VOICE BASED EMOTION RECOGNITION

Karthik Kurakula
P.No- 19990121-T434
email- kaku21@student.bth.se

Likhil Bavu
P.No- 19971219T536
email- liba21@student.bth.se

***Abstract*—We utilize machine learning (using Python) to examine several algorithms to choose characteristics that are significant to forecasting emotion, assess the accuracy, and then choose the method with the greatest accuracy. We work to detect different emotions such as happy, anger, sorrow, and neutral. After processing each audio file in the repository to create a feature vector with matching labels, we take the wave files from the audio clips and apply feature selection techniques to identify the most pertinent findings. The time-domain representation of sound is quite complicated, and in its original form, it does not give a great deal of insight into the important aspects of the signal, such as beneficial variations in volume or pitch.**

I. Introduction

The human voice is quite versatile and can express a variety of emotions. Speaking emotionally gives human behaviour greater perspective. We can better understand human behaviour by doing more research, whether we are studying adoring spectators or unhappy customers. Even though machine learning research is still in its early stages in this area, humans can instantly recognise when someone is speaking with emotion.

We may begin our examination of speech emotion by defining one emotion. We focus particularly on how to classify wrath in voice samples. We start by outlining the information that will be used to analyse emotion. Next, we discuss our methodology and examine the best methods for selecting traits that are crucial for mood prediction. We also consider several machine learning techniques for categorising emotion.

II. Motivation

Voice recognition is a young area with a lot of potential applications in business and commerce. You may tailor the user's interaction with the system by using their voice and attempting to discern the emotion it conveys. As a result, the system may be better able to comprehend the user and their usage habits and expectations.

III. Objective

This project's goal was to aid the system in bettering its comprehension of and aptitude for interacting with people. This allows different activities, like music suggestions, to be tailored according on the user's sentiment as heard in their voice.

IV. Architecture of Models

A. Model Used

- Convolution Layer: Create a CNN : In the example model below, a 2D Convolutional Layer (Conv2D) unit is the portion that learns the translation invariant spatial patterns and their spatial hierarchies. By downsampling feature maps to the maximum value possible within a window, the Max Pooling Layer reduces their size by half. Why downscale? Because if you did it another way, you'd end up with an enormous amount of parameters, your computer would crash, and the model would drastically overfit the data. A CNN can manage the massive quantities of data in pictures because of this mysterious layer. Max Pooling makes a nice model. To input into a Dense layer, which produces the 24 species that the model is expected to use to categorise the audio recordings, the Flatten Layer condenses all the feature map data into a single column. [7] [3]

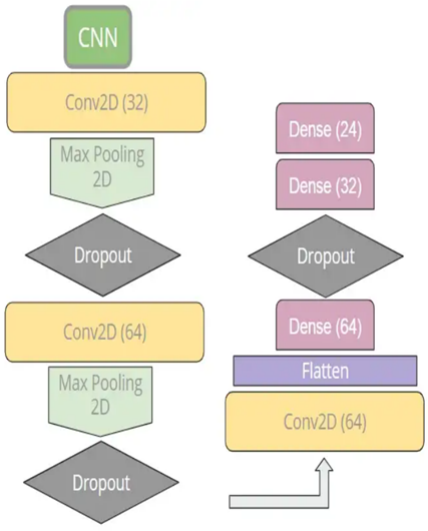

Fig. 1. Sample CNN model architecture

- CNN-2D(deep): CNN-2D (deep) This model was constructed in a similar format as VGG-16, but the last 2 blocks of 3 convolution layers were removed to reduce complexity. This CNN model had the following architectural complexity:

- 2 convolution layers of 64 channels, 3×3 kernel size and same padding followed by a max-pooling layer of size 2×2 and stride 2×2.
- 3 convolution layers of 256 channels, 3×3 kernel size and same padding followed by a max-pooling layer of size 2×2 and stride 2×2.
- Each convolution layer had the 'ReLu' activation function.
- After flattening, two dense layers of 512 units each were added and dropout layers of 0.1 and 0.2 were added after each dense layer.
- Finally, the output layer was added with a 'softmax' activation function.

[4]

- **Non Linearity (ReLU)**: ReLU stands for Rectified Linear Unit for a non-linear operation. The output is $f(x)$ = max(0,x). Why ReLU is important : ReLU's purpose is to introduce non-linearity in our ConvNet. Since, the real world data would want our ConvNet to learn would be non-negative linear values. There are other non-linear functions such as tanh or sigmoid that can also be used instead of ReLU. Most of the data scientists use ReLU since performance wise ReLU is better than the other two.
- **Neural Network with Softmax Layer**: One often wonders if it is possible to interpret the output, for instance y = [0.02, 0, 0.005, 0.975], as the probability of some input being in a class equal to the respective component values yi in the output vector when using neural network models like regular deep feedforward nets and convolutional nets for classification tasks over a set of class labels. No, unless you train the net using the cross-entropy loss function and use a softmax layer as the output layer, to cut to the chase. This topic about categorization using neural networks is crucial since it is occasionally overlooked in internet sources and even in certain textbooks. We'll examine how the softmax function is created in the following section.

