SPEECH EMOTION DETECTION
ML - 15

Ayush Awasthi
Atharva Nanoti
Shaurya Khetarpal
Jasmer Singh Sanjotra

1. The Problem Statement

The objective of this project was to develop a robust Speech Emotion Detection system. The main task was to analyze the speech signals and classify them into various emotional states like 'happy', 'angry', 'sad' etc. We needed to train the model and make it as accurate as possible in order to apply it in real-time. The ability to detect the emotion of a speaker from their speech is a valuable tool in a variety of applications. For example, in customer service, it can be used to identify frustrated or angry customers so that they can be quickly and effectively addressed. In healthcare, it can be used to monitor patients' emotional state and identify those who may be at risk of suicide or other self-harm.

2. Initial Approach

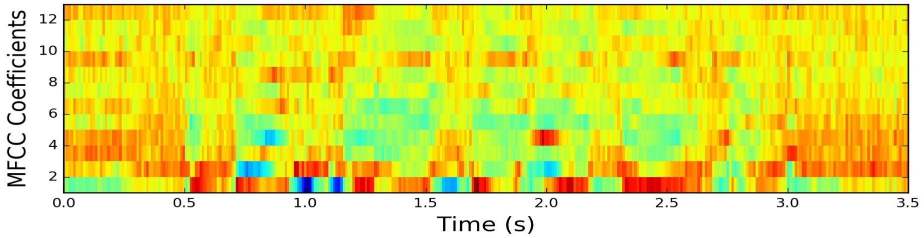
In the initial stages of the project, our plan was as follows:

- **Data Collection and Analysis:** We tested out several different datasets like RAVDESS, TESS, CREMA, SAVEE, IEMOCAP, LSSED etc. We tried to work things out like which dataset would be best to train so that our model works in real-time. Also, we checked the category-wise distribution of the different datasets so as to ensure that our datasets have equality among all the emotions.
- **Feature Extraction:** We also tried to learn about several features that an audio file might have like Mel-Frequency Cepstral Coefficients (MFCCs), Zero-Crossing Rate (ZCR), Chroma-STFT, Root Mean Square Value, Mel Spectrogram etc. We tested out several of these features and eventually ended with taking MFCCs as the main feature.
- **Model:** In the initial stages, we planned on developing a CNN model for training these datasets.

3. Final Iteration and Metrics

In the final iteration, we made the following decisions based on our experiments and evaluation metrics:

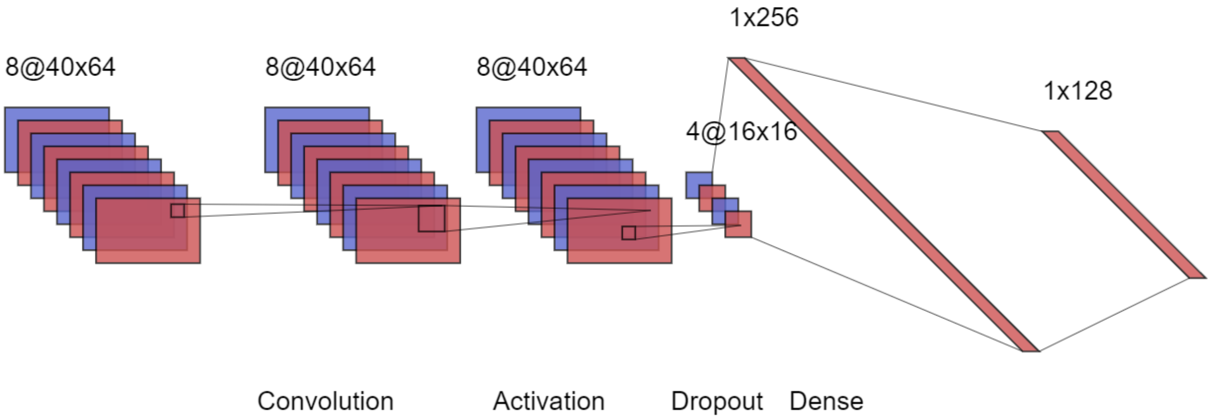
- **Data Preprocessing:** We loaded in all the audio samples of the TESS and SAVEE datasets by the librosa library of python and applied augmentation techniques like noise induction, pitch shift, time stretch etc. to increase the dataset's diversity and improve generalization.
- **Feature extraction:** After comparing different feature extraction methods, we found that MFCCs provided the best results in terms of capturing relevant speech characteristics for emotion recognition.



MFCCs aim to represent the spectral characteristics of audio signals in a more compact and informative way. The process involves computing the Mel-frequency spectrum of an audio signal, which is a representation of its frequency content. Then, the logarithm of the Mel-frequency spectrum is taken to convert it into the cepstral domain. Finally, a set of coefficients, known as MFCCs, are derived by performing a discrete cosine transform on the cepstral data. These coefficients effectively capture the essential features of the audio signal, making them valuable for tasks like speech recognition, speaker identification, and audio classification.

