AMITY UNIVERSITY

JHARKHAND

Submitted to:

Amity University Jharkhand

Sentiment Analysis on Audio Signal based on Deep Learning

By

Saransh Kumar

A35705221012

Under the guidance of

Dr. Soumen Kanrar

AMITY SCHOOL OF ENGINEERING AND TECHNOLOGY

AMITY UNIVERSITY JHARKHAND

RANCHI

2023-2024

DECLARATION

I, Saransh Kumar, student of Computer Science and Engineering hereby declare that the NTCC In-House Practical Training titled “Sentiment Analysis on Audio Signal based on Deep Learning” which is submitted by me to Department of Computer Science and Engineering , Amity School of Engineering and Technology, Amity University Jharkhand, in partial fulfilment of requirement for the award of degree of Bachelor of Technology, has not been previously formed the basis for the award of any degree, diploma or other similar title or recognition.

Amity University Jharkhand, Ranchi

Date:

Sign of the Student

Name of Student: Saransh Kumar

Enrolment Number: A35705221012

CERTIFICATE

On the basis of NTCC Term Paper submitted by Saransh Kumar, student of Bachelor of Technology, I hereby certify that the NTCC In-House Practical Training "Sentiment Analysis on Audio Signal based on Deep Learning" which is submitted to Department of Computer Science and Engineering, Amity School of Engineering and Technology, Amity University Jharkhand in partial fulfilment of requirement for the award of the degree of Bachelor of Technology is an original contribution with existing knowledge and faithful record of work carried out by him under my guidance and supervision.

To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Amity University, Ranchi

Date:

Signature of Guide (Internal)	Signature of Guide (External)
Dr. Soumen Kanrar	Name:
Assistant Professor	Designation:
Amity School of Engineering and Technology	
Amity University Jharkhand, Ranchi	

ACKNOWEDGEMENT

While conducting this report, I got support in many ways from many people. Firstly, I am deeply grateful to my project guide, Dr. Soumen Kanrar, who helped me with full devotion and always supported me earnestly whenever it was needed. Without his guidance, mental & moral support, and academic inputs this report was not possible.

This NTCC report could never have seen the light of the day without the co-operation of those clients who participated in this. I am thankful to all of them for giving me their valuable time.

My friends have been biggest support for me at every juncture of life. They manifested their great interest in my research work also and always tried to make things easy for me.

A word of gratitude goes to my family members whose love; affection and understanding have enabled me to complete this endeavour with ease. At the end, I thank to Almighty for giving me courage and strength to conduct this project report.

CONTENT

ABSTRACT

Audio sentiment analysis is a difficult job with many practical applications, such as tracking market mood in spoken interviews or figuring out consumer happiness in call centres. This study introduces a novel method for audio sentiment analysis by focusing just on acoustic characteristics to improve the calibre and nuance of emotional insights. This work expands the possibilities of sentiment analysis in the acoustic realm by eschewing conventional methods like audio-to-speech conversion.

Numerous auditory parameters were extracted in this work, such as contrast, spectral centroid, chroma features, zero-crossing rate (ZCR), root mean square error (RMSE), and Mel-frequency cepstral coefficients (MFCC).

We have created a deep learning model that not only outperforms existing models like LeNet, VGG16, AlexNet, and LSTM networks, but also broadens the scope of sentiment analysis for acoustic voice recognition. In addition to showcasing our model's efficacy, our research highlights the possibility of an inclusive and data-rich approach to audio sentiment analysis.

Keywords – Sentiment Analysis of Audio Data; Time Series Data; Acoustic Features; Deep Learning; LSTM; CNN

INTRODUCTION

A crucial area for machine learning and natural language processing is sentiment analysis in audio data. Its applications range from market trends analysis, customer service, to the study of patients' emotional states in hospital environments. Conventional methods for audio sentiment analysis have frequently been restricted to small datasets and a selection of characteristics. Using a comprehensive methodology this research produces a groundbreaking sentiment analysis model that performs better than existing techniques.

The first step in our journey is the careful gathering and compilation of datasets from a variety of sources, namely as the SAVEE **[8]**, TESS **[10]**, RAVDESS **[6]**, and CREMA-D **[5]**. This collection plays a crucial role in maintaining the depth and diversity of our dataset, which encompasses a wide range of emotional expressions. We used data augmentation techniques to further enhance our dataset and capture a wider spectrum of emotions. This increased the model's capacity to decipher various emotional expressions.

A further crucial component of our research is features extraction. We have chosen particularly a set of acoustic features so that they can comprehensively depict emotional content and capture important aspects of audio signals.

The creation of a deep learning model that advances the field of audio sentiment analysis is at the core of this study. This model outperforms its well-established competitors, such as LeNet, VGG16, AlexNet, and LSTM networks, in terms of accuracy and performance by utilizing the enriched dataset and a variety of acoustic parameters.

