Emotion Detection in Sound

Final project by Nimrod Dadush

Objective

The primary goal of this project is to train a model capable of accurately detecting emotions conveyed in short voice messages.

The Data

RAVDESS

Ryerson Audio-Visual Database of Emotional Speech and Song

TESS

Toronto emotional speech

CREMA

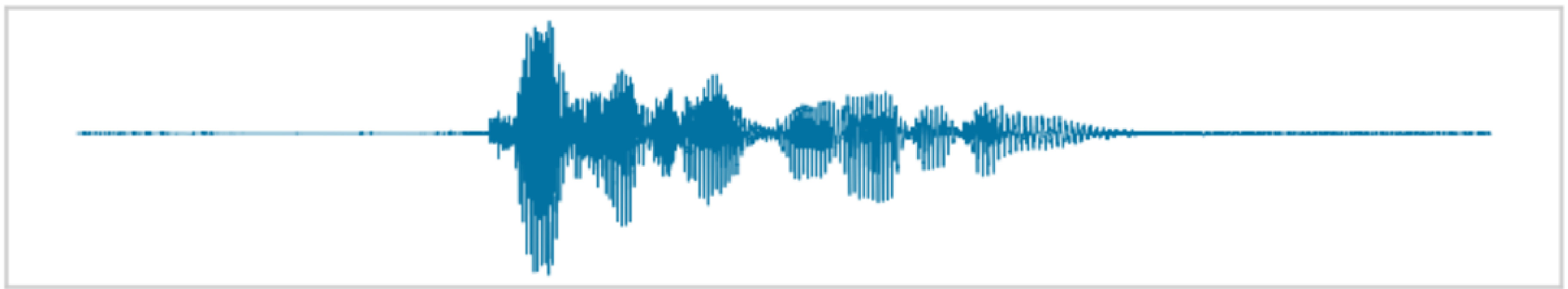
Crowd Sourced Emotional Multimodal Actors Dataset

