched in the database and will be confirmed with the user. Fig. 3 depicts a block diagram of the above process.

Text Classification for Google API

For the purpose of text classification, TF-IDF (Term Frequency - Inverse Document Frequency) is used.
It measures the uniqueness of a word by comparing the number of times a word appears in a document along with the number of documents the word appears in. It is calculated as shown in Equation(1).

$$TF - IDF = TF(t,d) * IDF(t) \qquad (1)$$

Where TF (t,d) stands for Term frequency where ‘t’ stands for the number of times the term appears in the document, ‘d’.

And IDF stands for Inverse Document frequency. It is calculated as shown in Equation (2).

$$IDF(t) = log(\frac{N}{DF+1}) \qquad (2)$$

Where N stands for the number of documents and DF stands for the document frequency of the term ‘t’.

- Using the NLTK WordNet database to serve as the primary general database, the transcribed text is retrieved as input string to extract the keywords.
- Next step is to remove the stop words present in the input string. Using the inflection string transformation library, it is mandatory to singularize the text.

- Using the corpus of grocery items as reference, the keywords which represent the items are matched. The grocery items extracted are segregated, mapped and returned as a key-value pair in the form of a dictionary for calculating associated costs.
- The initial step is to consider each grocery item as a distinct document, the following step is the TF scores are calculated for every based on the given word's frequency.
- We then multiply TF with IDF scores using sklearn's "TfIdfTransformer " and store it as a dataframe. The segregated input string is used as a query to retrieve related documents (food items).
- The TF-IDF scores are then calculated for the query term and stored as a dataframe. To rank the documents, the matching scores need to be calculated.
- The query is converted to a vector and then the pairwise cosine similarity score with each document is computed. Based on the cosine similarity scores, the maximum k documents are returned in descending order of scores to retrieve most relevant grocery items first.

In Fig.4, we can see the complete workings of the TF-IDF classification as a flowchart

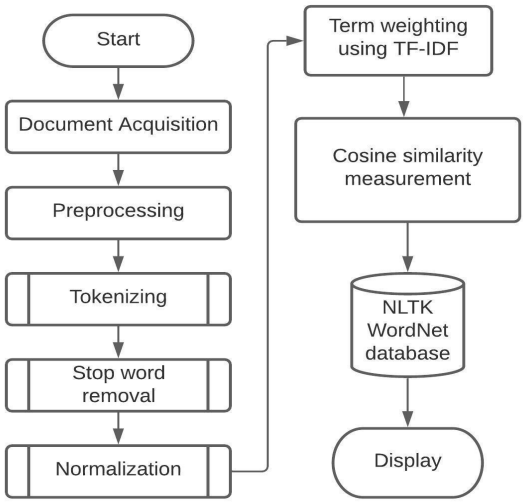

Fig.4. The workings of TF-IDF

CASE 2 (CNN Model with 1 input layer, 4 intermediate layers and 1 output layer):

