Emotion Detection System using CNN

Naveen Kumar Rajesh Steven Yang Shuban Ranganath

1. Introduction:

The project aims to develop an emotion detection system for speech data utilizing the RAVDESS and CREMA datasets. Emotion detection from speech data is a crucial task with numerous applications, including human-computer interaction and mental health monitoring. Our model selection leans towards a CNN architecture for its suitability in analyzing sequential data. Extracting meaningful parameters from audio files is crucial, so we focused on four key metrics: zero crossing rate, mel-frequency cepstrum coefficients (MFCC), mel spectrogram, and a root mean square value. These parameters offer insights into sound wave dynamics, frequency distributions, and volume, all vital for discerning emotional nuances in speech.

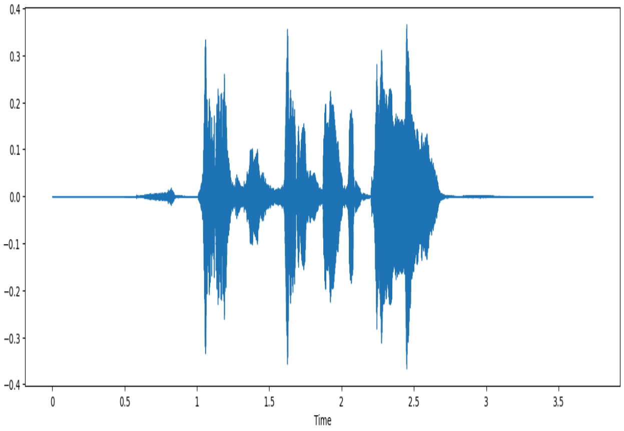
2. Data Exploration:

2.1 RAVDESS Dataset:

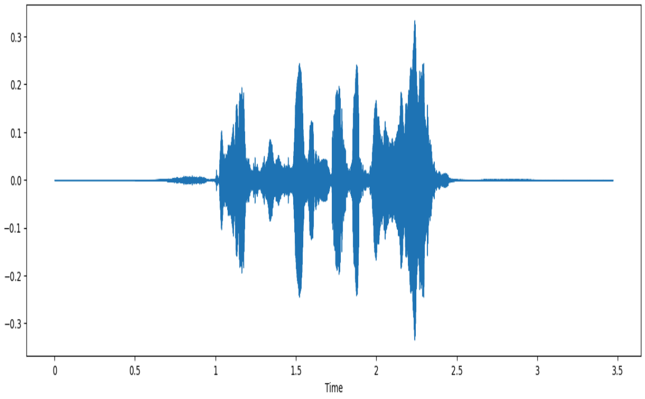
The RAVDESS dataset is a comprehensive collection of audio and video recordings designed for studying emotional expression. It features a range of vocalizations from 24 professional actors, both male and female, spanning different ages and ethnic backgrounds. It contains recordings of actors portraying different emotions, including neutral, calm, happy, sad, angry, fearful, disgust, and surprised, while speaking two specific prompts: "Kids are talking by the door" and "Dogs are sitting by the door". The richness of this dataset lies in the consistency and quality of recordings, making it a valuable resource for our application of creating an emotion recognition model.

The best way to understand the data is through visualizations, so below are the wave graph and spectrogram of 2 different sample audio recordings from the dataset:

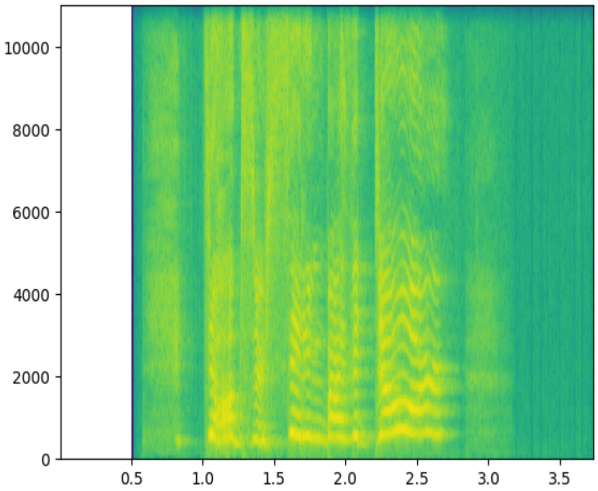
Wave-Graph of Actor-14 “happy” audio:



