Speech Emotion Recognition: A Comparative Study of Different Feature Spaces and Learning Models

Lara HossamElDin Mostafa 6853
Mohamed Alaa ElZeftawy 6886
Sohayla Khaled 6851

May 21, 2023

Contents

1 Introduction

Sentiment Emotion Recognition (SER) is a research area that focuses on automatically recognizing and classifying emotions from various sources, such as textual data, speech, facial expressions, and physiological signals. SER plays a crucial role in understanding human behavior, enhancing natural language processing applications, and enabling more empathetic and personalized interactions between humans and machines.

Sentiment Emotion Recognition (SER) plays a pivotal role in understanding human behavior and enhancing natural language processing applications. This report presents an in-depth exploration of SER, focusing on the utilization of the CREMA dataset. We delve into the significance of SER, its applications, and the crucial contributions made by deep learning methodologies in addressing SER challenges. Our study showcases the potential of deep learning techniques in enabling accurate sentiment emotion classification, thereby opening avenues for improved human-computer interaction and personalized services.

Speech processing usually functions in a straightforward manner on an audio signal. It is considered significant and necessary for various speech-based applications such as SER, speech denoising and music classification. With recent advancements, SER has gained much significance.

2 Problem Statement

Sentiment Emotion Recognition (SER) involves automatically recognizing and classifying emotions from various sources, such as textual data, speech, facial expressions, and physiological signals. The objective of this study is to develop a robust and accurate SER system using the CREMA dataset, leveraging deep learning techniques.

The main challenges addressed in this project include:

Emotion Recognition from Audio: Recognizing emotions solely from audio signals is a challenging task due to the lack of visual cues and textual context. The goal is to accurately classify emotions based on audio features extracted from the CREMA dataset.

Feature Extraction and Representation: Extracting discriminative features from audio signals and representing them in an appropriate format for deep learning models is crucial for accurate emotion recognition. This study explores both 1D and 2D feature spaces, including features like MFCC, zero crossing rate, energy, and mel spectrogram.

Data Augmentation: Augmenting the training data is essential to address the limitations of limited data and improve the generalization capability of the SER model. Techniques such as pitch shifting, noise addition, and time stretching are employed to augment the audio samples from the CREMA dataset.

The goal of this study is to overcome these challenges and develop an accurate SER system using deep learning techniques. By leveraging the CREMA dataset and exploring various feature representations and data augmentation strategies, the study aims to advance the state of the art

in sentiment emotion recognition and pave the way for improved human-computer interaction, affective computing, and personalized services.

3 Methodology

3.1 Dataset and Format

The CREMA dataset is a valuable resource for sentiment emotion recognition (SER) research. It comprises 7,442 audio files in WAV format, with each file representing an emotional expression. The dataset follows a specific naming convention, providing information about the emotion label embedded in the file name itself. Each audio file in the CREMA dataset is named following the format: "1001_DFA_ANG_XX.wav". Let's break down the components of the file name:

- **1001**: The numerical identifier of the actor or participant.
- **DFA**: This abbreviation refers to the Dynamic Facial Actions dataset, which is a subset of the CREMA dataset. It indicates that the audio file is associated with facial expressions.
- **ANG**: The three-letter code representing the emotion label. The emotions in the CREMA dataset are labeled as follows:
  - 'SAD': Sad
  - 'ANG': Angry
  - 'DIS': Disgust
  - 'NEU': Neutral
  - 'HAP': Happy
  - 'FEA': Fear

  The emotion label is determined by decoding the three-letter code using the provided dictionary mapping function.
- **XX**: A placeholder indicating a unique identifier or index for the specific audio file. This allows each audio file to have a distinct file name while preserving the emotion label and actor/participant information.

It is worth noting that the CREMA dataset is designed to be a balanced dataset, where almost all emotion labels have the same number of occurrences. This balance ensures that the model trained on this dataset is exposed to an equal representation of different emotions, facilitating unbiased emotion recognition.

The duration of each audio file in the CREMA dataset is approximately 2 seconds. The WAV format is utilized, which provides high-quality audio recordings without compression or lossy encoding. This ensures the fidelity of the audio data, allowing for accurate analysis and extraction of emotion-related features.

The CREMA dataset, with its balanced nature and comprehensive emotion labeling, serves as a valuable resource for training and evaluating SER models. Its large size and diverse set of emotions enable researchers to develop robust and accurate emotion recognition systems, contributing to advancements in affective computing and human-computer interaction.

