Speech Emotion Recognition using Deep Learning Algorithms

T SampathKumar [1, a)], Saikiran Anugam [2, b)], Apoorva Irukulla [2, c)],
Gaddam Huldah grace [2, d)], Meghana Daddanala [2, e)], Udaykiran Anugam [1, f)]

[1] *SR University, Warangal, India*
[2] *SR Engineering College, Warangal, India*

[a)] *Corresponding Author: t.sampathkumar@sru.edu.in*
[b)] *saikirananugam@gmail.com,* [c)] *irukullaapoorva@gmail.com,*
[d)] *huldahgraceg@gmail.com,* [e)] *meghanadaddanala@gmail.com,* [f)]*anguamudaykiran@gmail.com*

Abstract. Speech emotion recognition is an act of recognizing human emotions and pitch. The primary objective of this is to automatically detect, classify, and interpret emotions from spoken language, enabling machines to comprehend and respond appropriately to human emotions. Significant improvements in SER have been made with the introduction of deep learning methods and the availability of massive annotated speech datasets, leading to its wide applications across numerous fields. we can use some audio samples with this technique to anticipate emotions like sadness, rage, surprise, calmness, fear, neutrality, regret, and many others. It emphasizes the significance of SER in enabling machines to respond empathetically and adaptively to human emotions, ultimately enhancing user experience and interaction.

INTRODUCTION

Speech is a potent tool for intercultural communication. There has been a lot of interest in the capacity to identify and comprehend emotions from speech signals [8,9,13,15,16]. Emotions play a crucial role in human interactions, influencing our perceptions, decisions, and behaviors. The need to endow machines with the capacity to recognize and react to emotions becomes crucial as the area of artificial intelligence works to develop more natural and human-like interactions between people and machines. A promising path to achieving this objective is provided by SER, which is powered by deep learning algorithms.

The effects of utilizing deep learning for speech-emotion recognition (SER) have a wide reach, spanning various domains. These include areas such as affective computing, virtual assistants, interactions between humans and robots, and the monitoring of mental health. By enabling machines to accurately detect and respond to human emotions, we open the door to creating more empathetic and adaptable systems. This, in turn, can lead to an enhanced user experience and more meaningful interactions between humans and technology.

Deep learning, with its capacity to autonomously train and extract high-level representations from raw data, has transformed the area of SER, resulting in considerable improvements in accuracy and robustness. Convolutional Neural Networks (CNNs) [7,8,9,10,13], Multi-Layer Perceptron Classifier (MLPC) [18,19,20], and their variants such as Long Short-Term Memory (LSTM) [7,9,10,11] have demonstrated remarkable capabilities in capturing complex patterns and temporal dependencies [14,16] present in speech data. These models can accurately capture the ever-changing characteristics of emotions conveyed through speech. This paper also discusses the pre-processing techniques for feature extraction from speech signals using masking noise and using librosa, noise reduce libraries.

Identifying the emotions that the speaker elicits while speaking is the article's goal. These days, detecting emotions has become a crucial responsibility. Speaking from a position of low pitch [11] has a narrower range of pitch; conversely, speaking from a place of fear, fury, or excitement has a higher range of pitch. Speech recognition helps facilitate human-machine communication. To recognize the emotions in this situation, many categorization algorithms are being used. The audio features MFCC[14,19,20], MEL [18], and chroma were utilized, as well as multi-layer perception[17]. These emotional recognition models have been taught to distinguish between calm, neutral, surprised, happy, sad, furious, afraid, and disgusted. We will create a model with an MLP Classifier [14] and LSTM using the library sound file. This system will be able to identify emotions in audio files.

RELATED WORK

In [1] The proposed system presents an in-depth survey of speech emotion identification techniques developed over the last two decades, covering both classical and deep learning approaches. Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and their variants, such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs), are among the deep learning models described in the study. The authors also delve into topics like benchmark datasets, criteria for evaluation, and the challenges encountered within the field. Notably, the study brings attention to current trends and provides insights into future directions.

In [2] The suggested system mainly focuses on speech emotion recognition techniques, covering both traditional algorithms and deep learning methods. Deep learning models discussed in the paper include CNNs, RNNs (LSTM, GRU), hybrid models (such as Convolutional Recurrent Neural Networks - CRNNs), and attention mechanisms. It provides insights into Models for feature extraction, feature selection, and classification employed in SER. The authors also discuss the challenges and future trends, including multimodal emotion recognition and explainable AI in SER.

In [3] The suggested system places a strong emphasis on deep learning approaches for voice emotion identification. It examines the applications of several deep learning architectures, such as CNNs, RNNs, LSTMs, and GRUs, in SER. The authors provide a comprehensive analysis of feature extraction, preprocessing techniques, and datasets used in deep learning-based SER. They also highlight the challenges and future directions in the field.

In [4] The proposed framework introduces a multitask learning framework for speech emotion recognition that incorporates residual fusion and hierarchical attention mechanisms. The approach leverages the shared information between emotion recognition and auxiliary tasks to improve overall performance. The experimental evaluation shows promising results and contributes to the advancement of SER using deep learning techniques.

In [5] The paper delves into the merits and limits of each model, explores the influence of various architectures and training procedures, and emphasizes the problems and future prospects in the field of deep learning speech emotion recognition. It is a helpful resource for scholars and practitioners interested in learning cutting-edge SER methodologies and improvements.

The suggested model is introduced in [6] This study is a thorough review paper that investigates the application of deep learning techniques for voice emotion identification. It covers a range of deep learning models, including CNNs, RNNs (LSTM, GRU), attention-based models, and hybrid architectures, discussing their effectiveness and advancements in the field.

PROPOSED METHODOLOGY

Flow Chart

Audio Input: The first step in the flowchart is the audio input, which can be acquired via a microphone or an audio file.

Preprocessing: The audio input is next cleaned to get rid of any intrusive noise or artefacts that might impair the accuracy of the model. The audio signal may be subject to filtering, normalization, or resampling in this process.

The preprocessed audio stream is next processed to extract pertinent acoustic properties including pitch, intensity, and spectral content. These attributes are subsequently transformed into a numerical representation appropriate for examination by the machine learning algorithm. The diagram shown in Fig. 1 depicts the flowchart.

The retrieved characteristics are then chosen depending on their applicability to the job of emotion recognition. This can entail methods like principal component analysis or feature ranking algorithms.

The selected attributes are then employed in an emotion classification algorithm trained on a labeled dataset [11,16] comprising speech samples with well-known emotional states. This training data is used by the algorithm to categorize the input voice sample's emotional state as one of many potential emotions, such as joyful, sad, angry, or neutral.

The output, which shows the anticipated emotional state [8,13] of the input speech sample, is the last step in the flowchart. The results may be seen.

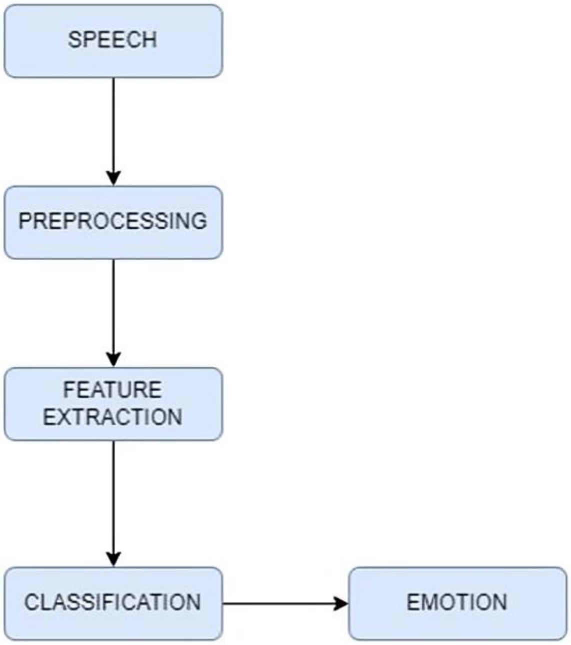


Figure 1. Block representation of the suggested system.

