SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V12I1P117

Volume 12 Issue 1, 216-235, January 2025
© 2025 Seventh Sense Research Group®

Original Article

Speech Emotion Recognition Using Hybrid Deep Learning and Ensemble Approaches

I. Manolekshmi[1], M.A. Mukunthan[2]

[1,2]*Department of Computer Science and Engineering,*
Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Tamil Nadu, India.

[1]*Corresponding Author : manolekshmi12@gmail.com*

Received: 21 November 2024 Revised: 27 December 2024 Accepted: 13 January 2025 Published: 30 January 2025

Abstract - *The technique of recognizing and classifying emotions expressed in language spoken using audio features is Speech Emotion Recognition (SER). Human-computer interaction must enable machines to accurately perceive and respond to human emotions. Numerous challenges, like capturing both spatial and temporal features in speech signals, impact the accuracy of emotion recognition models. Conventional emotion recognition systems heavily depend on manual feature extraction and classification, which require significant effort and often lead to errors in detection. Advances in image processing and Artificial Intelligence (AI) have introduced hybrid Deep Learning (DL) approaches to improve SER tasks. This study developed an efficient Speech Emotion Recognition (SER) system utilizing a hybrid DL model combined with an ensemble approach to accurately classify emotions expressed through speech. The models were evaluated on the CREMA dataset which contains 7,442 audio samples across six different emotions. After preprocessing and data augmentation, Mel Frequency Cepstral Coefficients (MFCC) were captured as features from speech data. The proposed models include CNN-LSTM and CNN-GRU to extract both spatial and temporal features. Outputs from these frameworks were combined using an ensemble learning approach with a Support Vector Machine (SVM) classifier as the meta-learner. Experimental results specify that the suggested model attained improved performance with an accuracy of 98.69%, precision of 98.70%, recall of 98.72% and an F1 score of 98.70%. The results highlight the effectiveness of combining advanced neural networks for achieving high performance in emotion detection from speech signals, providing valuable information for developing real-time emotion recognition systems and enhancing human-computer interaction.*

Keywords - *Speech emotion recognition, Support vector machine, MFCC, Convolutional neural network, CREMA dataset, Ensemble learning.*

1. Introduction

The most essential forms of human communication are speech signals. Many studies are continuously working in the field of human-machine interaction. Machines are able to understand human language and identify it in a meaningful way. Even though there have been significant advancements in speech recognition, it requires a lot of effort to make a natural interaction between humans and machines. One of the important factors is that it is difficult for machines to recognize the emotional states hidden in words spoken. Here, SER refers to identifying a speaker's emotional states through speech analysis [1].

SER can be utilized to enhance speech recognition system performance and extract relevant meanings from speech. SER focuses on identifying the emotions present in the voice signals in any case of their technical information. It can be used in many applications, such as in-car systems to monitor drivers' mental condition and initiate safety procedures, as a diagnostic tool for therapists, in cockpits of airlines, etc. The prime goal of a SER system is to identify different traits in speakers under various emotional circumstances. Typically, SER extracts information from voice signals, followed by a classification process to predict the emotions.

The researchers face several challenges, such as selecting proper speech features, assuring robustness to tone and style of speaking, and accounting for emotional expression across diverse cultures and situations. Extracting strong, effective, and discriminative features constitutes a primary research challenge. The advancement of effective SER models improves user experience in systems related to human-machine interactions, particularly in the domains of Artificial Intelligence (AI) and mobile health [2].

The capability to predict emotions from audio samples and mimic these emotions has a significant effect in the field of AI. DL models are currently used to address these

This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

recognition problems. Various approaches utilizing Deep Neural Networks (DNN) have been developed for SER, with some of the models focusing on determining important features directly from raw audio samples to enhance accuracy and efficiency [3].

The proposed approach employed a hybrid DL model using an ensemble method to recognize speech emotions. The model combines the strength of CNN-LSTM and CNN-GRU architectures to effectively collect both spatial and temporal features for speech signals, improving the accuracy and integrity of emotion recognition. By employing the advantages of these frameworks, the ensemble approach ensures improved generalization and robustness.

The primary contributions of the proposed research are as follows:

- To create a hybrid DL framework that combines the CNN-LSTM and CNN-GRU models to efficiently extract temporal and spatial features from the speech data.
- To propose an ensemble learning approach that combines the outputs of CNN-LSTM and CNN-GRU models, improving the accuracy and robustness of emotion classification from speech data.
- To examine the performance of the suggested hybrid ensemble framework employing performance metrics.
- To compare the suggested ensemble hybrid model with previous models, indicating the benefits of the proposed method for better emotion recognition accuracy.

The remaining portions of the paper are organized as outlined below: Section 2 presents a literature review highlighting existing works and identifying research gaps. Section 3 elaborates on the proposed model. Section 4 presents the findings of the study, while Section 5 provides the conclusion of the paper.

2. Related Works

Using a lightweight 1-D deep CNN, Bhangale and Kothandaraman (2023) [4] suggested an acoustic feature set for improving feature distinctiveness in speech emotion signals. The system's efficiency was examined using EMODB and RAVDEES datasets, achieving 94.18% accuracy on the RAVDEES dataset, outperforming conventional SER methods.

In order to capture global contextualized long-term dependencies in speech data, Kakuba et al. (2022) [5] developed an Attention-Based Multi learning Model (ABMD) that used Residual Dilated Casual Convolution (RDCC) blocks and dilated convolutional layers with multi-head attention. The framework performed well with fewer parameters than deeper models, confirmed its efficiency in SER and attained 95.3% accuracy on the EMODB dataset.

Aftab et al. (2022) [6] suggested a lightweight, Fully Convolutional Network (FCNN) for SER, which is designed for systems with limited hardware resources. The model used three parallel paths with different filter sizes to extract features, allowing deep convolutional blocks to capture high-level features. The model's integrity was verified by classifying emotions over the IEMOCAP and EMO-DB datasets, outperforming the modern SER systems.

Aggarwal et al. (2022) [7] studied two feature extraction methods to improve SER. They used Principal Component Analysis (PCA) and implemented a DNN with dense and dropout layers as an initial approach. On the second approach, images of a Mel spectrogram were utilized as an input to the pretrained VGG-16 framework, thereby achieving better accuracy on the RAVDEES dataset than using numeric features on DNN.

Li et al. (2022) [8] suggested a dense-DCNN framework to address the challenges that occurred due to limited speech datasets and lengthy training times in conventional and modern SER. The model was combined with StarGAN to extract features for better classification and generate numerous log-mel spectra with emotional labels. Various datasets (Emo-DB, SAVEE, RAVDESS, and CASIA) are used in this study, thereby attaining a classification accuracy of 97.36% on the RAVDESS dataset, showing strong generalization and robustness in multi-noise and multi-scene environments.

Mustaqeem and Kwon (2021) [9] created an end-to-end real-time SER framework based on a 1D Dilated Convolutional Neural Network (DCNN). A multi-learning strategy was employed by them to extract long-term contextual dependencies and spatial emotional features utilizing a fusion layer to combine these features for the recognition of emotion. The framework demonstrated its effectiveness in processing real speech signals by achieving 90% accuracy on EMO-DB datasets and 73% accuracy on IEMOCAP datasets.

For discrete SER, Zhao et al. (2021) [10] suggested an effective Deep Neural Network (DNN) design combining Connectionist Temporal Classification (CTC) loss. The model utilized a Self-Attention Residual Dilated Network (SADRN) for classification and integrated Parallel Convolutional layers (PCN) with a Squeeze and Excitation network (SEnet) to acquire relationships from 3D spectrograms acquiring 73.1% accuracy on the IEMOCAP dataset, presenting its effectiveness for discrete SER tasks.

In their study of DCNN for feature extraction, Amjad et al. (2021) [11] studied the drawbacks of handcrafted characteristics for identifying emotion from audio signals and examined its benefits. To determine the most discriminative features, they employed various classifiers for categorising

seven emotions, resulting in an accuracy of 93.6% over the RAVDEES dataset, surpassing conventional handcrafted feature-based frameworks.

An attention-based 3D CNN LSTM model developed by Atila and Sengur (2021) [12] was employed to recognise speech emotions. It uses speech images from MFCC spectrograms, fractal dimensions and cochleagrams. The input speech signals were preprocessed, resampled and transformed into speech images, then given to the proposed 3D CNN-LSTM model. The model was performed across RAVDESS, SAVEE and RML datasets, which acquired an accuracy of 96.18% over RAVDESS datasets, demonstrating its efficiency across these datasets.

Tuncer et al. (2021) [13] created a non-linear multilevel feature generation model that is utilized to recognize speech emotions from a cryptographic structure. The framework employed the Tunable Q Wavelet Transform (TQWT) to generate features, a twine shuffle pattern for feature extraction and an iterative neighborhood component for feature selection. By employing a 10-fold cross-validation method, the framework obtained 90.09% accuracy on the EMO DB dataset.

Wang et al. (2020) [14] created a dual-level framework for emotion recognition using features of MFCC in addition to Mel spectrograms from raw audio signals. The model consisted of a conventional LSTM for MFCC processing with a new Dual Sequence LSTM (DSLSTM) for instantaneous mel-spectrogram processing, which resulted in a weighted accuracy of 72% on the IEMOCAP dataset, highlighting the potential of frameworks that depend only on audio signals.

Nediyanchath et al. (2020) [15] suggested a Multi-Head Attention DL network employing Log Mel-Filter Bank Energies (LFBE) as the input features for SER. For capturing gender-specific emotional features, the model combined gender identification as an auxiliary task in addition to multi-task learning and position embedding, resulting in an overall accuracy of 76.4%, enhancing the SER efficiency and improving human-like conversational approaches with improved emotion recognition skills.

In order to simplify the categorical recognition of four unique emotions, Yao et al. (2020) [16] developed a model that combined three different classifiers: DNN, Recurrent Neural Networks (RNN), and CNN. They used Mel spectrograms, low-level descriptors and high-level statistical functions to train the single models and employed an attention mechanism-based weighted-pooling technique that integrates the outputs of RNN and CNN, thereby acquiring 58.3% accuracy on the IEMOCAP dataset, presenting the effectiveness of combining different classifiers for emotion recognition.

Sajjad et al. (2020) [17] presented a new framework on SER that utilized an essential sequence segment selection method. The selected sequence was transformed into a spectrogram, and salient features were gathered utilizing a CNN model, which was then regularized and given to a deep bi-LSTM for temporal analysis. The system showed better efficiency in emotion recognition with 85.57% accuracy on the EMO DB dataset.

In order to capture features from modern emotional speech datasets, Farooq et al. (2020) [18] used a DCNN network for SER by using a pre-trained network. A correlation-based selection framework was utilized to identify the most distinctive features and employed distinct classifiers such as SVM, RF, k-NN and neural networks that obtained 95.10% accuracy on the EMO-DB datasets, demonstrating its performance for speaker-independent SER.

Although several DL models have shown tremendous progress in SER, several gaps exist in the previous studies. The majority of the studies focused on increasing the accuracy of models using a single database. It often lacked generalization across multiple datasets or real-world scenarios involving noise and diverse environments.

