NLP PROJECT REPORT

SPEECH EMOTION IDENTIFICATION USING CREMA-D

Team Number: 26

Team Members

Gagana Atchula
Hyderabad Institute of Technology And Management
Computer Science and Engineering(Data Science)
gaganaatchula@gmail.com

Madhu Pathlavath
Hyderabad Institute of Technology And Management
Computer Science and Engineering(Data Science)
madhupathlavath442@gmail.com

N. Saketh Reddy
Hyderabad Institute of Technology And Management
Computer Science and Engineering(Data Science)
sakethreddy2903@gmail.com

Project Link:

https://drive.google.com/file/d/1A-K-gC75Og0diT6b1fIZQFQe887aKRhg/view?usp=sharing

Dataset Link:

https://www.kaggle.com/datasets/ejlok1/cremad

ANNEXURE

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

The human voice is very versatile and carries a multitude of emotions. Emotion in speech carries extra insight about human actions. Human speech conveys information and context through speech, tone, pitch and many such characteristics of the human vocal system. As human-machine interactions evolve, there is a need to buttress the outcomes of such interactions by equipping the computer and machine interfaces with the ability to recognize the emotion of the speaker. Emotions play a vital role in human communication. In order to extend its role towards the human-machine interaction, it is desirable for the computers to have some built-in abilities for recognizing the different emotional states of the user. Today, a large amount of resources and efforts are being put into the development of artificial intelligence, and smart machines, all for the primary purpose of simplifying human life. Research studies have provided evidence that human emotions influence the decision-making process to a certain extent. If the machine is able to recognize the underlying emotion in human speech, it will result in both constructive response and communication.

In order to communicate effectively with people, the systems need to understand the emotions in speech. Therefore, there is a need to develop machines that can recognize the paralinguistic information like emotion to have effective clear communication like humans. One important data in paralinguistic information is Emotion, which is carried along with speech. A lot of machine learning algorithms have been developed and tested in order to classify these emotions carried by speech. The aim to develop machines to interpret paralinguistic data, like emotion, helps in human-machine interaction and it helps to make the interaction clearer and natural. In this study different classification models such as CNN are used to predict in speech sample. The architecture of this CNN includes convolutional layers, batch normalization, max-pooling layers, and fully connected layers. The model is designed to process audio features extracted using Mel Frequency Cepstral Coefficients (MFCCs) for the task of Speech Emotion Recognition.To train the model CREMA– D dataset was used along with Data Augmention.

1.2 PROBLEM STATEMENT

In the era of advancing human-machine interactions, recognizing emotions in speech is crucial. The challenge is to enable machines to understand and respond to the emotional content embedded in human speech. Effective human-computer interactions require the development of systems capable of discerning paralinguistic information, especially emotions, to facilitate clear and natural communication. This study employs a CNN model to predict emotions in speech samples using the CREMA-D dataset and data augmentation techniques. The goal is to enhance machines' ability to interpret diverse emotional expressions in human speech for more nuanced and responsive interactions.

CHAPTER 2
DATASET

2.1 DATASET DETAILS

CREMA-D is a data set of 7,442 original clips from 91 actors. These clips were from 48 male and 43 female actors between the ages of 20 and 74 coming from a variety of races and ethnicities (African America, Asian, Caucasian, Hispanic, and Unspecified). Actors spoke from a selection of 12 sentences.

The sentences were presented using one of six different emotions (Anger, Disgust, Fear, Happy, Neutral, and Sad) and four different emotion levels (Low, Medium, High, and Unspecified).

This dataset is the sheer variety of data which helps train a model that can be generalised across new datasets. Many audio datasets use a limited number of speakers which leads to a lot of information leakage. CREMA-D has many speakers. For this fact, the CREMA-D is a very good dataset to use to ensure the model does not overfit.

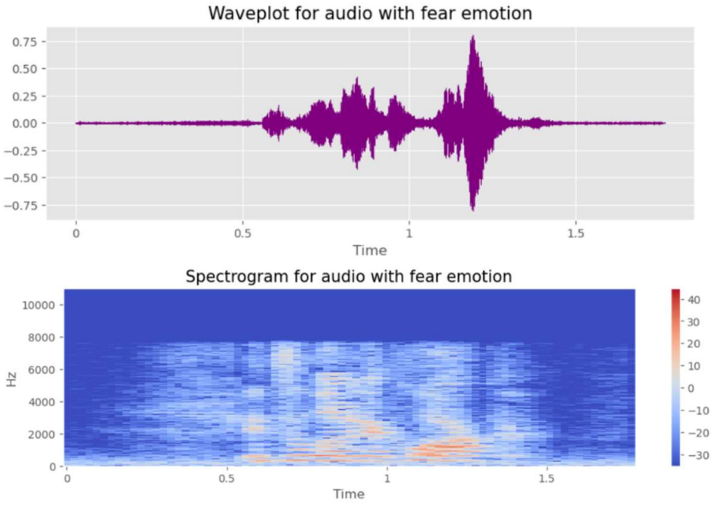
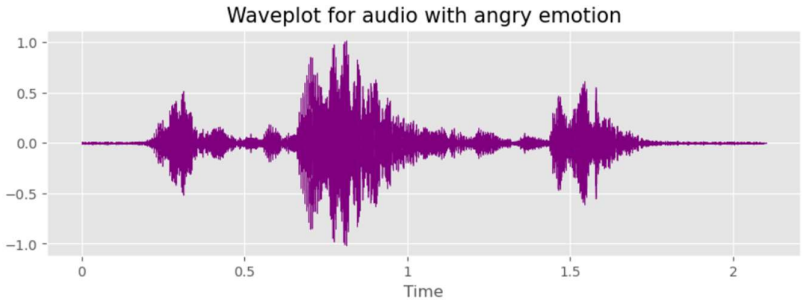