AUDIO DATASET

1. *What is audio dataset in sentiment analysis*

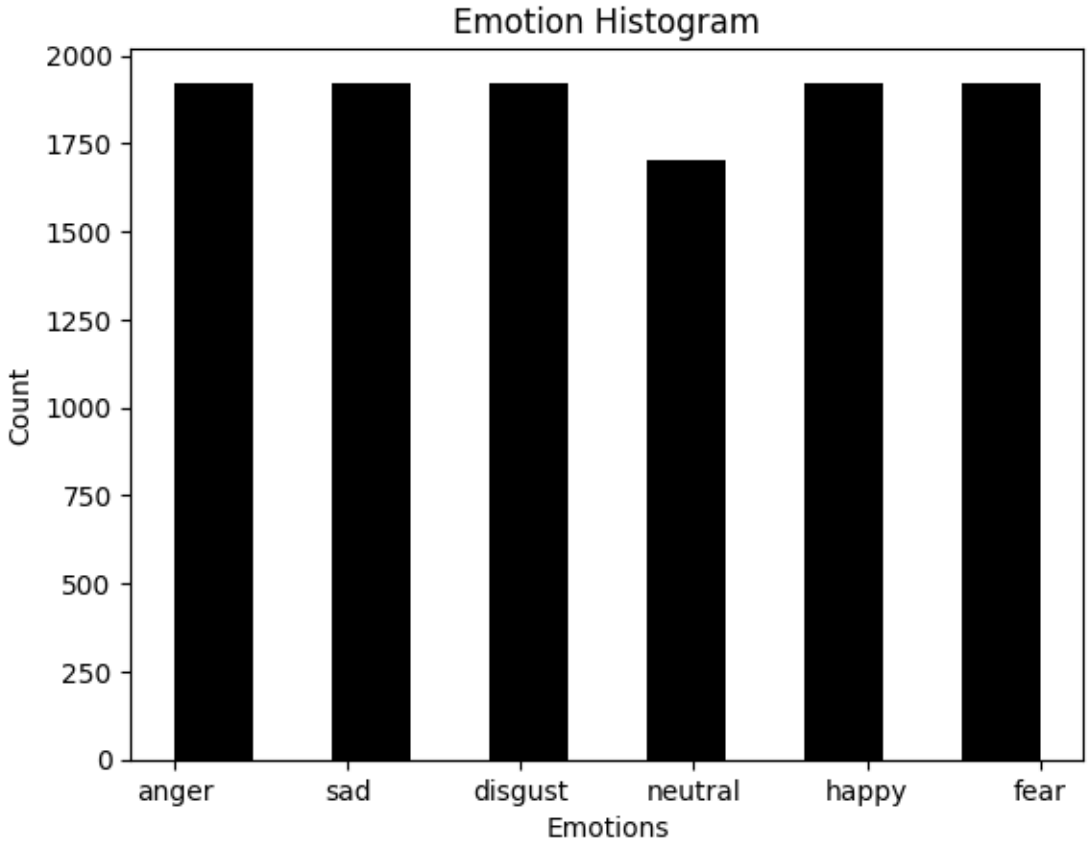
A collection of spoken language audio recordings that have been annotated with sentiments is known as an audio dataset for sentiment analysis. Providing a resource for training and assessing machine learning models or algorithms that can automatically analyse and categorize the emotional tone or sentiment indicated in the audio content is the aim of this kind of dataset.

2. *Compilation of Dataset*

We have collected audio recordings from four reputable and varied sources for this compilation of sentiment analysis datasets – *CREMA-D, RAVDESS, TESS, and SAVEE*.

This compilation's main goal is to provide a resource for researching the emotional content of audio. The six main emotions that we have selected to highlight are:

a. Anger
b. Sadness
c. Disgust
d. Neutrality
e. Happiness
f. Fear



These feelings cover a wide range of human emotions, which makes the dataset useful for machine learning, natural language processing, and affective computing research. Our compilation focuses on these six fundamental emotions to offer a targeted and useful tool for the creation of reliable emotion analysis algorithms.

The diversity and inclusivity of our compilation is one of its strong points. The dataset is applicable to a wide range of applications and cultural contexts because the emotions we have selected are basic and universal. We guarantee that our dataset is representative of a broad range of speakers and scenarios by utilizing recordings from different sources, which promotes more accurate and generalizable outcomes.

DATA PROCESSING

1. *Data Augmentation*

Data augmentation is a method that preserves the underlying meaning or qualities of the original data while applying various changes to artificially expand the size and diversity of a dataset. Data augmentation enhances the generalization and robustness of machine learning and deep learning models when applied to audio data, especially when performing tasks like sentiment analysis, audio classification, and voice recognition.

We have used 3 transformation techniques to augment our data:

a) Time Stretching

The transformation Time Stretch modifies the audio's speed by stretching or compressing it with respect to time. It requires the stretching factors of minimum and maximum rate, which specify a range of values of factors to stretch the data with. As a result, the audio may play more quickly or more slowly while keeping the similar qualities and substance.

To smooth out the unwanted transition noise we apply windowing as the first step.

$$\hat{\boldsymbol{x}}(\boldsymbol{t}) = \boldsymbol{x}(\boldsymbol{t}) \cdot \boldsymbol{win}(\boldsymbol{t})$$

We then apply Fast Fourier Transform to convert time domain signal to frequency domain signal.

$$\boldsymbol{x}(\boldsymbol{\omega}) = \boldsymbol{FFT}\big(\hat{\boldsymbol{x}}(\boldsymbol{t})\big)$$

After the transformation we apply the time stretching function having $\triangle\, \boldsymbol{t}$ as the factor of stretching.

$$\boldsymbol{x}'(\boldsymbol{\omega}) = \boldsymbol{x}(\boldsymbol{\omega}) + \boldsymbol{\omega} \cdot \triangle\, \boldsymbol{t}$$

Then at last we apply Inverse Fourier Transform

$$\boldsymbol{y}(\boldsymbol{t}) = \boldsymbol{IFFT}\big(\boldsymbol{x}'(\boldsymbol{\omega})\big)$$

b) Pitch Shifting

Pitch Shift alters the audio's pitch (or frequency) without changing the length of the file. Within the range delineated by minimum and maximum semitones, it modifies the pitch by a predetermined number of semitones. This alteration mimics variations in the pitch of the speaker's voice.

To smooth out the unwanted transition noise we apply windowing as the first step.

$$\hat{\boldsymbol{x}}(\boldsymbol{t}) = \boldsymbol{x}(\boldsymbol{t}) \cdot \boldsymbol{win}(\boldsymbol{t})$$

We then apply Fast Fourier Transform to convert time domain signal to frequency domain signal.

$$\boldsymbol{x}(\boldsymbol{\omega}) = \boldsymbol{FFT}\big(\hat{\boldsymbol{x}}(\boldsymbol{t})\big)$$

After the transformation we apply the pitch shifting function having $\boldsymbol{f}$ as the factor of pitch shifting.

$$\boldsymbol{x'(\omega) = f \cdot x(\omega)}$$

Then at last we apply Inverse Fourier Transform

$$\boldsymbol{y(t) = IFFT\big(x'(\omega)\big)}$$

c) Gaussian Noise Addition

It adds arbitrary Gaussian noise to the audio stream. Minimum and maximum amplitude regulate the noise's amplitude. Through the simulation of real-world noise and interference that may arise during recording or transmission, this noise addition serves to strengthen the audio data.

Let, $\boldsymbol{G(\mu, \sigma)}$ be the Gaussian noise with $\boldsymbol{\mu = 0}$ as mean and $\boldsymbol{\sigma}$ be the standard deviation.