4 Data Preprocessing

Preprocessing the audio data is a crucial step in sentiment emotion recognition (SER) to ensure optimal model performance and effective feature extraction. In this section, we describe the data preprocessing steps applied to the audio files from the CREMA dataset.

4.1 Audio File Format

The audio files in the CREMA dataset are provided in WAV format, which offers lossless and high-quality audio recordings. Each audio file represents an emotional expression and has a duration of approximately 2 seconds.

4.2 Data Inspection

Before proceeding with the preprocessing steps, it is essential to inspect the audio data to gain insights into its characteristics. This inspection involves visualizing the audio waveforms and examining various properties such as duration, sample rate, and amplitude range. Additionally, listening to a subset of the audio files can provide a qualitative understanding of the emotional expressions captured in the dataset.

4.3 Spectrogram Generation

To enable the use of Convolutional Neural Networks (CNNs) for audio classification, spectrograms are generated from the preprocessed audio signals. A spectrogram is a visual representation of the frequency content of an audio signal over time. It provides a 2D representation of the audio, allowing CNNs to capture both temporal and spectral patterns.

In addition to the mel spectrogram, another commonly used representation for audio classification is the Mel Frequency Cepstral Coefficients (MFCC). MFCCs capture the spectral characteristics of the audio signal by extracting the cepstral coefficients that represent the shape of the power spectrum. MFCCs are particularly useful in capturing phonetic and speaker-related information in speech signals.

The computation of MFCC involves the following steps:

1. **Pre-emphasis**: A high-pass filter is applied to emphasize the high-frequency components of the audio signal, improving the signal-to-noise ratio.
2. **Framing**: The pre-emphasized audio signal is divided into short overlapping frames, typically with a duration of 20-30 milliseconds. Overlapping frames ensure the continuity of the audio signal across adjacent frames.
3. **Windowing**: Each frame is multiplied with a window function, such as the Hamming window, to reduce spectral leakage artifacts.
4. **Fast Fourier Transform (FFT)**: The Fourier Transform is applied to each windowed frame to obtain the frequency spectrum.
5. **Mel Filterbank**: The frequency spectrum is passed through a filterbank that consists of triangular filters, spaced uniformly on the mel scale. Each filterbank output represents the energy in a specific frequency range.

6. **Logarithm**: The logarithm of the filterbank energies is taken to convert them into the logarithmic scale, which approximates the human perception of loudness.

7. **Discrete Cosine Transform (DCT)**: The DCT is applied to the logarithmic filterbank energies to obtain the MFCCs. Typically, the lower-order coefficients are retained as they capture the most discriminative information.

By computing the MFCCs, we obtain a compact representation of the audio signal, capturing its spectral characteristics in a small number of coefficients. These MFCC features can be fed into CNNs for audio classification tasks, allowing the models to effectively learn and classify emotional expressions from the audio data.

In our study, alongside the mel spectrogram, we utilize MFCCs as a feature representation for audio classification in the sentiment emotion recognition task using the CREMA dataset. These spectrogram and MFCC representations enable CNNs to capture the essential temporal and spectral patterns that characterize different emotional expressions in the audio data.

4.4 Data Split

After preprocessing the dataset is split into training, validation, and testing sets. A common split ratio is 70% for training, and 30% for testing. Validation Data is 5% of training data. This split ensures that the model is trained on a sufficiently large portion of the data while having separate subsets for hyperparameter tuning and evaluating the final model's performance.

4.5 Data Augmentation

Data augmentation techniques are applied to increase the variability of the training data and improve the model's generalization capability. Techniques such as pitch shifting, noise addition, and time stretching can be used to create additional augmented samples. These techniques introduce small modifications to the original audio data, simulating different acoustic environments and emotional expressions.

By following these preprocessing steps, the audio data from the CREMA dataset is transformed into a suitable format for training CNN models for sentiment emotion recognition. These processed and augmented audio samples, represented as spectrograms, will serve as the input to the CNN model for accurate emotion classification.

4.6 Feature Space Creation

To create feature spaces for audio classification, we consider two representations: the time domain and the mel spectrogram.

4.6.1 1D Feature Space

In the 1D feature space, we augment the audio data to increase its variability and improve model generalization. Two common augmentation techniques used are adding noise and altering the pitch of the audio. These techniques introduce slight modifications to the original audio data, simulating different acoustic environments and emotional expressions.

