PROFILING REGIONAL DIALECT

SUMMER INTERNSHIP PROJECT REPORT

Submitted by

Aishwarya PV(2016103003)

Prahanya Sriram(2016103044)

Vaishale SM(2016103075)

College of Engineering, Guindy

ANNA UNIVERSITY: CHENNAI 600 025

May-2018

1. PROBLEM STATEMENT:

In this project of Profiling Regional Dialect, our fundamental aim is to develop a Speech Recognition Algorithm that will process the human speech and also profile the speaker's particular dialect and thereby, convert it to text.

2. PROBLEM DESCRIPTION:

The problem being Speech-to-text conversion, our main goal was to replicate the Speech recognition process that takes place within the human body itself.The problem requires us to make the machine perceive speech just like the human ear. It has to understand the loudness and frequency factors of a speech signal in a way similar to the vibrations produced in the cochlea of the human ear. Machine learning techniques helped teach the machine to recognize words. After training, the model will start learning how to convert speech to text in an efficient manner. In this way, we can expand the problem to include more dialects, different accents etc. so that the model can convert speech to text in a more efficient manner.

3. WORKFLOW:

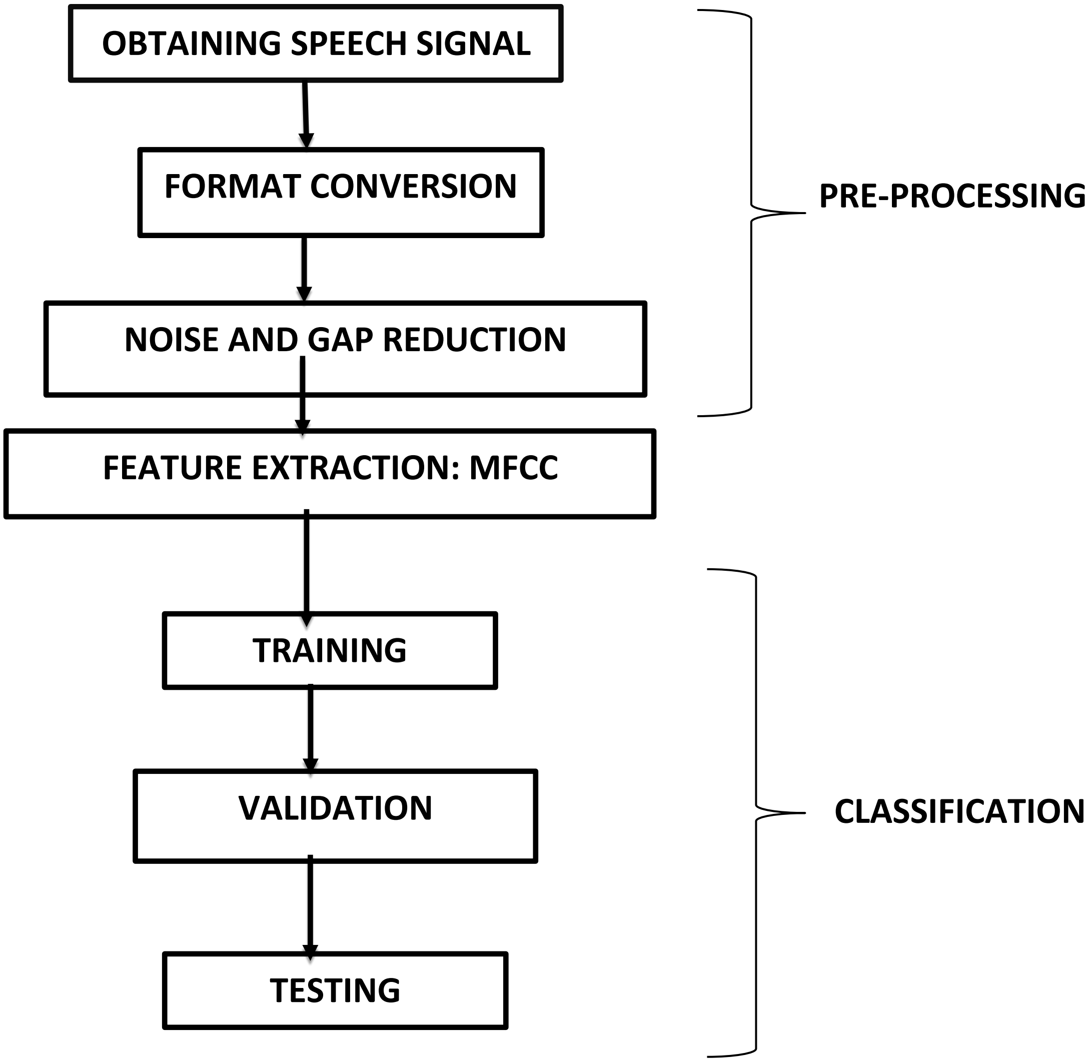


FIG 3.1: General workflow

3.1 Pre-processing:

The sound file is converted into a suitable format such as a .wav format. The signals may contain disturbances like noise and unnecessary pauses that are not desirable. These are removed with the help of tools like Audacity. The spectrograms of the speech signals before and after preprocessing can also be obtained using Audacity software. This step in preprocessing improves the accuracy of speech recognition.

The next step is to convert the signal which is in the time domain into its frequency domain. This is known as Discrete fourier transform(DFT). The transfer is done using an algorithm called Fast fourier transform, this helps speed up the transformation.

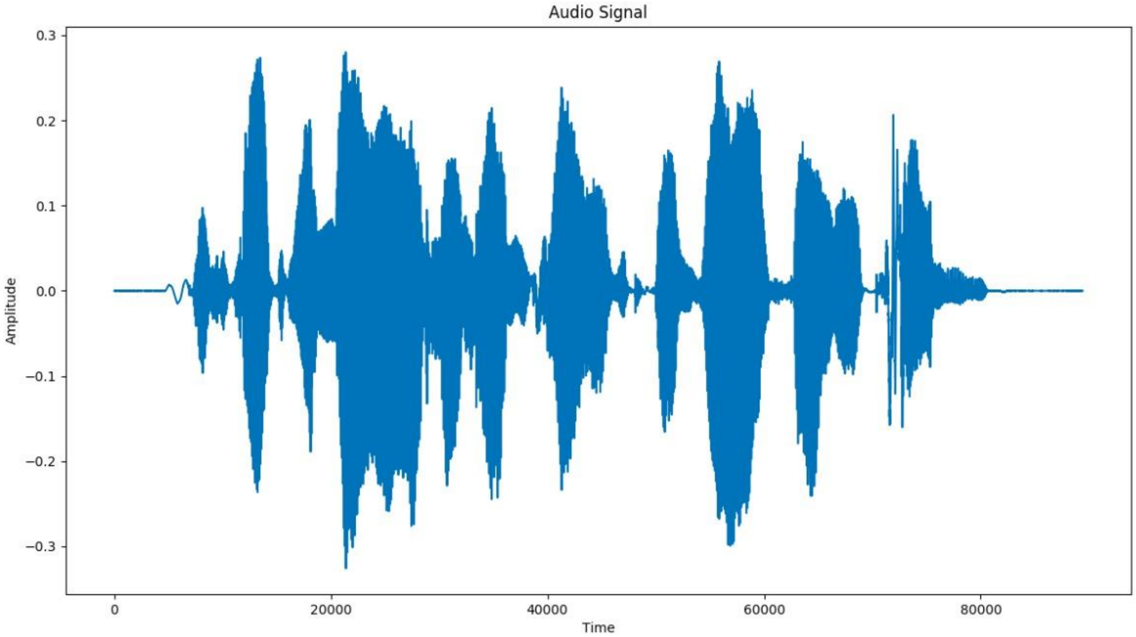


FIG 3.2: Audio signal in the time domain

3.2 Feature Extraction:

The next step in signal processing would be to extract the features of the pre-processed signal. Mel Frequency coefficient cepstrum was used as they are frequency domain features and are more accurate than time domain features. The following are the steps that are involved in obtaining MFCCs:

1. Frame the signal into short frames.
2. For each frame calculate the periodogram estimate of the power spectrum.
3. Apply the mel filterbank to the power spectra, sum the energy in each filter.
4. Take the logarithm of all filterbank energies.
5. Take the DCT of the log filterbank energies.
6. Keep DCT coefficients 2-13, discard the rest.