UML Diagrams

Sequence Diagram

It is a type of UML (Unified Modelling Language) diagram, that shows how objects or components interact with one another through time. The flow of messages or events between distinct objects or components, as well as the order in which these interactions occur, are depicted in Fig. 2.

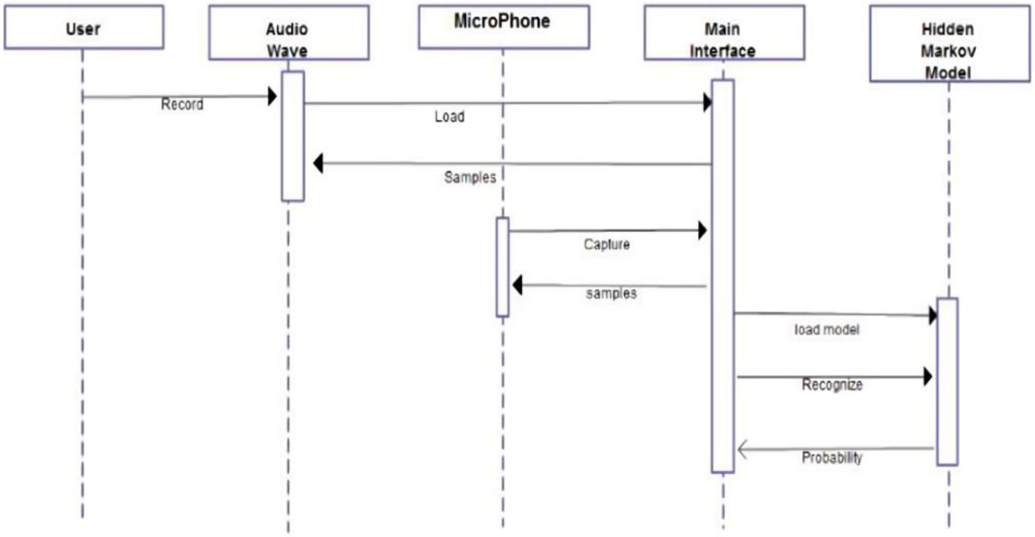


Figure 2. Sequence chart.

Use Case Diagram

It illustrates the relationships between objects or components across time. Fig. 2 shows the sequence in which these interactions take place as well as the flow of messages or events between various objects or components.

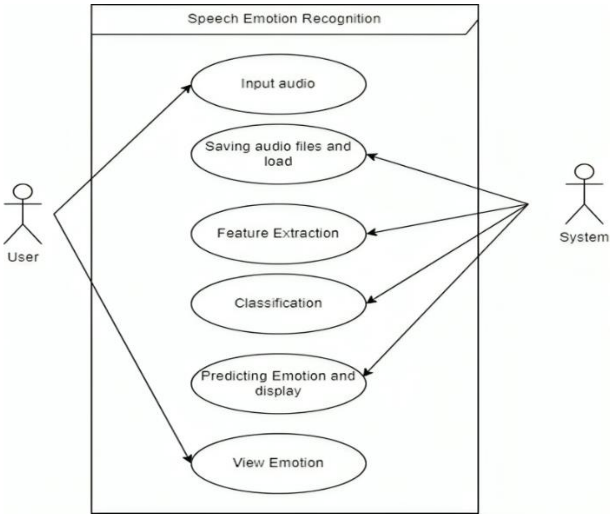

Speech Emotion Recognition
Input audio
Saving audio files and load
Feature Extraction
Classification
Predicting Emotion and display
View Emotion
User
System

Figure 3. Use case Diagram.

Dataset

From Kaggle, the Ravdness dataset is downloaded. In the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS), there are 7356 files totaling 24.8 GB. 24 professional actors, 12 males and 12 females, each read two lexically linked sentences in a neutral North American accent for the database. Expressions of calmness, delight, happiness, sadness, anger, terror, surprise, and disgust can be found in both speech and music. Two emotional intensity levels (normal, strong), as well as a third neutral expression, are used to generate each expression. All situations can access one of three modality formats: Audio-only (16bit, 48kHz.wav), Audio-Video (720p H.264, AAC 48kHz.mp4), or Video-only (without sound).

An example of the naming convention of the filename is 02-01-06-01-02-01-12.mp4

The file's modality [12] is represented by the first two digits; 01 denotes that it is a full-AV type file, 02 that it is only a video file, and 03 that it is only an audio file. In the above-given example, 02 specifies that it is a video file.

The voice channel is represented by the next two numbers. 01 denotes speech, and 02 denotes song. Here 01 suggests that it is of type speech.

In an audio recording, the fifth and sixth numbers are utilised to represent a person's emotional condition. 01 represents neutral, 02 represents calm, 03 represents happiness, 04 represents sorrow, 05 represents anger, 06 represents fear, 07 represents disgust, and 08 represents surprise. Here, 06 states that the speech's emotion is dread.

In an audio file, the seventh and eighth numbers signify the level of emotion. Strong is denoted by 02, whereas 01 stands for average intensity. No high intensity can be found in neutral feelings. Here 01 represents that the level of intensity is normal.

The statements are represented by the ninth and tenth digits. 01 represents the phrase "Kids are talking by the door." Furthermore, 02 stands for the phrase "Dogs are sitting by the door." Statement 2 is represented by 02 in this case.

The recurrence is shown by the eleventh and twelfth digits. The first repeat is represented by 01 in this sentence. The second repeat is represented by 02 in the sentence. Here 01 represents that the statement is repeated only once.

The Actor is represented by the final two numbers. The collection contains 24 actor files. Odd numbers are used to symbolize male performers, whereas even numbers are used to represent female actresses. Here, the number 12 stands for an actress, who is a female.

Data Preprocessing

Collecting data in the real world comes with its own set of issues. It is frequently exceedingly disorganized and contains missing data, outliers, an unstructured approach, etc. We must initially undertake preprocessing operations that only allow us to use the data for further observation and training our machine learning model, not for seeking for any insights into the data. Before submitting our data to the machine learning model, we do data processing using missing values treatment, outliers' identification, normalization, and data split. To clean the voice, we utilized the librosa and noise reduction packages.

Cleaning is done by downsampling audio files, applying a mask, and then directing it to a clean folder. The mask is used to eliminate extraneous empty voices surrounding the main audio voice.

Steps for Data Preprocessing:

Masking aims to remove unnecessary empty voices surrounding the main audio voice by applying a mask. This involves down-sampling the audio files and applying the mask based on a threshold value. We consider the sampling rate and threshold values as parameters for removing empty and unnecessary voices.

An empty list called a mask is initialized. This list will store Boolean values indicating whether each sample in the audio data should be kept (True) or discarded (False) based on the calculated mask. Next, absolute values are calculated by converting the audio file using the Pandas series function. This considers both positive and negative amplitudes and captures the overall magnitude of the audio signals.

