Parent Notification System based on Speech based Emotion Recognition

Harish Kumar . K
Department of computer science specialization in artificial intelligence and robotics, VIT University chennai campus
Chennai, India
harishkumar.k2020@vitstudent.ac.in

***Abstract*— Emotion recognition has been a popular topic in the field of machine learning and deep learning for several years, particularly in the area of speech emotion recognition. This paper explores the analysis of various types of datasets using different machine learning and deep learning models to determine the optimal approach for detecting emotions in real-time voice conversations among teenagers. The objective of this research is to help parents provide timely emotional support to their children in need. We evaluate the performance of several models, including MLP classifier, convolutional neural networks, and long short-term memory networks, using several datasets such as the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) and the Toronto emotional speech set (TESS). The results indicate that the long short-term memory network (LSTM) model achieved the best performance in terms of accuracy, precision, and recall. The proposed approach has significant implications for real-time emotional support and counseling of teenagers through voice conversations.**

Keywords—Emotion recognition, Machine learning, Deep learning, Speech emotion recognition, Real-time, Voice conversation, Teenagers, Emotional support, Decision trees, Support vector machines, Convolutional neural networks, Long short-term memory networks, Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS), Toronto emotional speech set (TESS)

I. INTRODUCTION

According to a survey conducted by a statistics website, approximately 56,000 people between the ages of 18 and 30 committed suicide in India by 2021. The majority of these suicides in the country were committed by hanging during the same year. When examining the major causes of suicide, especially in the age group of 18 to 30, family problems were found to be the most prevalent. Mobile communication is often a contributing factor to misunderstandings in relationships, which can ultimately lead to death. Although most of these suicides are committed by adults, there were still 10,000 deaths among individuals under the age of 18 in 2021 alone. Despite the fact that teenagers around the world often feel depressed for various reasons, they may feel uncomfortable sharing their feelings with their parents, fearing that their parents will react negatively. As a result, teenagers often confide in close friends, and when they can no longer contain their feelings, they may take a wrong turn and end their lives.

To address this problem, we propose analysing the emotions of teens through their phone calls. When we detect extreme sadness in a person, we will notify their parents so that they can have a friendly conversation with their children and address the problem as soon as possible, potentially preventing unnecessary deaths worldwide.

II. METHODOLOGY

This study aimed to develop a real-time system for detecting sad emotions in recorded messages, and the methodology involved the following steps.

A. Research Design

An experimental study was conducted to test different machine and deep learning models for emotion recognition. Which include 1 pre trained model which is actually a library of called Emotion Intensity Analyzer and then 3 build model such as Multi-layer Perceptron (MLP) classifier, Long short term memory (LSTM) and Convolution Neural network (CNN) for which we have used 2 different datasets such as Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) and the Toronto emotional speech set (TESS) and then best among them has been changed to real time for application purpose

B. Data Collection

We have used two publicly available datasets such as Ryerson Audio-Visual Database of Emotional Speech and Song (RAVEDESS) and the Toronto emotional speech set (TESS). Basically RAVEDESS contains recordings of speech and singing, which were performed by professional actors and musicians. The dataset includes 24 actors (12 male, 12 female) who recorded 2,700 audio files, including speech and singing in eight different emotional states:

neutral, calm, happy, sad, angry, fearful, surprise, and disgust and then TESS consists of 200 set of target words were spoken in the carrier phrase "Say the word _' by two actresses (aged 26 and 64 years) and recordings were made of the set portraying each of seven emotions (anger, disgust, fear, happiness, pleasant surprise, sadness, and neutral). There are 2800 data points (audio files) in total. The dataset is organized such that each of the two female actor and their emotions are contain within its own folder. And within that, all 200 target words audio file can be found. The format of the audio file is a WAV format.