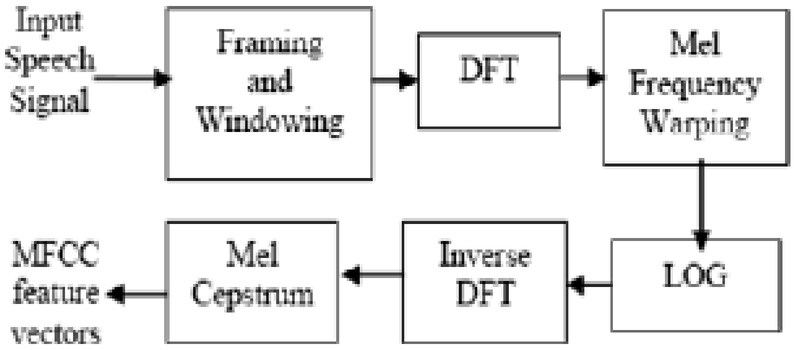


FIG 3.3: Steps in Feature Extraction

3.3 Neural Network and Machine Learning:

The obtained features are fed as input to the artificial neural network. An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems process information. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems (here, speech to text conversion).

3.4 Architecture Of Neural Networks:

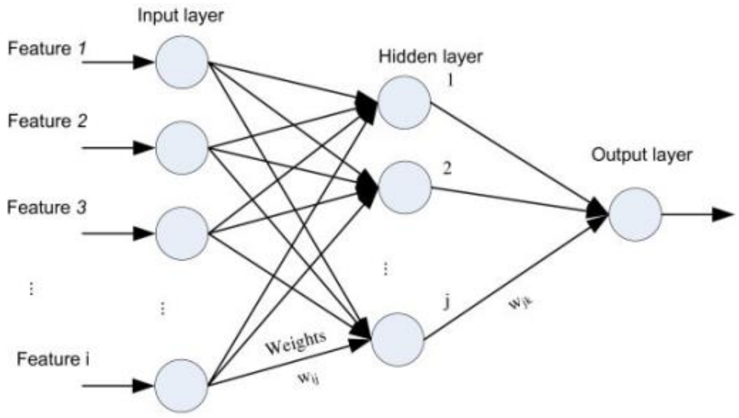


FIG 3.4: General Architecture of neural networks

The leftmost layer is called the input layer. The rightmost is the output layer. The middle layer is a hidden layer.

A combination of RNN and CNN was used in the project.

3.5 Recurrent Neural Networks:

Recurrent networks, take as their input not just the current input example they see, but also what they have perceived previously in time. The decision a recurrent network reached at time step affects the decision it will reach one moment later. So recurrent networks have two sources of input, the present and the recent past. Therefore, it can be considered that recurrent networks have memory.

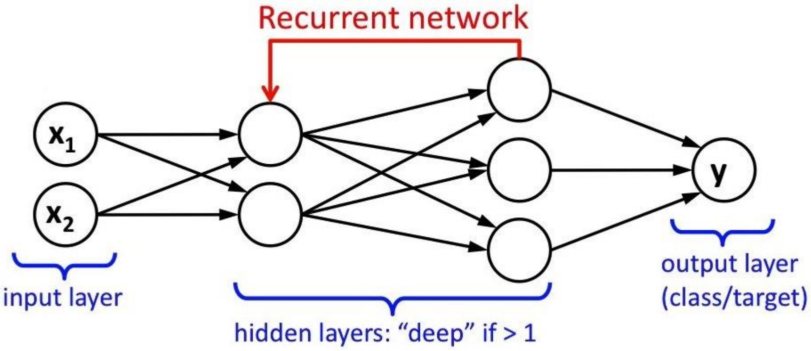


FIG 3.5: Recurrent neural network structure

3.6 Convolutional Neural network:

CNN, a Convolutional Neural network, is a biologically-inspired variant of MLPs (Multilayer Perception). The main component of this would be the convolutional layer. The convolution is a mathematical function used to merge two sets of information. The inputs of hidden units in layer **m** are from a subset of units in layer **m-1**, units that have spatially contiguous receptive fields.

3.7 Backpropagation:

Backpropagation is an algorithm used for supervised learning of artificial neural networks using gradient descent.