Fig 1.1 *Audio waveplot and spectrogram of fear emotion.*



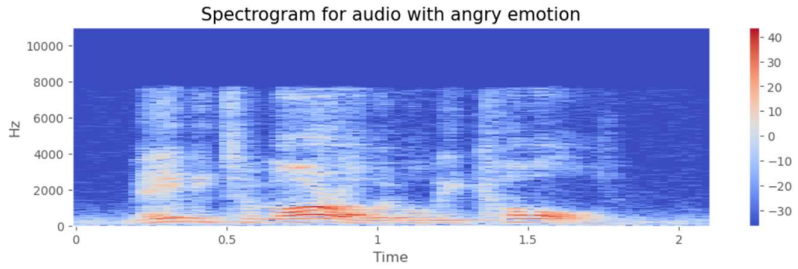


Fig 1.2 *Audio waveplot and spectrogram of angry emotion.*

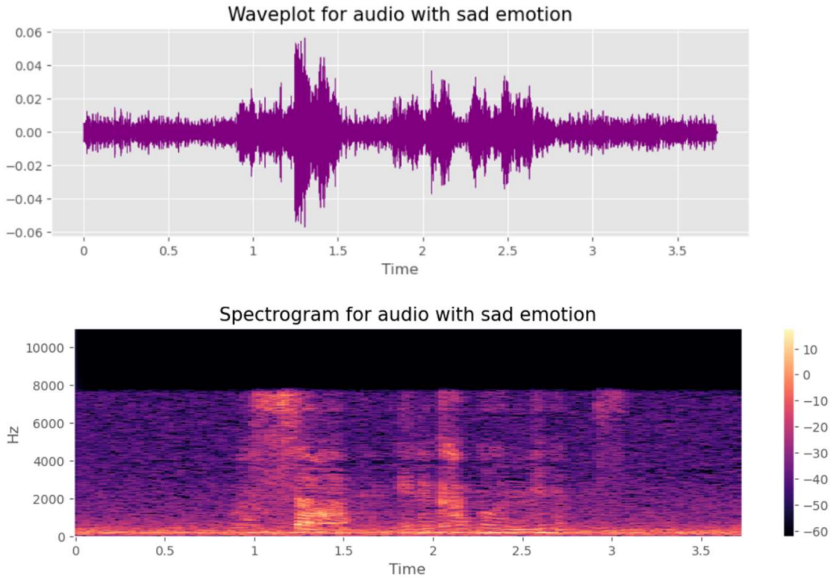


Fig 1.3 *Audio waveplot and spectrogram of sad emotion.*

CHAPTER 3
METHODOLOGY

3.1 METHOD/EXPERIMENTAL SETUP

In this section, we describe the steps we followed to preprocess and augment the audio data, extract and select the audio features, design and train the conv1D and RF models, and compare and analyze the results.

Data preprocessing and augmentation

We used four datasets of speech emotion recognition: CREMA, dataset contains speech samples of different actors expressing different emotions. We used eight emotion classes: calm, happy, sad, angry, fearful, disgust, surprised and neutral. All audios were converted to a common sampling rate of 16 kHz and a common bit depth of 16 bits.

To increase the size and diversity of our data, we augmented each audio file using techniques: adding noise, pitch shifting and time stretching. Noise defines a function that takes an audio data array as input and adds some random noise to it. The noise amplitude is proportional to the maximum value of the data and a random factor between 0 and 0.035. The noise is generated from a normal distribution with the same shape as the data array. The function returns the noisy data array as output.

Stretch defines a function called stretch that takes an audio data array and a rate factor as input and stretches or compresses the data in time. The rate factor determines how much the data is stretched or compressed. A rate factor less than 1 means that the data is stretched (slowed down), while a rate factor greater than 1 means that the data is compressed (sped up), where we used rate of 0.7.

Shift defines a function called shift that takes an audio data array as input and shifts it in time by a random amount. The shift range is determined by a random integer between -5 and 5 multiplied by 1000. The function uses the numpy. Roll function to perform the circular shift of the data array. The function returns the shifted data array as output.

Pitch defines a function called pitch that takes an audio data array, a sampling rate and a pitch factor as input and changes the pitch or frequency of the data. The pitch factor determines how much the pitch is changed. A pitch factor less than 1 means that the pitch is lowered, while a pitch factor greater than 1 means that the pitch is raised. The function uses the librosa.effects. pitch_shift function to perform the pitch shifting. The function returns the pitch-shifted data array as output. We used a pitch factor of 0.7.

Audio feature extraction and selection We extracted various types of audio features from the preprocessed and augmented data: MFCC, Chroma, Mel, SCF, Tonnetz and ZCR. These features capture different aspects of the spectral and temporal characteristics of the speech signals that are relevant for emotion recognition. We used the librosa library in Python to compute these features. All the audio processing techniques were implemented by Librosa package in Python. The following sections will briefly talk about the considered processed.