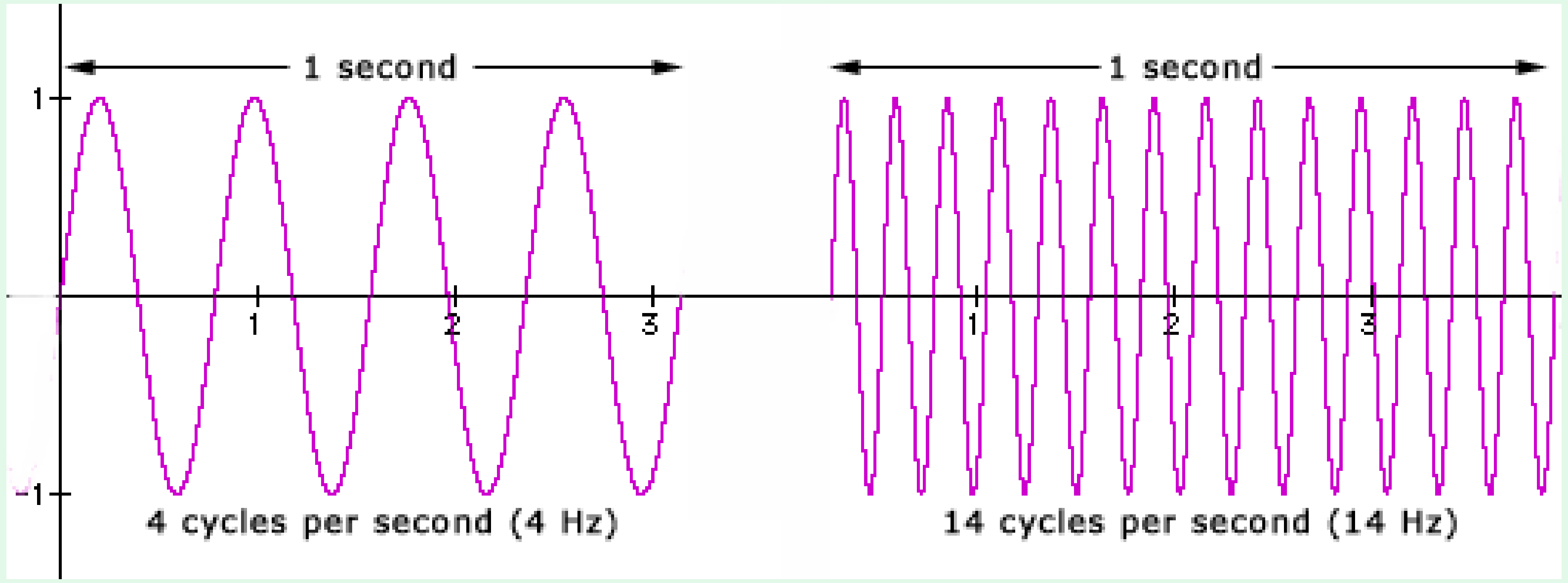
SAVEE

Surrey Audio-Visual Express Emotion

Consisting of 12,000+ wav files where various actors recorded short generic sentences in different emotions. **Neutral - Calm - Sad - Happy - Disgust - Angry - Surprised - Fear**