3.8 Optimisation:

Optimization algorithms helps us to ***minimize (or maximize)*** an error function **E(x)** which is dependent on the Model's **learnable parameters like weights and bias values** which are used in computing the target values(**Y**) from the set of *predictors*(**X**). What we'd like is an algorithm which lets us find weights and biases so that the output from the network approximates y(x) for all training inputs x. To quantify how well we're achieving this goal we define a cost function:

$$C(w,b) \equiv \frac{1}{2n} \sum_x \|y(x) - a\|^2.$$

Here, w denotes the collection of all weights in the network, b all the biases, n is the total number of training inputs, a is the vector of outputs from the network when x is input, and the sum is over all training inputs, x .Stochastic Gradient Descent(SGD) was used for optimising the model.

3.9 Validation:

Cross-validation is a method used to estimate the skill of machine learning models. Train/Test Splits the variation of validation that we have chosen. In this variation, the data we use is usually split into training data and test data. We have the test dataset (or subset) in order to test our model's prediction.

3.10 Predicting the output:

Finally the model is given a sample input and the true output is tested against the model's predicted output.

4. IMPLEMENTATIONS AND TOOLS:

4.1 Tools used:

Tools used	Description
Audacity	Audacity is a free audio editor and recorder. We used audacity to analyse a speech signal and for noise cancellation.
LibriSpeech	LibriSpeech is a corpus of approximately 1000 hours of sixteen kiloHertz read English speech.
Tensor flow	TensorFlow is a suite of software for developing deep learning models.

Keras	Keras is a minimalist Python library for deep learning that can run on top of TensorFlow.
Matplotlib	Matplotlib is a Python 2D plotting library which produces quality figures in a variety of hardcopy formats and interactive environments across platforms.
python_speech_features	This library provides common speech features for ASR including MFCCs and filterbank energies.
Data set 2	It contains multiple speech files of about 30 different words with different pronunciations of each word.

FIG 4.1: Tools used in the project

4.2 IMPLEMENTATION:

The LibriSpeech dataset was obtained containing 1GB of various speech signals in .flac format and the respective labels for those signals. First step was to convert the Flac files to wav files. Next, two JSON files were created containing the training data and the validation data.

The dataset was split into the training, validation/testing. 60% of the data was taken for training and 40% for validation. The features were then extracted and normalised. These extracted features were then fed into the model for training it. The training and validation sets were split into mini batches of size 20.

These data were then converted into FFT of the audio signal in fig 3.2 as seen in fig 4.2 and the features were extracted from the frequency domain and this is given as input to the neural network.