Many models achieved better accuracy on specific datasets but neglected to consider computational efficiency especially on systems with limited hardware resources. Also, there exist difficulties in effectively capturing long-term dependencies in speech signals and handling complex emotional states. There was also limited focus on the hybrid or ensemble approaches that combine the DL architectures to improve SER performance. Therefore, more research is required to develop robust, efficient and generalized models to fill these gaps in SER.

3. Materials and Methods

The diagrammatic illustration presented in Figure 1 demonstrates a structured procedure for recognising speech emotions utilizing an ensemble learning approach based on a hybrid DL model. Initially, a speech emotion dataset, “CREMA,” was preprocessed, and exploratory data analysis was performed to understand the features within the dataset. Data augmentation techniques enhance the dataset size and model generalization. After that, the feature extraction method MFCC was carried out to determine relevant features essential for emotion recognition. The split data were given to the two hybrid DL models, CNN-LSTM and CNN-GRU. These models were trained parallelly, and the outputs from these models were combined through a stacking mechanism, which improves the prediction accuracy. Finally, this stacked output was passed through an SVM classifier, and the model efficiency was examined based on the outputs from the classifier.

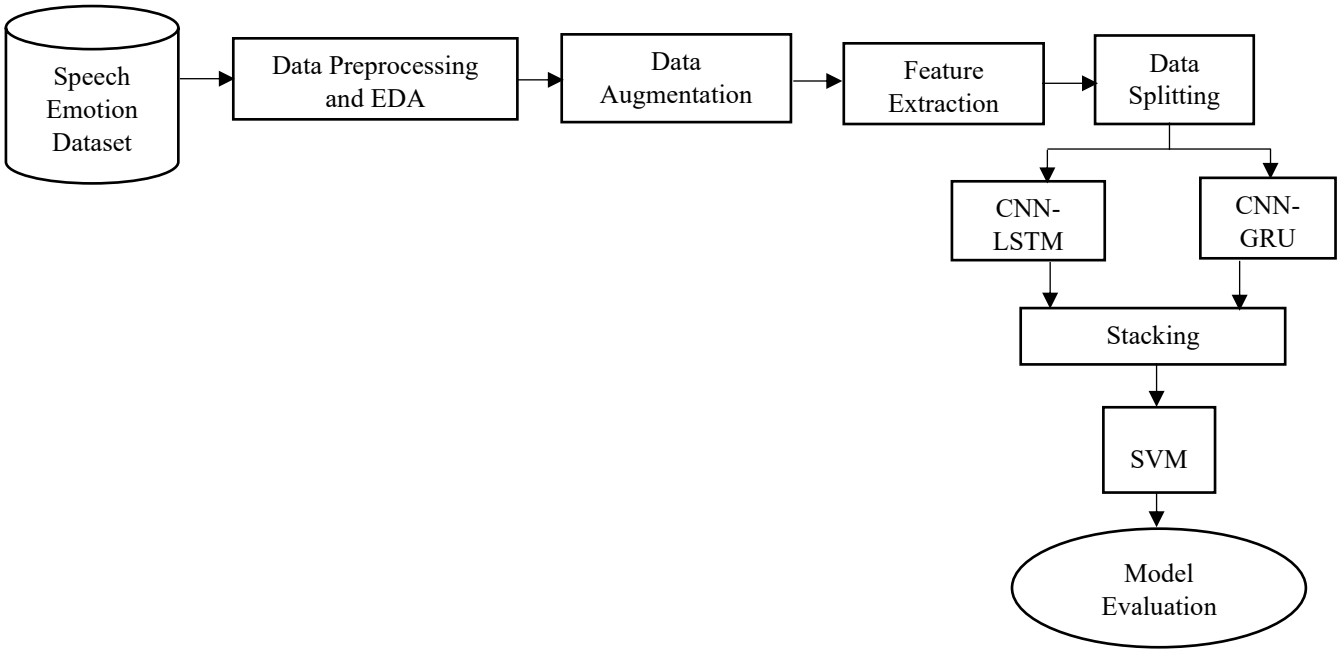


Fig. 1 Block diagram of SER

3.1. Dataset

The "CREMA" dataset, sourced from the Kaggle repository, was utilized for speech emotion recognition [19]. It consists of 7,442 audio samples collected from 91 actors, including 48 men and 43 women. The actors are between the ages of 20 and 74, which signifies a wide range of cultural backgrounds, consisting of American, Asian, Hispanic, African, Caucasian, and Unspecified.

Each actor portrays a set of 12 predefined sentences, delivering them with one among the six distinct emotions like: anger, disgust, happy, fear, sad and neutral. Also, these sentences reflect four different levels of emotional intensity, specified as low, medium, high and unspecified. Figure 2 represents the emotional Categories with File Paths Data Frame, and Figure 3 displays the histogram of speech emotion signals showing the count plot of six emotions.

	Emotion	**Path**
0	Angry	CREMA-D/AudioWAV//1001_DFA_ANG_XX.wav
1	disgust	CREMA-D/AudioWAV//1001_DFA_DIS_XX.wav
2	fear	CREMA-D/AudioWAV//1001_DFA_FEA_XX.wav
3	happy	CREMA-D/AudioWAV//1001_DFA_HAP_XX.wav
4	neutral	CREMA-D/AudioWAV//1001_DFA_NEU_XX.wav

Fig. 2 Emotional categories with file paths data frame

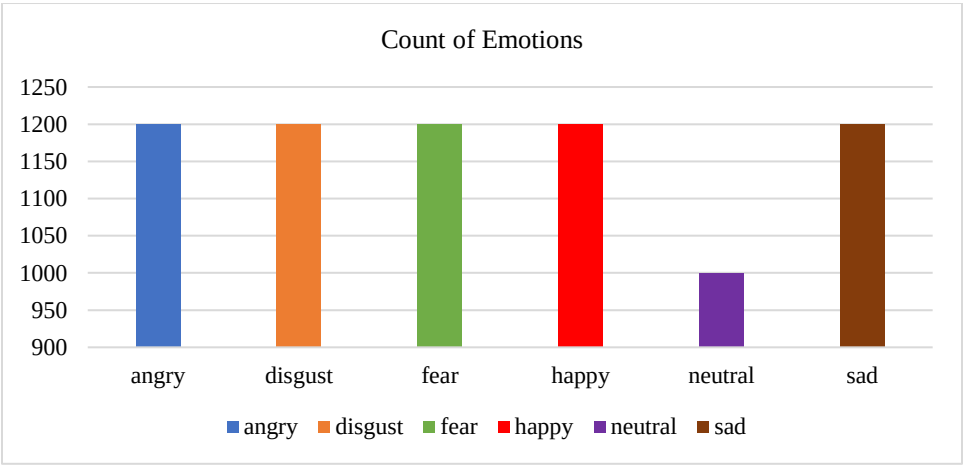
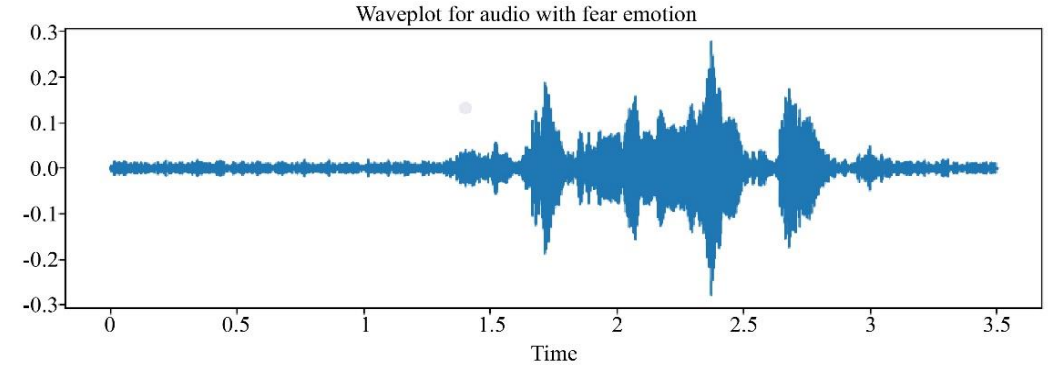


Fig. 3 Histogram of speech emotion labels

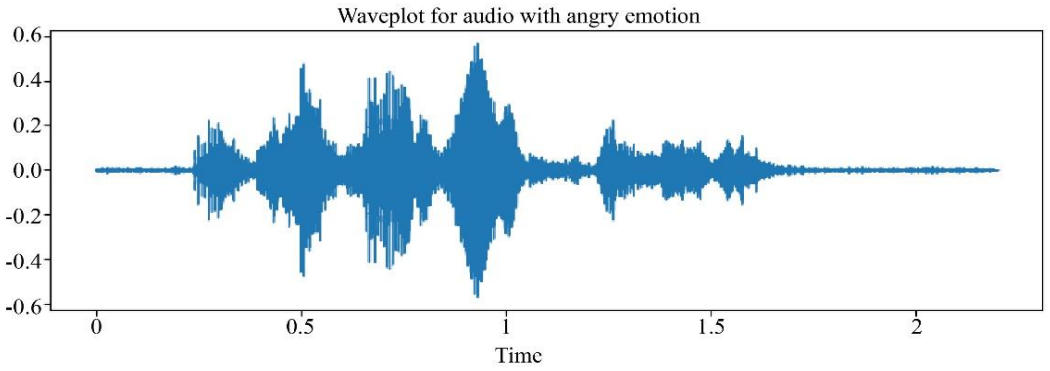
3.2. Data Preprocessing and Exploratory Data Analysis

In this study, preprocessing plays an essential role in SER to improve the quality of audio data. At first, an array of samples was obtained from the audio signal representing the raw sound signal. Then, the silence was trimmed from the beginning and end of the audio to ensure that only meaningful areas of the speech were retained. Lastly, padding is applied to ensure that every audio sample is of the same length, which is essential for consistency in model input and enabling batch processing during training. All these steps make the audio more standard for effective analysis.

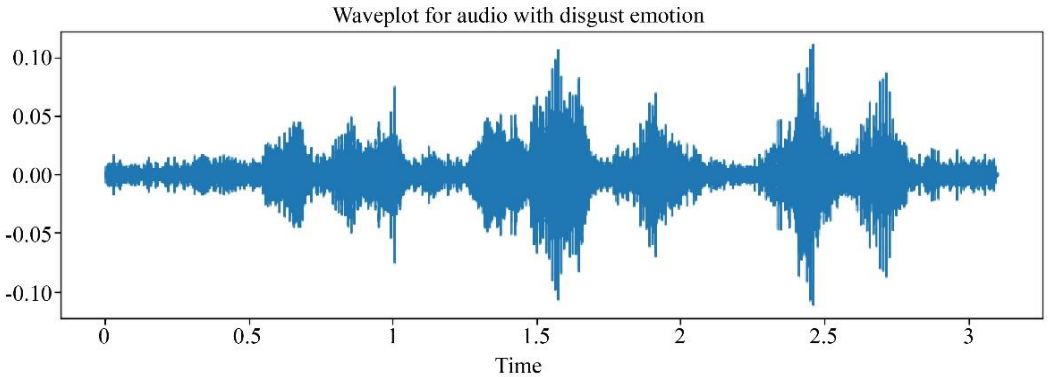
Wave plots present the amplitude and loudness of the audio signal, which helps understand the intensity and variations of sound. In SER, these plots provide information about the emotional intensity of speech. Emotions like anger or happiness result in higher amplitudes, while emotions like sadness or disgust result in lower amplitude waveforms. By visualizing the changes in loudness, wave plots provide a quick overview of how emotional expressions appear on the voice signals. The wave plots of six emotions are illustrated in Figure 4.