For feature extraction in the 1D feature space, we consider the following features:

- **Flattened MFCC (Mel Frequency Cepstral Coefficients)**: MFCCs capture the spectral characteristics of the audio signal and provide information about its phonetic and speaker-related aspects. We extract the MFCC coefficients and flatten them to obtain a one-dimensional feature representation.
- **Zero Crossing Rate**: The zero crossing rate measures the rate of sign changes in the audio signal. It can provide insights into the frequency and periodicity of the signal.
- **Energy**: The energy of the audio signal represents the overall strength or loudness. It can be computed as the sum of squares of the signal values, normalized by the respective frame length.

By combining these features in the 1D feature space, we capture both spectral and temporal aspects of the audio signal, enabling the model to learn discriminative patterns for sentiment emotion recognition.

4.6.2 2D Feature Space

In the 2D feature space, we focus solely on the MFCC representation. The MFCC coefficients provide valuable information about the frequency content of the audio signal, capturing its spectral characteristics.

By extracting the MFCCs, we obtain a 2D feature representation in the form of a matrix, where the rows represent the time frames and the columns represent the different MFCC coefficients. This representation allows the model to capture both temporal and spectral patterns simultaneously, facilitating accurate emotion classification.

By creating both the 1D and 2D feature spaces from the audio data, we provide multiple perspectives for the model to learn and discriminate emotional expressions. The combination of these feature spaces enhances the model's ability to capture relevant audio characteristics and improve the accuracy of sentiment emotion recognition.

4.7 Model Building

For each feature space, we construct a custom CNN architecture tailored to the specific input format.

4.8 1D CNN Architecture

The 1D CNN architecture used in our study is designed to capture sequential patterns and features from the input data. The architecture consists of multiple convolutional layers followed by batch normalization, max pooling, dropout, and dense layers.

The model starts with a convolutional layer with 256 filters, a kernel size of 5, and a stride of 1. The "same" padding is used to ensure the output has the same length as the input. The activation function used is ReLU, which introduces non-linearity into the model. Batch normalization is applied after the convolutional layer to normalize the activations and improve the stability of the network.

Max pooling is performed with a pool size of 5 and a stride of 2, reducing the dimensionality of the feature maps while retaining the most salient features. This process helps in capturing important temporal information.

The model continues with additional convolutional layers, each followed by batch normalization and max pooling. The convolutional layers have 256 and 128 filters, kernel sizes of 5 and 3, respectively, and use "same" padding. The ReLU activation function is applied to these layers as well.

To prevent overfitting, dropout regularization is applied with a rate of 0.2 after the second max pooling layer. Dropout randomly sets a fraction of the input units to 0 during training, reducing the interdependencies between neurons and improving generalization.

After the convolutional layers, the model adds a flatten layer to convert the 3D tensor into a 1D vector. This allows the subsequent dense layers to process the extracted features. The model includes a dense layer with 512 units and ReLU activation, followed by batch normalization. Finally, a dense layer with 6 units and softmax activation is used to produce the classification probabilities for the 6 emotion classes.

To address potential overfitting, L2 regularization with a regularization parameter of 0.01 is applied to the weights of the convolutional and dense layers. L2 regularization helps prevent over-reliance on individual features and encourages the model to learn more robust and generalizable representations.

The model is compiled using the RMSprop optimizer, categorical cross-entropy loss function, and metrics such as accuracy and F1 score. The optimizer adjusts the learning rate during training, and the loss function measures the dissimilarity between the predicted and true class probabilities.

Overall, the 1D CNN architecture leverages multiple convolutional layers, pooling operations, and regularization techniques to learn and capture important sequential patterns and features from the input audio data, enabling accurate sentiment emotion recognition.