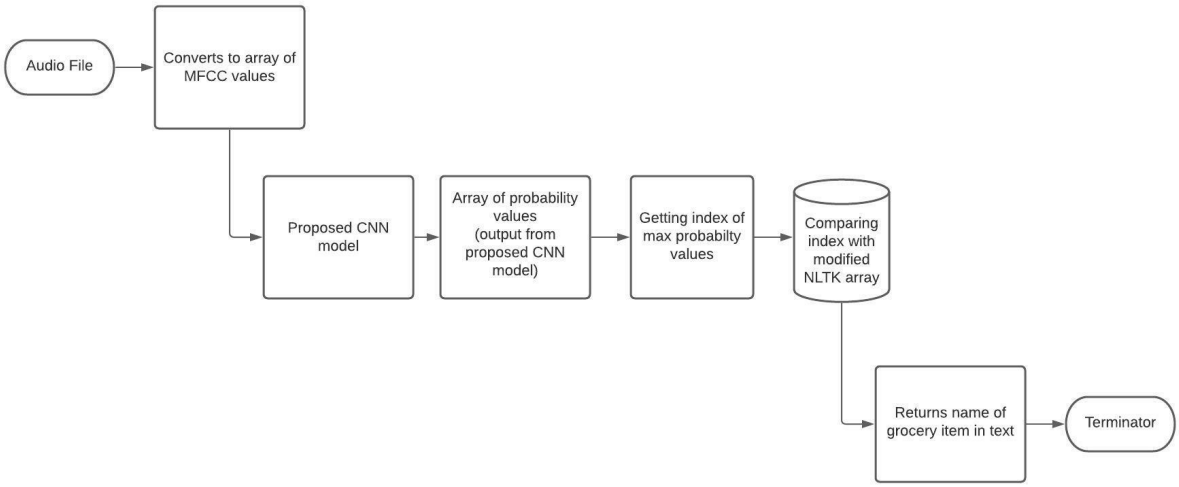

Fig.5. CASE 2 model architecture block diagram

1. The Convolutional neural network model (CNN) takes inputs of the audio of the speaker.
2. The audio file is converted to its array of Mel-frequency cepstral coefficients (MFCC) values and passed through the model.
3. We train our proposed CNN model on Hindi and Tamil separately using the dataset mentioned in section 4.1 and using three activation functions namely, relu, sigmoid, and softplus functions. The architecture of our proposed CNN model is mentioned in Table 2 and also shown in Fig.9.
4. The model returns an array of probabilities of each grocery item in the database. The probabilities are computed using the output layer with the softmax activation function. The softmax function takes the numerical outputs of the last linear layer and computes the exponent of it, which is further normalized to give the probability values. Each index of the array will contain a probability generated by the model which corresponds to a grocery item in our dataset. We discovered that the CNN models (for Hindi and Tamil) using relu activation function provided the highest training accuracy.
5. We train our CNN models with 160 audio clips for each Hindi and Tamil which amounts to 320 audio clips, which have been labelled as shown in Fig. 6. These audio clips are then converted into MFCC values which are fed to the model as input, it then uses the input and the labels to train and produces an output array of size 40 which is the size of our grocery list, containing the probability of each item.

```
print(target_val[0])
```

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

Fig.6. Target label array example for training (onion)

6. We test our CNN models with our 80 testing audios (1 for each of the 40 grocery items in Hindi and Tamil). An unlabelled audio recording is given as input to the trained CNN models. The output returned is an array of probabilities where the index corresponds to each grocery item in the database. We match the index of the highest probability value with the indices in the array of grocery items from the database to get the actual text of the item. The block diagram of the process is shown in Fig.5.

Speech To Text Methodology

The audio files need to be pre-processed and cannot be directly given as an input into any Speech-To-Text model as such. Our proposed work uses MFCC (Mel Frequency Cepstral Coefficient) (Mamatov et. al, 2021) of each audio file. These values are derived from the process shown in Fig.7. Cepstral is the rate of change of spectral bands and the information about them. A cepstral representation of a Hindi audio clip (this audio's script is "500-gram vasant pyaz daal do") is shown in Fig.8 where darker blue colour represents a lower cepstral coefficient value and the dark red colour represents a higher cepstral coefficient value. Each word pronounced or each sentence spoken will have an unique spectral waveform. This is done in Python using the mfcc function in the librosa library.

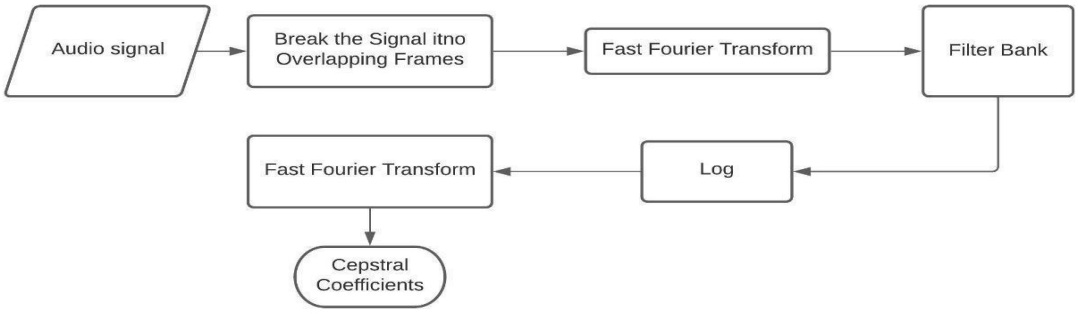

Fig.7. Transformation of audio signal to cepstral coefficients.