1. **Zero crossing rate**

Zero-crossing rate (ZCR) is a measure of the number of times that an audio signal crosses the zero amplitude level in a given time frame. ZCR captures the temporal information and noise level of speech that can be related to emotion expression. The process could be written as

$$zcr = \frac{1}{T-1}\sum_{t=1}^{T-1} 1_{R<0}\left(s_t s_{t-1}\right)$$

Where s is a signal of length T and $1_{R<0}$ is an indicator function.

For instance, assuming the sampling rate is 22,050 HZ, and frame size of 2,048 samples, then, frame size is 2,048/22,050 = 0.0929 seconds, and hop length is 512/22,050 = 0.0232 seconds, where 512 is the hop length in samples. Now the number for a 3 second audio is

$$\frac{audio\ length - frame\ size}{hop\ length} + 1 = \frac{3 - 0.0929}{0.0232} + 1$$

So we would have 126 column for this single value. We used the average of all and use that for our ZCR.

Again, the ZCR function returns a matrix with one row and as many columns as frames. By transposing this matrix, it becomes a matrix with one column and as many rows as frames. Then, we takes the mean value along the row axis.

2. **Mel-frequency cepstral coefficients**

The Mel-frequency cepstral coefficient (MFCCs) was used, where MFCCs describe the overall shape of the spectral envelope of an audio signal. The process could be summarized as extraction of short-time Fourier transform (STFT) magnitude spectrogram, mapping the STFT into Mel scale by use of triangular overlapping windows and getting its log, and finally application of the discrete cosine transform (DCT) to the previous stage.

Mel-frequency cepstral coefficients (MFCC) are a representation of the short-term power spectrum of an audio signal based on a nonlinear Mel scale of frequency. MFCC are widely used for speech recognition and analysis because they mimic the human auditory system and capture the timbral and harmonic information of speech.

MFCCs are widely used in various fields, such as speech recognition, speaker identification, music information retrieval, and audio compression. They capture the spectral envelope, the shape of the power spectrum of sound. It describes the variation of energy across different frequency bands of the sound and discard some irrelevant details, such as pitch and noise. They can also be combined with other features, such as delta and delta-delta coefficients, which represent the changes in MFCCs over time. A frequency measured in Hertz (f) can be converted to the mel scale using the following formula:

$$Mel\left(f\right) = 295\log_{10}\left(1 + \frac{f}{700}\right)$$

3. Roll-off frequency

The roll-off frequency for each frame is defined as the center frequency for a spectrogram bin where at least most of (85%) the spectrum energy of the frame spectrum is contained in the bin and the bins below. The technique is useful in distinguishing audios with different energy distributions.

In other words, roll-off frequency is a feature that measures the frequency below which a certain percentage of the total energy of the spectrum is contained. It can be used to distinguish between harmonic and noisy sounds, as harmonic sounds tend to have lower roll-off frequencies than noisy sounds. Roll-off frequency can also be used for emotion classification, as different emotions may have different spectral distributions.

Some studies have used roll-off frequency as one of the features for speech emotion recognition, along with other features such as MFCCs, pitch, and energy. It does not account for the temporal dynamics and prosodic variations of speech. Therefore, it may make sense to use roll-off frequency in combination with other features and methods for emotion classification.

The equation for roll-off frequency is

$$R = \underset{k}{argmin}\left(\sum_{i=0}^{k} s(i) \geq roll_{percent} \cdot \sum_{i=0}^{n} S(i)\right)$$

The equation works by finding the smallest value of k such that the sum of the power spectrum from 0 to k is greater than or equal to a certain percentage of the total sum of the power spectrum from 0 to n. This percentage is called roll_percent and it is usually set to 0.85. The value of k corresponds to the frequency bin that contains the roll-off frequency. The roll-off frequency is the frequency below which most of the energy of the spectrum is concentrated.

4. Spectral contrast

Spectral contrast feature is based on the past study which used the technique for music type classification, which present relative spectral distribution, instead of average spectral envelop. Although the technique was employed in music type classification in the past study here, we used that for the emotions associated with each audio. Librosa estimated this feature by dividing spectrogram frame of s into sub bands, and then estimating energy contrast by comparing mean energy in the peach energy or top quantile to that of the bottom quantile or valley energy.

Spectral contrast feature (SCF) is a representation of the spectral peak and valley structure of an audio signal based on the contrast between spectral subbands. SCF captures the spectral dynamics and texture of speech that can be related to emotion expression.

We compute the spectral contrast for each frame of the audio data and then take the mean value across all frames. The equation for computing spectral contrast is

$$C\left(k\right) = 10.log_{10}\left(\frac{V(k)}{P(k)}\right)$$

where $C(k)$ is the spectral contrast in the kth sub-band, $V(k)$ is the spectral valley in the kth sub-band, and $P(k)$ is the spectral peak in the k-th sub-band. The spectral valley and peak are computed by finding the minimum and maximum values of the spectrum within a specified quantile range in each sub-band. The equation works by capturing the relative energy distribution across different frequency regions of the spectrum.

5. **Harmonic change in tonal centroid features**

This measure detect harmonic changes in audios, by projecting chroma features into a 6 dimensional space. Tonnetz representation is a representation of the tonal space or harmonic relations of an audio signal based on a geometric model that maps pitches to points in a two-dimensional lattice. Tonnetz representation captures the tonal information of speech that can be related to emotion expression. It consists of six dimensions: fifth, minor third, major third, tonic, subdominant, and dominant. The equation for tonnetz is:

$$T = Q^{\intercal}C$$

where T is the tonnetz matrix with shape (6, t), Q is a constant matrix that maps chroma features to tonnetz features, and C is the chroma matrix with shape (12, t).