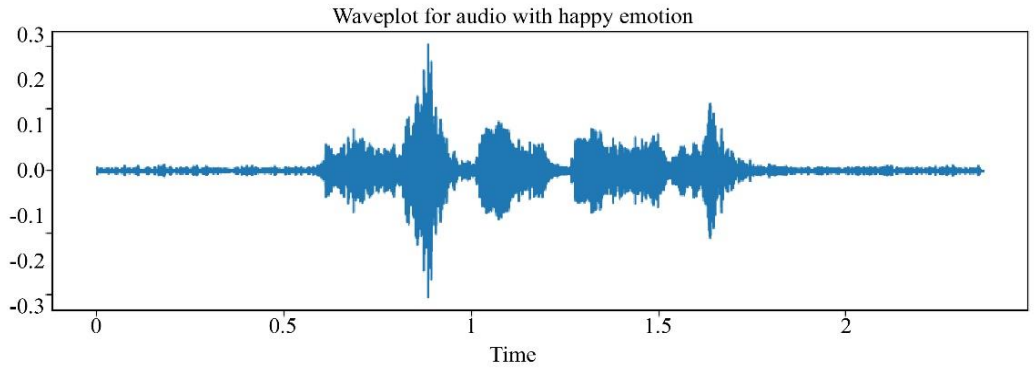
(a) Fear



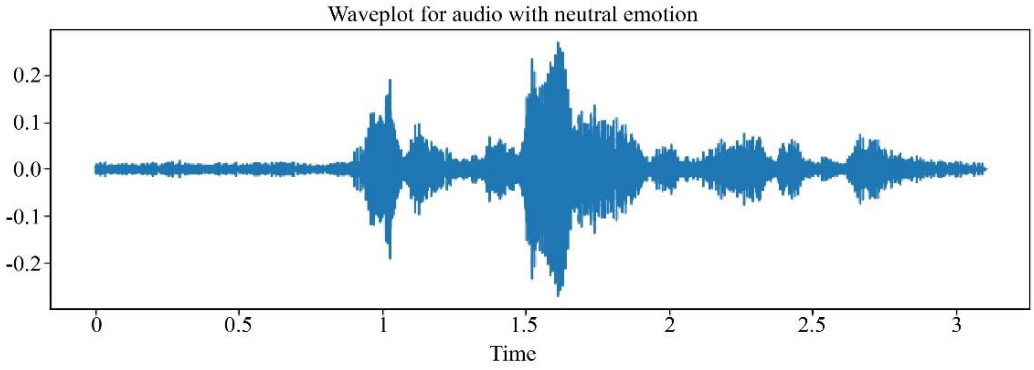
(b) Angry



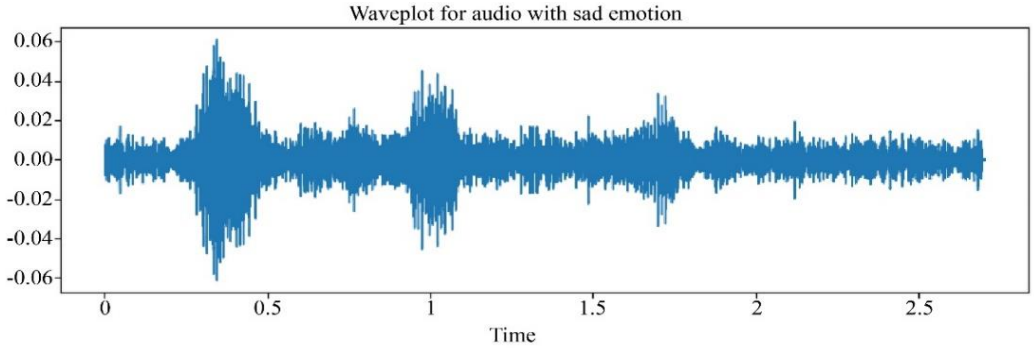
(c) Disgust



(d) Happy

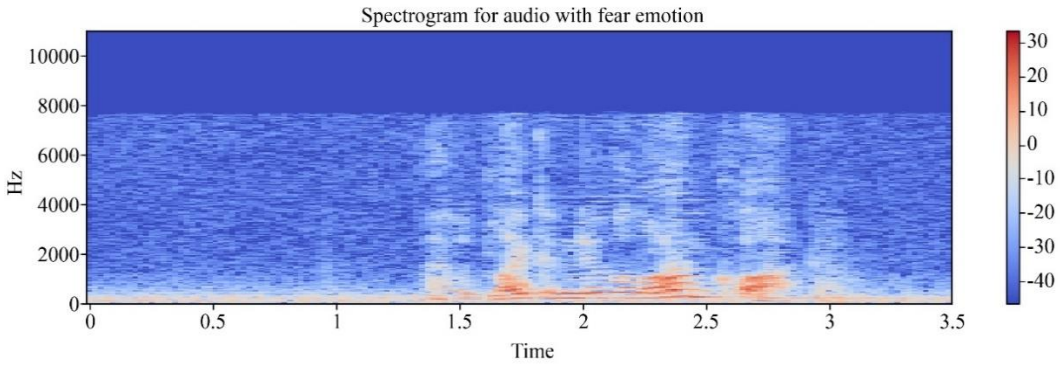


(e) Neutral

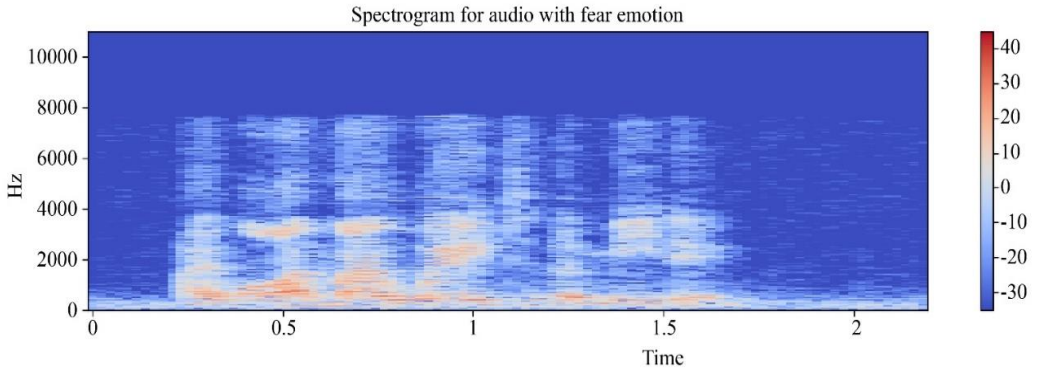


(f) Sad

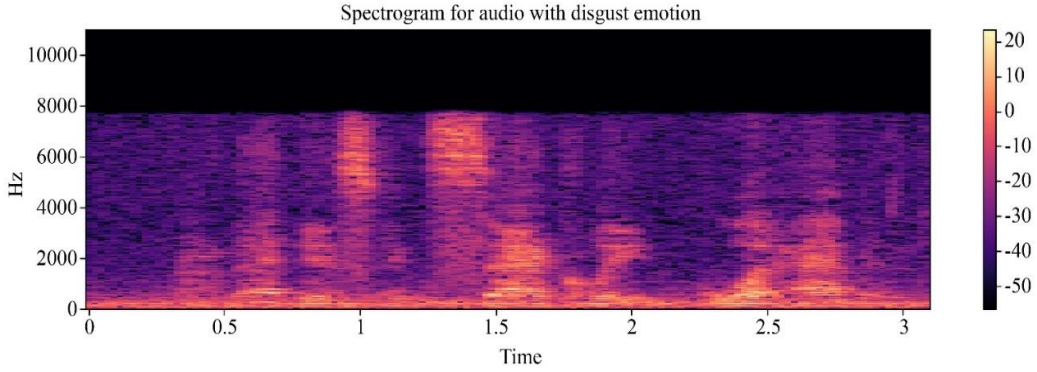
Fig. 4 Wave plot for emotions



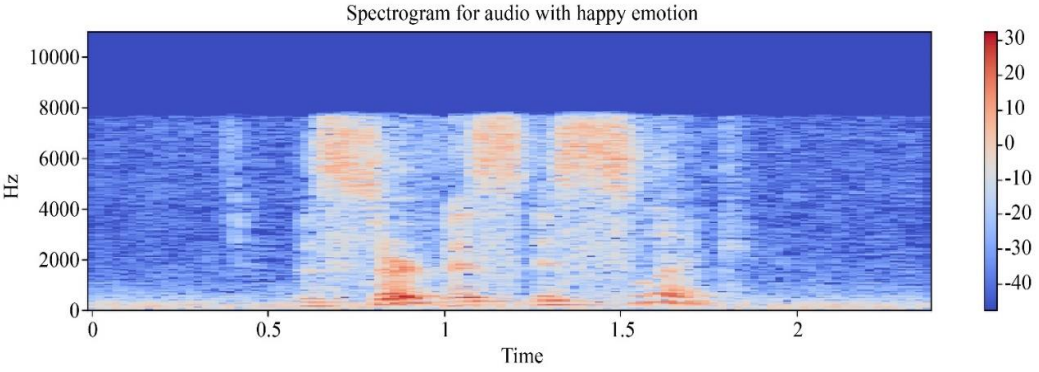
(a) Fear



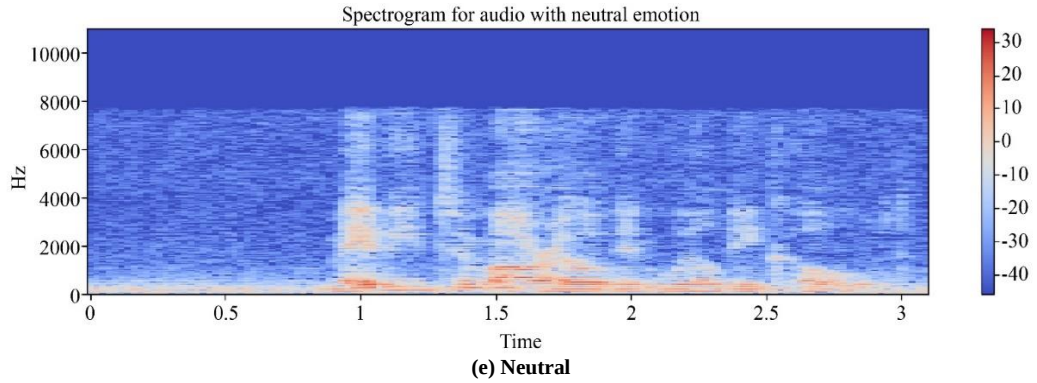
(b) Angry



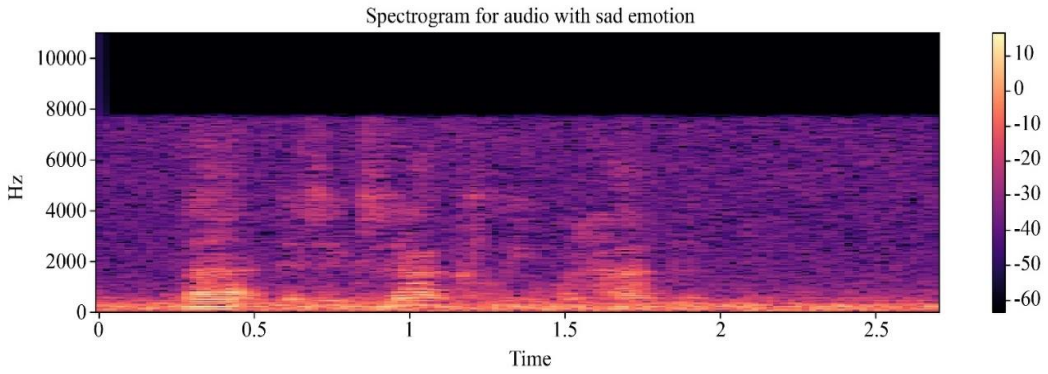
(c) Disgust



(d) Happy



(e) Neutral



(f) Sad

Fig. 5 Spectrogram of emotions

Spectrograms showed the detailed audio representation by displaying how the frequency spectrum changed over time. In the spectrogram, the x and y axes indicate the time and frequency, respectively, and the intensity of the color represents the energy at each frequency. It is mainly used to extract the tonal and pitch variations of the emotions. In cases of sadness or fear, the spectrogram shows lower frequencies, while for anger or happiness, the spectrogram shows higher frequencies. The spectrograms for six emotions are displayed in Figure 5.