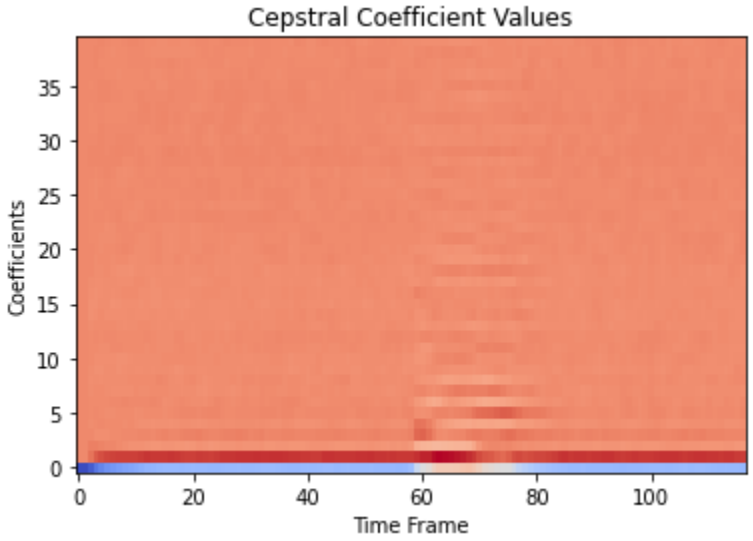

Fig.8. Cepstral Coefficient Values of a Hindi audio

The mean of the cepstral coefficients has been taken for each input audio and are the inputs for the input layer of our proposed CNN model.

The standard relu activation returns max(x, 0), the element-wise maximum of 0 and the input tensor where x is the input value.

Sigmoid activation function is shown in Equation (3) where x is the input value and for small values of x this function returns a value close to zero and for large values it will return a value close to 1.

$$sigmoid(x) = 1/(1 + exp(-x)). \quad (3)$$

Softplus activation function is shown in Equation (4) where x is the input value and as x value increases the function return value increases drastically.

$$softplus(x) = log(exp(x) + 1). \quad (4)$$

It is important to prevent overfitting so after these layers, a dropout layer is added and in this case 10 percent of the features will be lost. After this one more dense layer of size 30 is added with the relu activation function. The output layer is the size of the number of grocery items in the dataset with the activation function as softmax. The softmax of each vector x is computed as shown in Equation (5).

$$softmax(x) = exp(x)/sum(exp(x)). \quad (5)$$

Softmax function generates probability or likeliness values which helps to determine which audio or speech input corresponds to which grocery item. The number of epochs for which the model will run was chosen to be 250.

Our CNN model will take the audio as input and generate an array of probabilities for each food item as shown in Fig.10 and then the item with the highest probability is taken as a keyword. This keyword is then matched with the dataset and displays the item to the user on the screen.

Table.2. Architecture of proposed CNN model

Layer type	No. of neurons used	Activation function used
Convolution (Input)	40	Relu
Convolution	30	Relu
Convolution	30	Relu
Max Pooling	Na	Na
Dropout	Na	Na
Dense (Output)	30	Softmax