We then scale the noise with a factor of $\boldsymbol{\alpha}$.

$$\boldsymbol{noise(t) = \alpha \cdot G(\mu, \sigma)}$$

Then we add this noise to the original audio data.

$$\boldsymbol{y(t) = x(t) + noise(t)}$$

2. Feature Extraction

Feature extraction is a crucial step in audio signal processing and analysis, as it involves transforming raw audio data into a set of relevant and informative features that can be used as input for machine learning algorithms.

These extracted features serve as a more compact representation of the audio data, making it easier for models to understand and make predictions.

In our project we are extracting our data using a variety of audio features-

A. MFCCs (Mel-Frequency Cepstral Coefficients)

MFCCs are commonly utilized audio features that mimic how the human auditory system interprets sound by capturing the spectral properties of an audio source. These coefficients show the audio signal's short-term power spectrum and provide details on how energy is distributed among various frequency bands.

We have extracted a total of 40 MFCC features.

To give more stress on the higher frequencies we apply a filter of FIR.

$$\boldsymbol{x'(t) = x(t) - \alpha \cdot x(t-1)}$$

Then we convert these signals to $\boldsymbol{n}$ frames then we apply a windowing function to even out the irregularities.

$$\boldsymbol{\hat{x}(n) = x'(n) \cdot win(n)}$$

We then apply Fast Fourier Transform to calculate the spectrum of magnitudes of the signals windowed in the previous step.

$$\bar{x}(j) = FFT\left(\hat{x}(n)\right)$$

Then we create a Mel Filter Bank to apply to the transformed signal as follows.

$$h(j) = \begin{cases} 0, & j < f(m-1) \\ \dfrac{j - f(m-1)}{f(m) - f(m-1)}, & f(m-1) \le j \le f(m) \\ \dfrac{f(m+1) - j}{f(m+1) - f(m)}, & f(m) \le j \le f(m+1) \\ 0, & j > f(m+1) \end{cases}$$

Then we apply this filter bank to the transformed signal using natural log.

$$\widetilde{x_k} = \ln\left(\sum_{j=1}^{j} |\bar{x}(j)|^2 \cdot h(j)\right)$$

Then we apply the Discrete Cosine Transform and select the first ***40*** features only.

$$y_n = \sum_{k=1}^{K} \widetilde{x_k} \cdot \cos\left(\frac{\pi n}{K} \cdot (k - 0.5)\right)$$

B. Chroma Feature

The distribution of pitch classes in the audio signal is described by the chroma feature. It is helpful for deciphering the harmonic structure and tonal content of music.

To smooth out the unwanted transition noise we apply windowing as the first step.

$$\hat{x}(t, m) = x(m \cdot H + t) \cdot win(t)$$

We then apply short time Fourier Transform to convert time domain signal to frequency domain signal.

$$x'(k, m) = \sum_{t=0}^{T-1} win(t) \cdot \hat{x}(t, m) \cdot e^{-j\omega kn}$$

Then we do Chroma feature calculation, where $P(i, k)$ is the filter bank of pitch class.

$$y(k) = \sum_{i=0}^{11} P(i, k) \cdot |x'(i, m)|$$

C. Spectral Centroid

The location of the spectrum's centre of mass is indicated by the spectral centroid. It offers details on the "brightness" of the sound and may be connected to the audio's timbral qualities.

To smooth out the unwanted transition noise we apply windowing as the first step.

$$\hat{x}(t, m) = x(m \cdot H + t) \cdot win(t)$$

We then apply short time Fourier Transform to convert time domain signal to frequency domain signal.

$$x'(k,m) = \sum_{t=0}^{T-1} win(t) \cdot \hat{x}(t,m) \cdot e^{-j\omega kn}$$

Then we calculate the magnitude of the transformed signal.

$$|x'(k,m)| = \sqrt{\left(Re\left(x'(k,m)\right)\right)^2 + \left(Im\left(x'(k,m)\right)\right)^2}$$

Then we calculate the spectral centroid for each frame that is the weighted mean of frequencies.

$$y(m) = \frac{\sum_f f \cdot |x'(k,m)|}{\sum_f |x'(k,m)|}$$

D. Spectral Contrast

The amplitude difference between the audio spectrum's peaks and valleys is measured by spectral contrast. It can be helpful in differentiating between various instrumentations and sound textures.

To smooth out the unwanted transition noise we apply windowing as the first step.

$$\hat{x}(t,m) = x(m \cdot H + t) \cdot win(t)$$

We then apply short time Fourier Transform to convert time domain signal to frequency domain signal.

$$x'(f) = \sum_{t=0}^{T-1} win(t) \cdot \hat{x}(t,m) \cdot e^{-j\omega kn}$$

Then we calculate the Spectral energy for each frequency f in the band of frequencies.

$$E_k = \sum_{f}^{frequecy_{band}} |x'(f)|^2$$

Then we calculate the contrast between hight and mean amplitudes of each energy or frequency band.

$$y(k) = max(E_k) - mean(E_k)$$

E. Zero Crossing Rate (ZCR)

The rate at which the audio waveform crosses zero, or changes sign, is known as the ZCR. It serves as an indicator of the audio's level of noise.

First, we calculate the calculate the signum function.

$$signum\left(x(n)\right) = \begin{cases} -1, & x(n) < 0 \\ 0, & x(n) = 0 \\ 1, & x(n) > 0 \end{cases}$$

Then we calculate the difference of consecutive signums.

$$\boldsymbol{d(n) = signum\big(x(n)\big) - signum\big(x(n-1)\big)}$$

Then we calculate the ZCR.

$$\boldsymbol{cross(n)} = \begin{cases} \mathbf{1}, & \boldsymbol{d(n) \neq 0} \\ \mathbf{0}, & \boldsymbol{d(n) = 0} \end{cases}$$

$$\boldsymbol{y(k) = \frac{1}{2N} \sum_{n=0}^{N-1} cross(n)}$$

F. Root Mean Square Energy (RMSE)

The energy contained in the audio signal is measured by RMSE. It can be applied to gauge volume.