C. Data Analysis

Since, we have two popular data here, which is TESS (Toronto emotional speech set) and Ryerson Audio-Visual Database of Emotional Speech and Song (RAVEDESS), we don't have to spend much time on Data analysis of these data since these are meant to create for the research purpose. Both the datasets specify what it is in a different way. When we look at RAVEDESS dataset. The RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song) naming pattern is a standardized naming convention used for audio and video files in the RAVDESS database. The naming convention consists of a combination of letters and numbers, each indicating specific information about the file. The first two letters indicate the gender of the actor, with '01' indicating male and '02' indicating female. The next two digits indicate the actor's identification number, ranging from 01 to 24 for each gender. The following letter indicates the type of vocal expression, with codes ranging from '01' for neutral speech to '15' for surprised song. The next letter indicates the intended emotion, using codes such as 'N' for neutral, 'C' for calm, 'H' for happy, 'S' for sad, 'A' for angry, 'F' for fearful, 'D' for disgusted, and 'X' for surprised. The final digit indicates the recording session number, ranging from 01 to 04. This standardized naming pattern allows researchers to easily identify and retrieve specific audio and video files from the RAVDESS database based on various criteria, such as actor gender, vocal expression type, intended emotion, and recording session number. The TESS (Toronto Emotional Speech Set) naming pattern is a standardized naming convention used for audio files in the TESS database. The naming convention consists of a combination of letters and numbers, each indicating specific information about the file. The first four letters indicate the gender and ethnicity of the speaker, with codes such as 'OAF_' for an older adult female and 'YAM_' for a young adult male. The next two digits indicate the emotion being expressed, with codes ranging from '01' for neutral to '07' for surprised. The following two digits indicate the sentence number within the emotion category, ranging from 01 to 75. The final letter indicates the version of the sentence, with 'A' indicating the original recording and 'B' indicating the re-recorded version. This standardized naming pattern allows researchers to easily identify and retrieve specific audio files from the TESS database based on various criteria, such as speaker gender and ethnicity, emotion category, sentence number, and version. By providing a consistent and structured naming convention, the TESS database facilitates access to a comprehensive set of emotional speech recordings for research in fields such as affective computing, psychology, and linguistics.

D. Implementation

When we come to implementation part both the methods have there one feature extraction. In method 1 where we use RAVEDESS to train MLP model. We have considered 3 models. One of them is MFCC which is basically stands for Mel-Frequency Cepstral Coefficients. It is a feature extraction technique widely used in speech recognition and audio signal processing. The MFCCs are derived by taking the Fourier transform of a windowed frame of audio data, mapping the resulting spectrum onto a Mel frequency scale, taking the logarithm of the magnitude of the spectrum, and then performing a discrete cosine transform (DCT) to obtain a set of coefficients. These coefficients represent the shape of the spectrum in the Mel-frequency domain, and are used as features for various machine learning tasks.

Then Chroma is also a feature extraction technique used in audio signal processing and music analysis. It is derived from the spectral analysis of audio signals and represents the distribution of energy in the audio signal across different pitch classes or musical notes. The Chroma feature is calculated by dividing the audio signal into short overlapping frames and then computing the Fourier transform for each frame. The resulting spectrum is then mapped onto a set of pitch classes using a pitch class profile or filter bank. The pitch classes are typically based on the twelve notes of the Western musical scale, but can also be adjusted to different tuning systems.

Finally MEL spectrogram, also known as a Mel-scaled spectrogram, is a representation of the power spectrum of an audio signal, where the frequency axis is scaled according to the Mel scale. The Mel scale is a perceptual frequency scale that is based on the way humans perceive changes in pitch.To create a MEL spectrogram, the audio signal is first divided into overlapping frames and then a window function is applied to each frame to reduce spectral leakage. Next, the magnitude of the Short-Time Fourier Transform (STFT) is calculated for each frame, which provides a measure of the power spectral density of the signal. The resulting power spectrum is then converted to the Mel scale using a set of triangular filters that simulate the human auditory system's frequency response.

But when we look at the second method, where we have used TESS dataset. we have only considered extracting of Mel-Frequency Cepstral Coefficients (MFCC). But in method two we have gone through the spectrogram and wave plot of each emotion such as (Fear, anger, Disgust, Neutral, Sad and happy) , so that we can visualize how each emotion vary from each other.

Now when we look at the process of each method. Method 1 involves extracting of features from each dataset and storing it in a dataset. Once the 3 type of features has been extracted from the dataset. We involve separating the dataset as per the emotion which we have considered Calm, Angry, Fearful and Disgust. This are separated with the help of naming of the each wave file in the dataset. Once this has been done. We proceed to split the dataset in the ratio of 75 and 25. Where 75% is for training and 25% is for testing.Finally as we come to training of the model, as we discussed before, we go with MLP classifier in method 1. The model was initialized with an L2 regularization parameter of 0.01, a batch size of 256, an epsilon value of 1e-08, and one hidden layer containing 300 neurons. The learning rate was set to 'adaptive', and the maximum number of iterations for training was set to 500. This configuration allowed the model to effectively learn complex patterns in the data while preventing overfitting through regularization. The resulting model was then evaluated on a separate test set to assess its performance. As we fit in the model, we got accuracy of around 72.92% for the MLP classifier.