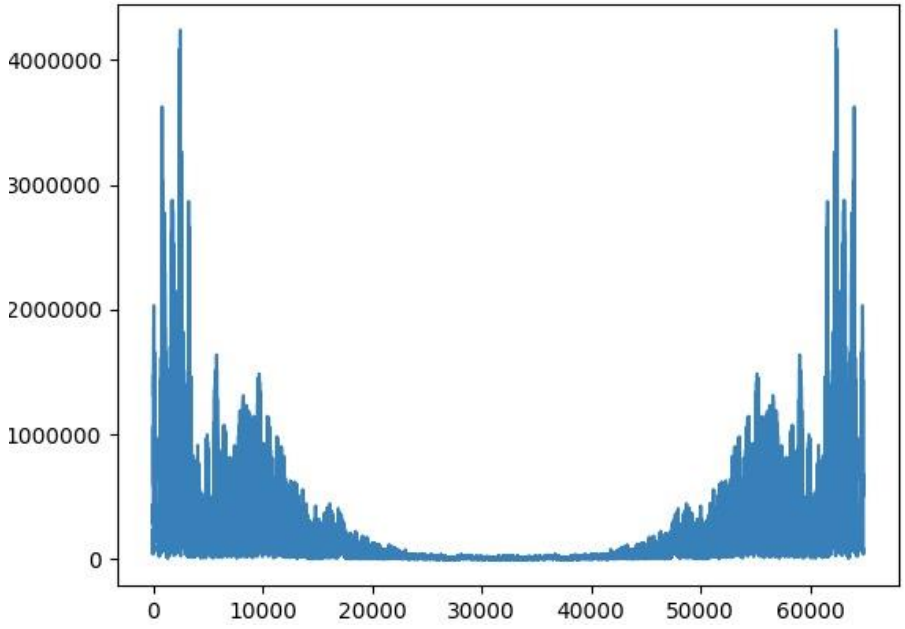
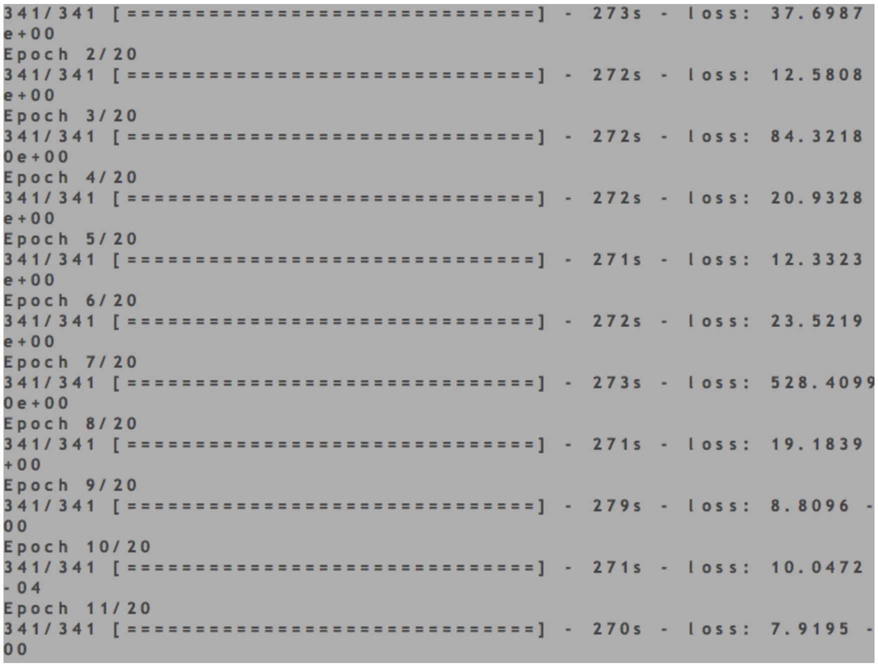


FIG 4.2: FFT

The model was trained for 20 epochs with each epoch containing 101 samples each. After training the model it was provided with a test sample to predict the output and evaluation took place. The model was trained with a different dataset containing speech signals of about 30 different words with different pronunciations. This dataset was used to limit the dictionary and to train the model to learn the different variations. This model was also trained for 20 epochs of 341 samples each. This can be seen in fig 4.3.



```
341/341 [==============================] - 273s - loss: 37.6987
e+00
Epoch 2/20
341/341 [==============================] - 272s - loss: 12.5808
e+00
Epoch 3/20
341/341 [==============================] - 272s - loss: 84.3218
0e+00
Epoch 4/20
341/341 [==============================] - 272s - loss: 20.9328
e+00
Epoch 5/20
341/341 [==============================] - 271s - loss: 12.3323
e+00
Epoch 6/20
341/341 [==============================] - 272s - loss: 23.5219
e+00
Epoch 7/20
341/341 [==============================] - 273s - loss: 528.4099
0e+00
Epoch 8/20
341/341 [==============================] - 271s - loss: 19.1839
+00
Epoch 9/20
341/341 [==============================] - 279s - loss: 8.8096 -
00
Epoch 10/20
341/341 [==============================] - 271s - loss: 10.0472
-04
Epoch 11/20
341/341 [==============================] - 270s - loss: 7.9195 -
00
```

FIG 4.3: The training process

The dataset was shuffled prior to training to avoid overfitting. Custom input was also used to test the model . Graphs were plotted to represent the audio signal in fig 3.2, its features and FFT. This is shown in fig 4.4 and 4.5.