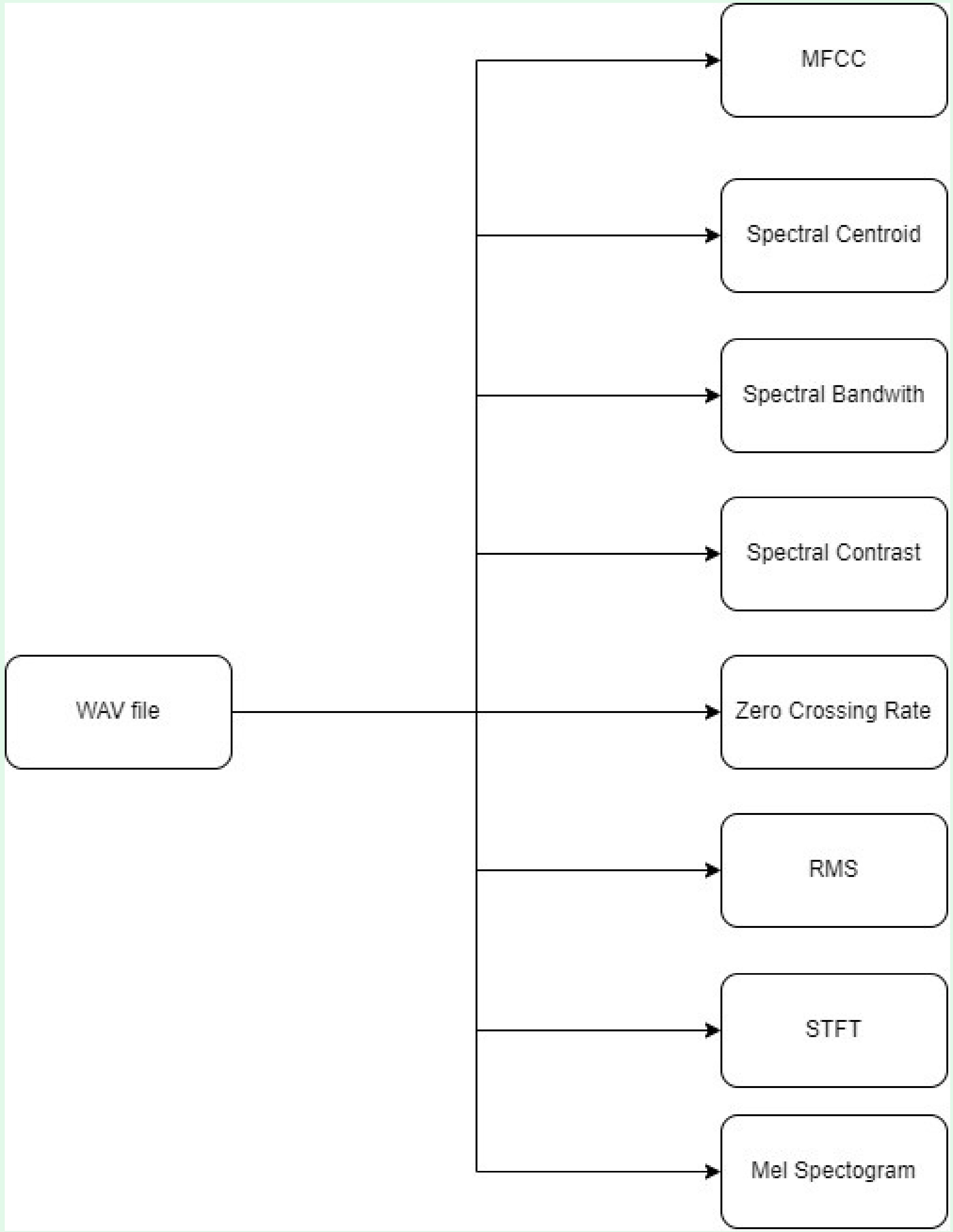
3 Second Sample Audio File





Challengs:

- **Temporal Dependency:** Audio features are inherently time-dependent
- **High Density:** Audio data yields dense feature sets
- **Vast Data Volume:** With a 44 kHz sampling rate over a 3-second audio clip, an extensive dataset of over 1.3 million samples