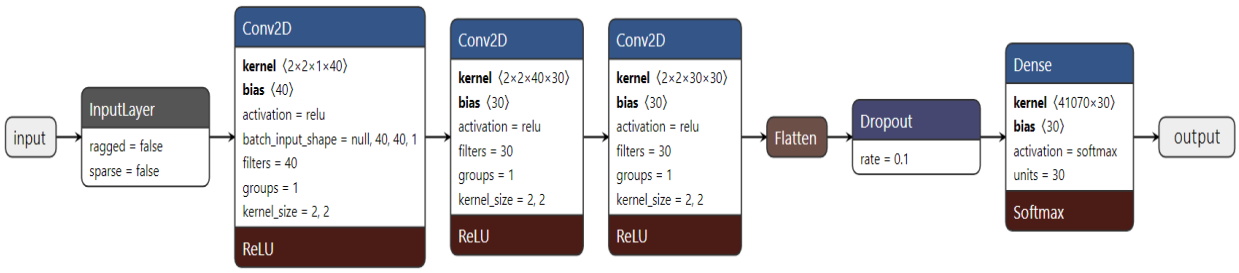

Fig.9. Proposed CNN Model Architecture

As shown in Fig.9, our proposed CNN model in total has 6 layers consisting of 1 input layer with input dimensions as 40 (indicating the number of cepstral coefficients) followed by 2 convolutional layers with the activation function as relu. The max pooling layer chooses the maximum element from a set of different features of the previous layer. This ensures that the model contains the most prominent features while going to the next layer. To prevent overfitting, a dropout layer with dropout rate of 0.1 is added to drop 10 percent of the features. Following this, another dense layer of 30 neurons is added with the activation function as Relu. The output layer is the size of the number of grocery items in the dataset with the activation function as softmax as shown in Table 2. Other activation functions were also tested namely sigmoid and softplus along with Relu. Relu was chosen as it gave the highest accuracy for the training and testing data as shown in Table 3.

```
In [20]: test[1]

Out[20]: array([-3.62333496e+02,  1.25694809e+02, -2.14186096e+01,  1.20067091e+01,
                -1.25998659e+01,  3.77031875e+00, -4.18639469e+00, -2.54375517e-01,
                -1.02168503e+01, -3.81593704e+00, -3.16860437e+00,  4.43091363e-01,
                -3.57281923e-01, -6.28106594e+00, -1.31959295e+00, -5.89414597e+00,
                -3.37643743e+00, -6.08278036e+00, -3.26495528e+00, -4.92254972e+00,
                -2.82059073e+00, -1.50573993e+00, -1.59171844e+00, -1.54666317e+00,
                -1.26217699e+00, -4.52000797e-01, -1.07674825e+00,  3.58589828e-01,
                 3.87932032e-01,  2.37139773e+00,  2.36600801e-01, -5.44529378e-01,
                 8.30730259e-01,  8.13699186e-01,  1.59878576e+00,  1.71491766e+00,
                -2.52684546e+00, -7.09456921e-01, -1.80718303e-02, -1.32757306e+00])
```

Fig.10. The array of probabilities of each food item in the database

5. Results and Discussion

Creation of the input data was done through extracting the MFCC values for each audio file. The training and testing data were split in an 80,20 ratio. Our proposed CNN model was created and for Hindi language, an accuracy of 100% with the training data consisting of 200 sentences and 94.5% with the testing data is obtained and for Tamil, an accuracy of 100% with the training data and 81% with the testing data is obtained from our CNN model. The model accuracy and model loss graphs of the training data are shown in Fig.11 and Fig.12 for Hindi and Tamil respectively. As the number of epochs increases, the accuracy increases until a certain threshold of 250 epochs. Similarly, for the model loss graph, as the number of epochs increases, the loss value decreases indicating how our CNN model improves for the same threshold. The proposed DNN and LSTM model (Singh et. al., 2019) produced an accuracy of 60% and 85% for the 1000 short and long sentences they provided for English to Hindi text

translation. In the system proposed, the POS-tagger tags 35% of the words incorrectly. The tagged words are used for translating the given input and so this 35% incorrect tagging greatly hinders the translation performance.

