Speech Emotion Recognition(SER) using CNN

Priyanka Bansal
203070005
dept. of Electrical Engineering
Indian Institute of Technology Bombay

Vedant Kandoi
203070003
dept. of Electrical Engineering
Indian Institute of Technology Bombay

Mikhil Gupta
203079016
dept. of Electrical Engineering
Indian Institute of Technology Bombay

***Abstract*—This project aims to build a fast, fairly accurate, and robust model for speech emotion recognition. We use 3 datasets as mentioned in section III for training and validation. We work with different kinds of features (Mel Spectrogram, MFCC, Chroma) and use CNN to classify the emotions. In the end, we achieve 60% accuracy with noise augmentation.**

***Index Terms*—Convolutional Neural Network (CNN), Mel Frequency Cepstral Coefficient (MFCC), Short Time Fourier Transform (STFT)**

I. Introduction

Human speech is one of the most common ways to express emotion [7]. Emotions are important to communicate so recognizing them in the modern world on a digital platform is vital.

It is challenging to find emotion from speech as the best features are not easily identifiable. The subjective nature of emotions also increases the difficulty.

We define a system that recognizes emotion from the frequencies and other sound features by feeding them into a Convolutional Neural Network(CNN). We use PyTorch [8] which is a python library for deep learning on irregular input data. The key feature is that it can use GPU which gives faster training and testing.

It is seen in the end that our model gives sufficient accuracy in determining emotions and takes relatively less time to train.

II. Background and Previous Work

CNN is a regularized version of a neural network. [9] Regularization is usually done by penalizing parameters or dropout. CNN takes advantage of patterns and local correlations in data thereby decreasing the fully connected nature of a neural network. This also avoids over fitting. CNN is generally used for image classification and gives very accurate results.

This work is inspired by [1]. In [1], Keras is used for speech emotion recognition on the same datasets that we used, whereas we use PyTorch as it can use faster GPU. Various other works have also been done on SER - [7], [10]

III. Datasets

We are using 3 datasets:

- The Ryerson Audio-Visual Database of Emotional Speech and Song(RAVDESS) [2].
  It contains 60 trials per actor and 24 actors(12 male, 12 female), 1440 total. There are 8 expressions - calm, happy, sad, angry, fearful, surprise, disgust, and neutral each with 2 levels of intensity.
- Crowd-Sourced Emotional Multimodal Actors Dataset (CREMA-D) [3]
  It contains 7442 clips from 91 actors expressing 6 emotions - anger, disgust, fear, happy, neutral, and sad.
- Toronto emotional speech set (TESS) [4]
  This contains 2800 total audio files by 2 actresses expressing 7 emotions - anger, disgust, fear, happiness, pleasant surprise, sadness, and neutral.

For consistency, we keep 6 emotions from all datasets - anger, disgust, fear, happy, neutral, and sad. The datasets were then combined into a csv file with path to file and emotion as columns.

IV. Procedure and Experiments

A. Data Processing

1) Visualization: The wave-plot and the log-frequency power spectrogram of an audio file can be seen in the figures 1 and 2. Fig. 1 shows the time vs amplitude plot of the audio. Fig. 2 shows the time and frequency vs amplitude plot.

2) Endpointing: The process of endpointing is done to remove the silence in the audio files. Fig. 3 shows the resulting wave-plot of the process. This is done to minimize unnecessary information.