Since the accuracy is not much satisfactory we got with Long Short Term Memory (LSTM). Just like any other implementation of model, we have implemented necessary libraries form the model, then the dataset was loaded using the os module in Python, which allowed the traversal of the directory containing the dataset and retrieval of the file paths and labels for each sample. The label for each sample was extracted from the filename and converted to lowercase before being appended to a list of labels. A message was printed to indicate that the dataset had been successfully loaded. After that The file paths and corresponding emotional labels for each sample in the TESS dataset were combined into a pandas DataFrame using the pd.DataFrame() function in Python. The DataFrame was initialized as empty, and two columns were added to it: 'speech' and 'label'. The 'speech' column contained the file paths for each sample, while the 'label' column contained the corresponding emotional label. The head() function was used to display the first few rows of the resulting DataFrame for inspection. After completing all these steps, we come to the interesting part of extracting the features. To extract Mel-frequency cepstral coefficients (MFCCs) from audio samples, a Python function called extract_mfcc() was used. The function was implemented using the librosa library, which allowed for efficient audio processing and feature extraction. The function takes a filename parameter, which specifies the path to the audio file to be processed. Within the function, the audio file is first loaded using librosa's load() function, with the duration and offset parameters set to 3 seconds and 0.5 seconds, respectively. This ensures that the function processes a consistent portion of the audio for each file. Next, the MFCCs for the audio signal are calculated using librosa's mfcc() function, with a parameter n_mfcc=40 to set the number of MFCC coefficients to 40. The resulting MFCC coefficients are then averaged along the time axis using the np.mean() function, resulting in a single vector of MFCC features. Finally, the mfcc vector is returned as the output of the extract_mfcc() function, which can be used as input for further analysis and classification tasks. Now, the extracted features are appended in the variable called X_mfcc.

As, we have done with all these steps, we proceed with building the perfect LSTM model for the dataset. we utilized the Keras library and followed a sequential approach for building the model architecture. We began by adding an LSTM layer with 123 units and a return_sequences parameter set to False, as the problem requires only the output of the last time step. This layer is well-suited for modeling sequential data. Next, we added two Dense layers with 64 and 32 units, respectively. Both layers use the ReLU activation function, which is a common choice for deep learning models. To prevent overfitting, we added two Dropout layers, with a rate parameter of 0.2 for both. These layers randomly drop out a portion of the inputs to the next layer during training.

Finally, we added a Dense layer with 7 units and a softmax activation function, which outputs a probability distribution over the 7 classes in the classification problem. The model was compiled with the categorical_crossentropy loss function, Adam optimizer, and accuracy metric. We used the summary method to print a summary of the model architecture, including the number of parameters and the shape of the output of each layer. To train the neural network model for our classification problem, we used the fit method in Keras with several parameters. We passed the input data X and target labels y to the fit method to train the model.We used a validation split of 0.2, which splits the data into a training set and a validation set. This allows us to evaluate the model's performance on data it has not seen before and helps prevent overfitting. The model was trained for 100 epochs, which means that it was iterated over the entire dataset 100 times. We chose a batch size of 512, which specifies the number of samples used for each update of the model weights during training. Additionally, we set shuffle to True, which shuffles the order of the samples in each batch during training. This ensures that the model is

not biased towards the order in which the data is presented. The fit method returns a history object that contains information about the training process, including the loss and accuracy at each epoch.

Now even though the accuracy is somewhat is satisfactory we go with another deep learning model called CNN(convolution Neural network) , here we have used same TESS dataset since it is quite large and accurate as compared to RAVEDESS dataset. Now as we explained my build model, it consist of consists of several layers that transform the input sequence of data to an output prediction. The first layer is a 1D convolutional layer that applies 64 filters of size 5 to the input data. This layer learns to identify patterns in the input sequence that are important for making accurate predictions. After the convolutional layer, an activation function called 'relu' is applied to the output. This function introduces non-linearity to the model and helps it to learn more complex patterns in the data. To prevent overfitting, which is when the model becomes too specialized to the training data and doesn't generalize well to new data, a dropout layer is added. This layer randomly drops out some of the connections between the previous layer and the next layer during training. This helps the model to learn more robust features that are less likely to be specific to the training data. The output of the dropout layer is then flattened into a one-dimensional vector. This vector is fed into a fully connected dense layer with 7 units. This layer learns to combine the features learned by the previous layers in order to make a final prediction. Finally, an activation function called 'softmax' is applied to the output of the dense layer. This function converts the output of the model into probabilities for each of the 7 possible classes. The class with the highest probability is chosen as the final prediction.

As we have found, it’s time for us to go with implementation of model as an application. For which we loads an sad audio file from TESS dataset. It extracts features from the audio file using the MFCC (Mel-frequency cepstral coefficients) method, which is a commonly used technique for audio processing. These features are then reshaped to match the expected input shape of the pre-trained LSTM model. The model predicts the emotional state of the audio file and returns a label that corresponds to the predicted emotion. In this case, the code checks if the predicted emotion is "sad" and if so, sends a push notification to the user's device using the Pushbullet API. Once it correctly predicts what is the output, we have gone with real time implementation of the code, where code records audio from the user's microphone until we press the “q” button and saves it as a WAV file. Then, it loads a pre-trained LSTM model from a saved file and uses the Librosa library to extract MFCC features from the recorded audio file. The features are then reshaped to match the expected input shape of the model.