WAV file

MFCC

Spectral Centroid

Spectral Bandwith

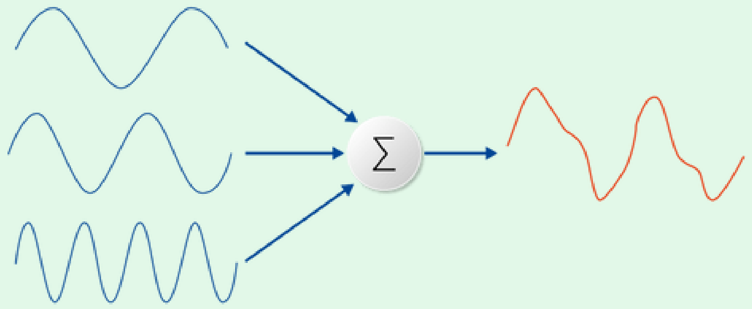
Spectral Contrast

Zero Crossing Rate

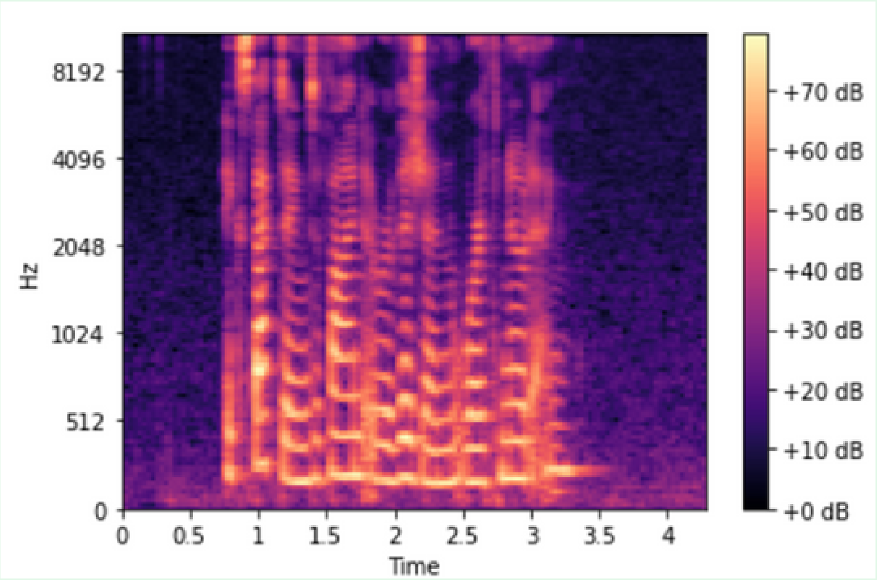
RMS

STFT

Mel Spectogram

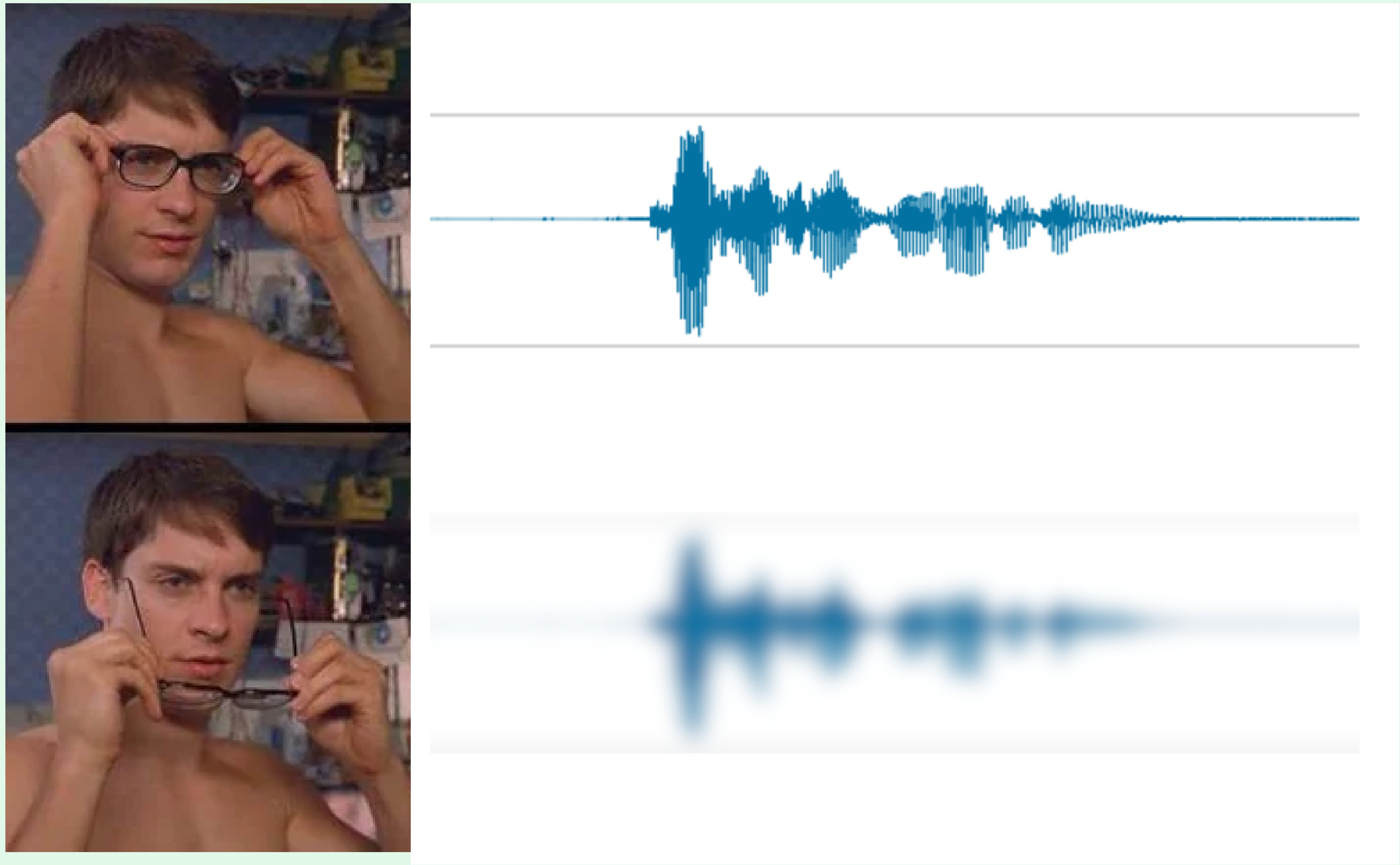


Two Perspectives on Audio: We can approach it from two angles: either as a detailed timeline of events or as a collection of different types of sound waves with specific frequencies.