3.3. Data Augmentation

The process of creating new synthetic instances by making minor modifications to the existing training dataset is known as data augmentation. The proposed research uses various methods, such as adding noise, time shifting, altering speed, and modifying pitch, to generate synthetic audio data. Adding noise, for example, involves incorporating background sounds into the audio signal.

This approach simulates real-world noise conditions, allowing the model to develop greater resilience to background disturbances. As illustrated in Figure 6, the normal audio signal presents a clear representation of sound, while Figure 7 depicts the same audio signal with added noise, showcasing the impact of this augmentation technique. This process not only enriches the dataset but also enhances the ability of the model to generalize by exposing it to a wider range of audio scenarios.

Time stretching was employed to adjust the duration of the audio signal while preserving its pitch, as illustrated in Figure 8. This technique allows for modifications in the speaking rate enabling the model to effectively accommodate various speech tempos. Such adaptability is particularly beneficial for applications like speech recognition, where the ability to understand different speaking speeds can enhance performance and accuracy.

Pitch shifting allows for altering an audio signal pitch without affecting its duration. This technique can produce variations in the speaker's pitch and intonation, enhancing the model's ability to generalize across different voice tones. Figure 9 illustrates the audio signal after pitch shifting, while Figure 10 showcases the original audio signal alongside its modified pitch. By leveraging this method, a more versatile model that effectively recognizes and processes diverse vocal characteristics can be created.

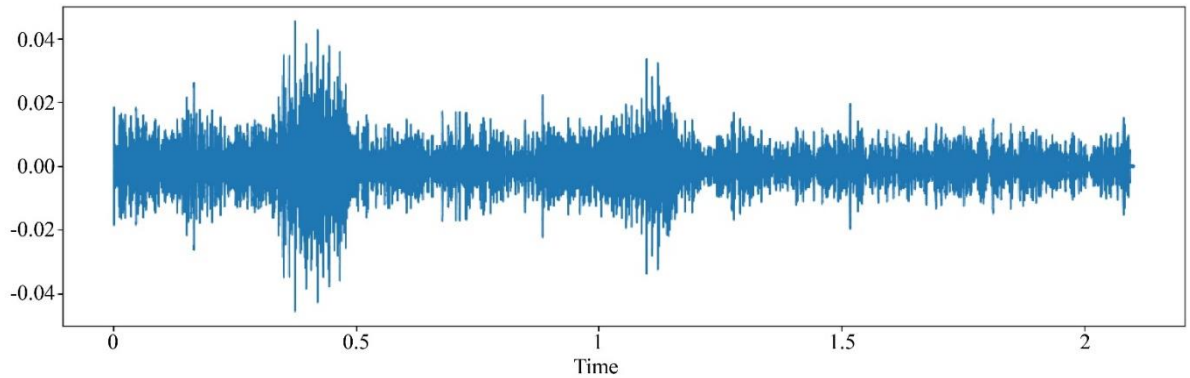


Fig. 6 Normal audio signal

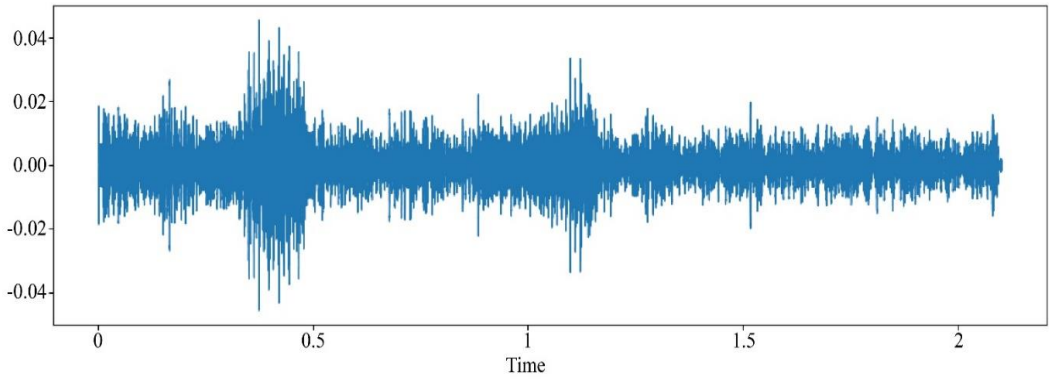


Fig. 7 Normal audio with noise

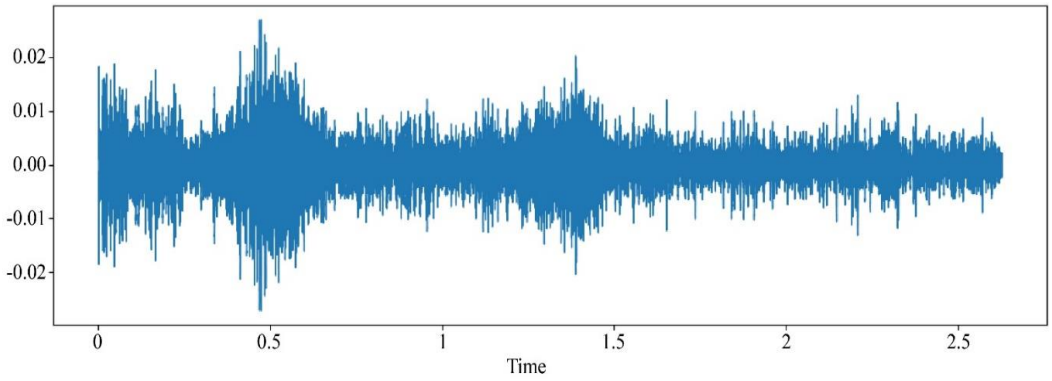


Fig. 8 Stretched audio

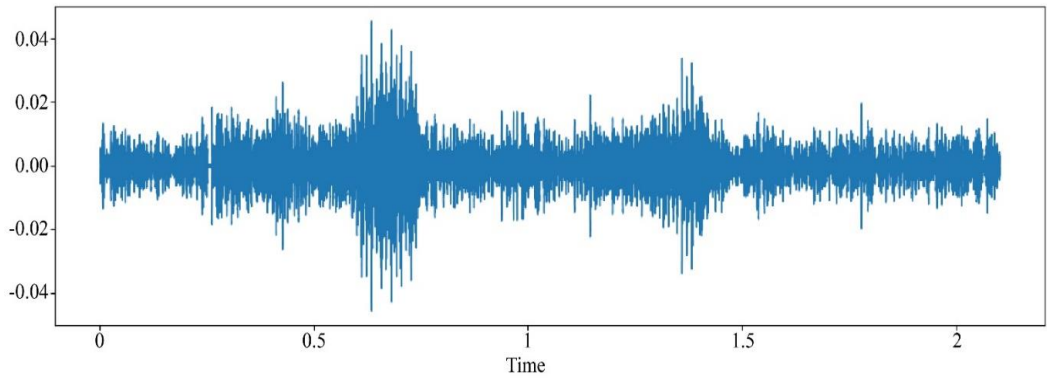


Fig. 9 Shifted audio

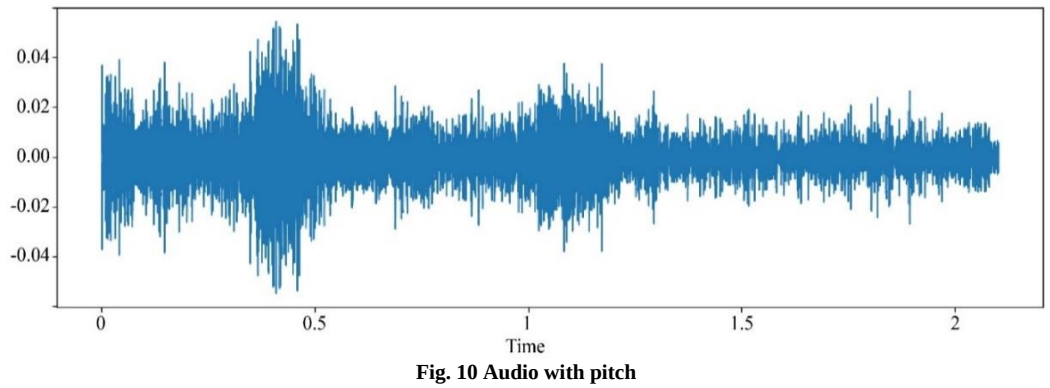


Fig. 10 Audio with pitch

3.4. Feature Extraction

MFCC is the commonly used feature extraction method in SER, as it efficiently extracts the features of speech signals. They are a representation of a sound signal's short-term power spectrum. The human vocal tract produces a spectrum that maps the known variation of the critical bandwidth of the ear using two filters: a logarithmic filter at a frequency above 1 kHz and a linear filter at a frequency below 1 kHz, which allows them to capture the features of speech [20]. The steps for MFCC feature extraction are depicted in Figure 11.