Next, the rolling mean of the absolute values is calculated using the rolling function from the panda's library. This mean is computed over a window of size int(rate/10) with a minimum of 1 period. The center parameter is set to True, ensuring that the calculated mean corresponds to the sample at the center of the window.

Subsequently, a loop iterates over each mean value in y_mean. For each mean value, if it exceeds the threshold, it suggests the presence of the main audio voice, so True is appended to the mask list. Otherwise, if the mean value is below the threshold, False is appended to the mask list.

After the loop has processed all the mean values, the function returns the resulting mask list.

The generated mask can be utilized as a filter to remove unnecessary empty voices or noise surrounding the main audio voice. By applying the mask to the audio data, the code effectively retains the relevant portions of the audio while discarding the unwanted segments, thus facilitating the cleaning process.

Feature Extraction

The process of translating raw speech signals into representative features that capture significant information for emotion classification is known as feature extraction [10,20]. MFCC (Mel-frequency cepstral coefficients), chroma, and mel spectrogram are three extensively utilised characteristics for speech emotion identification in this context.

MFCC (Mel-frequency cepstral coefficients):

MFCCs are commonly employed in speech processing because they give a concise representation of an audio signal's spectral envelope. Framing the audio signal into small segments, applying the Fourier Transform to obtain the power spectrum, mapping the spectrum onto the Mel scale to approximate human auditory perception, and finally computing the discrete cosine transform to obtain the cepstral coefficients are all steps in the MFCC extraction process. MFCCs effectively capture phonetic and auditory information by capturing the spectral structure of the speech stream.

MFCC coefficient
10
0
0
100
200
300
400
500
time

Figure 4. MFCC features.

The graphic in Fig.4 depicts the MFCC characteristics of an audio file, with the x-axis representing time and the y-axis representing MFCC coefficients.

Chroma

Chroma features represent the distribution of pitch classes or musical notes within an audio signal. They are calculated by taking the magnitude of the short-term Fourier Transform (STFT) and projecting the resultant spectrum onto a 12-dimensional chroma vector corresponding to the 12 musical notes. Chroma features are particularly useful for capturing tonal and harmonic information in speech signals, making them relevant for speech emotion recognition. They can reveal the presence of musical or melodic patterns in the speech, which can be indicative of certain emotional states.

Mel Spectrogram

The power spectral density of an audio source in the mel-frequency domain is shown visually in a mel spectrogram. It is created by segmenting the frequency spectrum into mel-scale bins, each of which stands for a particular range of frequencies. Insights into the spectrum properties of the spoken signal are provided by the mel spectrogram, which draws attention to the energy distribution across various frequency bands. We may generate a concise representation of the spectral content of the speech signal by averaging the mel spectrogram over time, which may be useful for identifying emotional signals.

These characteristics are calculated from the voice signal in speech emotion recognition using libraries like "librosa" or "python_speech_features." After the characteristics have been retrieved, machine learning algorithms can utilise them as input to build emotion categorization models. In order to recognise and categorise emotions in speech, these models learn to link particular patterns in the retrieved data with various emotional states.

The subplot in Fig.5 displays the waveform, MFCC, and mel spectrogram of the first four audio files. Understanding the temporal and spectral properties of the audio files with the assistance of this visualisation makes it easier to do further analysis and feature extraction for tasks like speech recognition or emotion categorization.

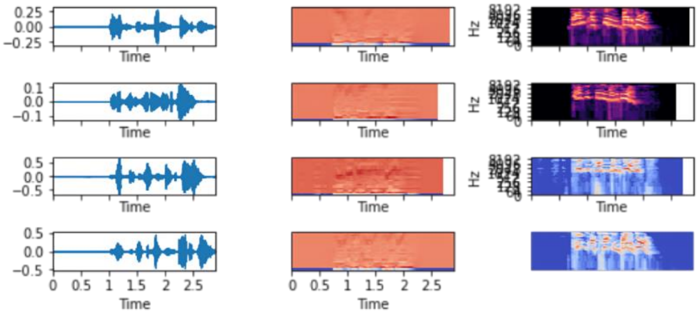


Figure 5. waveform, MFCC, and mel spectogram.

The sound file is opened using soundfile. SoundFile and the audio data are read into the X array, while the sample rate is stored in the sample rate.

The Short-Time Fourier Transform (STFT) is calculated using librosa.stft to take the absolute value of the STFT of X if the chroma parameter is set to True.

A result array is initialized as an empty numpy array. If the mfcc parameter is True, the MFCC features are computed using librosa.feature.mfcc, with X as the audio data, sample_rate as the sample rate, and n_mfcc set to 40. The resulting MFCCs are then averaged across time (axis=0) and concatenated to the result array.

Similarly, if the chroma parameter is True, the chroma feature is calculated using librosa.feature.chroma_stft on the STFT array stft, and the resulting chroma features are averaged across time and appended to the result array.

Finally, if the mel parameter is True, the mel spectrogram is computed using librosa.feature.melspectrogram on the audio data X, with sample_rate as the sample rate. The mel spectrogram is averaged across time and concatenated to the result array.

By combining these three features (MFCC, chroma, and mel spectrogram), speech emotion recognition systems can capture a wide range of acoustic characteristics relevant to emotion expression in speech. These features provide valuable information about the spectral shape, tonal patterns, and energy distribution in the speech signal, enabling effective emotion classification and analysis.

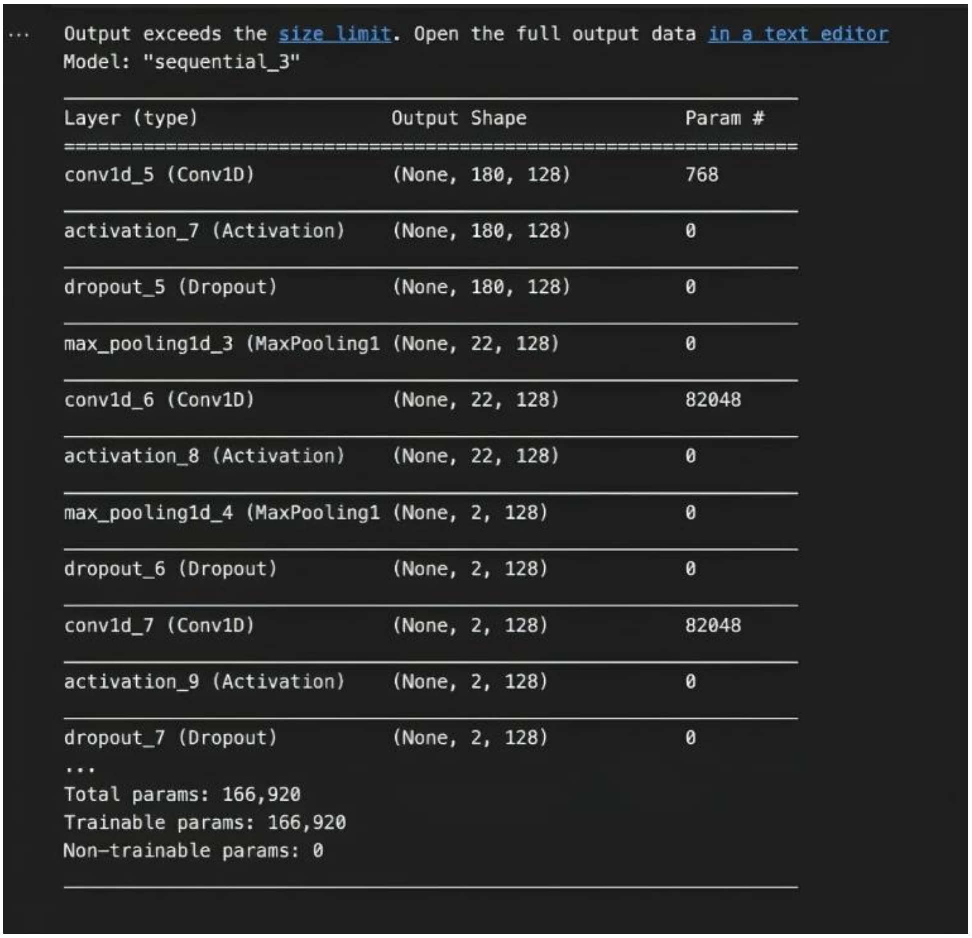
Models used

Multi-Layer Perceptron Classifier (MLPC)