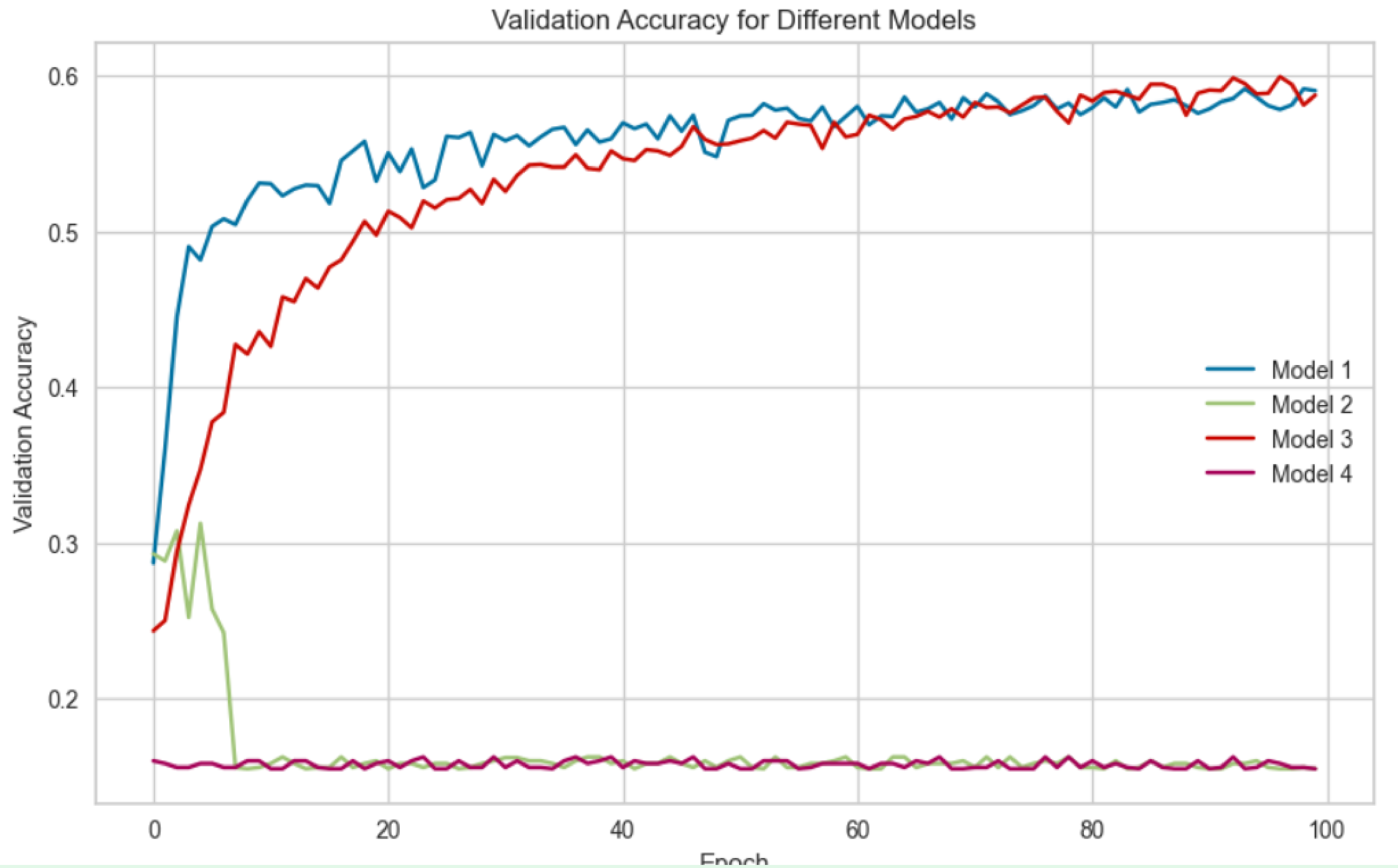
Spectrograms give us the best of both worlds, offering a representation that blends time and frequency details. They help us get both the flow of events over time and the various frequencies present in the audio.

Despite feature extraction reducing data volume, it's important to note that analysis still involves significant data