6. **Chromagram from a waveform or power spectrogram**

Chroma features are powerful representation for music audio where the entire spectrum is projected into 12 bins, representing 12 distinct chroma, where “Shepard Tones” which consist of a mixture of sinusoids carrying a particular chroma were used. Chroma features In the previous study chroma based audio features was carried out for dialect identification.

The value is a representation of the pitch content of a sound or music octave. This could be written as

$$C = \frac{F^{\intercal} S}{||F^{\intercal} S||_{\infty}}$$

Where S is the power spectrogram, and F is a chroma filter bank matrix that maps each frequency bin to a chroma bin.

7. **Root-mean-square**

The root-mean-square (RMS) value is a measure of the average energy or amplitude of a signal. It is computed by squaring each sample in a frame, taking the mean of the squares, and then taking the square root of the mean. The equation for RMS is:

$$RMS = \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} x[n]^2}$$

x is the audio signal, which is a sequence of samples, where $x[n]$ is the nth sample in a frame of length N. N is the length of the frame, which is a segment of the audio signal that we want to analyze.

8. **Mel-scaled spectrogram**

The function feature.melspectrogram uses both the STFT equation and the Mel filter bank equation to compute a mel-scaled spectrogram. It also uses the Mel scale conversion equations to construct the Mel filter bank.

The steps for calculating the function are as follows:

- 1- If the input is a time-domain signal y, compute the power spectrogram S, using the STFT equation:

$$S\left[m, k\right] = \left|\sum_{n=0}^{N-1} y\left[n\right] w\left[n - mH\right] e^{-j2\pi nkn/N}\right|^2$$

Where k is the index of frequency bin, ranging from 0 to K-1, where K is the number of frequency bins. Where the number of frequency bins is equal to 1-n_fft/2, where n_fft is the FFT window size. The FFT window size is the number of samples used to compute the discrete Fourier transform (DFT) of a segment of the signal. It is also called the frame length or the analysis window length. The FFT window size determines the frequency resolution and the time resolution of the STFT.

$j[n]$ is the nth sample of the signal, N is the window length, H is the hop length, $w[n]$ is the window function and j is the imaginary unit.

- 2- Now, construct Mel filter bank matrix M using the Mel scale conversion Eq 2

The Mel filter bank matrix M has shape (n_mels, 1 + n_fft/2), where n_mels is the number of Mel filters. Each row of M corresponds to a triangular filter that covers a certain range of frequencies on the Mel scale.

- 3- Apply the Mel filter bank matrix M to the power spectrogram S using the Mel filter bank equation:

$$S_m\left[m, l\right] = \sum_{k=0}^{K-1} S[m, k] M[l, k]$$

Where $S_m[m, l]$ is the mel-scaled spectrogram coefficient at time frame m and filter index l, and k is the number of frequency bins (1-n_fft/2). The mel-scaled spectrogram S_m has shape (n_mels, t).

We concatenated these feature types into one feature vector for each audio file, resulting in a feature vector with a wide dimensions. However, not all features may be equally relevant or useful for emotion recognition. Although conv1D might do feature selection itself, RF might not be able to do so. Therefore, we applied feature selection methods to reduce the dimensionality and complexity of the feature vectors and improve the performance of the models. We used random forest-based feature selection (RFS). RFS is a method that selects a subset of features based on their importance scores derived from a random forest model.

We designed and trained two models for SER: conv1D and RF. Conv1D is a type of convolutional neural network (CNN) that applies one-dimensional convolution filters to the input feature vectors to learn local and global patterns for emotion recognition. RF is a type of ensemble learning method that combines multiple decision trees to produce a robust and accurate prediction for emotion recognition. We briefly describe each model below:

We used a conv1D model with four convolutional layers, each followed by a batch normalization layer, a rectified linear unit (ReLU) activation function, and a max pooling layer.

Layer	Output Shape	Param #
Conv1D	(2376, 512)	3072
BatchNormalization	(2376, 512)	2048
MaxPooling1D	(1188, 512)	0
Conv1D_1	(1188, 512)	1311232
BatchNormalization_1	(594, 512)	2048
MaxPooling1D_1	(594, 256)	0
Conv1D_2	(594, 256)	655616
BatchNormalization_2	(297, 256)	1024
MaxPooling1D_2	(297, 256)	0
Conv1D_3	(297, 256)	196864
BatchNormalization_3	(149, 256)	1024
MaxPooling1D_3	(149, 128)	0
Conv1D_4	(149, 128)	98432
BatchNormalization_4	(75, 128)	512
MaxPooling1D_4	(75, 128)	0
Conv1D_5	(75, 128)	49280
BatchNormalization_5	(38, 128)	512
MaxPooling1D_5	(4864)	0
Flatten	(512)	0
Dense	(512)	2490880
BatchNormalization_6	(256)	2048
Dense_1	(256)	131328
BatchNormalization_7	(128)	1024
Dense_2	(128)	32896
BatchNormalization_8	(6)	512
Dense_3		774

Table 1

After the convolutional layers, we added a flatten layer, a dropout layer with a rate of 0.5, and a dense layer with 8 units and a softmax activation function. The dense layer produced the final output of the model, which was a probability distribution over the 8 emotion classes. We used categorical cross-entropy as the loss function, Adam as the optimizer, and accuracy as the

metric. We trained the model for 100 epochs with a batch size of 32, using early stopping with a patience of 10 epochs to prevent overfitting. Table 1 shows the details of the conv1D model architecture.