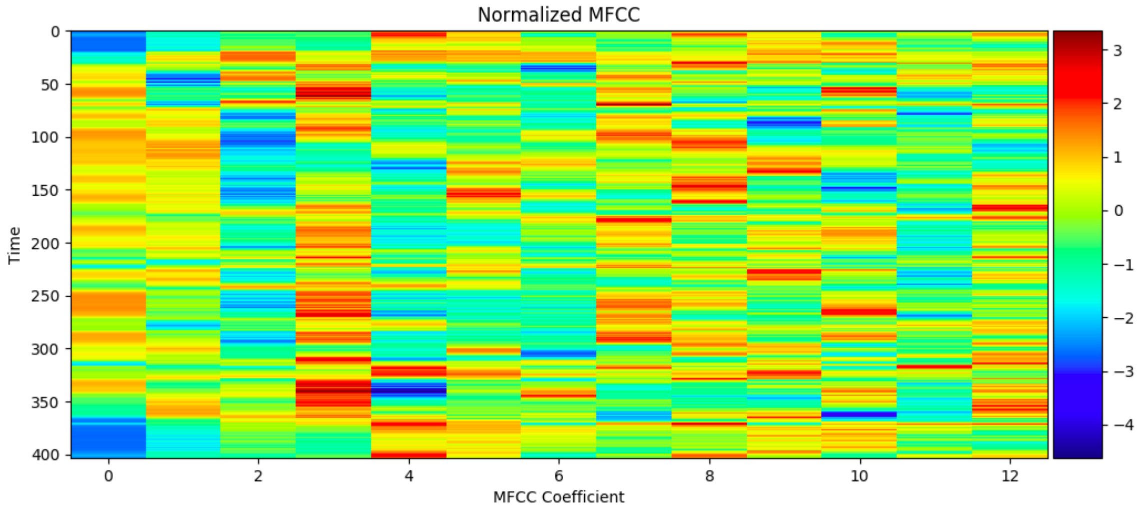


FIG 4.4: Normalized MFCC

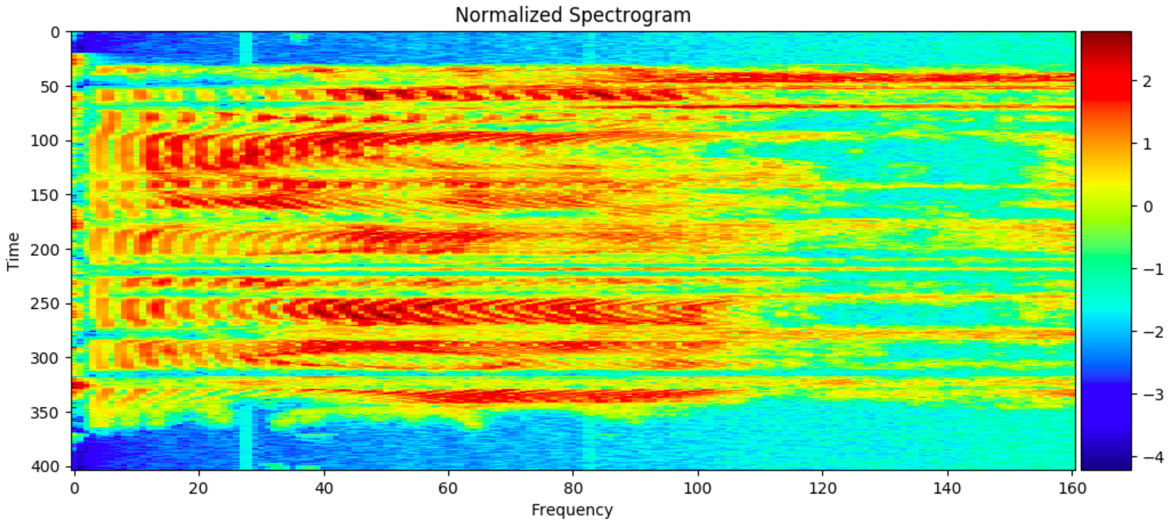


FIG 4.5: Normalized Spectrogram

5. ANALYSIS:

Initially both spectrogram features and the MFCC features were used. The model trained using the MFCC features proved to be more effective than the spectrogram, the reason for this is that a spectrogram uses a linear frequency scaling. The mel-frequency scale on the other hand, is a quasi-logarithmic spacing roughly resembling the resolution of the human auditory system. From this we can observe that the type and number of features drastically affect the accuracy of recognition as seen in fig 5.1.