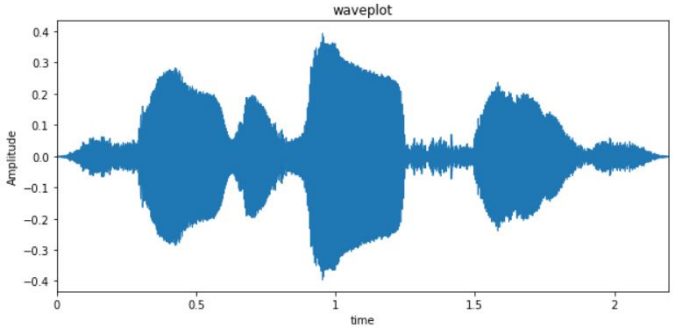

Fig. 1. Waveplot

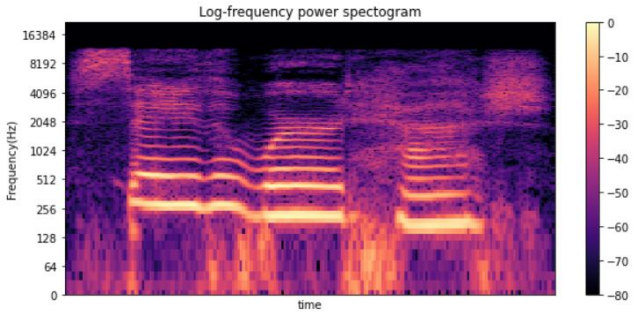

Fig. 2. Spectrogram

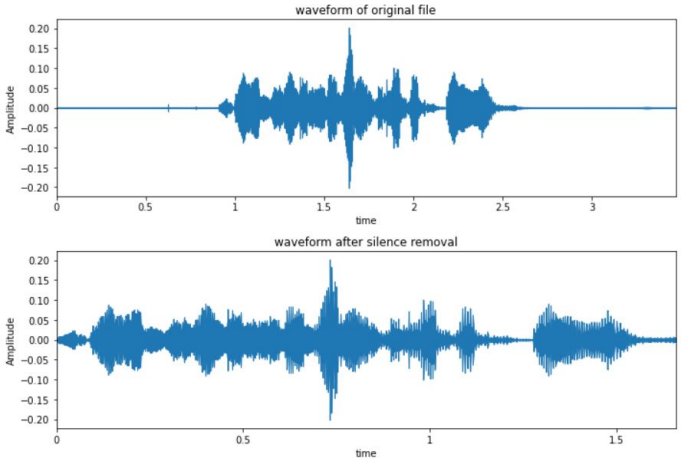

Fig. 3. Endpointing

3) Pre-emphasis: Most of the energy in the voice samples tends to be in the lower frequencies. Pre-emphasis is done to boost the high frequency energy. Speech from the mouth has a 6dB decay per octave, pre-emphasis removes this decay. [6]

B. Feature Extraction

1) MEL-Spectrogram: As mentioned before CNN can be used accurately to classify images. So, we convert the audio file into an image equivalent. An image is usually of the form 1920x1080 with 3 channels(RGB). For audio, a time sample x frequency sample with 1 channel image can be formed. For this, we are using the MEL spectrogram of an audio file as its feature. A MEL spectrogram is a 2-D representation (time x frequency) of audio. It gives us the power(in dB) of all frequencies at a certain time. Fig. 2 is an example of a MEL spectrogram.

2) MFCC [5]: Mel-frequency cepstrum is often used for speech or audio recognition. Mel Scale is a logarithmic scale of frequency given by

$$Mel(f) = 2595 \log_{10}(1 + \frac{f}{700})$$

The Mel-frequency Cepstral Coefficients (MFCCs) are calculated as

$$C_n = \sum_{k=1}^{K} \sqrt{\frac{2}{k}} (\log S_k) \cos\left[n(k-0.5)\pi/k\right]$$

for $n = 1, ..., N$ where S_k is the output of filter banks and N is the total number of samples in 1 audio unit (taken here as 20ms).

3) Chroma [11]: Chroma feature gives the tonal content of an audio signal. It is typically represented by a vector of length 12 which shows the energy of each pitch class. This vector is obtained by using short-time Fourier Transform(STFT), Constant-Q transform(CQT) and Chroma Energy Normalized (CENS).

All features are extracted and combined to form a large dataset of more than 10,000 audio files which is then used for training and validation.

C. Training and Validation

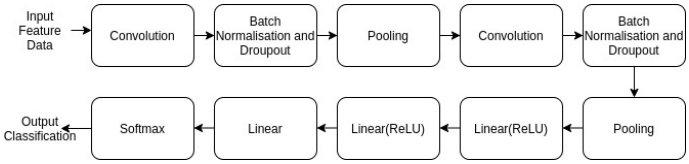

Fig. 4. CNN model layers

Convolutional Neural Network (CNN) model: Fig. 4 shows the topology of the model we have used.

For Model 1, MEL-Spectrogram is used to train the CNN model. Frame size of 20ms and window size of 30ms are used to extract features. This results in 128×134 feature size for a sample. Features are then fed to Conv2D layer consisting of 1 input channel and 20 output channels with kernel size 3. After passing through 2 convolution layers, the resultant is passed through 3 linear layers. For first two linear layers, ReLu is used as an activation function. For output, layer Softmax is used as an activation function. SGD optimizer and Cross-Entropy loss are used for training.

For Model 2 and 3, MEL-Spectrogram, MFCCs, and Chroma-stft are obtained for a sample file and then stacked together to be used as features. For all the features, the average value is taken over all the timesteps. This resulted in 160 features for a sample file. The Model used here is similar to model 1 except the fact that Conv1D layers are used here instead of Conv2D layers.

V. Results

A. Using only MEL spectrogram as feature

Firstly, we used only MEL spectrogram as a feature to get a basic overview of what we can expect from different models. After running the training model for 100 epochs, we observe that the validation accuracy saturates at approximately 48% as seen in Fig. 5. The loss saturates at about 1.53 as seen in Fig.