The developed system (Dharmale et. al., 2016) produced an accuracy of 90% with the speech inputs they have given, using the HMM (Hidden Markov Model) through the Google API. In their proposed model, they have focused only on Speech Recognition for English. In the system proposed (Manasa et. al., 2019), they propose two models, namely a trained acoustic model and a cross-language adapted model. The former gives a word error rate of 16.8% while the latter gives a word error rate of 50% which indicates that only half of the words predicted were matching. Our proposed methodology and model provides a better accuracy than the existing methods as shown in Table 3 and 4 for our dataset.

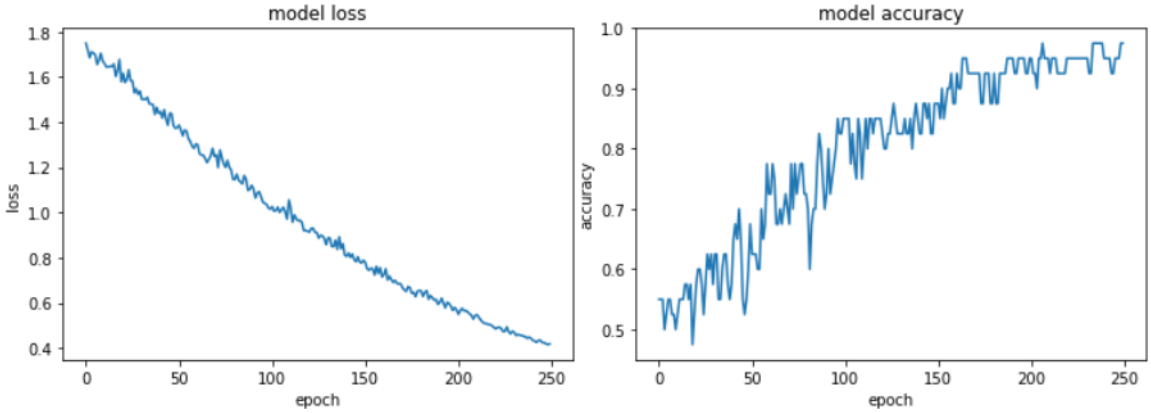

Fig. 11. Model accuracy and loss graphs vs epoch for Hindi dataset.

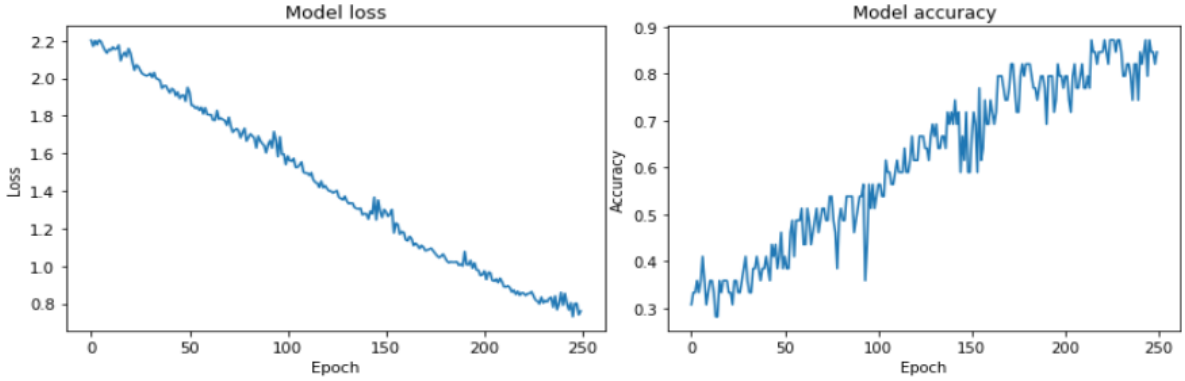

Fig.12. Model accuracy and loss graphs vs epoch for Tamil dataset

Table.3. Comparison of activation function based on Hindi and Tamil.