Wave-Graph of Actor-14 “fear” audio:



Spectrogram of Actor-14 “happy” audio:

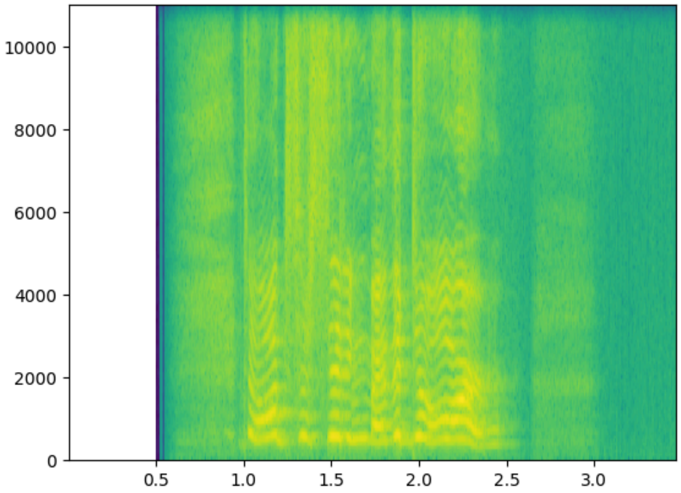


The spectrograms show us visually what the frequency of the audio looks like over time. While both the graphs here are very similar, looking closely at the light yellow color, we can see slight intricacies in the graphs and how they are different. This allowed our model to capture more of the intricacies in a human voice and also focus more on the relevant details and try to rule out any sort of distortions or background noises.

2.2 CREMA Dataset:

The CREMA dataset is a collection of both visual and audio recordings of different actors performing. The audio recordings capture the emotions through speech, while the video has additional visual cues. For this project, however, we only needed the audio recordings. The dataset also was very large, which would lead to better accuracy, but to save processing time we decided to only use half of it. Similar to the RAVDESS dataset, we can also visualize the audio files through wave graphs and spectrograms.

Spectrogram of Actor-14 “fear” audio:



3. Model Exploration:

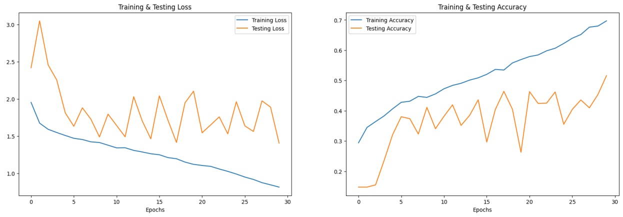
3.1 CNN Model:

One of the models we used is a Convolutional Neural Net. It has exactly 13 layers, including convolutional, batch normalization, max-pooling, dropout, flattening, and dense layers, designed to extract hierarchical features from one-dimensional sequential data. We chose this model because of its ability to capture local patterns, exploit translation invariance, and share parameters for efficient learning. They especially excel at extracting relevant information from spectrogram representations. This model ended up having close to 65% accuracy on unseen data. We also learned that as we add more data, the accuracy of the model tends to increase by 5-10%, however doing so adds too much time for feature extraction.

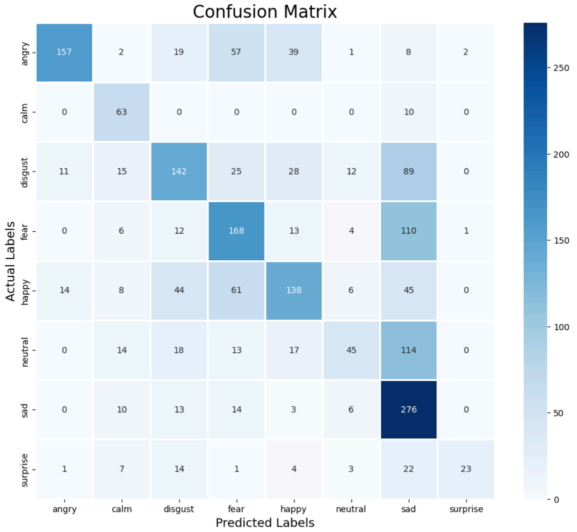
3.2 Underfitting vs. Overfitting:

To address underfitting, we tried to choose features that were sufficiently different from each other so that the model could handle variation among many different samples. We also ensured that our model wasn't too simple, adding more layers and neurons, but not too much to face overfitting issues. Our preprocessing also helped us account for issues with overfitting. By introducing noise and variation to our data, the model should become more generalized when we feed it new data.