First, we create frames for the audio signal, then we calculate the squared values of each frame.

$$\boldsymbol{\hat{x}(n) = \big(x(n)\big)^2}$$

Then the mean squared value of each frame is calculated.

$$\boldsymbol{\overline{x} = \frac{1}{N} \sum_{n=1}^{N} \hat{x}(n)}$$

Finally, we find the root mean squared value of each frame.

$$\boldsymbol{y = \sqrt{\overline{x}}}$$

G. Kurtosis, Bandwidth, Skewness, and Spectral Roll-Off

These are extra spectral characteristics that reveal details regarding bandwidth, higher-order statistical moments, and spectral energy distribution.

Each features give a vector or matrix having same number of frames so we can stack them along the columns of a single feature vector creating a concatenation of the retrieved features. This produces a matrix in which the different extracted features are represented by columns, and time frames or portions of the audio stream are represented by rows. Then, machine learning models can use this feature matrix as input data to carry out tasks like sentiment analysis, audio categorization, and other audio-related analyses.

3. Padding

Since each data point of the extracted features dataset have temporal frames of derived characteristics from audio data, to guarantee that every sequence or time frame has the same length we are using the technique of padding to the dataset.

Padding becomes important when working with audio features since we want to build fixed-length input sequences for our model. So, we have added the necessary zero vectors to correspond with the constant number of rows (time frames) that we have chosen.

Therefore, after extracting all the features and applying the necessary processing techniques we are getting a time series dataset which we can train our model on for better performance.

MODEL

As we are dealing with a time series dataset having a 2D shape for each data point, we cannot use Conv2D layers used in models such Le-Net-5, VGG-16, AlexNet. Therefore, we have used a modified version of these designs with layers for Conv1D and MaxPooling1D to address this problem.

1. *Proposed Model*

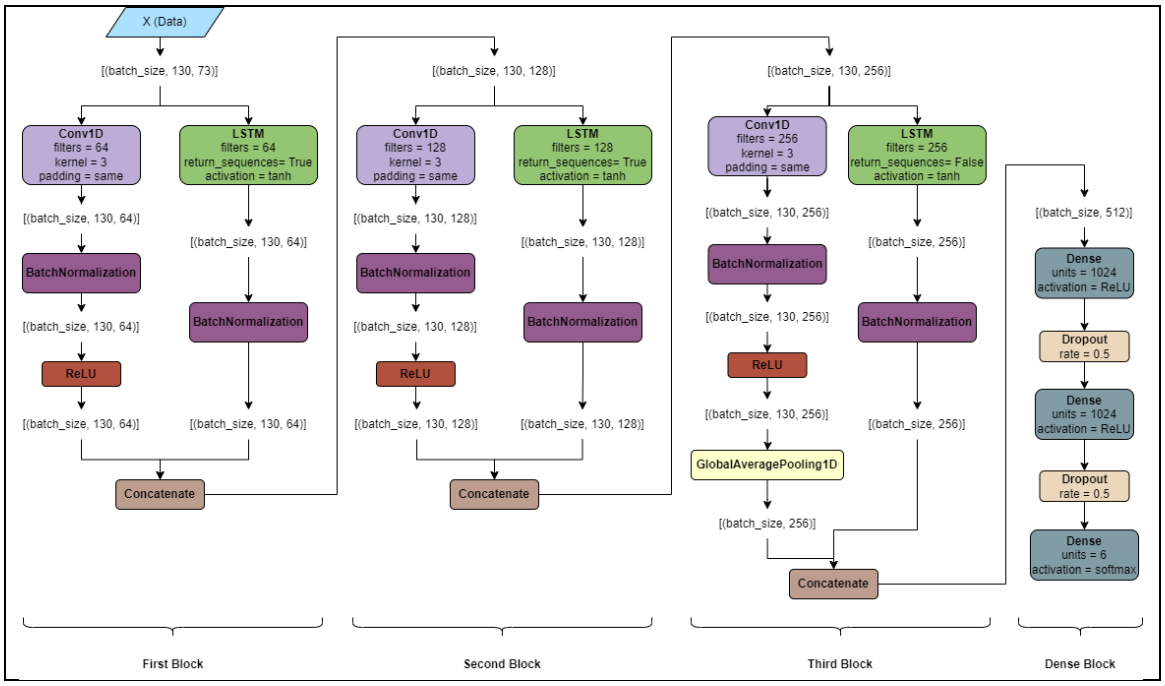


Fig 1: Model Architecture

The total distribution of parameters in the proposed model is given in the following table, having layers divided into blocks.

Block	*Layer*	*Params*
First Block	Conv1D	14080
	BatchNormalization	256
	ReLU	0
	LSTM	35328
	BatchNormalization	256
	Concatenation	0
Second Block	Conv1D	49280
	BatchNormalization	512
	ReLU	0
	LSTM	131584
	BatchNormalization	512
	Concatenation	0

Third Block	Conv1D	196864
	BatchNormalization	1024
	ReLU	0
	GlobalAveragePooling1D	0
	LSTM	525312
	BatchNormalization	1024
	Concatenation	0
Dense Block	Dense	525312
	Dropout	0
	Dense	1049600
	Dropout	0
	Dense	6150
	Total Params	2537094

Tab 1: Parameters in the layers of the model

2. *Analytic functions*

The main function we are going to use to describe our model are as follows:

A. Convolutional layer

$$\boldsymbol{Conv}(\boldsymbol{y}) = Conv1D(y, w) + b$$

$y \in \mathbb{R}^{(batch_{size}, time_{step}, input_{channel})}$
$w \in \mathbb{R}^{(kernel, input_{channel}, filters)}$
$b \in \mathbb{R}^{(filters)}$

B. Batch normalization

$$\boldsymbol{BatchNorm}(\boldsymbol{y}) = \gamma \cdot \frac{y - \mu}{\sqrt{\sigma^2 + \varepsilon}} + \beta$$

$$\mu = \frac{1}{batch_{size}} \cdot \sum_{i=1}^{batch_{size}} y^{(i)}$$

$$\sigma^2 = \frac{1}{batch_{size}} \cdot \sum_{i=1}^{batch_{size}} \left(y^{(i)} - \mu\right)^2$$

where:
$y^{(i)} = datapoint\ at\ position\ i\ in\ batch$
$\sigma^2 = variance\ of\ a\ batch$
$\mu = mean\ of\ a\ batch$
$\gamma = learnable\ scale\ parameter$
$\beta = learnable\ shift\ parameter$
$\varepsilon = small\ constant$