Activation functions	Training Accuracy		Testing Accuracy	
	Hindi	Tamil	Hindi	Tamil
Relu	100%	100%	94.5%	81%

Sigmoid	15%	12.8%	10%	7%
Softplus	90%	74.4%	85%	72.5%

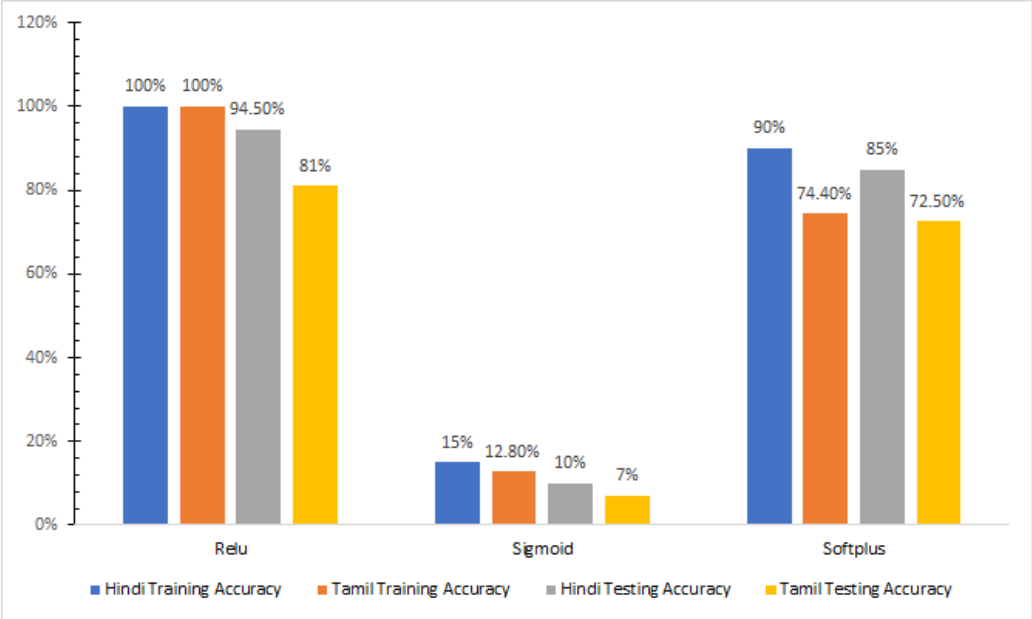

Fig.13. Training and Testing accuracies for Hindi and Tamil across Relu, Sigmoid and Softplus activation functions.

Our proposed CNN model has been experimented with 3 different activation functions as shown in Table 3. It is seen that the relu activation function gives the best accuracy. This is possibly because the sigmoid function returns values 0 and 1 for every input and the softplus activation function returns logarithmic output of the range from (-) infinity to 0 for each layer which may not be optimal in this case. This is because the cepstral coefficients are of a much larger range than what can be matched with a sigmoid or softplus function. Rectified Linear Unit (ReLu) activation function allows the max threshold to be changed accordingly which works well for the inputs. From Fig. 13 we can confirm that the ReLu activation function is much more suitable for this dataset than the Softplus and Sigmoid activation functions.

Table.4. Comparison between Google API and our proposed CNN model.

Speech-to-Text model	Hindi accuracy	Tamil accuracy
Google API	82%	75%
Our proposed CNN model	94.5%	81%