It is clear that the shape at each consecutive layer is estimated as follows:

$$output_{length} = (input_{length} + 2 * padding - dilation * (kernel_{size} - 1) / stride + 1$$

We used the default values for padding, dilation (spacing) and stride (step size) as 0, 1, and 1, respectively. As the same padding was used, it is assumed that padding is $(kernek_{size}-1)/2$, so after placing the values in the above equation we have 87, for instance, look at Table 1.

The non-linear activation function of Rectified Linear Unit (ReLU), f(x) = max(0, x) was used, due to its simplicity, avoiding vanishing gradient, and introducing sparsity to network resulting in reduction in overfitting. Softmax was used in the final layer due to its capability of converting k real numbers into k probabilities, summing up to 1.

We used an RF model with 100 decision trees, each with a maximum depth of None and a minimum samples split of 2. The RF model used the gini criterion to measure the quality of the splits, and bootstrap sampling with replacement to create the training subsets for each tree. The RF model produced the final output by averaging the predictions of all the trees, which was a probability distribution over the 8 emotion classes. We used accuracy as the metric to evaluate the performance of the model.

We split the data into two sets: training and testing. We used 80% of the data for training and 20% for testing. We used stratified sampling to ensure that each set had a balanced distribution of emotion classes. We used the training set to fit the models and the testing set to evaluate the performance of the models.

Few points should be highlighted regarding the choice of various values in Conv1d. The kernel size should be small enough to capture local patterns in the input but not too small to miss important info. On the other hand, the number of filters should be large enough to capture the features' diversities but not too high to result in computational inefficiency and overfitting. To prevent the possible loss of information, a lower stride value was used to preserve information but larger than 1 to address possible computational issues. Maxpooling1D works by taking the maximum value of spatial window of the size of pool size. Better performance of random forest is dependent on the data characteristics and the architects of the conv1d model. Fig 2 depicts the discussed methodological steps taken in this study.