The Multilayer Perceptron Classifier (MLPC) is a kind of artificial neural network used for supervised learning tasks like classification and regression. Each unit in MLPC conducts a simple mathematical operation to the incoming input as it travels through a series of layers that comprise neurons or units. Each layer passes its output on to the one below it until the final output is formed. As a result, MLPC may find complex relationships between input and output data.

Convolutional Neural Network (CNN)

The field of voice emotion recognition has also demonstrated the effectiveness of convolutional neural networks (CNNs). CNNs may be trained to learn how to extract pertinent characteristics that accurately capture the emotional content of speech by utilising the spectrogram representation of audio data. The audio input is split into brief time intervals to create the spectrogram, which is then represented as a 2D picture by performing the Fourier transform to each frame. Then, using convolutional operations on these spectrogram pictures, CNNs may identify patterns and correlations that represent various emotional states. CNNs may learn to categorise emotions like happiness, sorrow, rage, and more by training on labeled datasets. CNNs are a popular choice for voice emotion identification tasks due to their capacity to automatically acquire discriminative features from unlabeled input. This has led to the development of applications such as affective computing research, sentiment analysis in customer service, and emotion-aware virtual assistants. In Fig. 6, the CNN's model summary is shown.



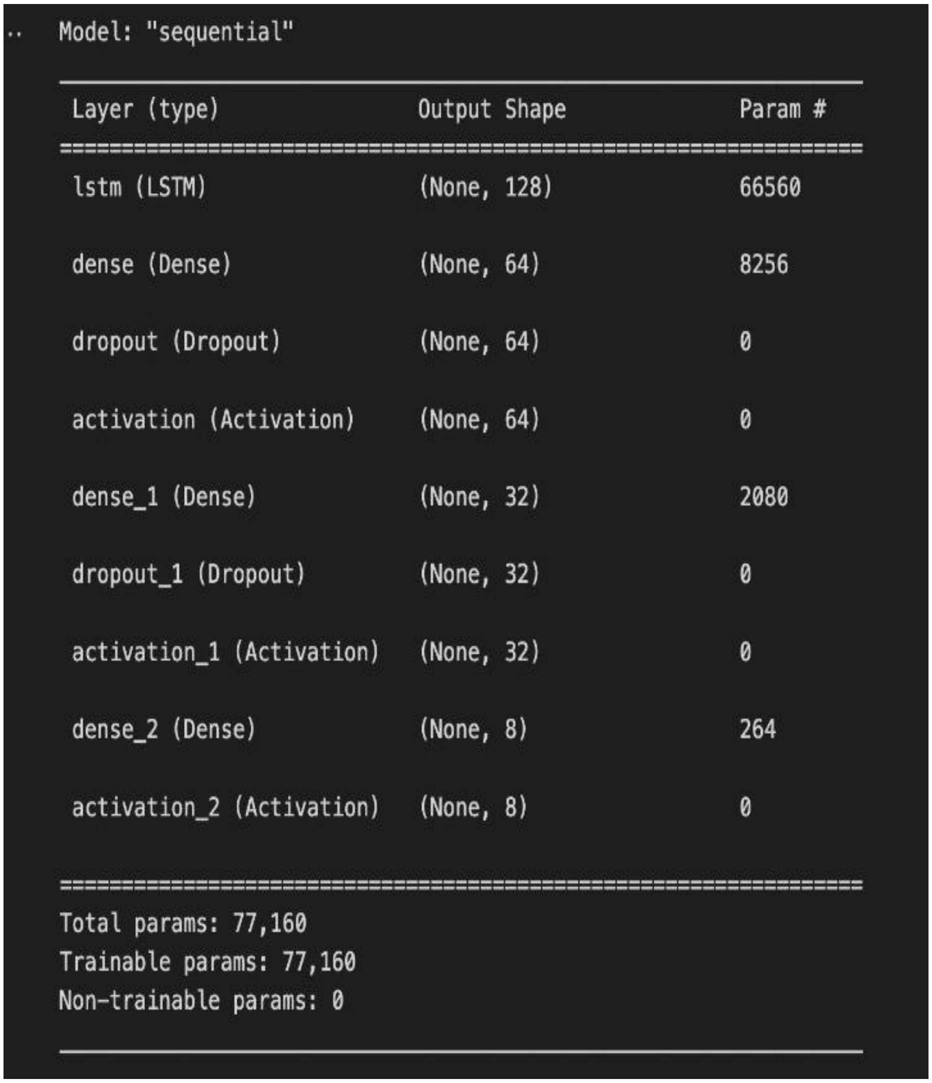
```
...  Output exceeds the size limit. Open the full output data in a text editor
     Model: "sequential_3"
     _________________________________________________________________
     Layer (type)                Output Shape              Param #
     =================================================================
     conv1d_5 (Conv1D)           (None, 180, 128)          768

     _________________________________________________________________
     activation_7 (Activation)   (None, 180, 128)          0

     _________________________________________________________________
     dropout_5 (Dropout)         (None, 180, 128)          0

     _________________________________________________________________
     max_pooling1d_3 (MaxPooling1 (None, 22, 128)          0

     _________________________________________________________________
     conv1d_6 (Conv1D)           (None, 22, 128)           82048

     _________________________________________________________________
     activation_8 (Activation)   (None, 22, 128)           0

     _________________________________________________________________
     max_pooling1d_4 (MaxPooling1 (None, 2, 128)           0

     _________________________________________________________________
     dropout_6 (Dropout)         (None, 2, 128)            0

     _________________________________________________________________
     conv1d_7 (Conv1D)           (None, 2, 128)            82048

     _________________________________________________________________
     activation_9 (Activation)   (None, 2, 128)            0

     _________________________________________________________________
     dropout_7 (Dropout)         (None, 2, 128)            0
     ...
     Total params: 166,920
     Trainable params: 166,920
     Non-trainable params: 0

     _________________________________________________________________
```

Figure 6. Model summary of CNN.

Convolutional layers with ReLU activations, pooling layers for downsampling, and a fully linked layer for classification with SoftMax activations make up this CNN model.

Longest Short-term Memory (LSTM)

Recurrent neural networks (RNNs) of the LSTM (Long Short-Term Memory) network type have demonstrated notable success in voice emotion identification tests. By selectively storing and retrieving data from previous time steps, LSTM models are excellent at capturing such long-range relationships. This enables them to handle sequential data, such as speech, particularly effectively. The temporal dynamics of emotional expressions may be modeled in order to teach LSTMs how to extract pertinent features and patterns from input audio sequences. The vanishing gradient problem is successfully solved by the memory cells and gates in LSTMs, which allow the network to store crucial information over extended periods of time. The ability of LSTMs to capture subtle variations and dependencies in speech signals has made them a popular choice in speech emotion recognition. They can be trained on labeled datasets to classify various emotional states, enabling applications such as emotion-aware human-computer interfaces, sentiment analysis in call centers, and affective computing research. Hence, the model summary of LSTM is shown in Fig.7.