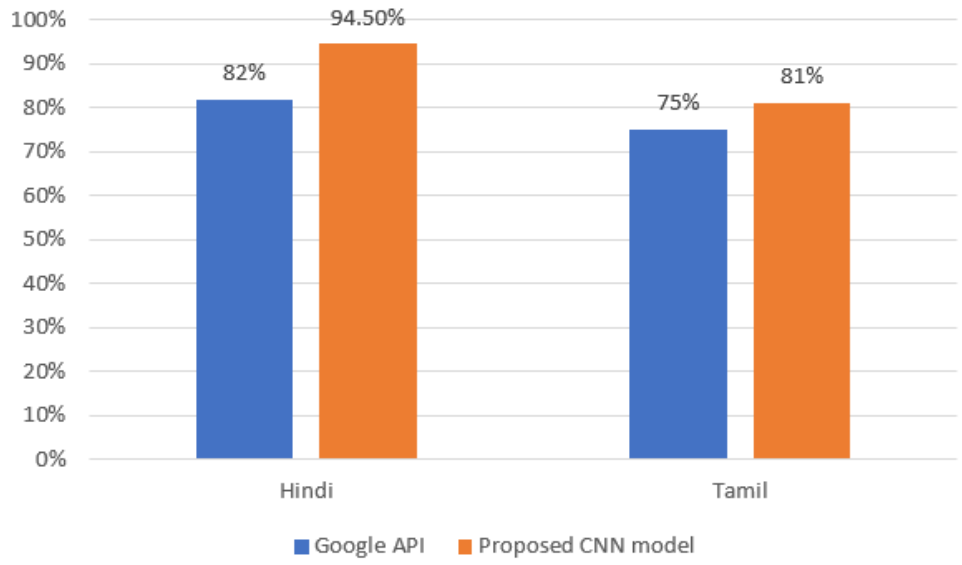

Fig.14. Testing accuracies for Hindi and Tamil with the Google API and our proposed CNN model.

From Fig. 14 and Table 4, it is seen that our proposed CNN model performs better than Google API for Hindi and Tamil languages in terms of classifying requests into their respective grocery item.

Fig.15 consists of the screenshots from the application using both Speech-to-Text classification models discussed in section 4.