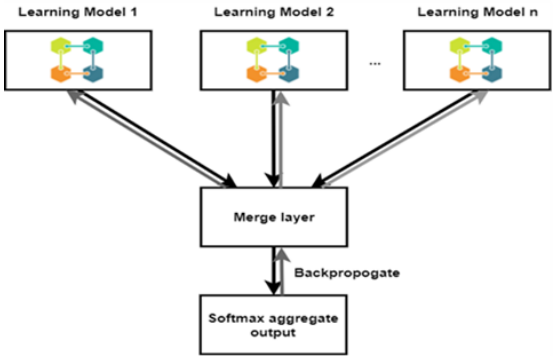

Fig. 2. Softmax Layer

We will examine the derivation of the softmax function in the multinomial logistic regression setting and its application to ensemble deep neural network models for robust classification.

V. Methodology

MFCC is an audio feature extraction technique which extracts speaker specific parameters from the speech.The MFCC feature extraction isthe most popular and dominant method to extract spectral features for speech by the use of perceptually based Melspaced filter bank processing of the Fourier Transformed signal. [5]

Speech recognition primarily concentrates on teaching the system to identify a person's distinctive vocal features. The Mel Frequency Cepstral Coefficients, often known as MFCC, are the most widely used feature extraction method because they are simpler to apply and more reliable under a variety of circumstances .

MFCC was created utilising an understanding of the human auditory system. It is a typical technique for voice recognition feature extraction. Pre-emphasis, Framing, Windowing, FFT, Mel filter bank, and calculating DCT are steps in MFCC.

The properties of the frequency of the signal represented on the Mel scale, which closely resembles the nonlinearity of human hearing, are captured by the Mel-frequency Cepstrum. The "spectrum of the spectrum" is, thus, represented by the Mel-frequency Cepstrum Coefficients (MFCC). By projecting the frequency spectrum's powers onto the mel scale, obtaining the log of those powers, and then performing the discrete cosine transform, it is possible to generate MFCCs. [6] The usage of MFCCs as features is widespread in voice recognition software. Mel-frequency cepstral coefficients (MFCCs)
y : np.ndarray [shape=(n,)] or None audio time series

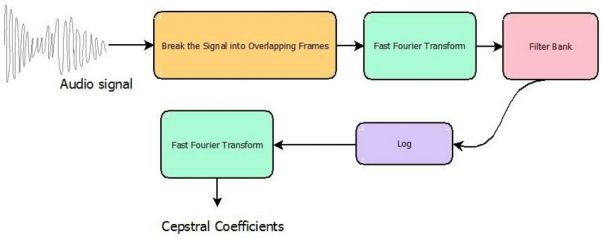

Fig. 3. MFCC Calculation

VI. Data Sets

If an audio recording contains furious emotions, we want to know about it. Each audio recording in the repository is analyzed in order to create a feature vector with labels that indicate whether or not the speaker is furious. Following feature extraction, we use feature selection algorithms to identify the most pertinent outcomes before applying different models to the features that received the highest scores. We use the dta sets The dataset we use for this project is "The RAVDESS" and "SAVEE" (Surrey Audio-Visual Expressed Emotion) .

- The RAVDESS is a validated multimodal database of emotional speech and song. The database is gender balanced consisting of 24 professional actors, vocalizing lexically matched statements in a neutral North American accent. [1]
- SAVEE (Surrey Audio-Visual Expressed Emotion) is an emotion recognition dataset. It consists of recordings from 4 male actors in 7 different emotions, 480 British English utterances in total. The sentences were chosen from the standard TIMIT corpus and phonetically balanced for each emotion [2].

A. Feature Extraction

It is challenging to decipher the essential characteristics of the signal from the time-domain representation of sound because it is so complex. In order to take into consideration this characteristic of sound sources, we transform this time domain representation into more illuminating properties. Calculating the signal's average energy is the simplest method. This number reflects the "volume" of the speaker as well as the total energy of the sound.
Statistics that shed light on emotions include the signal and spectrum's maximum, minimum, range, mean, standard deviation, and duration. These might be fluctuations in volume or pitch that could be a sign of shifting emotions. We also assess skewness, the signal's deviation from horizontal symmetry, and kurtosis, the signal's height and sharpness of the center peak relative to a standard bell curve, for both the signal and spectrum

B. Feature Selection

After processing the original sound data to extract features, we needed to filter the many characteristics to determine which ones contributed the most to the classifier, according to the considerable variance of our technique. After windowing the input speech sounds, 1131 samples were utilized for validation after about 485 had been learned. Given that there were many more characteristics than cases, the variance was rather substantial. It was clear that we needed to focus on the most important factors. Because there were so many features to take into account, we used algorithms to score each feature rather than using a brute force forward or backward search.