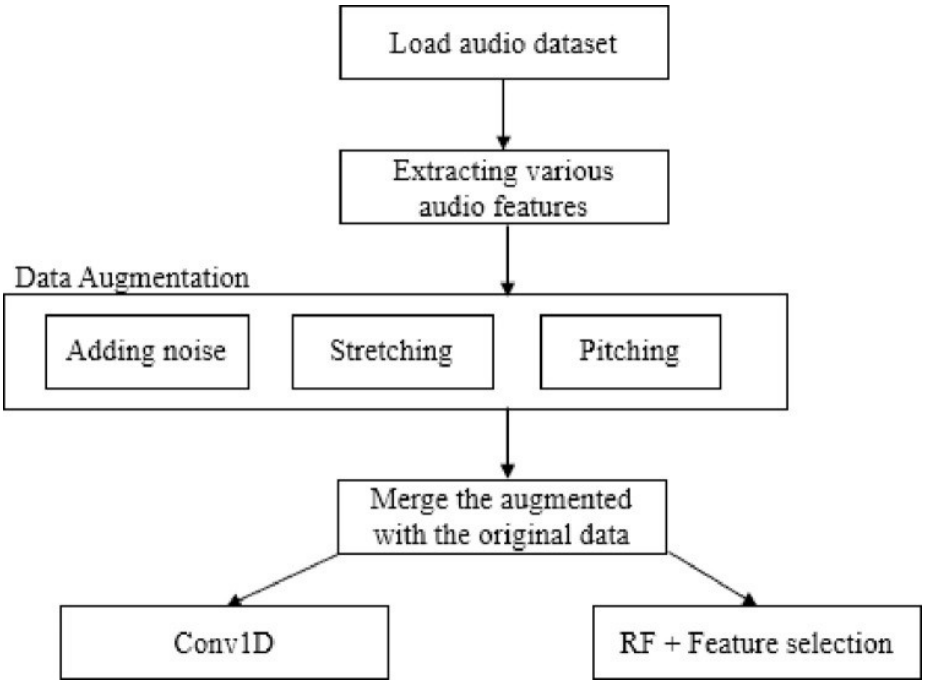


Fig .2 *Methodological steps for audio classification.*

3.2 DATA FLOW DIAGRAM

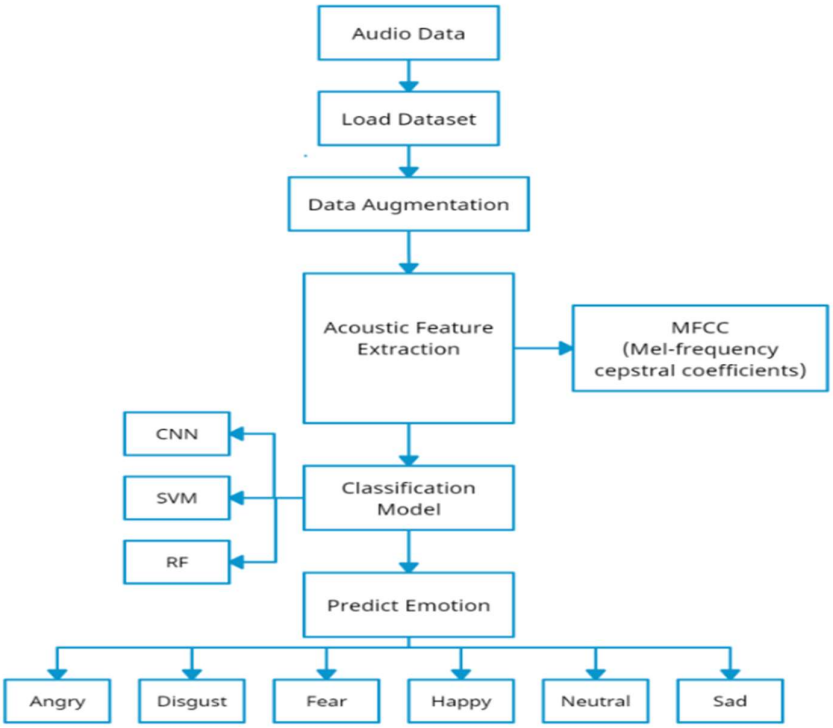


Fig.3 Data Flow Diagram

In the first step the Data Set is loaded, from which one audio file is taken. Then that audio file is passed through Data Augmentation, where one without Augmentation, Noise, High Speed, Low Speed, Stretch, Pitch are created. Then the Feature extraction is done from each data augmented audio file through MFCC.

This same process is being performed for all the audio files present in the dataset. Finally after the feature extraction from all the file is done . We randomly split our dataset into a train (80%) and test (20%) set. The same split is used for all the experiments to ensure a fair comparison. Here for Dependent variable (One Hot Encoding (CNN)).

Classification Models are trained and their Evaluation metrics are been noted.

3.3 CLASSIFICATION ALGORITHM

CNN

Convolution layer

A convolution layer is a fundamental component of the CNN architecture that performs feature extraction, which typically consists of a combination of linear and nonlinear operations, i.e., convolution operation and activation function.

Nonlinear activation function

The outputs of a linear operation such as convolution are then passed through a nonlinear activation function.The most common nonlinear activation function used presently is the rectified linear unit (ReLU).

Pooling layer

A pooling layer provides a typical down sampling operation which reduces the in-plane dimensionality of the feature maps in order to introduce a translation invariance to small shifts and distortions, and decrease the number of subsequent learnable parameters.

Fully connected layer

The output feature maps of the final convolution or pooling layer is typically flattened, i.e., transformed into a one-dimensional (1D) array of numbers (or vector), and connected to one or more fully connected layers, also known as dense layers, in which every input is connected to every output by a learnable weight. Once the features extracted by the convolution layers and down sampled by the pooling layers are created they are mapped by a subset of fully connected layers to the final outputs of the network, such as the probabilities for each class in classification tasks. The final fully connected layer typically has the same number of output nodes as the number of classes.

Last layer activation function

The activation function applied to the last fully connected layer is usually different from the others. An activation function applied to the multiclass classification task is a softmax function which normalizes output real values from the last fully connected layer to target class probabilities, where each value ranges between 0 and 1 and all values sum to 1.

4. EXPERIMENTAL RESULTS

In the culmination of our investigation, the implemented Convolutional Neural Network (CNN) with one-dimensional convolutional layers (Conv1D) has demonstrated a notable capability in discerning and categorizing emotional states embedded in speech samples. The model underwent comprehensive testing on the designated test dataset, and the ensuing results provide valuable insights into its performance.

Accuracy on Test Data: 95.52%

The obtained accuracy of 95.52% indicates that our model successfully classified emotional states in the test dataset with a commendable precision. This metric showcases the proportion of accurately predicted emotional states relative to the total instances in the test data.

The CNN's one-dimensional convolutional layers played a pivotal role in effectively capturing intricate patterns within Mel Frequency Cepstral Coefficients (MFCCs), contributing to the model's ability to discern subtle nuances in speech that correspond to different emotional expressions. This result underscores the suitability of the chosen architecture for Speech Emotion Recognition tasks.

Furthermore, the robustness of the model in handling the complexities of paralinguistic information, such as emotion conveyed through speech, underscores its potential applicability in human-machine interaction scenarios. The integration of such models into systems holds promise for enhancing the clarity and naturalness of communication between humans and machines. Overall, the achieved accuracy substantiates the efficacy of the Conv1D CNN architecture for Speech Emotion Recognition.