C. ReLU activation function

$$\boldsymbol{ReLU}(\boldsymbol{y}) = max(0, y)$$

D. LSTM

a. forget gate

$$f_g = \sigma\left(w_f \cdot [a^{<t-1>}, x^{<t>}] + b_f\right)$$
$\sigma = sigmoid\ activation\ funcion$

b. update gate

$$u_g = \sigma(w_u \cdot [a^{<t-1>}, x^{<t>}] + b_u)$$
$$\sigma = sigmoid\ activation\ funcion$$

c. output gate

$$o_g = \sigma(w_o \cdot [a^{<t-1>}, x^{<t>}] + b_o)$$
$$\sigma = sigmoid\ activation\ funcion$$

d. cell state

$$\tilde{c}_g{}^{<t>} = \tanh(w_c \cdot [a^{<t-1>}, x^{<t>}] + b_c)$$
$$c_g^{<t>} = u_g * \tilde{c}_g{}^{<t>} + f_g * c_g^{<t-1>}$$

e. hidden state

$$a^{<t>} = o_g * \tanh c_g^{<t>}$$

The output of LSTM is different for 2 cases:

CASE I - return_sequence = True

This parameter returns $\boldsymbol{a}$ as a sequence of many-to-many model having the same dimension as the input sequence.

$$\boldsymbol{LSTM(y) = a}$$
$$y \in \mathbb{R}^{(batch_{size}, time_{step}, input_{channel})}$$
$$a \in \mathbb{R}^{(batch_{size}, time_{step} filters)}$$

CASE II - return_sequence = False

This parameter returns $a^{<,time_{step}-1>}$ as output of many-to-one model of only the last state of *hidden state*.

$$\boldsymbol{LSTM(y) = a^{<time_{stamp}-1>}}$$
$$y \in \mathbb{R}^{(batch_{size}, time_{step}, input_{channel})}$$
$$a^{<time_{step}-1>} \in \mathbb{R}^{(batch_{size}, filters)}$$

E. Global average pooling 1D

$$\boldsymbol{GlobalAveragePooling1D(y)} = \frac{1}{time_{step}} \cdot \sum_{i=1}^{time_{step}} y^{<i>}$$
$$y^{<i>} = feature\ vector\ at\ time\ step\ i$$

F. Concatenate

$$\boldsymbol{Concatenate([y_1, y_2]) = y_1 \oplus y_2}$$
$$y_1 \in \mathbb{R}^{(batch_{size}, time_{step}, input_{channel_1})}$$
$$y_2 \in \mathbb{R}^{(batch_{size}, time_{step}, input_{channel_2})}$$
$$Concatenate([y_1, y_2]) \in \mathbb{R}^{(batch_{size}, time_{step}, input_{channel_2} + input_{channel_2})}$$

3. Analytic model

A. Input

$$\boldsymbol{X = represents\ the\ extracted\ feature\ dataset}$$
$$X \in \mathbb{R}^{(batch_{size}, time_{step}, feature_{size})}$$

B. *First block*

a. *Conv1D layer (Convolutional)*

$$\boldsymbol{Z_{conv_1} = Conv(X)}$$
$$\boldsymbol{Z_{conv_1} = BatchNorm(Z_{conv_1})}$$
$$\boldsymbol{Z_{conv_1} = ReLU(Z_{conv_1})}$$
$$Z_{conv_1} \in \mathbb{R}^{(batch_{size}, time_{step}, 64)}$$

b. *LSTM layer (Long short-term memory)*

$$with\ return\_sequence = True$$
$$\boldsymbol{Z_{lstm_1} = LSTM(X)}$$
$$Z_{lstm_1} \in \mathbb{R}^{(batch_{size}, time_{step}, 64)}$$

c. *Concatenate layer*

$$\boldsymbol{a_{concat_1} = Concatenate([Z_{conv_1}, Z_{lstm_1}])}$$
$$a_{concat_1} \in \mathbb{R}^{(batch_{size}, time_{step}, 128)}$$

C. *Second block*

a. *Conv1D layer (Convolutional)*

$$\boldsymbol{Z_{conv_2} = Conv(a_{concat_1})}$$
$$\boldsymbol{Z_{conv_2} = BatchNorm(Z_{conv_2})}$$
$$\boldsymbol{Z_{conv_2} = ReLU(Z_{conv_2})}$$
$$Z_{conv_2} \in \mathbb{R}^{(batch_{size}, time_{step}, 128)}$$

b. *LSTM layer (Long short-term memory)*

$$with\ return\_sequence = True$$
$$\boldsymbol{Z_{lstm_2} = LSTM(a_{concat_1})}$$
$$Z_{lstm_2} \in \mathbb{R}^{(batch_{size}, time_{step}, 128)}$$

c. *Concatenate layer*

$$\boldsymbol{a_{concat_2} = Concatenate([Z_{conv_2}, Z_{lstm_2}])}$$
$$a_{concat_2} \in \mathbb{R}^{(batch_{size}, time_{step}, 256)}$$

D. *Third block*

a. *Conv1D layer (Convolutional)*

$$\boldsymbol{Z_{conv_3} = Conv(a_{concat_2})}$$
$$\boldsymbol{Z_{conv_3} = BatchNorm(Z_{conv_3})}$$
$$\boldsymbol{Z_{conv_3} = ReLU(Z_{conv_3})}$$
$$\boldsymbol{Z_{conv_3} = GlobalAveragePooling1D(Z_{conv_3})}$$
$$Z_{conv_3} \in \mathbb{R}^{(batch_{size}, 256)}$$

b. *LSTM layer (Long short-term memory)*

$$with\ return\_sequence = False$$
$$\boldsymbol{Z_{lstm_3} = LSTM(a_{concat_2})}$$
$$Z_{lstm_3} \in \mathbb{R}^{(batch_{size}, 256)}$$

c. *Concatenate layer*

$$\boldsymbol{a_{concat_2} = Concatenate([Z_{conv_3}, Z_{lstm_3}])}$$

$$a_{concat_2} \in \mathbb{R}^{(batch_{size},512)}$$

E. Dense block

$$\boldsymbol{a_{dense_1}} = \boldsymbol{ReLU}\left(\boldsymbol{a_{concat_2}} \cdot \boldsymbol{w_1} + \boldsymbol{b_1}\right)$$

where:

$w_1 \in \mathbb{R}^{(512,1024)}$

$b_1 \in \mathbb{R}^{(1024)}$

$a_{dense_1} \in \mathbb{R}^{(batch_{size},1024)}$

$$\boldsymbol{a_{drop_1}} = \boldsymbol{Dropout}\left(a_{dense_1}, rate = 0.5\right)$$

$a_{drop_1} \in \mathbb{R}^{(batch_{size},1024)}$

$$\boldsymbol{a_{dense_2}} = \boldsymbol{ReLU}\left(\boldsymbol{a_{drop_1}} \cdot \boldsymbol{w_2} + \boldsymbol{b_2}\right)$$

where:

$w_2 \in \mathbb{R}^{(1024,1024)}$

$b_2 \in \mathbb{R}^{(1024)}$

$a_{dense_2} \in \mathbb{R}^{(batch_{size},1024)}$

$$\boldsymbol{a_{drop_2}} = \boldsymbol{Dropout}\left(a_{dense_2}, rate = 0.5\right)$$

$a_{drop_2} \in \mathbb{R}^{(batch_{size},1024)}$

F. Output

$$\boldsymbol{y} = \boldsymbol{softmax}\left(\boldsymbol{a_{drop_1}} \cdot \boldsymbol{w_2} + \boldsymbol{b_2}\right)$$

where:

$w_2 \in \mathbb{R}^{(1024,n_{class})}$

$b_2 \in \mathbb{R}^{(n_{class})}$

$a_{dense_2} \in \mathbb{R}^{(batch_{size},n_{class})}$

$n_{class} = 6; number\ of\ emotions$

For our model we have used:

A. **Optimizer** = Adam
B. **Loss** = Categorical Cross entropy

$$Loss = -\sum_{i=1}^{n_{class}} y_i \cdot \log \hat{y}_i$$

RESULT

A distinct set of difficulties arises when analysing sentiment in audio data, particularly when trying to do so without the aid of conventional methods like audio-to-speech conversion. But our strategy of using solely acoustic characteristics improves the analysis quality and expands the range of information that can be extracted from audio data.

Choosing the right number of features to extract and then organizing the resulting dataset to train our model was one of the main problems we ran into with this project. The process of selecting features required considerable thought because it affected the model's capacity to represent the crucial aspects of audio that elicit emotion.

Pitch, tempo, and spectral characteristics are important parts of audio signals that are captured by acoustic features and are crucial for identifying emotional content. We were able to offer a more comprehensive and sophisticated analysis of auditory feelings by depending on these aspects.

1. Evaluation

We are using modified Le-Net-5, VGG-16, AlexNet and Bidirectional-LSTM Network to serve as a comparison base for our proposed architecture.

We have used the following metrics for the evaluation of our model and create a comparative analysis with the other models.

A. Categorical accuracy

$$Categorical\ Accuracy = \frac{1}{K} \times \sum_{k=1}^{K} \frac{n(TP^{[k]})}{n(Total^{[k]})}$$

B. Precision

$$Precision = \frac{1}{K} \times \sum_{k=1}^{K} \frac{n(TP^{[k]})}{n(TP^{[k]} + FP^{[k]})}$$

C. Recall

$$Recall = \frac{1}{K} \times \sum_{k=1}^{K} \frac{n(TP^{[k]})}{n(TP^{[k]} + FN^{[k]})}$$

D. F1 score

$$F1^{<k>} = 2 \times \frac{Precision^{[k]} \times Recall^{[k]}}{Precision^{[k]} + Recall^{[k]}}$$

Where:

$\boldsymbol{K}$: $Total\ number\ of\ classes\ (emotions)$
$\boldsymbol{n}(\boldsymbol{TP}^{[k]})$: $number\ of\ \boldsymbol{True\ Possitives}\ of\ class\ k$
$\boldsymbol{n}(\boldsymbol{FP}^{[k]})$: $number\ of\ \boldsymbol{False\ Possitives}\ of\ class\ k$
$\boldsymbol{n}(\boldsymbol{FN}^{[k]})$: $number\ of\ \boldsymbol{False\ Negative}\ of\ class\ k$
$\boldsymbol{F1}^{<k>}$: $F1\ Score\ of\ class\ k$
$k \in K\ ; 1 \leq k \leq K$

Table for Comparative analysis:

Model	*Metric*	*fear*	*anger*	*happy*	*disgust*	*sad*	*neutral*
LeNet-5	*Categorical accuracy*	0.514					
	Precision	0.46	0.67	0.40	0.49	0.58	0.55
	Recall	0.36	0.56	0.60	0.51	0.55	0.51
	F1 Score	0.40	0.61	0.48	0.50	0.56	0.53
VGG-16	*Categorical accuracy*	0.558					
	Precision	0.53	0.84	0.47	0.65	0.56	0.48
	Recall	0.47	0.51	0.60	0.43	0.60	0.78
	F1 Score	0.50	0.64	0.53	0.52	0.58	0.60
AlexNet	*Categorical accuracy*	0.562					
	Precision	0.56	0.61	0.49	0.64	0.58	0.53
	Recall	0.41	0.70	0.59	0.47	0.59	0.62
	F1 Score	0.47	0.65	0.53	0.54	0.58	0.57
LSTM	*Categorical accuracy*	0.608					
	Precision	0.71	0.79	0.53	0.56	0.59	0.56
	Recall	0.40	0.67	0.59	0.62	0.66	0.74
	F1 Score	0.51	0.72	0.56	0.59	0.62	0.64
Proposed Model	*Categorical accuracy*	0.623					
	Precision	0.74	0.77	0.55	0.75	0.55	0.54
	Recall	0.41	0.63	0.70	0.51	0.71	0.82
	F1 Score	0.52	0.70	0.61	0.61	0.62	0.66

Tab 2: Comparative Analysis

2. Confusion Matrix

In machine learning and statistics, a confusion matrix is a vital tool for assessing how well classification models perform. It gives a straightforward summary of how well a model predicts the ground truth or target labels.

Essentially, a confusion matrix aids in calculating the proportion of accurate and inaccurate predictions a model makes.