6. This can still be improved. This model is not very robust either.

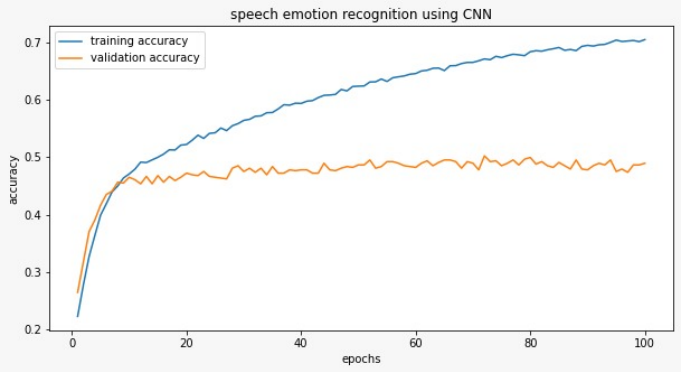

Fig. 5. Accuracy using MEL spectrogram as feature

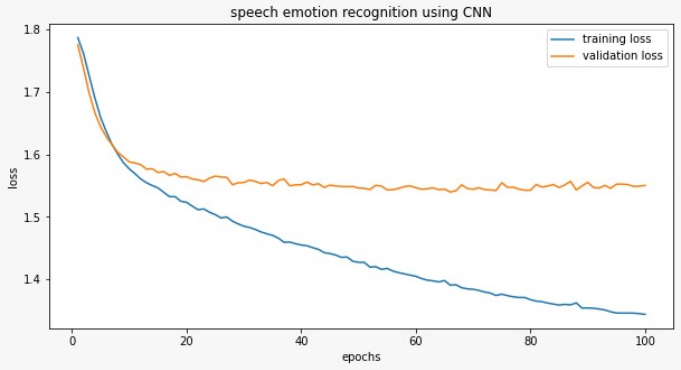

Fig. 6. Loss using MEL spectrogram as feature

B. Using combined MFCC, MEL spectrogram and Chroma as features

We then use all 3 extracted features - MFCC, MEL spectrogram, and Chroma. After running this for 500 epochs, we see that the validation accuracy saturates at about 57% (Fig. 7) and loss at 1.44 (Fig. 8). This shows a significant improvement in accuracy. We can also see that the robustness has significantly improved. The training and validation accuracy is very close.

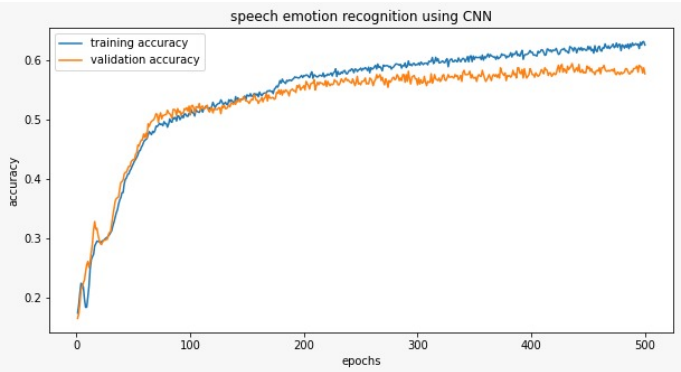

Fig. 7. Accuracy using MEL spectrogram, MFCC and Chroma as feature

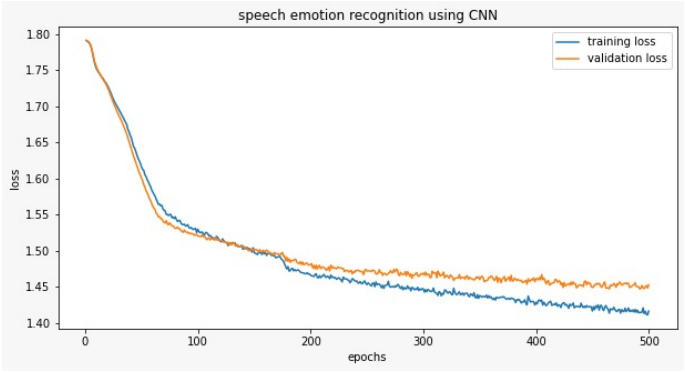

Fig. 8. Loss using MEL spectrogram, MFCC and Chroma as feature

C. Using combined features with noise augmentation

For increasing the dataset and to avoid the overfitting problem, data augmentation is carried out. White noise is added to all the sample files. Original wav files along with noise augmented files were then used together to train the model. From Fig. 9, we can see that validation accuracy saturates around 60%.

