SPEECH EMOTION RECOGNITION

M.Madhusudhan[1], R.Mahesh kumar[2], S.Ambareesh[3], CH.Jayanth[4]

[1]*AssistantProfessor,CSE,CMR Technical campus,Hyderabad, India*

[2]*Student, CSE, CMR Technical Campus, Hyderabad, India*

[3]*Student, CSE, CMR Technical Campus, Hyderabad, India*

[4]*Student, CSE, CMR Technical Campus, Hyderabad, India*

***ABSTRACT*: In this paper, a real time Deep learning based system was built for the Emotion Recognition using speech that have been given as input with the help of a PC microphone. The main purpose of this project is to design a model that can record live speech from computer microphone, analyse the audio file, detect and recognize the particular emotion. After recognition is done, the particular user can view their emotion through an interface with appropriate emojis. Speech Emotion Recognition is an rapidly increasing research domain in recent years. In this paper eight basic emotions (Anger, Happy, Fear, Neutral, sad, calm, disgust, surprise) are analyzed from emotional speech signals. In Our project we have used Long Short Term Memory(LSTM) network which is an artificial Recurrent Neural Network. The reason we used LSTM is that it provides higher accuracy while dealing with emotion recognition using speech. Our model was trained using RAVDEES data set which contains 7356 audio files . I have took 2880 files for training my model, in order to increase our accuracy and also to detect and recognize emotion from different speech. Our model scored an accuracy of 93% for training data set.**

***Keywords*: Long Short Term Memory (LSTM), Recurrent neural network, RAVDEES data set.**

1. INTRODUCTION

As we know Emotion plays a crucial role in daily human interactions. It helps us to know other people's feelings while we communicate with them through directly or indirectly.

when we communicate through social media platform like whats app, face book etc we use emojis to express our emotions, but while communicating through mobiles or while exchanging audio files to communicate we cannot detect the emotion of the person while conversation.

To overcome this problem, we developed a deep learning model which is capable recognizing emotions by just through the audio of the speaker.This speech emotion recognition is very useful in various fields like call centers, entertainment, voice assistance , good human computer interactions and education systems.

We have achieved speech emotion recognition using Recurrent neural networks. Our model detects certain emotions from the sound and shows users through the Tinker window. Therefore, it will reduce the problem of not being able to see emotions during a communication through audio exchange only.

This process of emotional processing and speech recognition usually consists of three parts, which were the selection of the emotional expression database, the feature extract, and the emotional recognition.

2. LITERATURE SURVEY

This section provides an overview of the major speech recognition techniques developed in recent years. Because of the importance of emotion recognition in human communication with the computer and the construction of artificial intelligence systems, there are many other recent publications and surveys in which it is conducted with the SER. In this section, we review the most recent studies related to current work.

In 2018, Swain et al. revised studies between 2000 and 2017 in the SER programs according to the three retained methods, input domain, and separators. An important phase of data research research and feature releases; however, only traditional machine learning methods such as CNN, KNN, SVM etc are considered a distinguishing tool, and the authors feel remorse for the neural networks and deep learning methods.

A year later, Khalil et al. reviewed comprehensible approaches to the SER using in-depth reading. Many in-depth, updated learning methods include deep neural network (DNN), convolution neural network (CNN), repetitive neural network (RNN), and auto encoder, spoken and some of their issues and strengths in the study. However, research cannot address accessible ways to overcome weaknesses.

Recently, Anjali et al. has published a review as a summary of how to identify speech emotions. A comprehensive discussion of various factors used in the sensitivity of speech and in the review of the various approaches used for this purpose from 2009 to 2018 is provided in the review. The retrieval of this paper was the depth of the analysis. Still, it can be considered a start.

In 2020, Bas et al. published a brief review of the importance of data sets and features of speech recognition, audio removal; finally, they analyzed the importance of differentiated approaches involving SVM and HMM. The power of the study was to identify a number of factors related to the recognition of speech emotions; however, its weakness is the leak of modern research methods and is briefly mentioned in the interaction of repetitive neural networks as an in-depth learning method.

In our model, the data set we needed to train was the RAVDEES data set. In this training process, we were able to achieve 93% accuracy. In our project we have chosen the LSTM RNN network because it is best suited to solve speech prediction problems.

3. EMOTIONAL SPEECH DATA

The effectiveness and robustness of emotion recognition systems will be easily affected if they are not properly trained with the appropriate database. Therefore, it is imperative to have sufficient and appropriate clauses in the database to train the speech emotion recognition system and to evaluate and verify its effectiveness. In this section, we provide detailed data about the database we used which is the RAVDEES data set.