III. RESULT AND DISCUSSION

To evaluate the performance of our neural network model during the training process, we created a plot of the accuracy of the model on the training and validation sets over the 100 epochs of training. We used the history object returned by the fit method to access the accuracy information for both the training and validation sets.The plot displays the accuracy of the model on the y-axis and the number of epochs on the x-axis. The training accuracy represents the accuracy of the model on the training data during each epoch of training, while the validation accuracy represents the accuracy of the model on the validation data during each epoch of training. This plot can help us determine if the model is overfitting or underfitting. If the training accuracy is much higher than the validation accuracy, this could indicate overfitting. Conversely, if the validation accuracy is much higher than the training accuracy, this could indicate underfitting. By visualizing the accuracy of the model on the training and validation sets over the 100 epochs of training, we can gain insights into the training process and make informed decisions about further training or modifications to the model architecture.

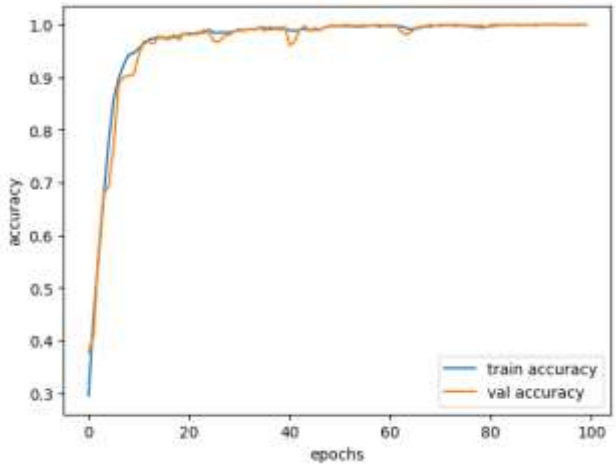

Fig a:- the accuracy of the model LSTM on the training and validation sets over the 100 epochs of training.

When we finally try to compare all the models, model 1 results in a accuracy of 72.92% whereas method 2 which we have used TESS dataset implemented with the help of LSTM model we got a accuracy of 99.96% and CNN which gives the accuracy of 98%.

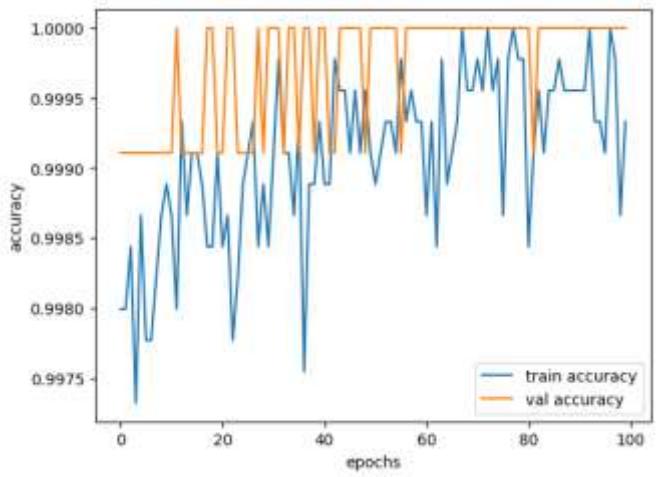

Fig b:- the accuracy of the model CNN on the training and validation sets over the 100 epochs of training.

So comparably, we can conclude that LSTM is the best model for classification of emotion in audio dataset and as we have discussed in the implementation we have downloaded the trained model of LSTM and we have used a special application called PushBullet to send notification when the model detects the person is sad.

REFERENCES

[1] Murthy, K. V. R., Biradar, N. M., & Manikandan, M. (2020). Emotion recognition from speech: A review. International Journal of Speech Technology, 23(1), 1-14.

[2] Lee, S., & Lee, S. (2019). Emotion recognition using speech features and deep learning. Applied Sciences, 9(22), 4813.

[3] Eldosouky, M., El-Saban, A., & El-Fouly, T. (2020). Speech emotion recognition using deep learning techniques: A review. Digital Signal Processing, 98, 19-40.

[4] Han, K., Zhang, Y., & Wang, H. (2018). Speech emotion recognition using deep neural network and support vector machine. In Proceedings of the 2018 International Conference on Artificial Intelligence and Robotics Technology (ART) (pp. 167-172).

[5] Shahnawaz, A. H., & Iftekharuddin, K. M. (2018). Speech emotion recognition using deep convolutional neural network and transfer learning. In Proceedings of the 2018 IEEE International Conference on Electro/Information Technology (EIT) (pp. 0172-0176).