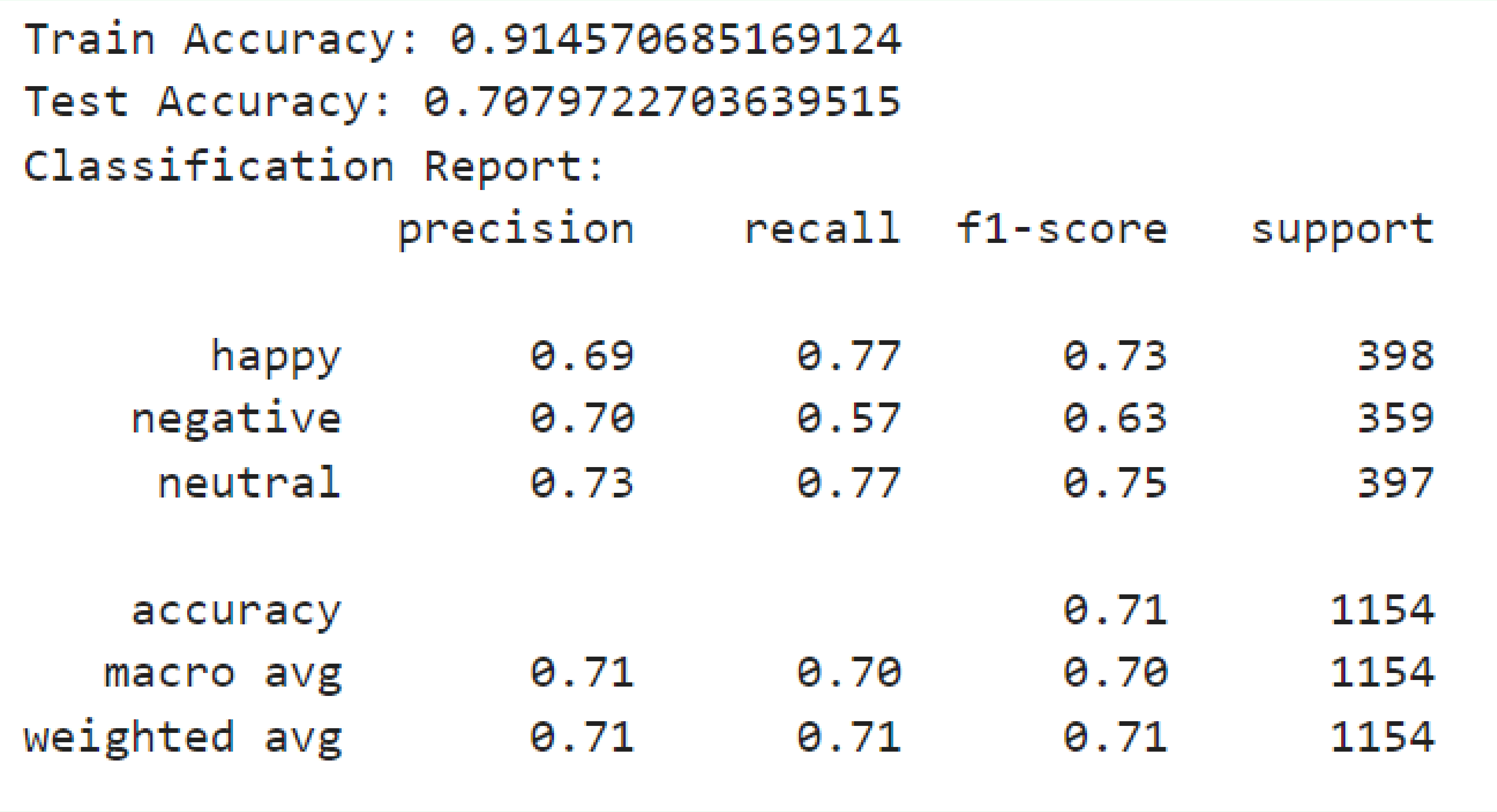
Approach 1:
Simplify the data: used means instead of entire features, which sacrifice granularity.

Approach 2:
Use a more complex and robust model - train a CNN.



Used a CNN model, as it is effective in analyzing data by learning patterns and features directly from it. CNNs are suited for tasks like recognizing and classifying sounds due to their advanced capabilities in processing spatial information.