3.1 RAVDEES DATA SET

The full form of RAVDEES is the Ryerson Audio-Visual Database of Emotional Speech and Song. It is a data set with eight different emotions such as joy, sadness, anger, fear, surprise, disgust, calmness, and neutrality performed by 24 characters (12 male actors and 12 female actors). The total size of this data set is 24.8 GB where we spent about 1.10 GB on our project.

The total audio files set by this data are 7356 audio files. These files are a mixture of both standard sentences and songs. RAVDESS is very rich in a variety of samples; and each feeling is made to performed in two different intensity and both with a normal voice and singing voice. RAVDEES data set consists of North American English accent.

4. FEATURE EXTRACTION

The speech signal contains of many number of parameters that reflects and are related to the emotional characteristics. One of the difficult task in emotion recognition is to decide what features should be used for building model. In recent research, there are many common features are extracted, such as energy, pitch, tone, and some spectrum features such as Linear Prediction Coefficient (LPC), Mel-Frequency Cep-strum Coefficient (MFCC) and Modulation spectral features. In this work, we have selected Mel Frequency Cep-strum Coefficient (MFCC), to extract the emotional features.

4.1 MFCC FEATURES

The Mel-Frequency Cep-strum Coefficient is widely used representation of the spectral property of the voice signals.These are the best in terms of speech recognition as it takes sensitivity to human perceptions of frequency into consideration. In each frame, the Fourier transform and power spectrum are estimated and map on Mel-frequency scale . There are various techniques used for MFCC features extraction some windowing the signal, applying the DFT, taking the log of the magnitude, and then warping the frequencies on a Mel scale.

In our project LSTM classifier will extract all the MFCC features from the given data set and then all the feature vectors are used for training the classifier.

5. SYSTEM ARCHITECTURE

The next phase after completing feature extraction is creating our LSTM mode land training our model and saving it.

The training of our LSTM Network model is done using TensorFlow and Keras. Since our model involves recognition of audio data, we used Recurrent Neural Networks. Recurrent Neural Networks is an extension to Neural Networks which was developed to deal with speech data.

Once the data set is imported and pre-processing of data is completed, we trained our model using TensorFlow and Keras. To avoid the problem of training our data set every time we want to use our model, we saved our trained LSTM model using Keras. Any deep learing model should be well trained in order to get accurate outputs, so in our project we are training our LSTM model with 100 numbers of epochs so as to get more accuracy for our model.

Our LSTM model consists of one hidden LSTM layer with 128 neurons and next we have three dense layers with continuous dropouts. First dense layer consists of 64 neurons with RELU as Activation function and second dense layer consists of 32 neurons with RELU as Activation function and the final dense layer which is an output layer consists of 8 neurons which denotes 8 different emotion states and here we have used SoftMax as an Activation function.

The below figure shows the visual diagram of LSTM classifier used in our project.

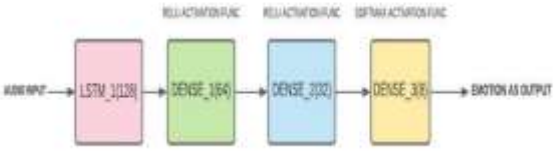


Fig. 1. visual diagram of LSTM

The below figure depicts the system architecture of our Deep learning project.

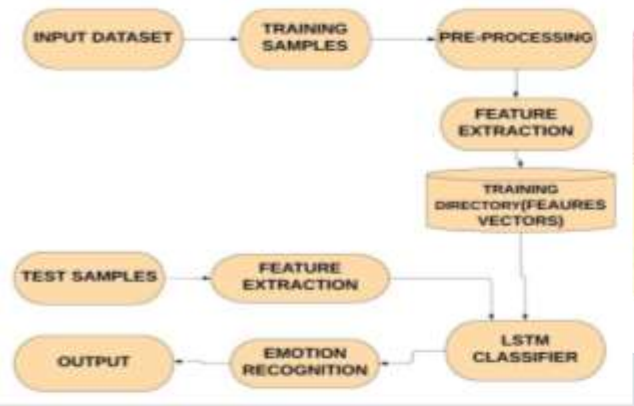


Fig. 2. System Architecture

In order to make our model to recognize the emotion of audios present in our dataset, we need to first train our model. We use TensorFlow module to train.

TensorFlow is an open-source library for numerical computation and large-scale machine learning that ease Google Brain TensorFlow, the process of acquiring data, training models, serving predictions, and refining future results.