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 128)               66560

 dense (Dense)               (None, 64)                8256

 dropout (Dropout)           (None, 64)                0

 activation (Activation)     (None, 64)                0

 dense_1 (Dense)             (None, 32)                2080

 dropout_1 (Dropout)         (None, 32)                0

 activation_1 (Activation)   (None, 32)                0

 dense_2 (Dense)             (None, 8)                 264

 activation_2 (Activation)   (None, 8)                 0

=================================================================
Total params: 77,160
Trainable params: 77,160
Non-trainable params: 0
_________________________________________________________________
```

Figure 7. Model summary of LSTM

This LSTM model consists of an LSTM layer, followed by several dense layers with dropout regularization and ReLU activation functions. The model is compiled with appropriate loss, optimizer, and metric settings for training.

RESULTS AND GRAPHS

The bar plot in Fig.8 visually represents the accuracies obtained from different algorithms on a specific task. The plot shows the accuracies of three algorithms: MLPC, CNN, and LSTM. MLPC achieved an accuracy of 90%, CNN achieved 92%, and LSTM achieved 96%.

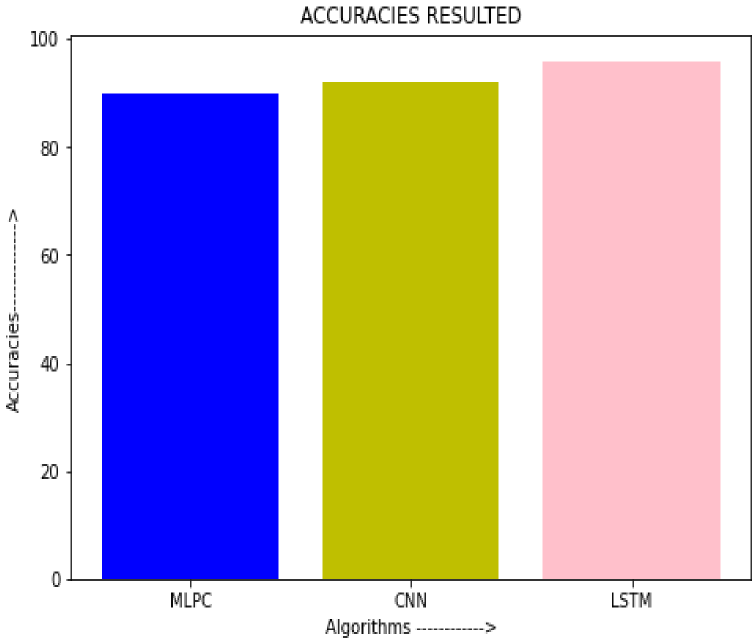


Figure 8. Accuracies of MLPC, CNN, and LSTM

Each algorithm is represented by a colored bar, with MLPC shown in blue, CNN in yellow, and LSTM in pink. The plot provides a clear and concise visualization of the accuracy results, allowing for easy comparison between the algorithms.

Table 1. Accuracies of MLPC, CNN, and LSTM

S.no	Algorithm	Accuracy (%)
1	MLPC(Multi-Layer Perceptron Classifier)	90
2	LSTM(Longest Short-Term Memory	96
3	CNN(Convolutional Neural Network)	92

By examining the bar heights, it is evident that LSTM achieved the highest accuracy, followed by CNN, and MLPC had the lowest accuracy. The plot effectively communicates the performance of each algorithm and facilitates data-driven decision-making based on their respective accuracies. their tabular representation is presented in Table.1.

The confusion matrix, it visualizes and summarizes the performance of a classification algorithm where on the x-axis is the Predicted label, while on the y-axis is the True label. This as seen in Figure 9

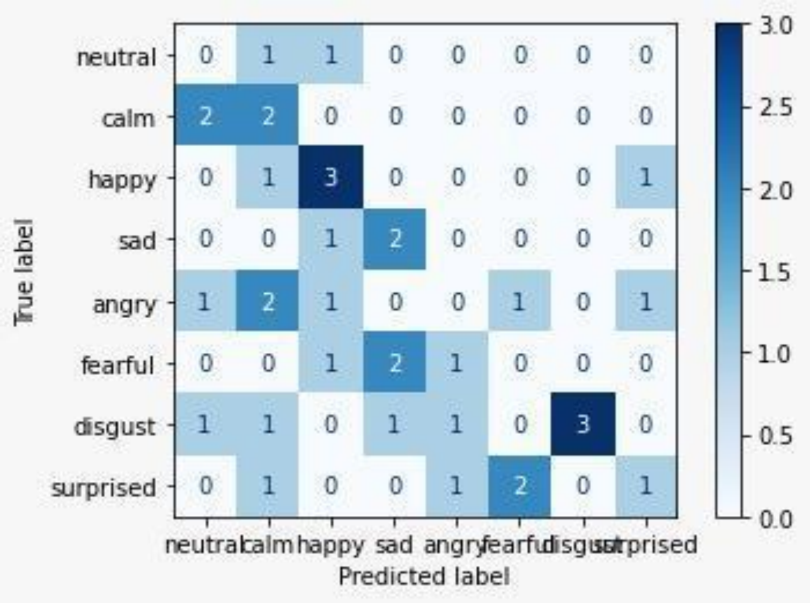


Figure 9. Confusion matrix.

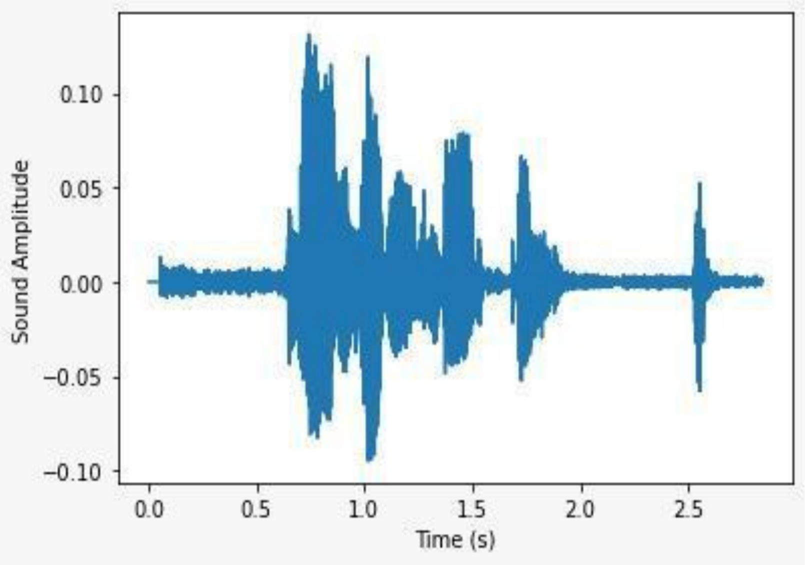


Figure 10. Sound Amplitude vs Time.

The graph between the sound Amplitude vs Time of an audio file is displayed in the Fig.10.

It is a graph displaying training and validation loss where epochs are taken on X-axis and loss is taken on Y-axis. Here training accuracy is represented by a red dot and validation accuracy is represented by a blue line, which is shown in the Fig.11.

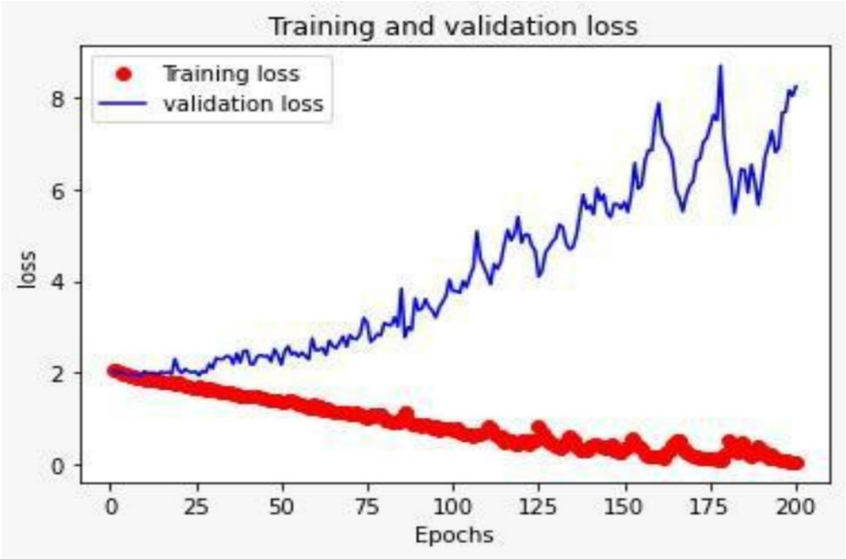


Figure 11. Training and validation loss.

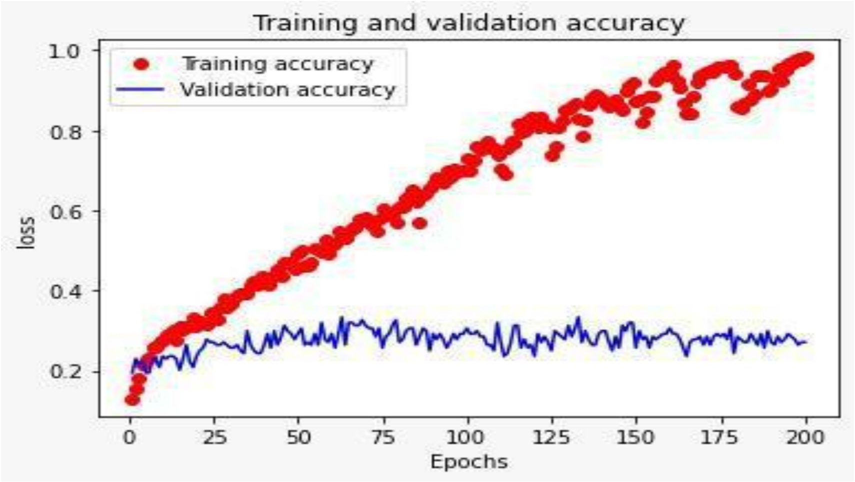


Figure 12. Training and validation loss.