VII. Results

The below figure shows the result of the evaulation model where the testing data accuracy . The below graph's shows the Acurracy and Loss for both traning and testing. Overall, we got 78.61% of the test data right, which is respectable, but we can do better by employing more augmentation techniques and other feature extraction strategies. As seen in the below fig, our model is better at predicting surprise and anger than other emotions.

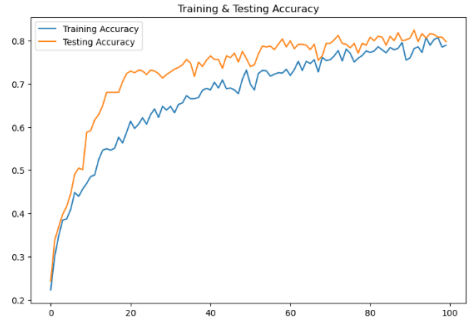

Fig. 4. Training and Testing accuracy

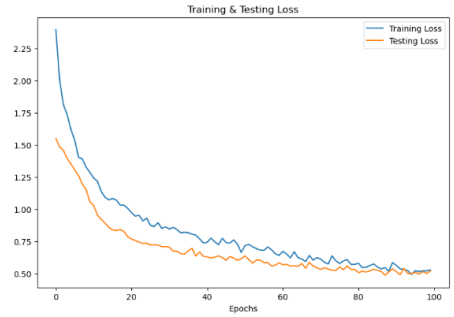

Fig. 5. Testing and Training Loss

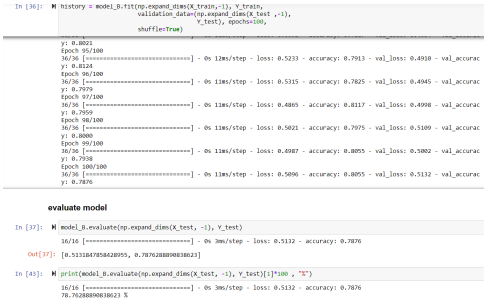

Fig. 6. Accuracy

Overall, we got 78.61% of the test data right, which is respectable, but we can do better by employing more augmentation techniques and other feature extraction strategies.

```
if __name__ == "__main__":
    # load the saved model (after training)
#     model = pickle.load(open("result/mlp_classifier.model", "rb"))
    print("Please talk")
    filename = "test.wav"
    # record the file (start talking)
    record_to_file(filename)
    # extract features and reshape it
    features = extract_mfcc(filename)
    # predict
    result = model_B.predict(features.reshape(1,40,1))[0]
    # show the result !
#     print("result:", result)
    print(intemotion[np.argmax(result)])
```

```
Please talk
1/1 [==============================] - 0s 278ms/step
angry
```

Fig. 7. Result

VIII. Conclusion and Future Work

- This project can be used in different areas and has wide range of applications.

- In a way, it can help in constructing machines that can better understand human emotions hence making them more human-like, aiding our quest to build better human-like systems.
- This project can also be used for market applications like:
  - Understanding customer satisfaction of a company's service.
  - Building recommendation systems based on mood or emotion of the person using voice.
  - Understanding the emotion through voice and helping in creating better support for people in the area of mental health.

REFERENCES

[1] RAVDESS Emotional speech audio — kaggle.com. https://www.kaggle.com/datasets/uwrfkaggler/ravdess-emotional-speech-audio. [Accessed 15-Jan-2023].

[2] Surrey Audio-Visual Expressed Emotion (SAVEE) Database — kahlan.eps.surrey.ac.uk. http://kahlan.eps.surrey.ac.uk/savee/Download.html. [Accessed 15-Jan-2023].

[3] Noushin Hajarolasvadi and Hasan Demirel. 3d cnn-based speech emotion recognition using k-means clustering and spectrograms. *Entropy*, 21(5):479, 2019.

[4] Dias Issa, M Fatih Demirci, and Adnan Yazici. Speech emotion recognition with deep convolutional neural networks. *Biomedical Signal Processing and Control*, 59:101894, 2020.

[5] Chadawan Ittichaichareon, Siwat Suksri, and Thaweesak Yingthawornsuk. Speech recognition using mfcc. In *International conference on computer graphics, simulation and modeling*, volume 9.

[6] Parwinder Singh. An approach to extract feature using mfcc. *IOSR Journal of Engineering*, 4:21–25, 08 2014.

[7] Keras Team. Keras documentation: Convolution layers — keras.io. https://keras.io/layers/convolutional/. [Accessed 15-Jan-2023].