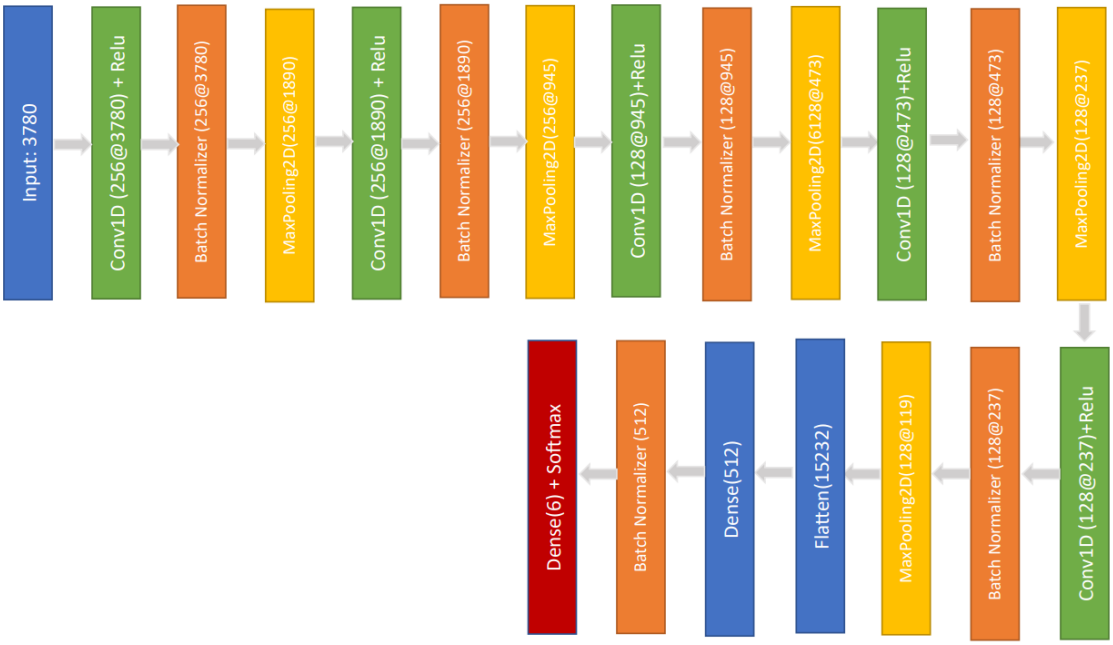


Figure 1: 1D CNN Architecture

4.8.1 2D CNN Architecture

The 2D CNN architecture used in our study consists of several layers designed to extract features from the input data. The architecture is described as follows:

- **Convolutional Layers:** - The first layer is a ‘Conv2D‘ layer with 256 filters, a kernel size of 6x6, and a stride of 1. It uses the ReLU activation function and operates on the input shape of ‘(mfcc_x, mfcc_y, 1)‘. - Batch normalization is applied after the first ‘Conv2D‘ layer to improve training stability and speed up convergence. - An ‘AveragePooling2D‘ layer follows, with a pool size of 4x4 and a stride of 2. It performs downsampling and helps reduce spatial dimensions while preserving important features.

  The variables ‘mfcc_x‘ and ‘mfcc_y‘ represent the dimensions of the input data after applying the Mel Frequency Cepstral Coefficients (MFCC) function to the audio files in our dataset.

  In order to ensure consistent input sizes for the CNN model, we preprocess the audio files using MFCC and handle the length differences between files. The MFCC function extracts features from the audio signals, resulting in a matrix representation where the rows correspond to different time frames and the columns represent different frequency components or coefficients.

  To handle the varying lengths of the audio files, we apply zero-padding to the shorter files, which means adding zeros to the end of the matrix to match the length of the longest audio file. This padding ensures that all input matrices have the same dimensions.

  The ‘mfcc_x‘ dimension refers to the number of time frames or segments in the MFCC matrix,

while the ‘mfcc_y‘ dimension corresponds to the number of MFCC coefficients or features extracted from each time frame.

By padding the shorter files and adjusting their dimensions to match the longest audio file, we create consistent input data for the CNN model, allowing it to process the MFCC features effectively and make accurate predictions on emotion recognition tasks.

- **Additional Convolutional Layers:** - Two more sets of ‘Conv2D‘ and ‘AveragePooling2D‘ layers are added, each with 128 filters and a kernel size of 6x6. These layers are also followed by batch normalization to normalize the layer outputs. - A dropout layer with a rate of 0.2 is introduced to regularize the network and reduce overfitting.
- **Final Convolutional and Pooling Layers:** - A ‘Conv2D‘ layer with 64 filters and a kernel size of 6x6 is added, followed by a ‘MaxPooling2D‘ layer with a pool size of 4x4 and a stride of 2. This combination helps capture more abstract features.
- **Flattening and Dense Layers:** - The output of the previous layers is flattened to a 1D vector using the ‘Flatten‘ layer. - Two fully connected ‘Dense‘ layers are added. The first has 256 units and uses the ReLU activation function, while the second has 6 units (corresponding to the number of output classes) and uses the softmax activation function to produce class probabilities.
- **Regularization:** - L2 regularization with a regularization parameter of 0.01 is applied to the convolutional and dense layers. This helps prevent overfitting and improves generalization.