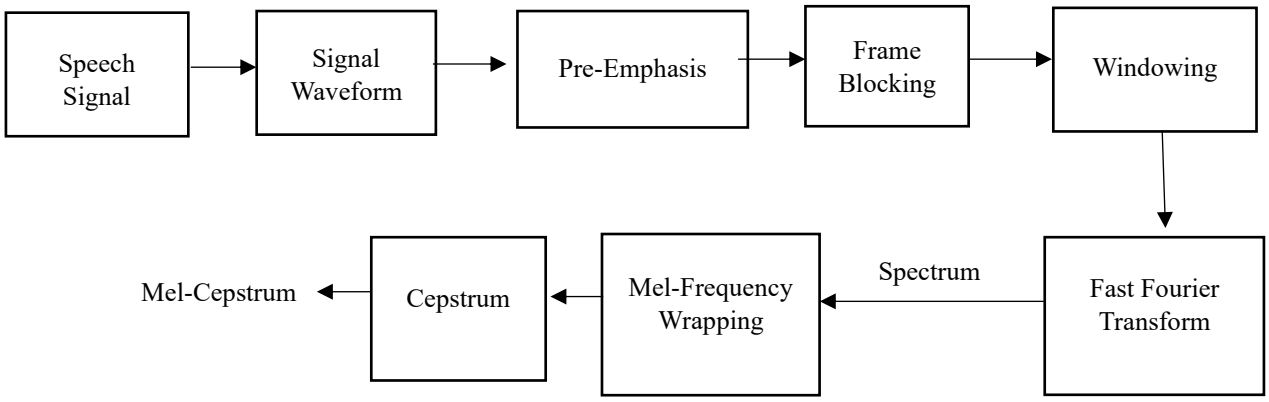


Fig. 11 MFCC feature extraction

Pre-emphasis was used on the speech signals to compensate for the high-frequency components that were blocked during human sound production. It is achieved by applying a high-pass filter to the audio signal, amplifying the higher frequencies shown in Equation (1).

$$p(x) = r(x) - \alpha * r(x-1) \quad (1)$$

Where, $p(x)$ represents the output signal, $r(x)$ and $r(x-1)$ denotes the present and past signals. The value of α is between 0.9 and 1. The audio signal is split into N samples of small frames, with M samples separating $N-M$ samples overlapping the adjacent frames. This procedure continued until the entire signal was divided into small frames.

Through windowing, the spectral distortion was decreased by reducing the signal to zero at the beginning and end of every frame. Once the input signal $r(x)$, is multiplied with a window $w(x)$, at time x, the extracted signal is obtained and shown by Equation (2).

$$q(x) = r(x) * w(x),\ 0 \le x \le N-1 \quad (2)$$

Where N is the sample count in every frame, a Hamming window was employed in this case as it lowers the frequency resolution of spectral analysis while reducing the sidelobe levels in the window transfer function. Equation (3) represents the Hamming window function.

$$w(x) = 0.54 - 0.46\ \cos\left[\frac{2\pi x}{N-1}\right], 0 \le x \le N-1 \quad (3)$$

Fast Fourier Transform (FFT) is a broadly utilized algorithm that proficiently calculates Discrete Fourier Transform (DFT). It is employed to transform N samples from the time domain to the frequency domain. The output obtained from FFT is referred to as a periodogram or spectrum. DFT is defined as a set of N samples $\{q_x\}$, which is shown in Equation (4).

$$Q_k = \sum_{x=0}^{N-1} q_x\, e^{-j2\pi kn/N}, k = 0,1,2,\dots\dots\dots, N-1 \quad (4)$$

Where, Q_k, represents the signal frequency components. Mel frequency is the study of sound frequencies experienced by humans. Human hearing shows variation in sensitivity across different frequency bands. The sensitivity decays at frequencies beyond 1000 Hz, and the spacing of the Mel frequency scale is linear, below 1 kHz, while above this threshold, the scale becomes logarithmic. The speech signal is given in Equation (5).

$$Mel(f) = 2595 * \log_{10}(1 + \frac{f}{700}) \quad (5)$$

The last step in the MFCC process is the calculation of the cepstrum, where the log Mel spectrum is transformed back to the time domain. This transformation is performed by DCT and captures the essential energy components of the signal. The output obtained from DCT is known as MFCC and is represented by Equation (6).

$$C_x = \sum_{x=0}^{N-1} \log\left|\sum_{x=0}^{N-1} r(x)\exp\left(\frac{-j2\pi kx}{N}\right)\right| \exp\left(\frac{j2\pi kx}{N}\right) \quad (6)$$

Where, $x = 0,1,2,\dots\dots\dots N-1$, C_x represents the MFCC, and n represents the coefficients. These factors extract essential information about speech signal frequency, making them effective for recognizing emotions based on vocal patterns and tone.

3.5. Model Development

3.5.1. Convolutional Neural Network

A CNN network is a DL model especially created to process and analyze images. It has several advantages: its resemblance to a human visual processing system, its highly structured design for processing both 2D and 3D images and its efficiency in learning and extracting features [21]. It is composed of multiple layers, as illustrated in Figure 12. Convolutional layers apply kernels or filters all over the image to obtain features from the input images.

These layers capture textures, edges, and shapes. A pooling layer is employed to down sample the feature maps produced by the convolutional layer, hence reducing the spatial dimensions of the data. Ultimately, the fully connected layer analyses the extracted features and performs tasks such as regression or classification. CNN proved effective in identifying objects from images due to their capability to share parameters and connect the pixels nearby, allowing them to learn patterns at distinct levels such as textures and shapes.

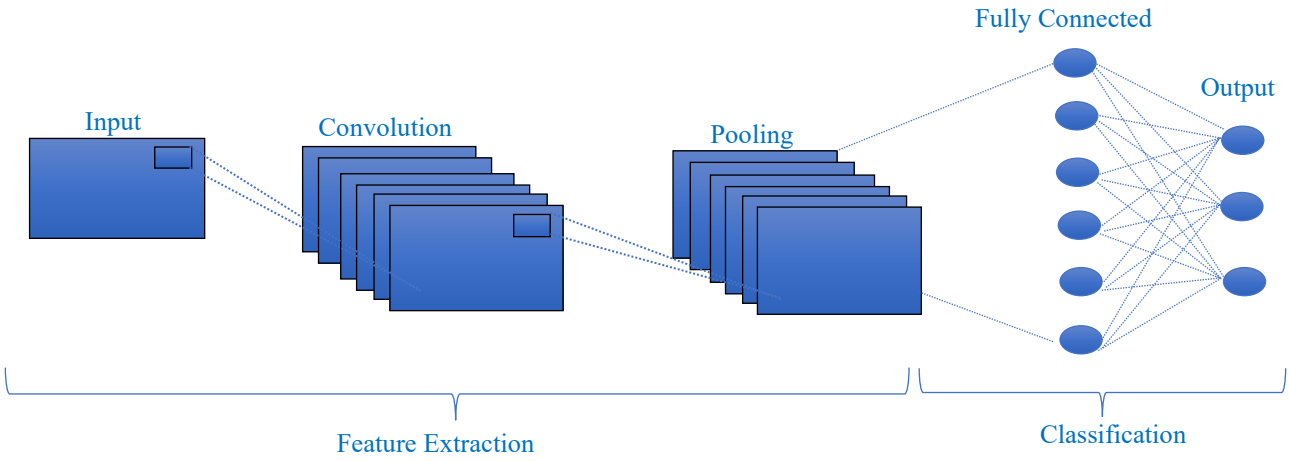


Fig. 12 Architecture of CNN

During the convolution operation, a filter moves across the input image, performing element-wise multiplication with the corresponding pixel values and then summing the results to produce a feature map. This process effectively extracts important features from the image. The mathematical representation of this convolution operation is illustrated in Equation (7).

$$(I * K)(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(m, n) K(x + m, y + n) \quad (7)$$

Where K denotes the convolution kernel, I is the input image, and x and y are the coordinates in the output feature map. After applying convolution, the resulting feature map undergoes a non-linear transformation through an activation function, commonly Rectified Linear Unit (ReLU), defined in Equation (8).

$$f(x) = \max(0, x) \quad (8)$$

To effectively reduce dimensionality while preserving essential features in a neural network, pooling layers are utilized. Among the various pooling techniques, max pooling is the most commonly employed method.

This approach entails choosing the greatest value from a specified region of the feature map, hence highlighting the most significant features while discarding less critical information. The mathematical representation of max pooling is articulated in Equation (9) as follows:

$$P(x, y) = \max_{(m,n) \epsilon R} F(m, n) \quad (9)$$

Where R is the region over which max pooling is applied. The resulting feature maps are then flattened into a one-dimensional vector and passed through fully connected (dense) layers. This is where classification or regression tasks are performed. The computations for the dense layer are expressed in Equation (10).

$$z = W \cdot x + b \quad (10)$$

Where W denotes the weight matrix, x is the input vector from the previous layer, b is the bias term, and z is the fully connected layer output. Finally, the output layer produces a probability distribution from a softmax function, as shown in Equation (11).

$$\sigma(z)_x = \frac{e^{z_x}}{\sum_{y=1}^{K} e^{z_y}} \quad (11)$$

Where, $\sigma(z)_x$ is the probability of class x, and K represents all classes.

3.5.2. Long Short-Term Memory Networks (LSTM)

LSTM is one of the many variations of Recurrent Neural Networks (RNNs) and is widely recognized for its effectiveness in feature extraction. Figure 13 showcases the architecture of an LSTM cell. The forget gate plays a crucial role in determining whether to retain or discard data from both the current input and previous states. This decision is influenced by a sigmoid function, yielding an output value between 0 and 1. A score of 0 signifies the elimination of prior information, whereas a value of 1 denotes preserving that information. The input gate evaluates the significance of the current input required for the task, whereas the output gate determines the output according to the hidden state. Another sigmoid function is applied to determine what information should be transmitted through the output gate. At last, the tanh function is employed to activate the cell state, which is then multiplied to regulate the flow of information within the cell [22].

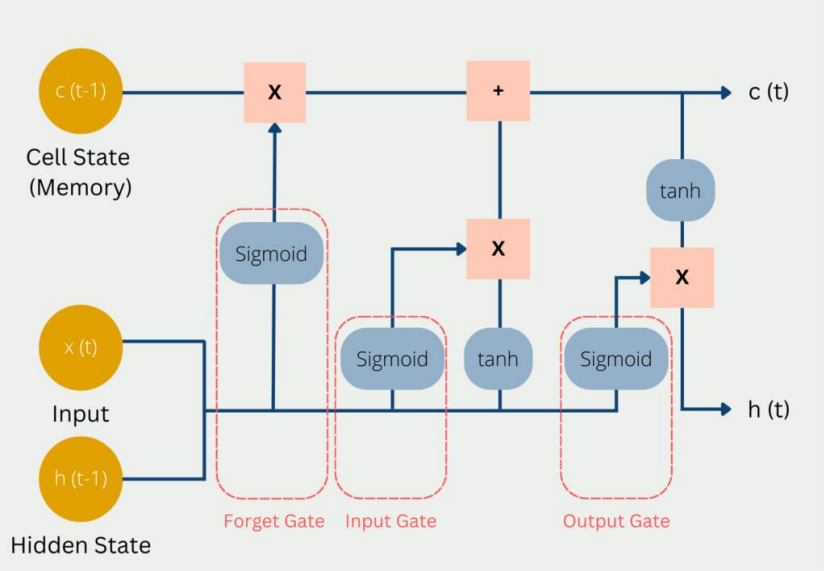


Fig. 13 Architecture of LSTM

The basic equation for calculating the hidden state at time step t is shown in Equation (12).

$$h_t = f(Ux_t + Vh_{t-1}) \tag{12}$$

Where U and V are the weight vectors of the hidden state, x_t is the current input state, and the previous hidden state is given by h_{t-1} from the time steps t and $t-1$.