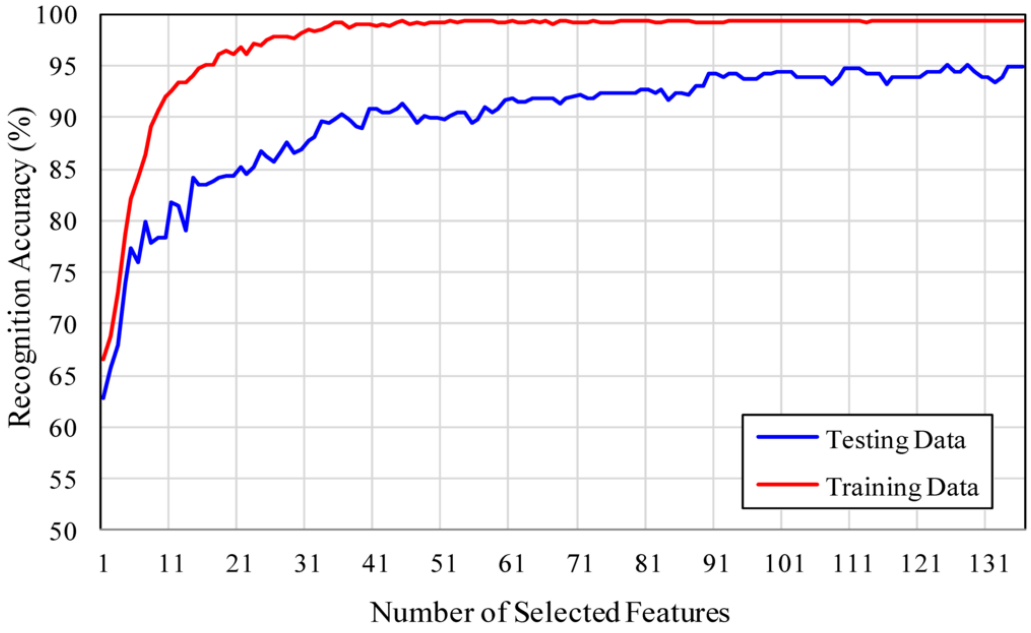


FIG 5.1: Accuracy vs features.

Different neural network models were used. First a simple RNN model was implemented, but the model was inefficient. Then a deeper RNN model was used, but were faced with the exploding gradient problem. In deep networks or RNNs, error gradients can accumulate and result in very large gradients. These in turn result in large updates to the network weights, and in turn, an unstable network.

The final model included a 1D convolution layer followed by batch normalisation and a recurrent layer and a time distributer wrapper. This wrapper allows us to apply a layer to every temporal slice of an input. The input layer is normalised by adjusting and the activations. Batch normalization reduces the amount by what the hidden unit values shift. We have used the Connectionist Temporal Classification which is a way to get around not knowing the alignment between the input and the output. For optimization, different algorithms like Adam, standard gradient descent and stochastic gradient descent were tried. The latter was used as it was the most suited algorithm.

To avoid over-fitting or under-fitting, cross validation is required. Overfitting would lead to the model fit too closely to the training dataset. But when a model is underfitted, it means that the model does not fit the training data and therefore misses the trends in the data. The method used for avoiding this is the train/test split. There is also a more efficient k-fold cross validation which is quite time consuming.

6. CONCLUSION:

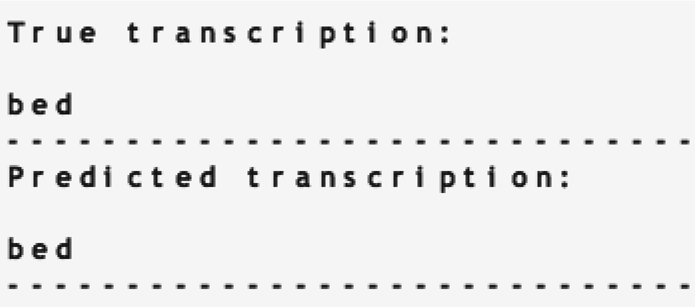
The speech-to-text Converter which was built in the duration of one month is capable of recognising basic speech with an acceptable level of accuracy due to the limited resources and the short timespan. There is, of course, a scope of betterment as is in any commodity. In future, with the access to increased amounts of speech datasets, better hardware and a larger timespan, further enhancements can be made by training our model with more datasets that will incorporate varieties in speech and thus, enable us to effectively deal with a variety of accents of speech. It can also be developed into a speech translator with the help of advanced resources.

The hope is to build a more advanced version of the project in the future.