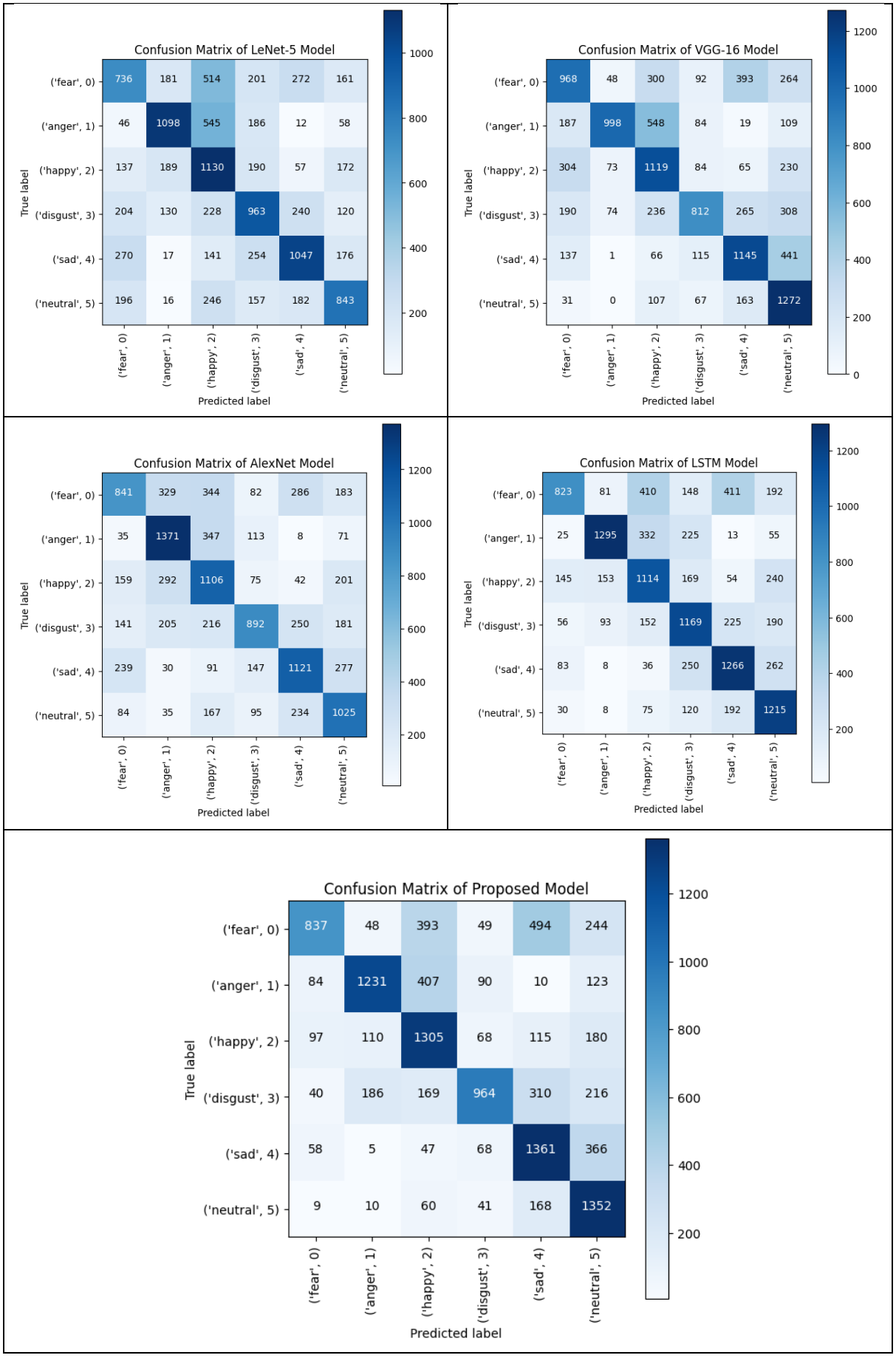


Fig 2: Confusion Matrix

3. *AUC-ROC*

ROC curve is the graphical visualization of a model performance as per each class in the model classification. And the AUC score summarizes this curve using a numerical value.