Training and test performance:



Confusion matrix showing the accuracy of predictions:



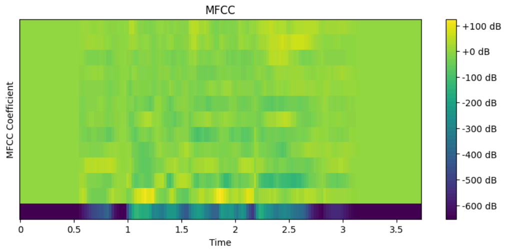
It seems the model/features we chose are particularly good at identifying sad audio. It also seems that the model overcorrects for sad audio, as it runs into confusion with fear, disgust, and neutrality, misclassifying them as fear. Overall, the model showed around 70% accuracy over 30 epochs using the 2 datasets.

4. Data Preprocessing and feature design:

We extracted four different features: zero crossing rate, mel-frequency cepstrum coefficients (MFCC), mel spectrogram, and the root-mean-square value. The zero crossing rate detects the rate at which a sound wave moves from positive to zero to negative and vice versa, which can help detect different intensities in speech. The Mel-frequency cepstrum coefficients measure rate changes in audio frequencies on the Mel Scale, which is the normalized audio frequency scale that aims to separate frequencies equally. It also creates a chart of audio frequencies over time. The root-mean-square value measures the average loudness of the recording. We chose features that would help us visualize each audio recording in different ways so that the model can compare their differences and learn in the training phase. The selected features should show how the audio frequencies change over time, as well as how the audio frequencies and volume are distributed in the recording.

In addition to these features, we perform some preprocessing on our data to make it more generalizable to different scenarios where audio quality may not be perfect, or a person's expression/voice may differ from the data we have. This includes adding noise to the audio files and changing the audio's pitch. Additionally, we crop each audio file to exclude periods of silence at the start and end of the recordings.

Example of MFCC data for a 'happy' track

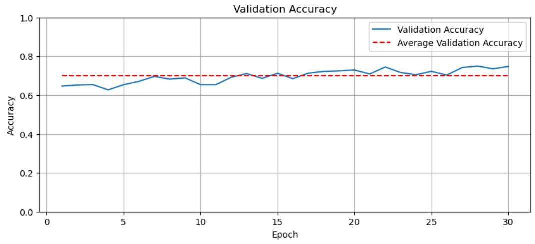


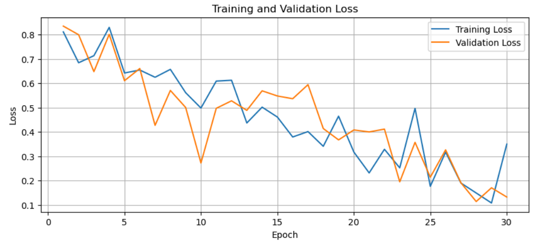
5. Performance Validation:

We conducted a K-fold cross-validation with five folds for our performance validation. This achieved an average validation accuracy of 70% over 30 epochs across the dataset, aligning with our initial model testing. Most importantly, we saw consistent model performance as the mean validation loss remained stable across the epochs. Our training history plots also depict a slight decrease in training and validation losses, displaying effective learning without significant overfitting concerns.

Overall, this validation process highlights our model's capabilities and generalization. By incorporating K-fold cross-validation over diverse data subsets, we ensured an unbiased performance evaluation. We gained a deeper understanding of the model's learning dynamics and predictive behavior through the validation mean accuracy charts and loss graphs. However, access to hardware that can train on more data would greatly help the model's generalizability and accuracy. As a result, we are confident in the reliability of our model for applications in emotional speech classification.

Training/Validation Accuracy and Loss graphs of K-fold:


Validation Accuracy
Validation Accuracy
Average Validation Accuracy
Accuracy
Epoch


Training and Validation Loss
Training Loss
Validation Loss
Loss
Epoch