- **Model Selection:** Among the various CNN architectures tested (e.g., CNN with different numbers of layers, kernel sizes, and pooling strategies), a specific CNN configuration achieved the maximum accuracy and we used the same.

The working of CNN models is explained below:
Convolutional Neural Networks (CNNs) are deep learning models commonly used for image recognition and computer vision tasks. They employ convolutional layers to extract features from input data by sliding small filters over the input and creating feature maps. Pooling layers reduce the spatial dimensions of these maps, retaining essential information. The output is then flattened and passed through fully connected layers to make the final decision about the input's class. Non-linear activation functions add complexity to the model, and backpropagation is used to adjust internal parameters during training, enabling the network to learn and make accurate predictions. Overall, CNNs excel in learning hierarchical patterns and features from data, making them powerful tools for various computer vision applications.



Other models we have tried:

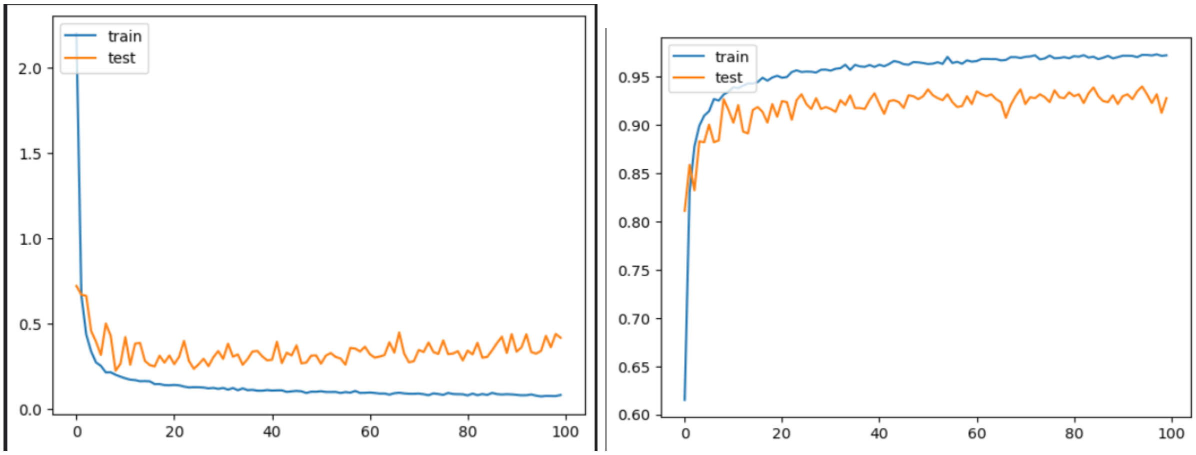
Model	Accuracy
LSTM	54%

Bi-directional LSTM	61%

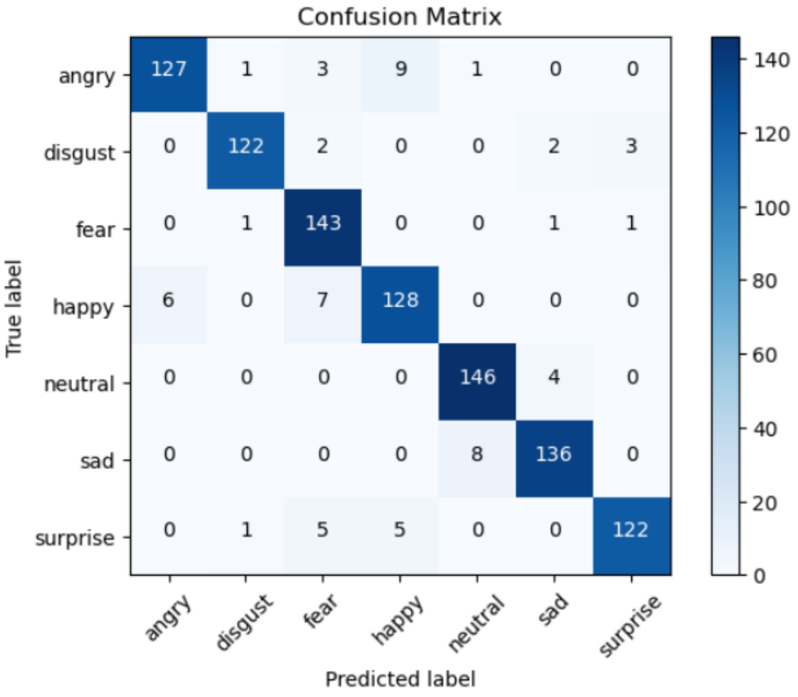
- **Model Training:** We trained the selected CNN model using rmsprop optimizer and categorical cross-entropy loss. We used early stopping and learning rate reduction strategies to prevent overfitting and improve convergence. We trained the model with 100 epochs with EarlyStopping enabled.