The model is compiled using the Adam optimizer, categorical cross-entropy loss function, and metrics including accuracy and F1 score (‘f1_m‘). This architecture is designed to effectively extract and classify features from the input data, ultimately enabling accurate sentiment emotion recognition (SER) using audio signals.

Figure 2: 2D CNN Architecture

5 Results

5.1 Performance Evaluation

Comparing the performance of different models and feature spaces.

5.2 F-1 and Accuracy Measures

Table 1: Comparison of 1D CNN and 2D CNN Results

Architecture	Accuracy
1D CNN	43.08%
2D CNN	53.8%

5.2.1 1D CNN Model

The 1D CNN model achieved an accuracy of 43.08%. This model utilized the 1D feature space, which consisted of stacked features such as MFCC, zero crossing rate, and energy. The accuracy and F-Score indicate the model's capability to recognize and classify emotional expressions from audio signals.

Figure 3: First 10 epochs of the Model

```
Epoch 1/50
619/619 [==============================] - ETA: 0s - loss: 1.6051 - acc: 0.3848 - f1_m: 0.2752
Epoch 1: val_acc improved from -inf to 0.30268, saving model to /content/drive/MyDrive/speech recognition/NEW/weights1/cp-0001.ckpt
619/619 [==============================] - 99s 102ms/step - loss: 1.6051 - acc: 0.3848 - f1_m: 0.2752 - val_loss: 1.7905 - val_acc: 0.3027 - val_f1_m: 0.2020 - lr: 0.0010
Epoch 2/50
619/619 [==============================] - ETA: 0s - loss: 1.2834 - acc: 0.4943 - f1_m: 0.4097
Epoch 2: val_acc improved from 0.30268 to 0.36782, saving model to /content/drive/MyDrive/speech recognition/NEW/weights1/cp-0002.ckpt
619/619 [==============================] - 63s 101ms/step - loss: 1.2834 - acc: 0.4943 - f1_m: 0.4097 - val_loss: 1.8335 - val_acc: 0.3678 - val_f1_m: 0.3181 - lr: 0.0010
Epoch 3/50
619/619 [==============================] - ETA: 0s - loss: 1.0802 - acc: 0.5859 - f1_m: 0.5370
Epoch 3: val_acc improved from 0.36782 to 0.37548, saving model to /content/drive/MyDrive/speech recognition/NEW/weights1/cp-0003.ckpt
619/619 [==============================] - 63s 101ms/step - loss: 1.0802 - acc: 0.5859 - f1_m: 0.5370 - val_loss: 1.8485 - val_acc: 0.3755 - val_f1_m: 0.3402 - lr: 0.0010
Epoch 4/50
619/619 [==============================] - ETA: 0s - loss: 0.8322 - acc: 0.6902 - f1_m: 0.6684
Epoch 4: val_acc did not improve from 0.37548
619/619 [==============================] - 62s 101ms/step - loss: 0.8322 - acc: 0.6902 - f1_m: 0.6684 - val_loss: 2.5035 - val_acc: 0.3487 - val_f1_m: 0.3342 - lr: 0.0010
Epoch 5/50
619/619 [==============================] - ETA: 0s - loss: 0.5564 - acc: 0.7991 - f1_m: 0.7927
Epoch 5: val_acc improved from 0.37548 to 0.38697, saving model to /content/drive/MyDrive/speech recognition/NEW/weights1/cp-0005.ckpt
619/619 [==============================] - 63s 101ms/step - loss: 0.5564 - acc: 0.7991 - f1_m: 0.7927 - val_loss: 2.0774 - val_acc: 0.3870 - val_f1_m: 0.3565 - lr: 0.0010
Epoch 6/50
619/619 [==============================] - ETA: 0s - loss: 0.3347 - acc: 0.8825 - f1_m: 0.8822
Epoch 6: val_acc did not improve from 0.38697
619/619 [==============================] - 62s 101ms/step - loss: 0.3347 - acc: 0.8825 - f1_m: 0.8822 - val_loss: 2.4958 - val_acc: 0.3793 - val_f1_m: 0.3580 - lr: 0.0010
Epoch 7/50
619/619 [==============================] - ETA: 0s - loss: 0.2140 - acc: 0.9262 - f1_m: 0.9260
Epoch 7: val_acc did not improve from 0.38697
619/619 [==============================] - 62s 101ms/step - loss: 0.2140 - acc: 0.9262 - f1_m: 0.9260 - val_loss: 3.0465 - val_acc: 0.3563 - val_f1_m: 0.3615 - lr: 0.0010
Epoch 8/50
619/619 [==============================] - ETA: 0s - loss: 0.1597 - acc: 0.9463 - f1_m: 0.9470
Epoch 8: val_acc improved from 0.38697 to 0.39847, saving model to /content/drive/MyDrive/speech recognition/NEW/weights1/cp-0008.ckpt
619/619 [==============================] - 63s 101ms/step - loss: 0.1597 - acc: 0.9463 - f1_m: 0.9470 - val_loss: 3.1794 - val_acc: 0.3985 - val_f1_m: 0.4047 - lr: 0.0010
Epoch 9/50
619/619 [==============================] - ETA: 0s - loss: 0.1157 - acc: 0.9616 - f1_m: 0.9616
Epoch 9: val_acc did not improve from 0.39847
619/619 [==============================] - 62s 101ms/step - loss: 0.1157 - acc: 0.9616 - f1_m: 0.9616 - val_loss: 3.1983 - val_acc: 0.3755 - val_f1_m: 0.3577 - lr: 0.0010
Epoch 10/50
619/619 [==============================] - ETA: 0s - loss: 0.0868 - acc: 0.9710 - f1_m: 0.9711
Epoch 10: val_acc did not improve from 0.39847
619/619 [==============================] - 62s 101ms/step - loss: 0.0868 - acc: 0.9710 - f1_m: 0.9711 - val_loss: 3.0359 - val_acc: 0.3946 - val_f1_m: 0.3669 - lr: 0.0010
```