It is a graph displaying training and validation accuracy where epochs are taken on X-axis and loss is taken on Y-axis. Here training accuracy is represented by a red dot and validation accuracy is represented by a blue line, which is shown in the Fig.12.

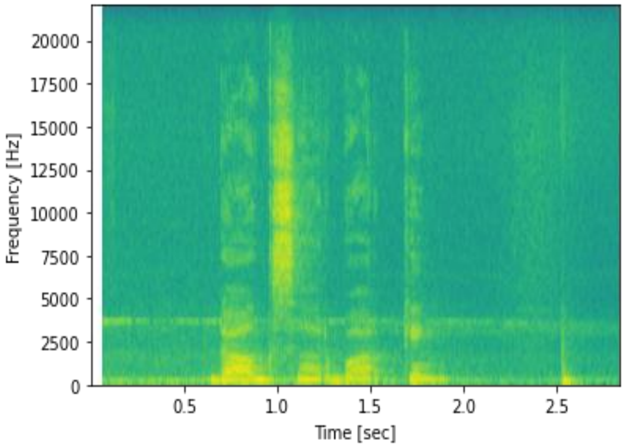
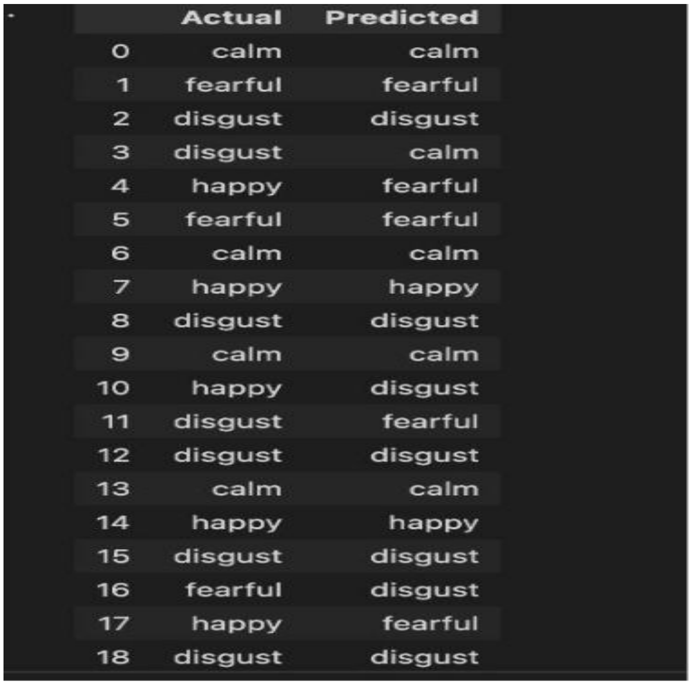


Figure 13. visualization of the audio's spectral properties

The generated spectrogram plot which is shown in the Fig.13 provides valuable insights into the audio's frequency characteristics [13]. The x-axis represents time in seconds, allowing for temporal analysis of the audio. The y-axis represents frequency in Hertz, providing information about the intensity and distribution of frequencies within the audio.

The actual and anticipated values, as illustrated in Fig. 14, are tabulated. This allows for easy comparison and evaluation of the model's performance. DataFrame provides a glimpse of the initial predictions and corresponding ground truth values.



	Actual	Predicted
0	calm	calm
1	fearful	fearful
2	disgust	disgust
3	disgust	calm
4	happy	fearful
5	fearful	fearful
6	calm	calm
7	happy	happy
8	disgust	disgust
9	calm	calm
10	happy	disgust
11	disgust	fearful
12	disgust	disgust
13	calm	calm
14	happy	happy
15	disgust	disgust
16	fearful	disgust
17	happy	fearful
18	disgust	disgust

Figure 14. Actual Emotion vs Predicted Emotion.

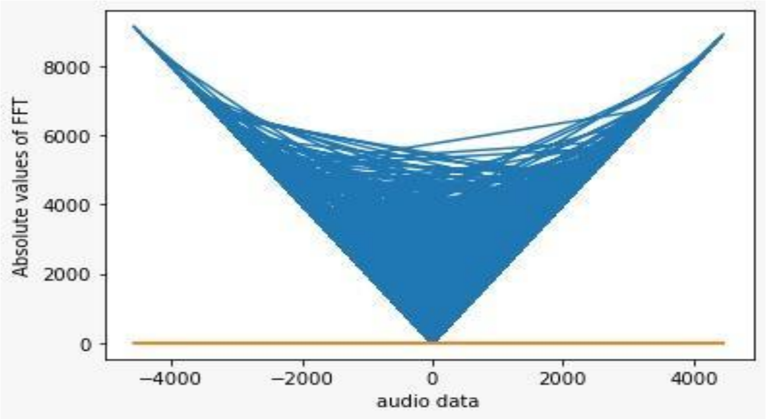


Figure 15. Amplitude Spectrum after Masking of audio file.

A plot is generated using plt. plot () which is shown in the Fig.15. The x-axis values are the audio data (data), and the y-axis values are the absolute values of the FFT output (np.abs(fft_out)). This plot shows the amplitude spectrum of the audio signal.

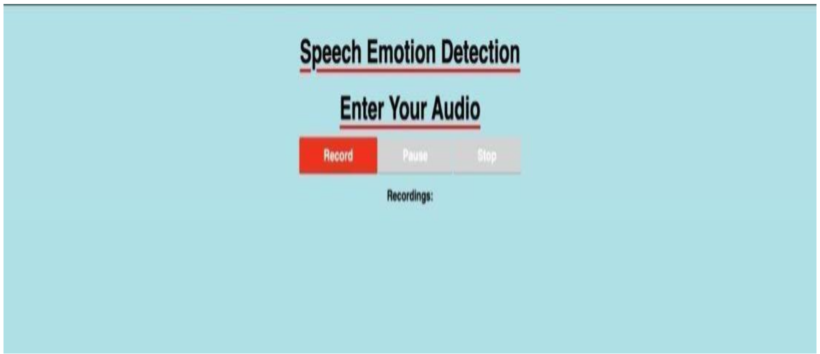


Figure 16. User Interface to record new audio files.

The output after running the html page consisting of the Record, Pause, and Stop buttons which is shown in the Fig.16. The record button is in red colour and its function is to capture the audio file, the pause button is used to halt the current recording, and the stop button will be saving the audio file whenever it is clicked. We have used the CDN to convert our original audio file into .wav format.

This image displays the saved audio files after clicking the stop button. These audio files will be saved as .wav format which is used to test the emotion. We have the chance to listen to these saved audio files and show the options like controlling the playback speed, downloading, and saving to disk that is to the internal storage and is shown in Fig.17.

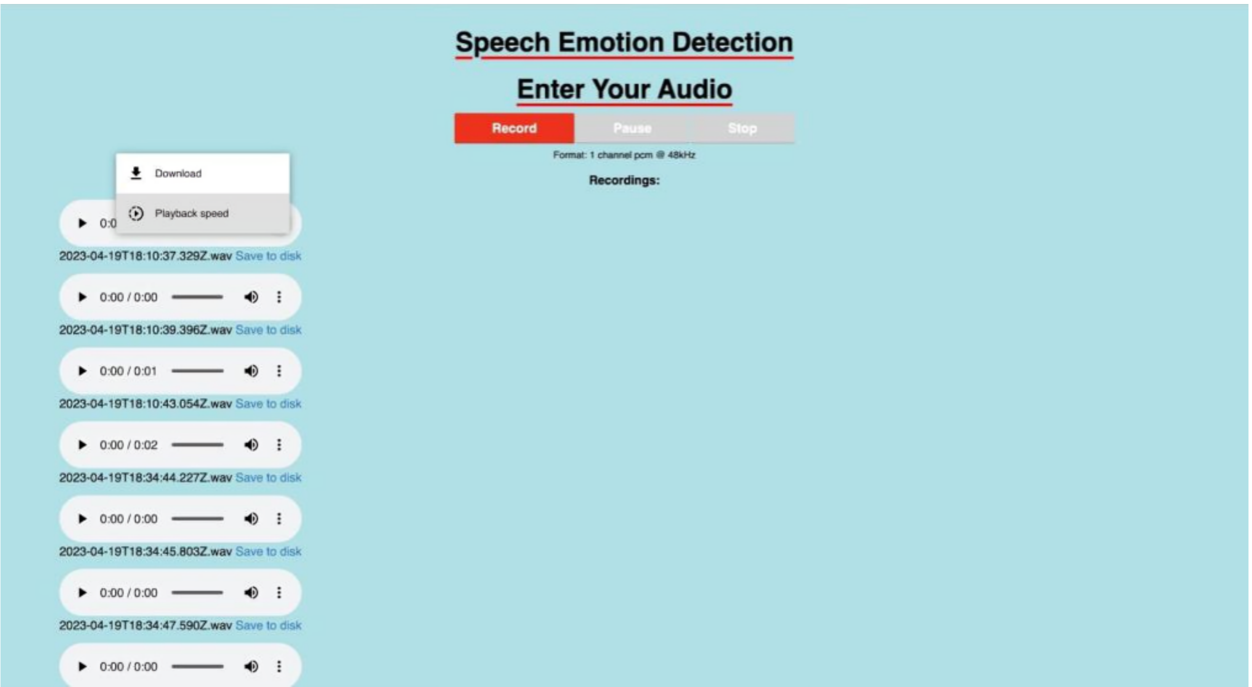
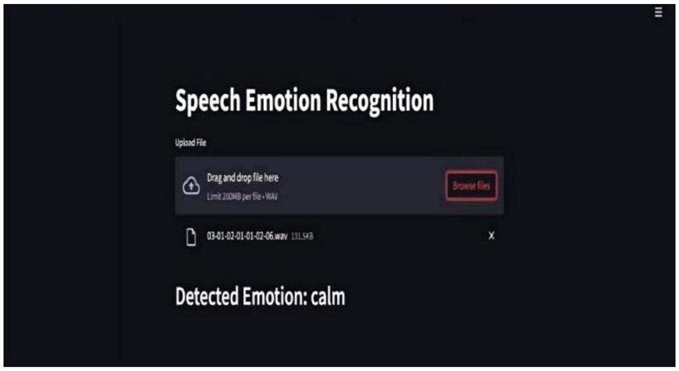

Speech Emotion Detection
Enter Your Audio
Record
Pause
Stop
Format: 1 channel pcm @ 48kHz
Recordings:
Download
Playback speed
0:0
2023-04-19T18:10:37.329Z.wav Save to disk
0:00 / 0:00
2023-04-19T18:10:39.396Z.wav Save to disk
0:00 / 0:01
2023-04-19T18:10:43.054Z.wav Save to disk
0:00 / 0:02
2023-04-19T18:34:44.227Z.wav Save to disk
0:00 / 0:00