TensorFlow bundles together both Machine Learning and Deep Learning models and algorithms. It uses Python as a convenient front-end and runs it efficiently in optimized C++. TensorFlow allows developers to create a graph of computations to perform. Each node in the graph represents a mathematical operations and each connection represents data. Hence, instead of dealing with low-details like figuring out proper ways to hitch TensorFlow allows developers to create a graph of computations to perform. Each node in the graph represents a mathematical operations and each connection represents data. Hence, instead of dealing with low-details like figuring out proper ways to hitch the exact output of one function to the input of another, the developer can simply focus on the overall logic of the application. TensorFlow is for the backend of keras. Thus this tensorFlow plays an major role in almost all deep learning based projects.

A loss function is used to optimize the deep learning algorithm. The loss is calculated on training and testing of model and its interpretation is based on how well the model is performing in these two sets. It is the sum of errors made for each example in training or testing sets. Loss value implies how inefficently or well a model behaves after each iteration of optimization. The loss at each iteration of our deep learning model has been decreasing which indicates a better accuracy of model for detection.

The loss of our model is shown in the below figure.

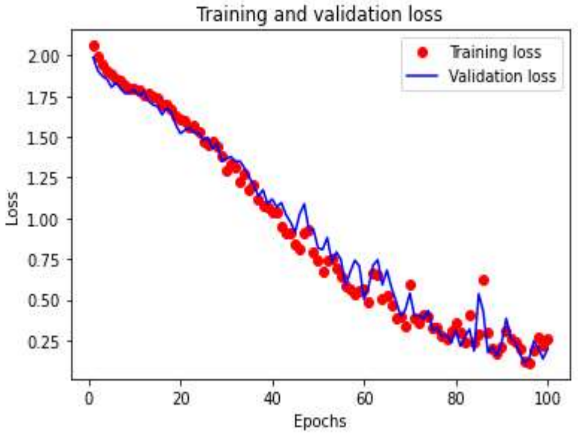


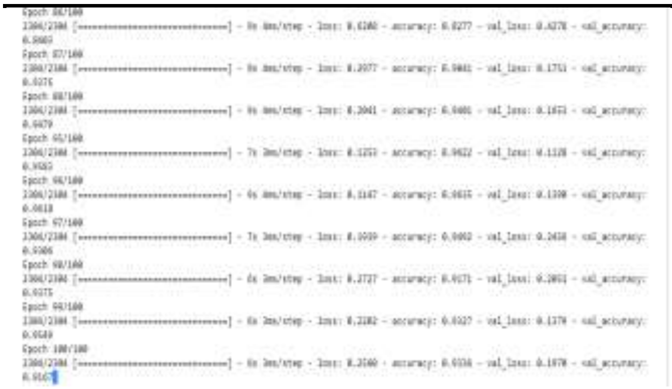
Fig. 3. Loss Graph of our model

The loss function we used is “Categorical_Crossentropy”. This loss function is used when there are more than two label classes in our dataset. The formula related to it is as follows

$$\mathcal{L}(\hat{y}, y) = -\frac{1}{N}\sum_{i}^{N}[y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)]$$

In our project we have trained our LSTM model with 100 number of epochs so as to get more accurate results as we know that any deep learning models accuracy depends upon how many number of times it is being trained. We can set epochs to any number of times depending upon our requirement. In first epoch the loss will be more and accuracy will be less but when it comes to last epoch loss and accuracy are vice versa; loss is less and accuracy is more.

The below figure shows the loss after each epoch in our training process.



```
Epoch 92/100
2304/2304 [==============================] - 9s 4ms/step - loss: 0.4260 - accuracy: 0.8277 - val_loss: 0.4278 - val_accuracy:
0.8603
Epoch 93/100
2304/2304 [==============================] - 9s 4ms/step - loss: 0.2077 - accuracy: 0.9041 - val_loss: 0.1751 - val_accuracy:
0.9276
Epoch 94/100
2304/2304 [==============================] - 9s 4ms/step - loss: 0.2041 - accuracy: 0.9401 - val_loss: 0.1673 - val_accuracy:
0.9479
Epoch 95/100
2304/2304 [==============================] - 7s 3ms/step - loss: 0.1253 - accuracy: 0.9622 - val_loss: 0.1128 - val_accuracy:
0.9583
Epoch 96/100
2304/2304 [==============================] - 6s 4ms/step - loss: 0.1147 - accuracy: 0.9615 - val_loss: 0.1390 - val_accuracy:
0.9618
Epoch 97/100
2304/2304 [==============================] - 7s 3ms/step - loss: 0.0939 - accuracy: 0.9602 - val_loss: 0.2438 - val_accuracy:
0.9306
Epoch 98/100
2304/2304 [==============================] - 6s 3ms/step - loss: 0.2727 - accuracy: 0.9171 - val_loss: 0.2061 - val_accuracy:
0.9375
Epoch 99/100
2304/2304 [==============================] - 6s 3ms/step - loss: 0.2182 - accuracy: 0.9327 - val_loss: 0.1370 - val_accuracy:
0.9549
Epoch 100/100
2304/2304 [==============================] - 6s 3ms/step - loss: 0.2160 - accuracy: 0.9336 - val_loss: 0.1870 - val_accuracy:
0.9167
```