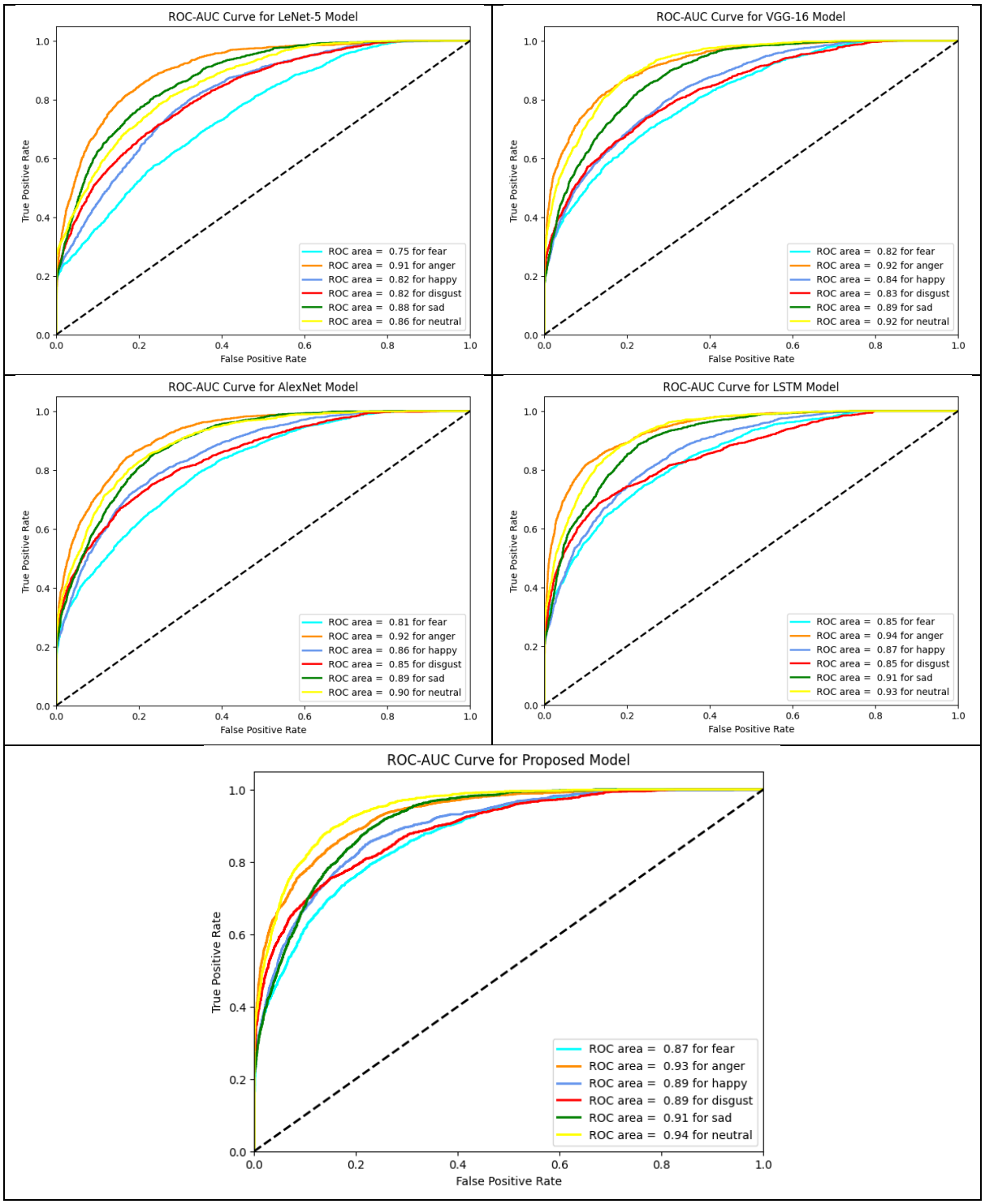


Fig 3: AUC-ROC Curves

We have created a state-of-the-art model and cleared the path for a more thorough and precise interpretation of the emotions portrayed in audio data by tackling the difficulties of feature selection and comprehending the complexities of data flow. Sentiment analysis's exclusive reliance on auditory qualities opens new avenues for application and provides previously hard-to-get insights into a variety of domains.

CONCLUSION

To sum up, our effort has pushed the limits of what is feasible in this field while also producing a superior model for audio sentiment analysis. Our study represents a substantial improvement in this sector by exceeding well-established models.

The uses and applications for our model are numerous. And ranging from bettering customer experiences, automating helpdesks, learning more about the attitudes of the market to boosting healthcare results such as mental health care, fast diagnosis, etc.

The potential for expanding the capabilities of audio sentiment analysis utilizing acoustic characteristics through further optimizations and enhancements is quite promising for the future. Our effort is proof of the creative potential and the boundless potential of artificial intelligence. We anticipate a future in which the emotions portrayed in audio data may be comprehended more thoroughly and precisely than ever before as we continue to push the boundaries of audio sentiment analysis.

REFRENCES

[1] Sarah A. Abdu, Ahmed H. Yousef, Ashraf Salem,
Multimodal Video Sentiment Analysis Using Deep Learning Approaches, a Survey,
Information Fusion,
Volume 76,
2021,
Pages 204-226,
ISSN 1566-2535,
https://doi.org/10.1016/j.inffus.2021.06.003.

[2] Luo, Z., Xu, H., & Chen, F. (2018). Audio Sentiment Analysis by Heterogeneous Signal Features Learned from Utterance-Based Parallel Neural Network. AffCon@AAAI.

[3] Tembhurne, J.V., Diwan, T. Sentiment analysis in textual, visual and multimodal inputs using recurrent neural networks. *Multimed Tools Appl* **80**, 6871–6910 (2021). https://doi.org/10.1007/s11042-020-10037-x

[4] U. Sehar, S. Kanwal, K. Dashtipur, U. Mir, U. Abbasi and F. Khan, "Urdu Sentiment Analysis via Multimodal Data Mining Based on Deep Learning Algorithms," in IEEE Access, vol. 9, pp. 153072-153082, 2021, doi: 10.1109/ACCESS.2021.3122025.

[5] Cao H, Cooper DG, Keutmann MK, Gur RC, Nenkova A, Verma R. CREMA-D: Crowd-sourced Emotional Multimodal Actors Dataset. IEEE Trans Affect Comput. 2014 Oct-Dec;5(4):377-390. doi: 10.1109/TAFFC.2014.2336244. PMID: 25653738; PMCID: PMC4313618.

[6] Livingstone SR, Russo FA (2018) The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English. PLoS ONE 13(5): e0196391. https://doi.org/10.1371/journal.pone.0196391.

[7] S. Haq and P.J.B. Jackson, "Multimodal Emotion Recognition", In W. Wang (ed), Machine Audition: Principles, Algorithms and Systems, IGI Global Press, ISBN 978-1615209194, chapter 17, pp. 398-423, 2010.

[8] S. Haq and P.J.B. Jackson. "Speaker-Dependent Audio-Visual Emotion Recognition", In Proc. Int'l Conf. on Auditory-Visual Speech Processing, pages 53-58, 2009.

[9] S. Haq, P.J.B. Jackson, and J.D. Edge. Audio-Visual Feature Selection and Reduction for Emotion Classification. In Proc. Int'l Conf. on Auditory-Visual Speech Processing, pages 185-190, 2008.

[10] Pichora-Fuller, M. Kathleen; Dupuis, Kate, 2020, "Toronto emotional speech set (TESS)", https://doi.org/10.5683/SP2/E8H2MF, Borealis, V1

[11] Ellis, Daniel P.W. "Chroma feature analysis and synthesis"
2007/04/21
https://www.ee.columbia.edu/~dpwe/resources/matlab/chroma-ansyn/

[12] Jiang, Dan-Ning, Lie Lu, Hong-Jiang Zhang, Jian-Hua Tao, and Lian-Hong Cai. “Music type classification by spectral contrast feature.” In Multimedia and Expo, 2002. ICME’02. Proceedings. 2002 IEEE International Conference on, vol. 1, pp. 113-116. IEEE, 2002.

[13] Klapuri, A., & Davy, M. (Eds.). (2007). Signal processing methods for music transcription, chapter 5. Springer Science & Business Media.