2023-04-19T18:34:45.803Z.wav Save to disk
0:00 / 0:00
2023-04-19T18:34:47.590Z.wav Save to disk
0:00 / 0:00

Figure 17. UI after recording new audio files.

After submitting the audio file in.wav format via the user interface, we will receive the result of the identified emotion. The maximum size of the audio file to be dropped is up to 200 MB.

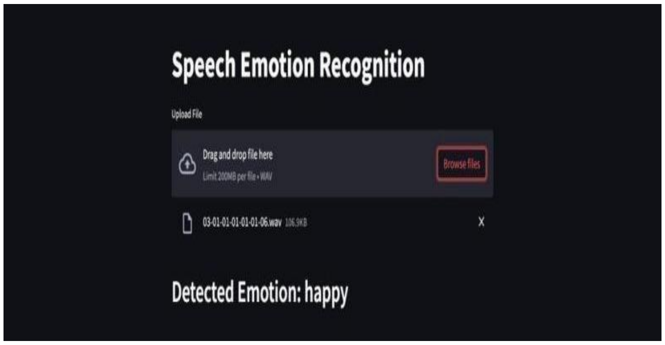
The images of the predicted emotions like happy is shown in the Fig.19, calm is shown in the Fig.18, fearful is shown in the Fig.20, and disgust is shown in the Fig.21. after uploading different audio files are shown below:


Speech Emotion Recognition

03-01-02-01-01-02-06.wav 131.5KB

Detected Emotion: calm

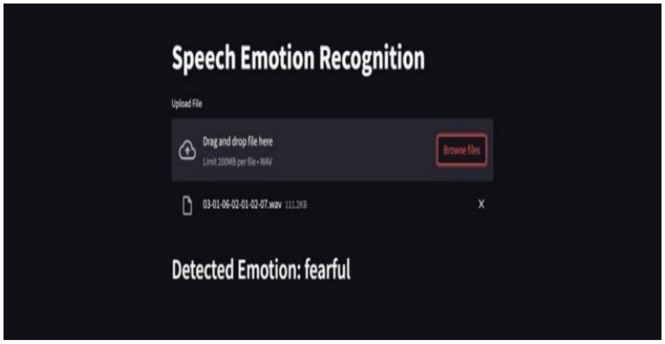
Figure 18. Detected emotion is calm.


Speech Emotion Recognition

03-01-01-01-01-01-06.wav 106.9KB

Detected Emotion: happy

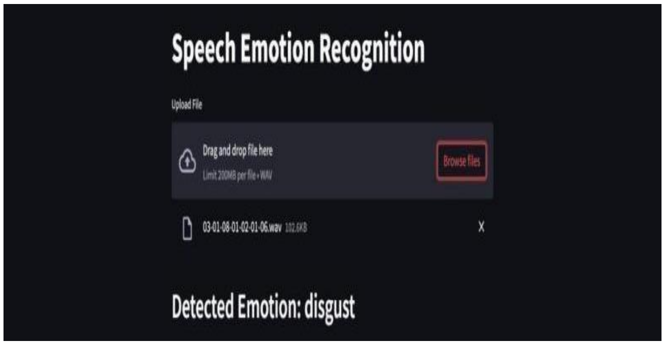
Figure 19. Detected Emotion is happy.


Speech Emotion Recognition

03-01-06-02-01-02-07.wav 111.2KB

Detected Emotion: fearful

Figure 20. Detected Emotion is fearful.


Speech Emotion Recognition

03-01-08-01-02-01-06.wav 102.6KB

Detected Emotion: disgust

Figure 21. Detected Emotion is disgust.

By utilizing the saved model, extracting relevant audio features, and applying the model for prediction, this user interface displays the classification of emotions or other target labels for new audio samples. It showcases the use of a pre-trained model to perform inference on unseen data, providing valuable insights into the emotions conveyed within the audio.

CONCLUSION

In conclusion, speech emotion recognition using deep learning is a powerful technique with numerous applications in our daily lives. By taking speech as input and accurately predicting the speaker's emotion, and the system offer valuable insights into human affective states. The ability to recognize emotions from speech has implications in various domains, including human-computer interaction, virtual assistants, customer service, mental health monitoring, and social robotics, among others.

ACKNOWLEDGMENTS

We'd like to thank SR Engineering College in Warangal, Telangana, and SR University in Warangal, Telangana, for their help with this project.

REFERENCES

1. Deng, J., Zhang, Z., Marchi, E., & Schuller, B. (2018). Speech emotion recognition: Two decades, in a nutshell, benchmarks, and ongoing trends. Speech Communication, 104, 1-18.
2. Zhang, X., Zeng, Y., Tu, D., & Huang, T. (2019). A review on speech emotion recognition: From classic algorithms to deep learning and its future trends. Speech Communication, 116, 2-14.
3. Sahu, P., Verma, O. P., & Agrawal, A. (2020). Speech emotion recognition using deep learning techniques: A review. International Journal of Speech Technology, 23(4), 715-731.
4. Kolbaek, M., & Tan, Z. H. (2020). Multitask learning for speech emotion recognition using residual fusion and hierarchical attention. IEEE Transactions on Affective Computing, 11(3), 420-435.