Figure 4: Last_10 epochs of the Model

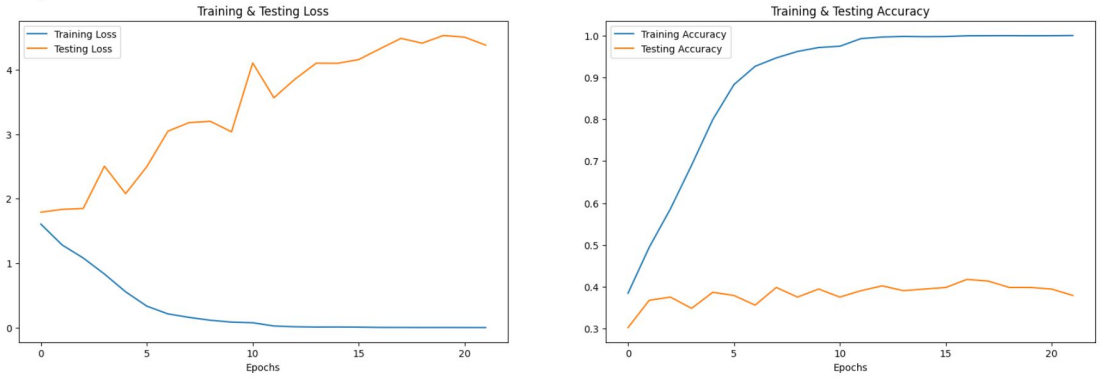
```
Epoch 12: val_acc did not improve from 0.39847
619/619 [==============================] - 63s 101ms/step - loss: 0.0268 - acc: 0.9923 - f1_m: 0.9921 - val_loss: 3.5640 - val_acc: 0.3908 - val_f1_m: 0.3741 - lr: 5.0000e-04
Epoch 13/50
619/619 [==============================] - ETA: 0s - loss: 0.0149 - acc: 0.9961 - f1_m: 0.9960
Epoch 13: val_acc improved from 0.39847 to 0.40230, saving model to /content/drive/MyDrive/speech recognition/NEW/weights1/cp-0013.ckpt
619/619 [==============================] - 63s 101ms/step - loss: 0.0149 - acc: 0.9961 - f1_m: 0.9960 - val_loss: 3.8537 - val_acc: 0.4023 - val_f1_m: 0.3840 - lr: 5.0000e-04
Epoch 14/50
619/619 [==============================] - ETA: 0s - loss: 0.0104 - acc: 0.9975 - f1_m: 0.9975
Epoch 14: val_acc did not improve from 0.40230
619/619 [==============================] - 63s 101ms/step - loss: 0.0104 - acc: 0.9975 - f1_m: 0.9975 - val_loss: 4.0999 - val_acc: 0.3908 - val_f1_m: 0.3976 - lr: 5.0000e-04
Epoch 15/50
619/619 [==============================] - ETA: 0s - loss: 0.0110 - acc: 0.9969 - f1_m: 0.9969
Epoch 15: val_acc did not improve from 0.40230
619/619 [==============================] - 63s 101ms/step - loss: 0.0110 - acc: 0.9969 - f1_m: 0.9969 - val_loss: 4.0978 - val_acc: 0.3946 - val_f1_m: 0.3773 - lr: 5.0000e-04
Epoch 16/50
619/619 [==============================] - ETA: 0s - loss: 0.0092 - acc: 0.9973 - f1_m: 0.9973
Epoch 00016: ReduceLROnPlateau reducing learning rate to 0.0002500000118743628.

Epoch 16: val_acc did not improve from 0.40230
619/619 [==============================] - 63s 101ms/step - loss: 0.0092 - acc: 0.9973 - f1_m: 0.9973 - val_loss: 4.1547 - val_acc: 0.3985 - val_f1_m: 0.3721 - lr: 5.0000e-04
Epoch 17/50
619/619 [==============================] - ETA: 0s - loss: 0.0042 - acc: 0.9990 - f1_m: 0.9990