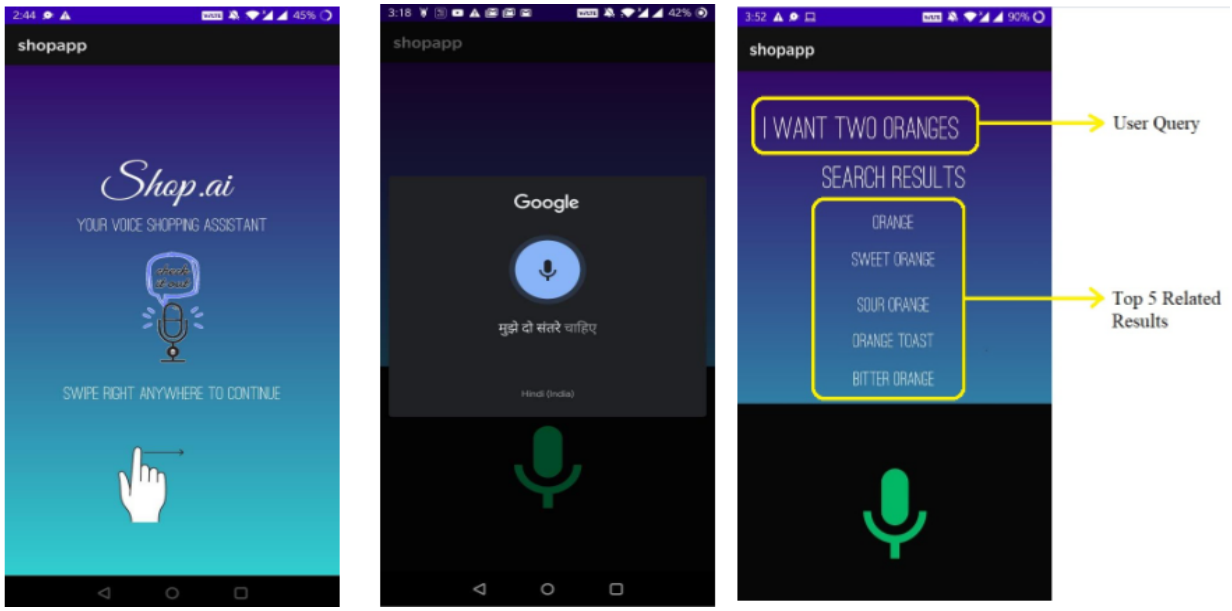

Fig.15. Screenshots of the final application

6. Conclusion

The objective of creating a total voice-based implementation of online shopping using native Indian languages (Hindi and Tamil) were fulfilled and also its extension into an Android application was completed. The testing was carried out with 400 audio clips for both Hindi and Tamil combined. Our proposed CNN model provides good results in terms of accuracy for languages which have not been tested in such a scenario (Hindi and Tamil). It is also clear that speech recognition for Hindi and Tamil work better with relu activation function, giving an training accuracy of 100% for the given dataset and the testing accuracy were found out to be 94.5% and 81% for Hindi and Tamil respectively. We have also concluded that our proposed CNN model outperforms Google API's while classifying grocery requests with the Google API giving 82% and 75% accuracy for Hindi and Tamil respectively, and our proposed CNN model giving 94.5% and 81% training accuracy for Hindi and Tamil respectively. These ideas and models can be extended for not just shopping items but a variety of other applications which require human computer interaction. Furthermore, it can be extended to a variety of other languages which are not as popular which can help the minorities only familiar with that certain language.

References

Adi, DP., Gumelar, AB., Meisa, RP. 2019. Interlanguage of Automatic Speech Recognition. International Seminar on Application for Technology of Information and Communication, 88-93.

Anoop, VS., Asharaf, S. 2017. A Topic Modeling Guided Approach for Semantic Knowledge Discovery in e-Commerce. International Journal of Interactive Multimedia & Artificial Intelligence, 4,6.

Bano, S., Jithendra, P., Niharika, GL., Sikhi, Y. 2020. Speech to text translation enabling multilingualism. IEEE International Conference for Innovation in Technology, 1-4.

Chand, M., Mulchandani, S., Mirkar, S. 2019. Visually Impaired Friendly E-commerce website. 4th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques , 191-196.

Dharmale, G., Thakare, V., Patil, DD. 2016. Intelligent hands free speech based sms system on android. International Conference on Advances in Human Machine Interaction, 1-5.

Jiang, J., Wang, HH. 2021 Application intelligent search and recommendation system based on speech recognition technology. International Journal of Speech Technology, 1, 23-30.

Kandhari, MS., Zulkemine, F., Isah, H. 2018. A Voice Controlled E-Commerce Web Application. IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference , 118-124.

Këpuska, V., Bohouta, G. 2017. Comparing speech recognition systems (Microsoft API, Google API and CMU Sphinx). International Journal of Engineering Research and Applications, 3, 20-4.

Kiran, R., Nivedha, K., Subha, T. 2017. Voice and speech recognition in Tamil language. 2nd International Conference on Computing and Communications Technologies, 288-292.

Kumar, A, Aggarwal, RK. 2020. Hindi speech recognition using time delay neural network acoustic modeling with i-vector adaptation. International Journal of Speech Technology, 1-2.

Mamatov, N., Niyozmatova, N., Samijonov, A. 2021. Software for preprocessing voice signals. International Journal of Applied Science and Engineering, 1-8.

Manasa, CS., Priya, KJ., Gupta, D. 2019. Comparison of acoustical models of GMM-HMM based for speech recognition in Hindi using PocketSphinx. 3rd International Conference on Computing Methodologies and Communication, 534-539.

Miao, C., Liang, S., Chen, M., Ma, J., Wang S., Xiao J. 2020. Flow-TTS: A non-autoregressive network for text to speech based on flow. International Conference on Acoustics, Speech and Signal Processing, 7209-7213.

Rallabhandy, S., Rodda, S. 2020. Keyboard-less online shopping for the visually impaired using natural language processing and face recognition mechanism. Smart Intelligent Computing and Applications, 253-260.

Singh, SP., Darbari, H., Kumar, A., Jain, S., Lohan, A. 2019. Overview of Neural Machine Translation for English-Hindi. International Conference on Issues and Challenges in Intelligent Computing Techniques, 1-4.

Trivedi, A., Pant, N., Shah, P., Sonik, S., Agrawal, S. 2018. Speech to text and text to speech recognition systems-A review. IOSR Journal of Computer Engineering, 2, 36-43.

WHO, 2020, Vision2020 report, Blindness, https://www.who.int/blindness/Vision2020_report.pdf, Access on: 16th April 2021.