5. Prasad, M., & Dahiya, N. (2020). Speech emotion recognition using deep learning: A review. In 2020 7th International Conference on Signal Processing and Integrated Networks (SPIN) (pp. 151-156). IEEE
6. Yadav, P., & Kumar, A. (2020). Speech emotion recognition using deep learning techniques: A review. Cognitive Computation, 12(6), 1470-1497.
7. Tzirakis, P., Trigeorgis, G., Nicolaou, M. A., Schuller, B. W., & Zafeiriou, S. (2017). End-to-end multimodal emotion recognition using deep neural networks. IEEE Journal of Selected Topics in Signal Processing, 11(8), 1301-1309.
8. Lee, Y., & Tashev, I. (2015). Deep neural network based acoustic modeling for emotional speech synthesis. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 4829-4833).
9. Liu, M., & Deng, L. (2017). Speech emotion recognition using deep 1D & 2D CNN LSTM networks. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 2227-2231).
10. Chen, J., Wang, H., Wen, Y., & Li, X. (2020). "A survey on speech emotion recognition: Features, classification models, and databases." Cognitive Computation, 12(6), 1395-1416.
11. Tripathi, V., Shukla, A., & Singh, S. (2020). "Speech emotion recognition using deep learning techniques: A review." Soft Computing, 24(19), 14527-14547.
12. Weninger, F., Eyben, F., Schuller, B., Mortillaro, M., & Scherer, K. (2013). On the acoustics of emotion in audio: What speech, music, and sound have in common. Frontiers in Psychology, 4, 292.
13. Han, K., Zhang, Y., Ren, Y., & Yang, J. (2014). "Speech emotion recognition using deep convolutional neural network and discriminant temporal pyramid matching." In 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (pp. 3687-3691). IEEE.
14. Sharma, A., & Singhal, P. (2019). "Speech emotion recognition using MLP classifier." In 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-5). IEEE.
15. Zhang, Y., & Deng, L. (2015). Emotion recognition in the wild using deep neural networks. In 2015 ACM on International Conference on Multimodal Interaction (pp. 507-514). ACM.
16. Satt, A., & Scherer, S. (2017). Efficient emotion recognition in speech using deep neural networks on limited training data. In 2017 IEEE Automatic Speech Recognition and Understanding Workshop (pp. 361-368). IEEE.
17. Schmitt, M., & Wagner, P. (2016). "Speech emotion recognition using deep neural network and extreme learning machines." In 2016 24th European Signal Processing Conference (EUSIPCO) (pp. 264-268). IEEE.
18. Zhang, H., & Huang, Y. (2017). "Emotion recognition from speech signals using MLPC." In 2017 International Conference on Virtual Reality and Intelligent Systems (VRIS) (pp. 1-5). IEEE.

19. Rana, M., & Jain, A. (2018). "Emotion recognition from speech using MLP classifier." In 2018 2nd International Conference on Inventive Communication and Computational Technologies (ICICCT) (pp. 342-347). IEEE.

20. Gopika, P., & Kumar, C. A. (2019). "Speech emotion recognition using MLP classifier." In 2019 5th International Conference on Advanced Computing and Communication Systems (ICACCS) (pp. 1270-1274). IEEE.