Fig. 4. Loss after each epoch of our model

Since our model is a predictive algorithm as it predicts the emotions of human speech, we must need to evaluate the model before using in real time. Evaluation is nothing but checking the extent up to which we can use our developed model. There are many evaluation criteria present in Keras. We chose our evaluation metric as 'Accuracy'

The reason we chose accuracy is that it will give accurate results when all the classes in our dataset has same number of records in each folder. Since our dataset contains equal number of records in each folder, we proceeded using this as our evaluation metric. The accuracy graph of our model is as shown in below figure.

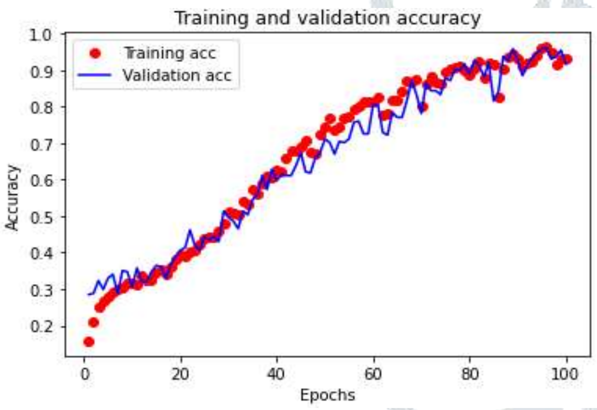


Fig. 5. Accuracy Graph of our model

The mathematical formula behind 'Accuracy metric' is as follows.

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)}$$

The activation functions we used in our two Dense layers is 'ReLU'. ReLU stands for rectified linear unit, and it is a type of activation function. Mathematically, it is defined as *y = max (0, x)*. Visually, it looks like the following:

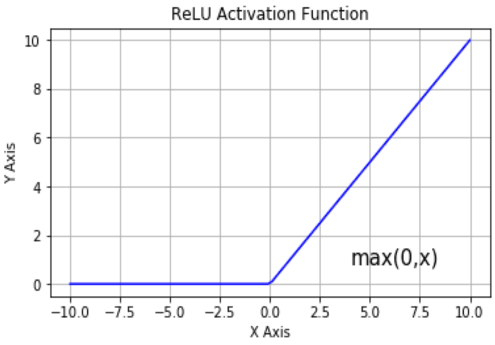


Fig. 6. ReLU Activation Function Graph

The activation function we used in our last layer i.e.; Dense layer is 'SoftMax'. SoftMax activation function is used when there are more than two classes present in our dataset or when our model should predict more than two class labels and our model need to predict eight emotion so we have used SoftMax Activation Function. The mathematical formula related to 'SoftMax' is as follows.

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

6. REAL TIME EMOTION RECOGNITION

The next phase after training our model is recognizing the emotions using real time Audio feed from computer microphone. Emotion recognition from real time data is done by importing the trained model using Keras. We need to access the microphone of our laptop/computer with the use of Sounddevice. The live audio recording using sounddevice module is sent for pre- processing of audio. Our system microphone is capable of recording 44,100 audio samples per second. However it is not unusual to see 96,000 samples **a second** with some digital audio formats. So, our algorithm sends around 48,000 samples per second for pre-processing. Usually, the audio extracted from computer microphone is of size 44.1 kHZ with 3 channels. Initially while training our model, we set the size of our audio to 48,000 multiplied by 0.8 sec . So, we need to resize the raw feed in order to sent it to Detection algorithm.