The forget gate functions like a filter for previous memories, monitoring how much information is retained or discarded. When the forget gate is closed, it makes sure that no prior memories are saved, effectively blocking any influence from past experiences. On the other hand, if the gate is fully open, all previous memories can move through, making them contribute to current decisions and actions. This mechanism is necessary for sustaining an optimal balance among acquiring relevant information and obstructing the overload of unnecessary memories, thereby enhancing the system's overall efficiency. Equation (13) represents this functionality, mathematically illustrating how the forget gate controls memory retention based on its state.

$$f_t = \sigma(U_f * x_t + V_f * h_{t-1}) \tag{13}$$

Where U_f and V_f are the weight vectors for the forget gate.

If the previous memory is multiplied by a vector that is close to zero, it results in the erasure of most of the previous memory. On the other hand, if the forget gate is set to 1, it allows all previous memory to pass through without any modification. This conduct can be mathematically shown in Equations (14) and (15).

$$C_{t-1} * f_t = 0, \ if \ f_t = 0 \tag{14}$$

$$C_{t-1} * f_t = C_{t-1}, \ if \ f_t = 1 \tag{15}$$

Where, C_{t-1} represents the cell's previous memory.

The input gate plays a crucial role in defining how much new information should be combined into the current memory. By altering the parameters of this gate where, we can influence how current and past memories are affected. The input gate, as stated in Equation (16), is specifically designed to assess the implication of incoming data, making sure that only the most pertinent evidence is retained and incorporated.

$$i_t = \sigma(U_i * x_t + V_i * h_{t-1}) \tag{16}$$

Where, U_i and V_i are the weight vectors for current input and previous hidden states. The cell state in LSTM represents the network memory that flows through the whole sequence of data. C_t is the cell state at the time step t. Equation (17) represents the cell state.

$$\widehat{C_t} = \tanh\left(U_c * x_t + V_c * h_{t-1}\right) \quad (17)$$

Therefore, the updated internal memory state at the time step t is given in Equation (18).

$$C_t = C_{t-1} * f_t + \widehat{C_t} * I_t \quad (18)$$

The output gate regulates the current input, the previous output and the new memory. It determines how much additional memory should be incorporated into the following LSTM unit. The Equation (19) describes the function of the output gate. This mechanism ensures that the LSTM network retains relevant information while discarding what is no longer needed, thus maintaining an effective flow of information through the network.

$$o_t = \sigma(U_o * x_t + V_o * h_{t-1}) \quad (19)$$

Where, U_o and V_o are the weight vectors for current input and previous hidden states. As a result, the sigmoid function value lies in the range between 0 and 1 so that the cell states are modified, o_t and the tanh function is employed to identify the current hidden state, presented in Equation (20).

$$h_t = o_t * \tanh\left(C_t\right) \quad (20)$$

3.5.3. Proposed CNN-LSTM Model

The proposed CNN-LSTM model is designed to effectively process sequential data by integrating convolutional layers for feature extraction and LSTM layers for temporal pattern recognition. The architecture begins with a 1D convolutional layer comprising 200 filters, followed by ReLU activation to capture essential features from the input data. A max-pooling layer then reduces the spatial dimensions while preserving important information. Successive convolutional layers consisting of 100 and 50 filters, respectively, are each convoyed by max-pooling layers to further down-sample the data. The features captured from these layers are then given into an LSTM layer with 64 units, making the model acquire temporal data from the sequences. To reduce the hazard of overfitting, a dropout layer is unified after the first LSTM layer.

A second LSTM layer, also with 64 units methods the sequential data further. The model then includes two dense layers, each with 50 units and ReLU activation, enabling it to learn more complex patterns. The output from these layers is flattened and passed to a 100-unit dense layer that employs L2 regularization to reduce overfitting. At last, a softmax function transforms the model output into a probability distribution across six classes of speech emotions, making the model well-suited for classification tasks. The architecture of the proposed CNN-LSTM model is illustrated in Figure 14.

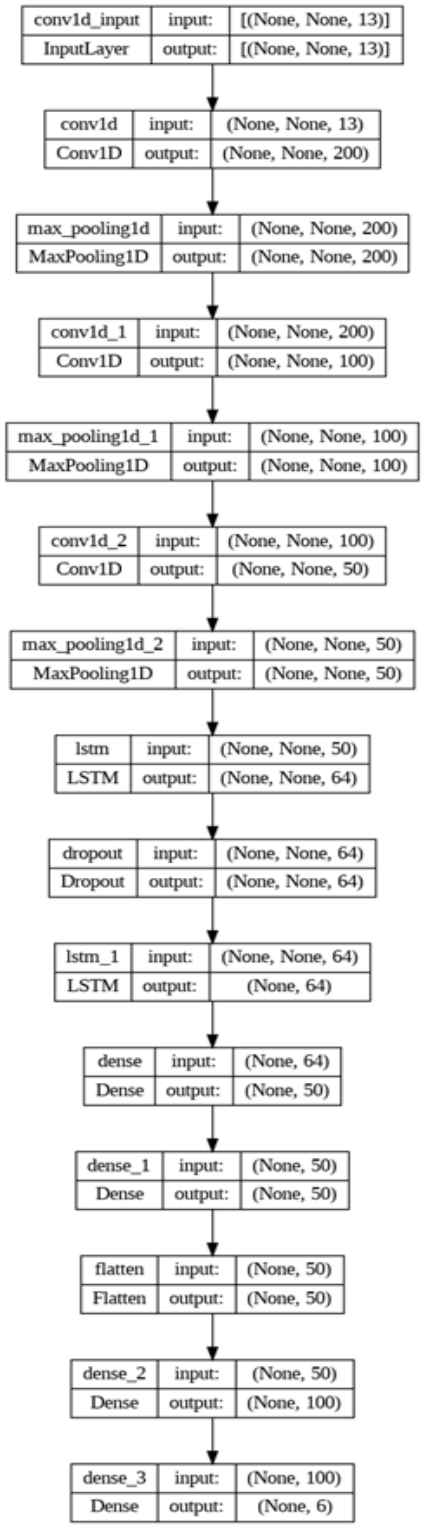


Fig. 14 Proposed CNN-LSTM model architecture

3.5.4. Gated Recurrent Network (GRU)

GRU is an advanced form of normal RNN that resolves short-term memory problems by utilizing a combined gating mechanism similar to an LSTM. The information flow inside the GRU is managed and even circulated by the internal gate mechanisms. These gates help the GRU cell to decide if the data should be retained or deleted, allowing relevant information to be passed to make accurate predictions [23]. Figure 15 shows the basic architecture of GRU.

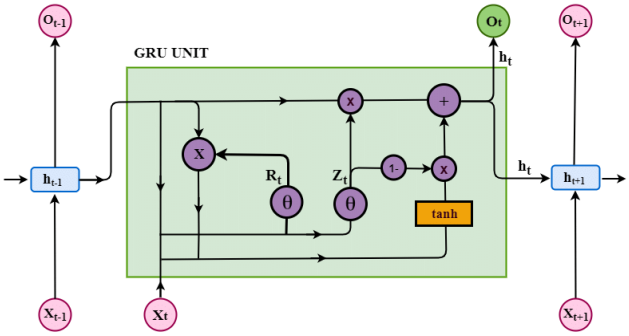


Fig. 15 Architecture of GRU

An update gate, z_t is created by connecting the forget and input gates. The amount of memory that can retain both new and previous data is maintained by the update gate. x_t is the current input vector, and h_{t-1} is the value obtained from the previous neighboring layer. Therefore, the learnable weight matrix for the update gate w_z is shown in Equation (21).

$$z_t = \sigma(w_z \cdot [h_{t-1}, x_t] \tag{21}$$

GRU uses the reset gate r_t represented in Equation (22) to integrate the current input x_t, with the previous memory h_{t-1}. The reset gate is responsible for identifying the integration of the Equation with the previous state and new output.

$$r_t = \sigma(w_r \cdot [h_{t-1}, x_t] \tag{22}$$

Where the learnable weight matrix for the reset gate is denoted by w_r. The output range of a hyperbolic tangent function $tanh$ varies from -1 to 1. Besides, h_t is the computed value for the current cell is shown in Equations (23) and (24).

$$h_t = \tanh\,(r_t * [h_{t-1}, x_t]) \tag{23}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * h_t \tag{24}$$

3.5.5. Proposed CNN-GRU Model

The CNN-GRU model is designed to effectively manage sequential data by integrating convolutional layers for feature extraction and GRU layers for recognizing temporal patterns. It starts with a 1D convolutional layer consisting of 200 filters to capture low-level input features, followed by a ReLU activation function to introduce non-linearity. A max-pooling layer is employed to lessen the spatial dimensions of the feature maps. This extraction process is repeated in the second and third convolutional layers, which utilize 100 and 50 filters, respectively, and are followed by max pooling to further down-sample the data.

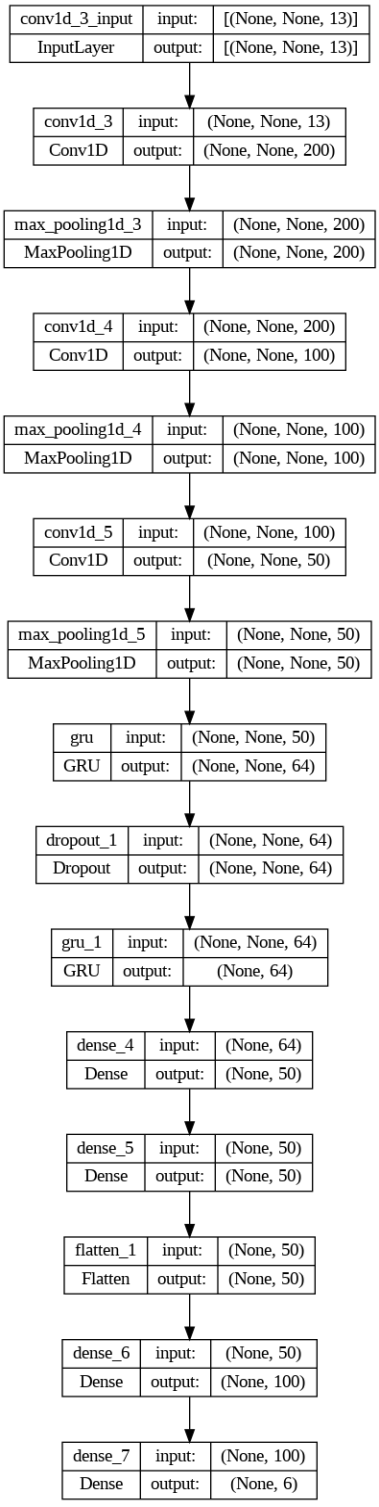


Fig. 16 Proposed CNN-GRU model architecture

After completing the feature extraction phase, a GRU layer with 64 units is introduced to capture the temporal information present in the sequence. To combat over fitting, a dropout layer with a rate of 0.5 is included. This is succeeded by another GRU layer, again with 64 units, which refines the extraction of sequential patterns.

The extracted features are then processed through two dense layers containing 50 units and employing ReLU activation before being flattened. The final stages of the model include a dense layer with 100 units incorporating L2 regularization to mitigate overfitting, followed by a softmax output layer with 6 units that classifies the data into one of six emotional categories. The architecture of the CNN-GRU framework is illustrated in Figure 16.