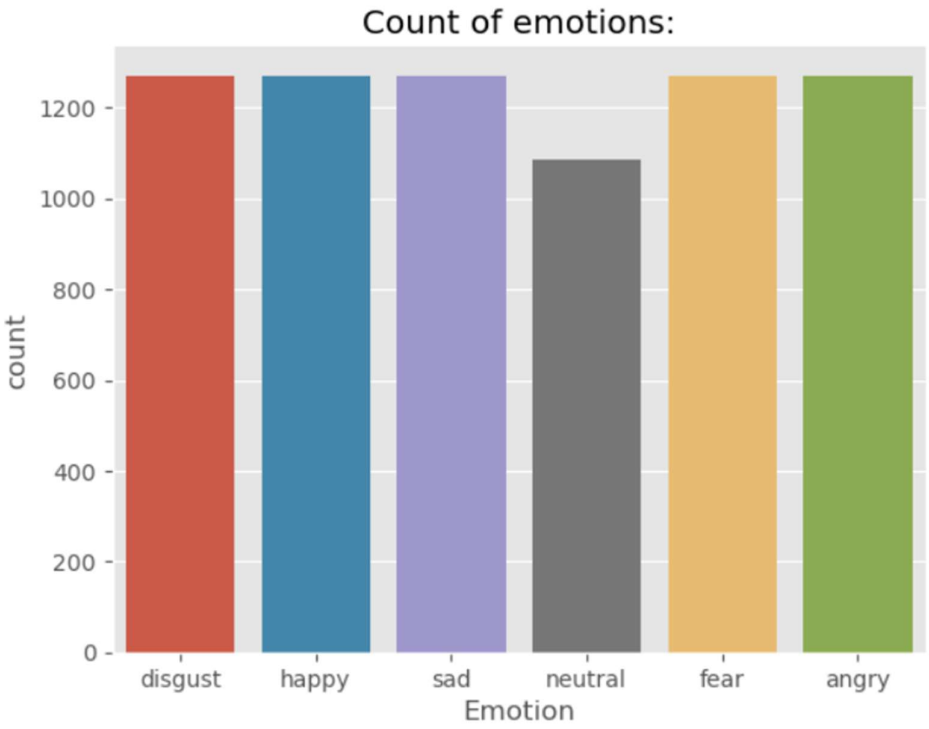


Fig.4 *Count of emotions*

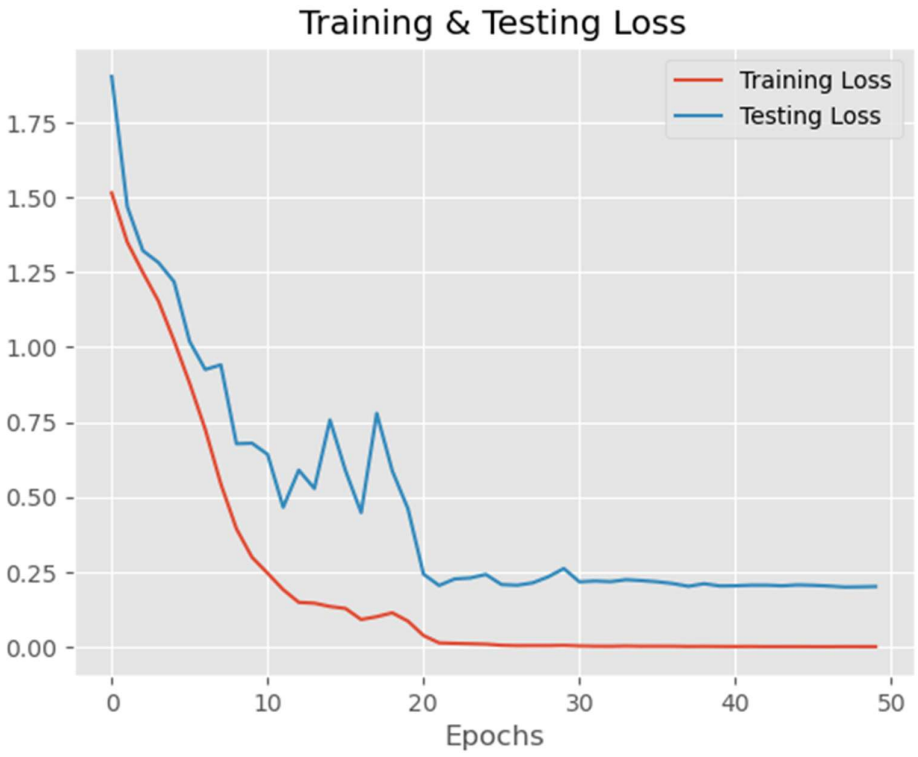


Fig .6 *Training and Testing Loss*

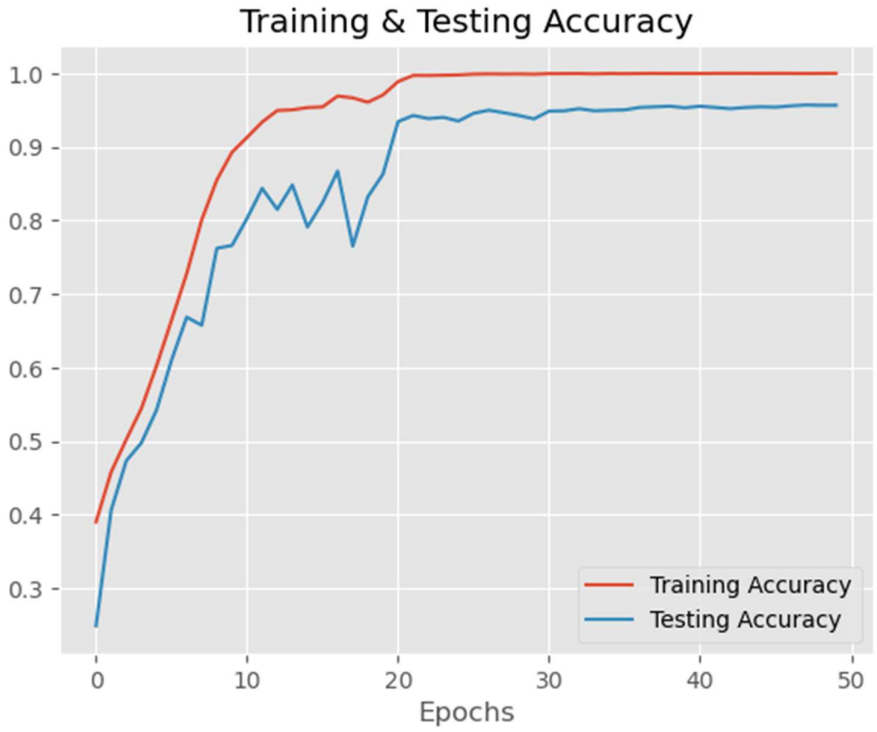


Fig .7 *Training and Testing Accuracy*

4.1 Actual and Predicted Values

The table below illustrates the outcomes of our model's predictions on the test dataset.