APPENDIX:

SCREEN SHOTS OF RESULTS:

Custom given input:

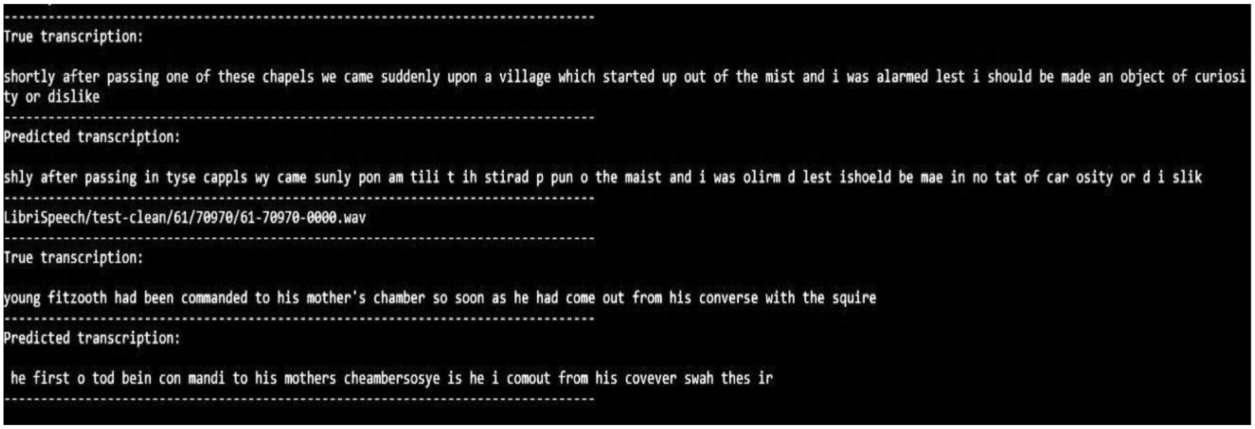


```
True transcription:

bed
-----------------------------
Predicted transcription:

bed
-----------------------------
```

Input given from part of dataset:



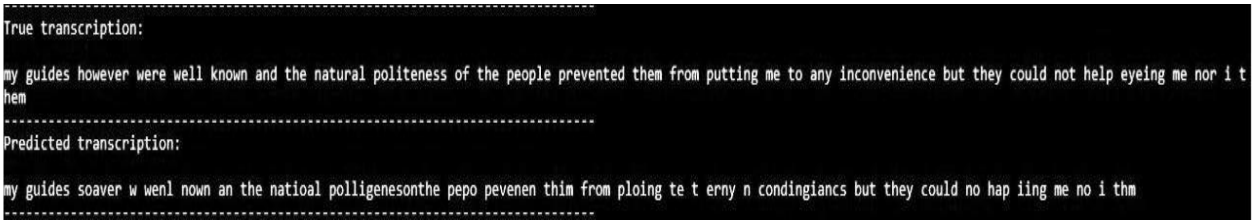
```
--------------------------------------------------------------------------------
True transcription:

shortly after passing one of these chapels we came suddenly upon a village which started up out of the mist and i was alarmed lest i should be made an object of curiosi
ty or dislike
--------------------------------------------------------------------------------
Predicted transcription:

shly after passing in tyse cappls wy came sunly pon am tili t ih stirad p pun o the maist and i was olirm d lest ishoeld be mae in no tat of car osity or d i slik
--------------------------------------------------------------------------------
LibriSpeech/test-clean/61/70970/61-70970-0000.wav
--------------------------------------------------------------------------------
True transcription:

young fitzooth had been commanded to his mother's chamber so soon as he had come out from his converse with the squire
--------------------------------------------------------------------------------
Predicted transcription:

 he first o tod bein con mandi to his mothers cheambersosye is he i comout from his covever swah thes ir
--------------------------------------------------------------------------------
```



```
--------------------------------------------------------------------------------
True transcription:

my guides however were well known and the natural politeness of the people prevented them from putting me to any inconvenience but they could not help eyeing me nor i t
hem
--------------------------------------------------------------------------------
Predicted transcription:

my guides soaver w wenl nown an the natioal polligenesonthe pepo pevenen thim from ploing te t erny n condingiancs but they could no hap iing me no i thm
--------------------------------------------------------------------------------
```