Ensemble learning is a widely used machine learning technique aimed at enhancing predictive accuracy by integrating multiple individual models. The core principle of this approach is that combining predictions from several models often yields better performance than relying on a single model. One effective ensemble method is Stacked SVM, which improves classification performance by amalgamating multiple SVM classifiers.

In this study, we employ CNN-LSTM and CNN-GRU models as base learners. The CNN-LSTM model is designed to extract spatial features while capturing temporal dependencies in speech signals. Meanwhile, the CNN-GRU framework is utilized for effective feature extraction. The predictions generated by both models are then fed into a meta-learner, another SVM classifier.

This meta-learner synthesizes the outputs from the base models to achieve a final emotion classification. This stacked architecture allows the system to uncover compound relationships within the data, resulting in greater performance in emotion recognition tasks. The algorithm for the suggested system is given below.

Algorithm : SER Recognition Using Ensemble Model

Input : Speech Signals for various emotions (Anger, Fear, Disgust, Neutral, Happy, Sadness)

Output : Emotion Recognition Model

Begin :

- Load and preprocess data:
  1. Collect dataset: $D = \{(X_i, y_i)\}_{i=0}^{N-1}$, were X_i is the Speech signal and $y_i \in \{0, 1, ...N-1\}$.
  2. Preprocess:
     - Normalize: $B_i' \rightarrow \frac{B_i' - \mu}{\sigma}$
     - Resize: $B_i \rightarrow B_i' \in R^{224 \times 224}$
     - Data Augmentation: $B_i' \rightarrow \{B_i''\}$
- Define CNN-LSTM Model:
  1. Input: $224 \times 224 \times 3$
     Block 1: Conv1D (200, (3,3), activation='relu')
     MaxPooling2D (pool size= (2, 2))
     Block 2: Conv1D (100, (3,3), activation='relu')
     MaxPooling2D (pool size= (2, 2))
     Block 3: Conv1D (50, (3,3), activation='relu')
     MaxPooling2D (pool size= (2, 2))
     LSTM (64)
     Dropout (0.5)
     LSTM (64)
     Dense (50, activation='relu')
     Dense (50, activation='relu')
     Flatten ()
     Dense (100, activation='relu')
     Dense (6, activation='softmax')
- Define CNN-GRU Model:
  1. Input: $224 \times 224 \times 3$
     Block 1: Conv1D (200, (3,3), activation='relu')
     MaxPooling2D (pool size= (2, 2))
     Block 2: Conv1D (100, (3,3), activation='relu')
     MaxPooling2D (pool size= (2, 2))
     Block 3: Conv1D (50, (3,3), activation='relu')

MaxPooling2D (pool size= (2, 2))
GRU (64)
Dropout (0.5)
GRU (64)
Dense (50, activation='relu')
Dense (50, activation='relu')
Flatten ()
Dense (100, activation='relu')
Dense (6, activation='softmax')

- Define Ensemble model:
  stacked_predictions = np. concatenate ([y_{pred} cnn_lstm_model, y_{pred} cnn_gru_model], axis=1)
  svm_model.fit (stacked_predictions, y_{test})
  ensemble_predictions=svm_model.fit. predict(stacked_predictions)
- Model Compilation and Training:
  1. Compile each model P:
     optimizer=Adam ()
     loss=sparse_ categorical _crossentropy
     metrics=[accuracy]
  2. Train: P.fit (X_{train} , y_{train} ,validation_data= (X_{val}, y_{val}), batch size= (32,64), epochs= (200,100)
- Model Evaluation and Comparison:

1. Evaluate:
   metrics=P.evaluate(X_{test} , y_{test}), where metrics contain accuracy recall precision.

- Save the Model

End

3.6. Hardware and Software Setup

The study utilized a high-performance computational configuration comprising an Intel Core i7 CPU, 32GB of RAM and an NVIDIA GeForce GTX 1080Ti GPU, facilitating the better management of challenging computational workloads. The framework was accomplished with the Keras library a high-level neural network API based on TensorFlow known for its user-friendly interface and robust functionalities. The training practice was conducted on Google Colab, a cloud-based Python notebook platform that provides easy availability to substantial computational resources, hence enabling model training. An essential element of this research was the selection of hyper parameters that profoundly influence model performance during training. Unlike model parameters derived from the data, hyper parameters are predetermined by the user and are crucial in shaping the configuration of the training process to optimize the efficiency of the SER model. The precise hyper parameter selections and model configuration are detailed in Table 1.

Table 1. Hyperparameter specifications

Model	Optimizer	Dropout	Learning Rate	Batch Size	Loss Function	Number of Epochs	Activation Function
CNN-LSTM	Adam	0.5	0.001	32	Sparse Categorical Cross-Entropy	200	ReLU, Softmax
CNN-GRU				64		100	

4. Results and Discussion

The accuracy and loss plots are crucial for assessing the effectiveness of the SER framework. The accuracy plot illustrates the model's performance over time, showcasing its ability to predict outcomes accurately in comparison to actual results. Likewise, the loss plot demonstrates the model's learning process by tracking the decrease in the loss function over time, that lower loss values signify improved performance. Together, these plots highlight the model's proficiency in identifying various emotions from speech data.
.

Figures 17 and 18 present the CNN LSTM accuracy and loss plots. Initially the system's accuracy is relatively low at approximately 81.76%. However, as the epochs progress, accuracy improves, ultimately reaching around 85.83% by the final epoch. This upward trend signifies the model's enhanced capability to classify data correctly over time. Regarding loss, the system starts with a value of around 0.50. As training continues, this loss value gradually decreases, finishing at approximately 0.42 in the final epoch. This reduction indicates that the model's predictions increasingly align with the actual values reflecting its learning effectiveness.

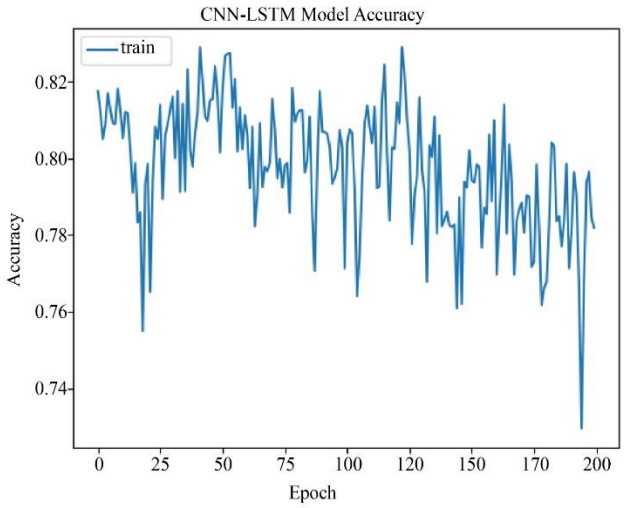


Fig. 17 Accuracy plot of CNN-LSTM model

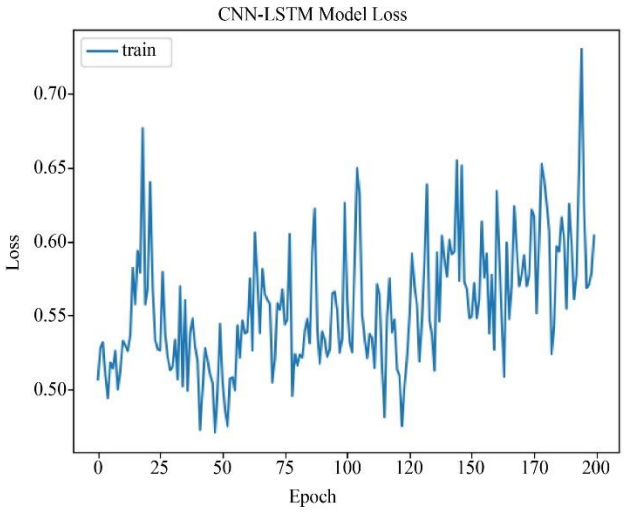


Fig. 18 Loss Plot of CNN-LSTM model

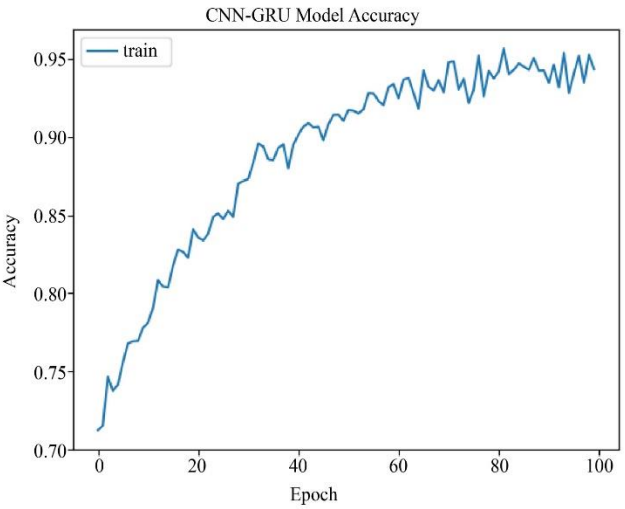


Fig. 19 Accuracy plot of CNN-GRU model

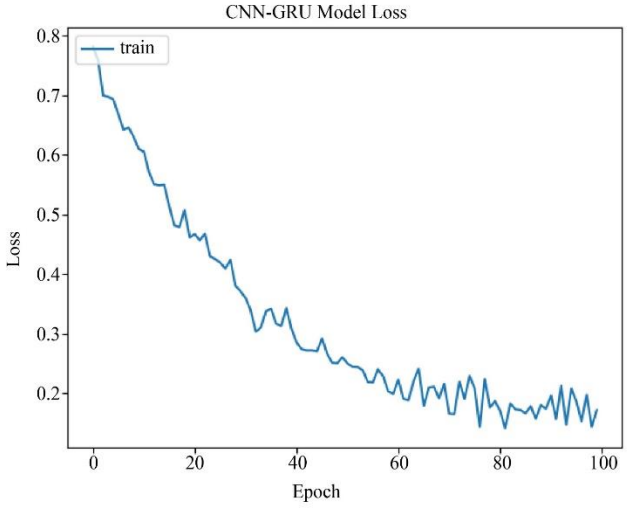


Fig. 20 Loss plot of CNN-GRU model

Figures 19 and 20 illustrate the accuracy and loss plots for the CNN GRU model. Initially, the model demonstrates relatively low accuracy at around 71.22% during the early epochs. As training progresses, there is a significant improvement in accuracy, ultimately reaching approximately 97.74% by the final epoch. This increase indicates the model's effectiveness in accurately classifying the data over time.

The initial loss value is high, approximately 0.78. Whereas as the epochs advance the loss decreases steadily, culminating at 0.07 by the final epoch. This loss reduction indicates that the model's predictions closely align with the actual values, showcasing improved performance and reliability throughout the training process.