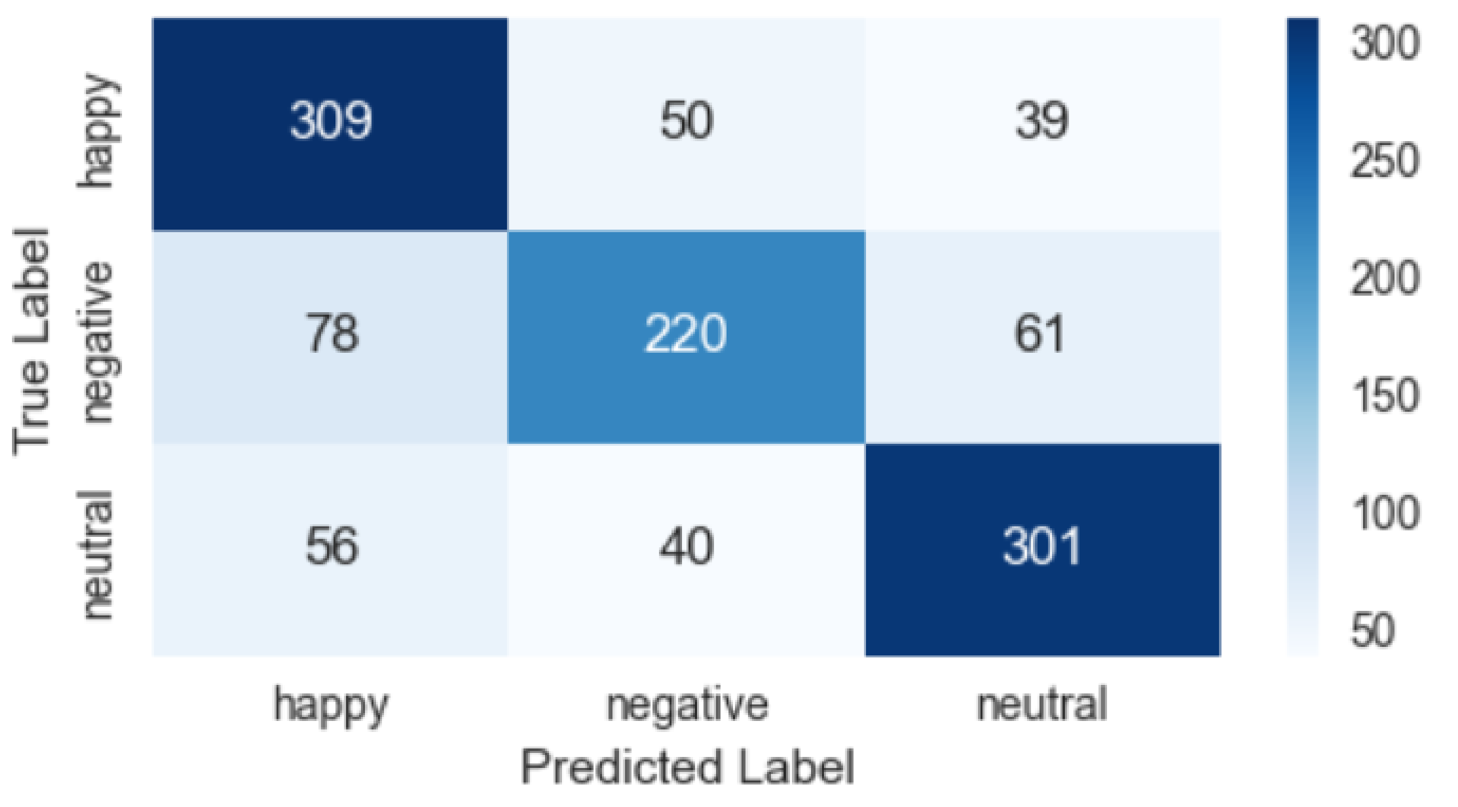
- Created a basic CNN model
- Ran the model four times with different hyperparameter configurations.
- Evaluated the performance of each model and compared their accuracies.
- Despite efforts, failed to achieve a satisfactory level of accuracy.



```
Train Accuracy: 0.914570685169124
Test Accuracy: 0.7079722703639515
Classification Report:
              precision    recall  f1-score   support

       happy       0.69      0.77      0.73       398
    negative       0.70      0.57      0.63       359
     neutral       0.73      0.77      0.75       397

    accuracy                           0.71      1154
   macro avg       0.71      0.70      0.70      1154
weighted avg       0.71      0.71      0.71      1154
```



True Label \ Predicted Label	happy	negative	neutral
happy	309	50	39
negative	78	220	61
neutral	56	40	301

- Transitioned to a Random Forest model
- Reduced the number of emotion classes from 8 to 3 as both models performed similarly
- improved accuracy by simplifying the classification task but shows overfit

Thank you