Evaluation Metrics: The final model achieved an accuracy of 92% on the test set, which indicated its effectiveness in recognizing emotions in speech. The following are the graphs of the metrics.

Loss:



Confusion Matrix:



	angry	disgust	fear	happy	neutral	sad	surprise
angry	127	1	3	9	1	0	0
disgust	0	122	2	0	0	2	3
fear	0	1	143	0	0	1	1
happy	6	0	7	128	0	0	0
neutral	0	0	0	0	146	4	0
sad	0	0	0	0	8	136	0
surprise	0	1	5	5	0	0	122

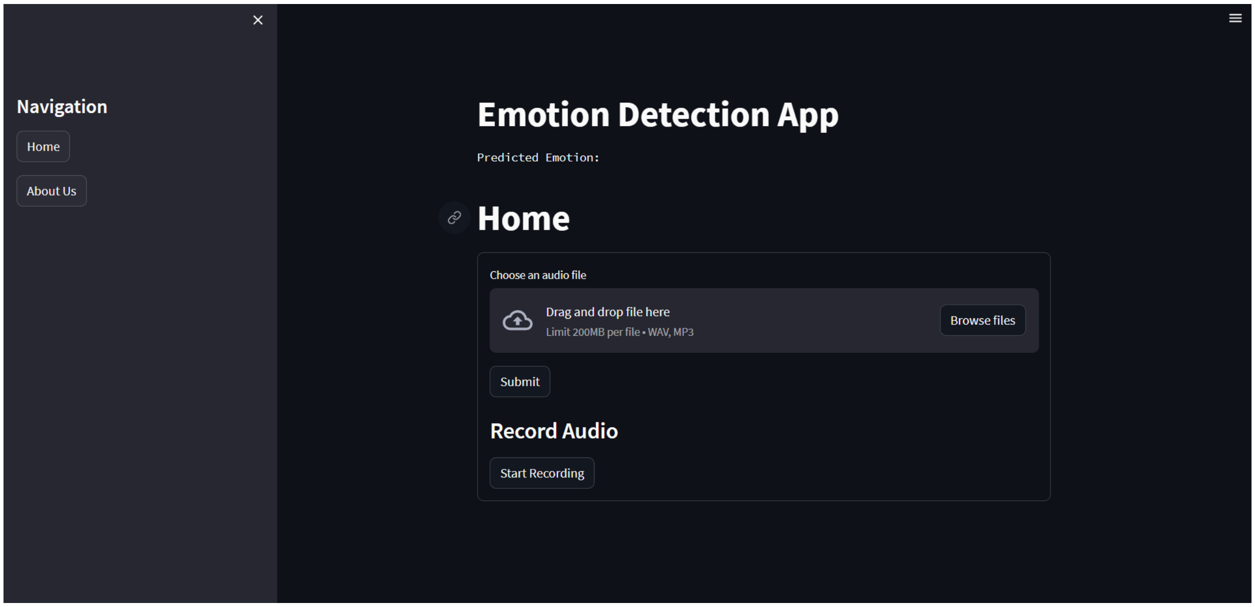
4. Sample Results and Demo

Here are some sample results from the Speech Emotion Detection system:

Audio File	Actual Emotion	Predicted Emotion
TESS ‘OAF_back_happy.wav’	Happy	Happy
TESS ‘OAF_bought_disgust.wav’	Disgust	Disgust
SAVEE ‘DC_a01.wav’	Angry	Angry

Demo (Streamlit Deployed App)

We deployed the model on a streamlit website which has the feature of using the audio files from local storage or recording the audio in real-time.



Emotion Detection App

Predicted Emotion:

Home

Choose an audio file

Drag and drop file here

Limit 200MB per file • WAV, MP3

Record Audio

5. Possible Improvements and Conclusion:

There could’ve been following improvements in our project:

1. Though our model provides a pretty decent accuracy on the validation dataset but it’s results are not up to mark on our voices. This can be improved if we make our own diverse dataset with our accents and voice modularities. This will help the model to learn more diverse features.
2. We can also consider using pre-trained models for this task and fine-tune them on dataset made by us or any other dataset so as to achieve more promising results. We did try some pre-trained models on huggingface but sadly their results on our voices were also not promising.

6. References

- **SAVEE Dataset**
- **TESS Dataset**
- **MFCCs**
- **Basics of CNN**
- **CNN - Pooling Layers**
- **CNN - Padding and Stride**

7. Conclusion:

The Speech Emotion Detection project successfully implemented a CNN-based model and deployed it on Streamlit for emotion recognition in speech signals. The model achieved a very good validation accuracy, demonstrating its potential in practical applications, such as sentiment analysis and human-computer interaction. With further improvements and enhancements, the system can be optimized for even more accurate and efficient emotion detection in speech.