Initially, several factors are defined to quantify essential performance parameters as represented in the following Equations. These metrics, based on the principles of False Positive (FP), True Negative (TN), False Negative (FN) and True Positive (TP), are crucial for evaluating the efficacy of the model.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (25)$$

$$Precision = \frac{TP}{TP+FP} \quad (26)$$

$$Recall = \frac{TP}{TP+FN} \quad (27)$$

$$F1-score = 2 \times \frac{precision \times Recall}{Precision+Recall} \quad (28)$$

The graphical representation of the performance comparison of the proposed system is depicted in Figure 21, summarizing the performance of three different models: CNN-LSTM, CNN-GRU, and an ensemble model. The CNN-LSTM model attained an accuracy of 85.83%, with precision at

86.19%, recall at 85.97%, and an F1 score of 85.79%, indicating a balanced performance across these metrics. In contrast, the CNN-GRU model significantly outperformed the previous model, reaching 97.74% accuracy, 97.78% precision, 97.79% recall, and an F1 score of 97.76%. This improvement highlights its strong ability to generalize and accurately detect emotions. Finally, the ensemble model, which leverages the strengths of multiple models, excelled further, attaining 98.69% accuracy, 98.70% precision, 98.72% recall, and an F1 score of 98.70%. This demonstrates its superior efficiency in emotion recognition, confirming the effectiveness of combining model strengths for enhanced performance.

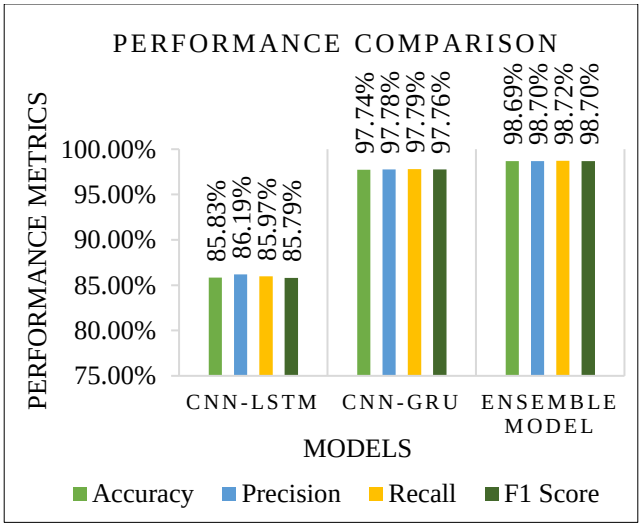


Fig. 21 Performance comparison of proposed model

The comparison of the ensemble model with various existing approaches, assessing the accuracy of Speech Emotion Recognition (SER) across different datasets, is presented in Table 2. The CNN-based model utilizing an acoustic feature set achieved an impressive 94.18% accuracy on the RAVDEES dataset. Meanwhile, a dilated CNN framework designed to capture both spatial emotional features and long-term dependencies reached 90% accuracy on the EMO-DB dataset. The DCNN combined with Random Forest (RF) to identify discriminative features also performed well, attaining an accuracy of 93.6% on the RAVDEES dataset. In addition, a cryptographic structure using the Time-Warping Quantization Technique (TWQT) attained an accuracy of 90.09% on EMO DB. The LSTM and multi-head attention network models yielded lower accuracies of 72% and 76% on the IEMOCAP dataset. In contrast, the proposed ensemble model significantly outperformed these existing methods, achieving an accuracy of 98.69% in classifying emotions from speech signals. Figure 22 provides a graphical representation of the performance comparison between the existing approaches and the proposed ensemble model, highlighting the superior performance of the ensemble method.

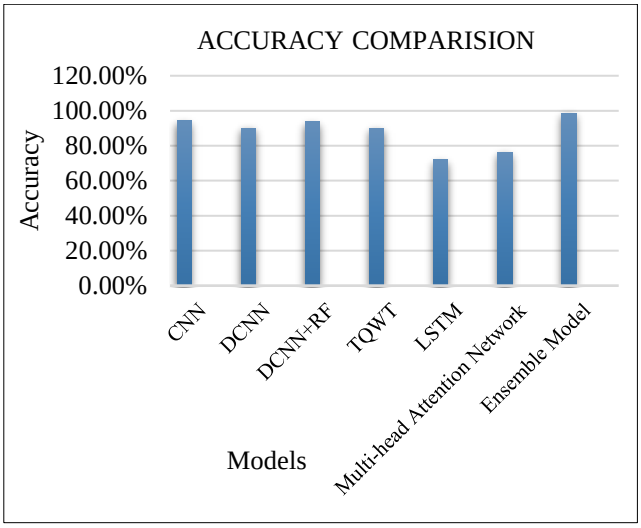


Fig. 22 Accuracy comparison of ensemble model with existing approaches

Table 2. Accuracy comparison of existing methods and proposed model

AUTHOR & REF	MODEL	DATASET	ACCURACY
Bhangale and Kothandaraman [4]	CNN	RAVDEES	94.18%
Mustaqeem and Kwon [9]	DCNN	EMO-DB	90%
Amjad et al. [11]	DCNN + RF	RAVDEES	93.6%
Tuncer et al. [13]	TQWT	EMO-DB	90.09%
Wang et al. [14]	LSTM	IEMOCAP	72%
Nediyanchath et al. [15]	Multi-head Attention Network	IEMOCAP	76.4%
Proposed Ensemble Model		**CREMA**	**98.69%**

5. Conclusion

Speech Emotion Recognition (SER) is a method that identifies and categorizes emotions such as anger, happiness, and sadness expressed through speech. It employs Deep Learning (DL) techniques to analyze features like tone, pitch and rhythm, enabling the accurate detection of emotional states in both real-time and recorded speech. This study developed an efficient SER system using a hybrid deep-learning model with an ensemble approach. Utilizing the "CREMA" dataset, which comprises 7,442 audio samples from various actors, the system recognized six distinct emotional states. The methodology involved several key steps: preprocessing, data augmentation

and feature extraction using Mel-Frequency Cepstral Coefficients (MFCC). Two hybrid models were designed, CNN-LSTM and CNN-GRU which effectively capture both temporal and spatial features from the speech data.

The outputs from these models were then combined through an ensemble learning approach utilizing a Support Vector Machine (SVM) classifier as the meta-learner. The proposed system achieved impressive performance metrics, including an accuracy of 98.69%, precision of 98.70%, recall of 98.72% and an F1 score of 98.70%.

These results demonstrate the superior performance of the ensemble system compared to both the individual CNN-LSTM and CNN-GRU models and several existing methods in previous studies. Overall, the findings underscore the effectiveness and superiority of the proposed ensemble system in accurately identifying emotions in speech.

Acknowledgements

The author conveys deep gratitude to the supervisor for contributing guidance and steadfast support during this study.

References

[1] Abdul Malik Badshah et al., "Speech Emotion Recognition from Spectrograms with Deep Convolutional Neural Network," *International Conference on Platform Technology and Service*, Busan, Korea, pp. 1-5, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[2] Dias Issa, M. Fatih Demirci, and Adnan Yazici, "Speech Emotion Recognition with Deep Convolutional Neural Networks," *Biomedical Signal Processing and Control*, vol. 59, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[3] Abdelaziz A. Abdelhamid et al., "Robust Speech Emotion Recognition Using CNN+ LSTM Based on Stochastic Fractal Search Optimization Algorithm," *IEEE Access*, vol. 10, pp. 49265-49284, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[4] Kishor Bhangale, and Mohanaprasad Kothandaraman, "Speech Emotion Recognition Based on Multiple Acoustic Features and Deep Convolutional Neural Network," *Electronics*, vol. 12, no. 4, pp. 1-17, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[5] Samuel Kakuba, Alwin Poulose, and Dong Seog Han, "Attention-Based Multi-Learning Approach for Speech Emotion Recognition with Dilated Convolution," *IEEE Access*, vol. 10, pp. 122302-122313, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[6] Arya Aftab et al., "Light-Sernet: A Lightweight Fully Convolutional Neural Network for Speech Emotion Recognition," *IEEE International Conference on Acoustics, Speech and Signal Processing*, Singapore, pp. 6912-6916, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[7] Apeksha Aggarwal et al., "Two-Way Feature Extraction for Speech Emotion Recognition Using Deep Learning," *Sensors,* vol. 22, no. 6, pp. 1-11, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[8] Lu-Qiao Li et al., "Emotion Recognition from Speech with StarGAN and Dense-DCNN," *IET Signal Processing*, vol. 16, no. 1, pp. 62-79, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[9] Soonil Kwon Mustaqeem, "MLT-DNet: Speech Emotion Recognition using 1D Dilated CNN Based on Multi-Learning Trick Approach," *Expert Systems with Applications*, vol. 167, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[10] Ziping Zhao et al., "Combining a Parallel 2D CNN With a Self-Attention Dilated Residual Network for CTC-Based Discrete Speech Emotion Recognition," *Neural Networks*, vol. 141, pp. 52-60, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[11] Ammar Amjad, Lal Khan, and Hsien-Tsung Chang, "Effect on Speech Emotion Classification of a Feature Selection Approach Using a Convolutional Neural Network," *PeerJ Computer Science*, vol. 7, pp. 1-28, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[12] Orhan Atila, and Abdulkadir Şengür, "Attention Guided 3D CNN-LSTM Model for Accurate Speech-Based Emotion Recognition," *Applied Acoustics*, vol. 182, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[13] Turker Tuncer, Sengul Dogan, and U. Rajendra Acharya, "Automated Accurate Speech Emotion Recognition System Using Twine Shuffle Pattern and Iterative Neighborhood Component Analysis Techniques," *Knowledge-Based Systems*, vol. 211, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[14] Jianyou Wang et al., "Speech Emotion Recognition with Dual-Sequence LSTM Architecture," *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing*, Barcelona, Spain, pp. 6474-6478, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[15] Anish Nediyanchath, Periyasamy Paramasivam, and Promod Yenigalla, "Multi-Head Attention for Speech Emotion Recognition with Auxiliary Learning of Gender Recognition," *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing,* pp. 7179-7183, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[16] Zengwei Yao et al., "Speech Emotion Recognition Using Fusion of Three Multi-Task Learning-based Classifiers: HSF-DNN, MS-CNN and LLD-RNN," *Speech Communication*, vol. 120, pp. 11-19, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[17] Mustaqeem, Muhammad Sajjad, and Soonil Kwon, "Clustering-Based Speech Emotion Recognition by Incorporating Learned Features and Deep BiLSTM," *IEEE Access*, vol. 8, pp. 79861-79875, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[18] Misbah Farooq et al., "Impact of Feature Selection Algorithm on Speech Emotion Recognition Using Deep Convolutional Neural Network," *Sensors*, vol. 20, no. 21, pp. 1-18, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[19] Crowd Sourced Emotional Multimodal Actors Dataset (CREMA-D), Kaggle. [Online]. Available: https://www.kaggle.com/datasets/ejlok1/cremad

[20] Kharibam Jilenkumari Devi, Ayekpam Alice Devi, and Khelchandra Thongam, "Automatic Speaker Recognition using MFCC and Artificial Neural Network," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, pp. 39-42, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[21] Md. Zahangir Alom et al., "A State-of-the-Art Survey on Deep Learning Theory and Architectures," *Electronics*, vol. 8, no. 3, pp. 1-66, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[22] M. Kalpana Chowdary, J. Anitha, and D. Jude Hemanth, "Emotion Recognition from EEG Signals Using Recurrent Neural Networks," *Electronics*, vol. 11, no. 15, pp. 1-20, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[23] Iram Bibi et al., "A Dynamic DL-Driven Architecture to Combat Sophisticated Android Malware," *IEEE Access*, vol. 8, pp. 129600-129612, 2020. [CrossRef] [Google Scholar] [Publisher Link]