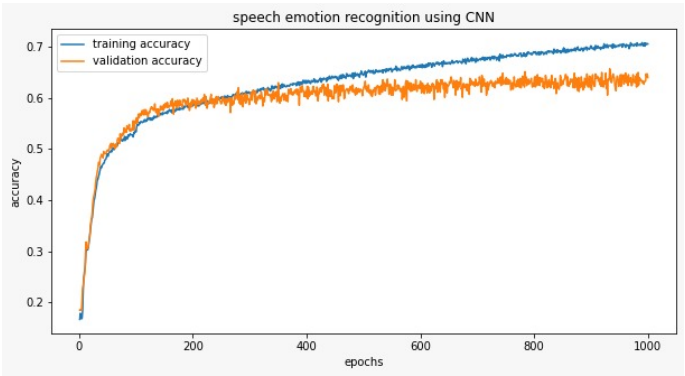

Fig. 9. Accuracy using MEL spectrogram, MFCC and Chroma as feature with augmentation

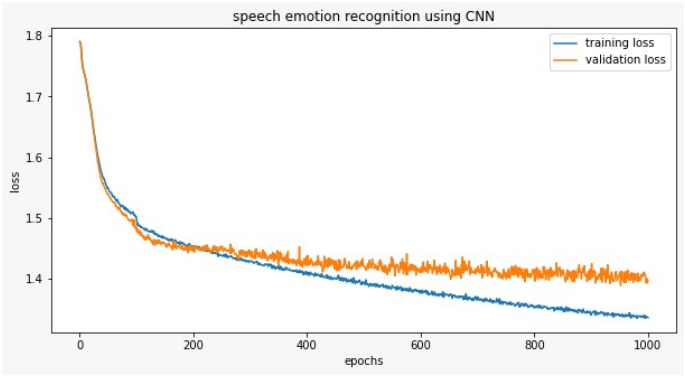

Fig. 10. Loss using MEL spectrogram, MFCC and Chroma as feature with augmentation

VI. Conclusions

In this project, we used 3 models with different kinds of features - Mel spectrogram, combined 3 features, and

combined 3 features with noise augmentation. We see that the accuracy of the models are 48%, 57% and 60% respectively. We also achieved good robustness.

We also compared our work to [1] which used Keras. Our model proved to have better accuracy(60% over 49%) and trained much faster.

The limitation of our work is that our model is too basic and data is too limited. There are many more datasets which we could not use due to resource constraints. Attention or stacked long short-term memory(LSTM) might perform better and give better accuracy.

VII. Statement of contributions

P. Bansal researched on audio features MFCC and chroma. M. Gupta researched on Mel Spectrogram. V. Kandoi researched on Pytorch and Keras models for CNN. V. Kandoi and M. Gupta dived into the creation and pre-processing of the dataset. P. Bansal created the training model. The team worked together for training and validation of the data. Report writing was done by V. Kandoi with input from others. Video making was done by P. Bansal. The team also looked into various alternatives and possible improvements(Keras, LSTM, etc.) We shared thoughts and ideas often using virtual meets and the above is just an approximation of contribution. All 3 of us were engaged in all activities from beginning to end.

References

[1] Ritzing, "Speech Emotion Recognition with CNN" https://www.kaggle.com/ritzing/speech-emotion-recognition-with-cnn

[2] "The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)" by Livingstone and Russo is licensed under CC BY-NA-SC 4.0. https://doi.org/10.1371/journal.pone.0196391

[3] The Crowd-sourced Emotional Mutimodal Actors Dataset (CREMA-D) is made available under the Open Database License. https://github.com/CheyneyComputerScience/CREMA-D

[4] TESS is published under Creative Commons license Attribution-NonCommercial-NoDerivatives 4.0 International. https://doi.org/10.5683/SP2/E8H2MF

[5] Min Xu; et al. (2004). "HMM-based audio keyword generation" (PDF). In Kiyoharu Aizawa; Yuichi Nakamura; Shin'ichi Satoh (eds.). Advances in Multimedia Information Processing – PCM 2004: 5th Pacific Rim Conference on Multimedia

[6] Himani Chauhan et al,"Voice Recognition" International Journal of Computer Science and Mobile Computing, Vol.4 Issue.4, April- 2015, pg. 296-301 https://ijcsmc.com/docs/papers/April2015/V4I4201563.pdf

[7] "Speech Emotion Recognition(SER) through Machine Learning by Analytics Insight https://www.analyticsinsight.net/speech-emotion-recognition-ser-through-machine-learning/

[8] Pytorch Documentation https://pytorch.org/docs/stable/index.html

[9] Wikipedia - Convolutional Neural Network https://en.wikipedia.org/wiki/Convolutional_neural_network

[10] Speech Emotion Recognition with Convolutional Neural Network by Reza Chu https://towardsdatascience.com/speech-emotion-recognition-with-convolution-neural-network-1e6bb7130ce3

[11] "Chroma Feature Extraction" by M. Kattel, A. Nepal, A. K. Shah, D. Shrestha, Department of Computer Science and Engineering, School of Engineering, Kathmandu University, Nepal https://www.researchgate.net/publication/330796993_Chroma_Feature_Extraction