	Actual	**Predicted**
0	angry	angry
1	angry	angry
2	fear	fear
3	sad	sad
4	neutral	neutral
…	…	…
5949	neutral	neutral
5950	disgust	disgust
5951	angry	angry
5952	happy	happy
5953	happy	happy

Table-2

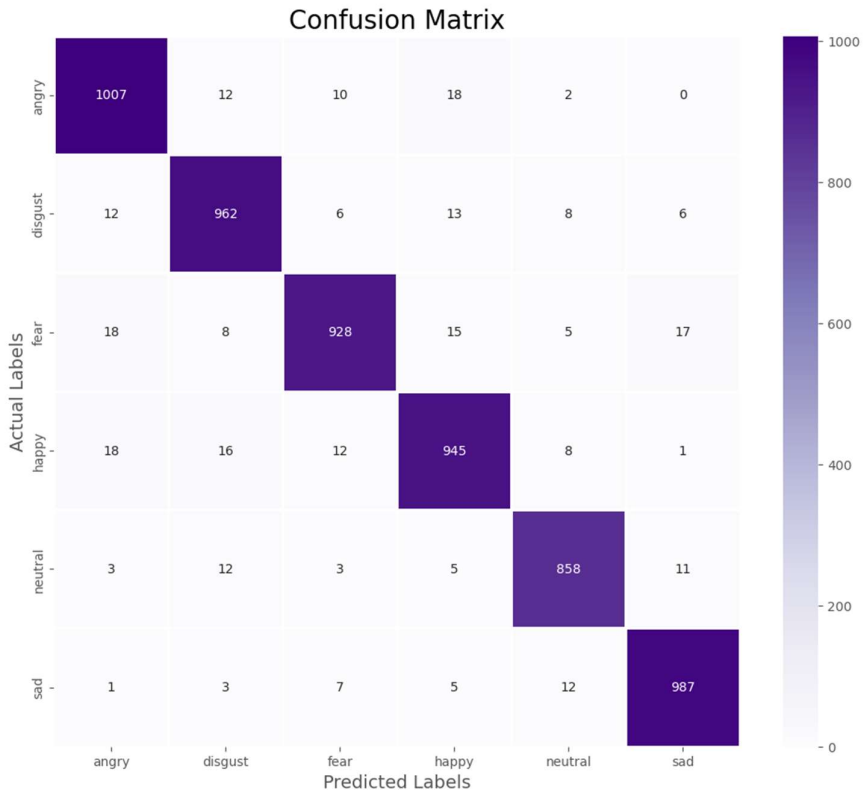


Fig.5 *Confusion Matrix For CNN*

Precision and recall scores for each emotion using Conv1D method

	Precision	**Recall**	**F1-score**	**Support**
Angry	0.95	0.96	0.96	1049
Disgust	0.95	0.96	0.95	1007
Fear	0.96	0.94	0.95	991
Happy	0.94	0.94	0.94	1000
Neutral	0.96	0.96	0.96	892
Sad	0.97	0.97	0.97	1015
Accuracy			0.96	5954
Macro avg	0.96	0.96	0.96	5954
Weighted avg	0.96	0.96	0.96	5954

CONCLUSION

An approach to emotion recognition in speech based upon CNN classifiers was presented. CREMA – D dataset was used and a total of six in 6 emotion output classes namely angry, fear, disgust, happy, sad and neutral. Feature extraction was done using MFCC, total of 58 features were extracted for emotion prediction. Data Augumention was performed on the audio samples to increase the variety of dataset so that the model could perform better. For CNN we tried multiple epochs, and the loss function used is categorical cross entropy, the optimizers used was adam. For all the models accuracy, precision, recall, F1 score and support was noted. Hence CNN performed the best with an accuracy of 95.52%. The report presents only the prediction of six human emotions using speech. It can be expanded to predict more human emotions. The CNN classification algorithm predicted some of the samples belonging to sad class predicted to fear class. This can be rectified by extracting more features to better distinguish between these two class. The plan to further make the Speech Emotion Recogniton system more roboust & real time analysis would be done.

REFERENCE

[1] K.V .Krishna Kishore, P.Krishna Satish, "Emotion Recognition in Speech Using MFCC and Wavelet Features", 3rd IEEE International Advance Computing Conference (IACC) , 2013.

[2] Yixiong Pan, Peipei Shen and Liping Shen, "peech Emotion Recognition Using Support Vector Machine ", International Journal of Smart Home, 2012

[3] Ashish B. Ingale and Dr.D.S.Chaudhari,, " peech Emotion Recognition Using Hidden Markov Model and Support Vector Machi e", International Journal of Advanced Engineering Research and Studies,Vol. 1,Issue 3 , 2012.

[4] Davood Gharavian, Mansour Sheikhan, Alireza Nazerieh, Sahar Garoucy, " peech emotion recognition using FCBF feature selection method and GA-optimized fuzzy ARTMAP neural etw r ", NeuralComputing and Applications , Volume 21, Issue 8 , pp 2115–2126, 2011.