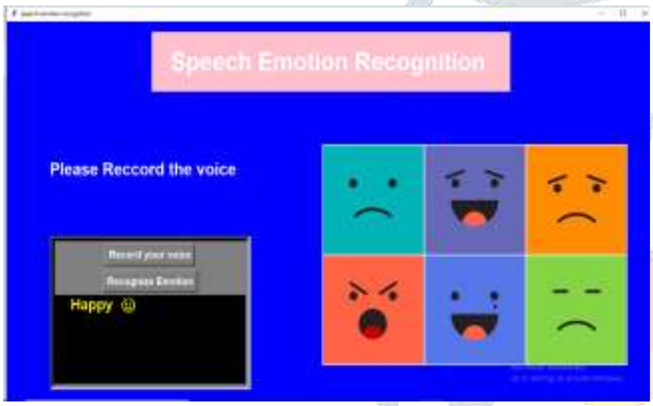
After pre-processing of audio samples is completed, then it will extract the mfcc and all other related features from audio, then the detection algorithm will be able to analyse the features and then finally it will detect the exact emotion of the speaker in real time.

7. RESULTS AND DISCUSSIONS

A real-time Speech Emotion Recognition using Neural Networks named LSTM is introduced. In this paper,Speech Emotion recognition is done to help easily find out the emotion of customers in call centers and many other wide applications. This system showed good results in recognizing Emotions of particular person and shows results with appropriate Emojis. With the help of Neural Networks, in particular Recurrent Neural Networks we were able to develop a LSTM model and thereby easing the work of emotion detection.

Neural Networks results in more accurate recognition predictions compared to other methods of speech emotionl recognition. I strongly believe neural networks as an ideal way to solve this problem.

Below figure shows the execution of our Speech emotion recognition model.



Happy 😐

Fig. 7. Emotion recognition of real time audio

As soon as the recognition of emotion is done, the user will be able to check his/ her emotion through an tkinter interface along with related emojis.

8 CONCLUSION

In this paper, a real-time Deep learning based Speech Emotion Recognition system was built. The use of Recurrent Neural Networks that is an LSTM network helps our model to recognize the emotion of an audio file or emotion from particular persons speech.

The system achieved a maximal accuracy of about 93% for training and 91% for the validation set. Our model will also recognize emotions of speech files that aren't trained by our model.

Through this project we have shown that how to implement Speech Emotion Recognition through deep learning. Our project give ability to a machine to determine the emotions of human through his speech so that there will be a better communication between humans and machines.

If we merge speech emotion recognition and facial emotion recognition in the future, then we can get more precise emotional response and we can build a perfect project for emotion detection.

speech emotion recognition and understanding will eventually show the way to true artificial intelligence.

8. REFERENCES

1. Booth, P.A. An Introduction to Human-Computer Interaction; Psychology Press: Hove, UK, 1989.

2. Harper, E.R.; Rodden, T.; Rogers, Y.; Sellen, A. Being Human: Human-Computer Interaction in the Year 2020; Microsoft Research: Redmond, WA, USA, 2008; ISBN 0955476119.

3. Cambria, E.; Hussain, A.; Havasi, C.; Eckl, C. Sentic computing: Exploitation of common sense for the development of emotionsensitive systems. In Development of Multimodal Interfaces: Active Listening and Synchrony; Springer: Berlin/Heidelberg, Germany, 2010; pp. 148–156.

4. Patil, K.J.; Zope, P.H.; Suralkar, S.R. Emotion Detection From Speech Using Mfcc and Gmm. Int. J. Eng. Res. Technol. (IJERT) 2012, 1, 9.

5. Hassan, A.; Damper, R.I. Multi-class and hierarchical SVMs for emotion recognition. In Proceedings of the INTERSPEECH 2010, Makuhari, Japan, 26–30 September 2010; pp. 2354–2357.

6. Lin, Y.L.; Wei, G. Speech emotion recognition based on HMM and SVM. In Proceedings of the 2005 International Conference on Machine Learning and Cybernetics, Guangzhou, China, 18–21 August 2005; Volume 8, pp. 4898–4901.

7. Nicholson, J.; Takahashi, K.; Nakatsu, R. Emotion Recognition in Speech Using Neural Networks. In Proceedings of the 6th International Conference on Neural Information Processing (ICONIP '99), Perth, Australia, 16–20 November 1999.

8. Schüller, B.; Rigoll, G.; Lang, M. Speech emotion recognition combining acoustic features and linguistic information in a hybrid support vector machine-belief network architecture. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Montreal, QC, Canada, 17–21 May 2004.

9. France, D.J.; Shiavi, R.G.; Silverman, S.; Silverman, M.; Wilkes, D.M. Acoustical Properties of Speech as Indicators of Depression and Suicidal Risk. IEEE Trans. Biomed. Eng. 2000, 47, 829–837. [CrossRef] [PubMed]

10. Hansen, J.H.; Cairns, D.A. ICARUS: Source generator based real-time recognition of speech in noisy stressful and Lombard effect environments. Speech Commun. 1995, 16, 391–422. [CrossRef]