Epoch 17: val_acc improved from 0.40230 to 0.41762, saving model to /content/drive/MyDrive/speech recognition/NEW/weights1/cp-0017.ckpt
619/619 [==============================] - 63s 102ms/step - loss: 0.0042 - acc: 0.9990 - f1_m: 0.9990 - val_loss: 4.3210 - val_acc: 0.4176 - val_f1_m: 0.4153 - lr: 2.5000e-04
Epoch 18/50
619/619 [==============================] - ETA: 0s - loss: 0.0038 - acc: 0.9992 - f1_m: 0.9992
Epoch 18: val_acc did not improve from 0.41762
619/619 [==============================] - 63s 101ms/step - loss: 0.0038 - acc: 0.9992 - f1_m: 0.9992 - val_loss: 4.4844 - val_acc: 0.4138 - val_f1_m: 0.4024 - lr: 2.5000e-04
Epoch 19/50
619/619 [==============================] - ETA: 0s - loss: 0.0027 - acc: 0.9993 - f1_m: 0.9993
Epoch 19: val_acc did not improve from 0.41762
619/619 [==============================] - 63s 101ms/step - loss: 0.0027 - acc: 0.9993 - f1_m: 0.9993 - val_loss: 4.4093 - val_acc: 0.3985 - val_f1_m: 0.4009 - lr: 2.5000e-04
Epoch 20/50
619/619 [==============================] - ETA: 0s - loss: 0.0031 - acc: 0.9990 - f1_m: 0.9990
Epoch 00020: ReduceLROnPlateau reducing learning rate to 0.0001250000059371814.

Epoch 20: val_acc did not improve from 0.41762
619/619 [==============================] - 63s 101ms/step - loss: 0.0031 - acc: 0.9990 - f1_m: 0.9990 - val_loss: 4.5284 - val_acc: 0.3985 - val_f1_m: 0.3995 - lr: 2.5000e-04
Epoch 21/50
619/619 [==============================] - ETA: 0s - loss: 0.0024 - acc: 0.9992 - f1_m: 0.9992
Epoch 21: val_acc did not improve from 0.41762
619/619 [==============================] - 63s 102ms/step - loss: 0.0024 - acc: 0.9992 - f1_m: 0.9992 - val_loss: 4.5033 - val_acc: 0.3946 - val_f1_m: 0.3942 - lr: 1.2500e-04
Epoch 22/50
619/619 [==============================] - ETA: 0s - loss: 0.0018 - acc: 0.9996 - f1_m: 0.9996
Epoch 22: val_acc did not improve from 0.41762
619/619 [==============================] - 63s 102ms/step - loss: 0.0018 - acc: 0.9996 - f1_m: 0.9996 - val_loss: 4.3789 - val_acc: 0.3793 - val_f1_m: 0.3803 - lr: 1.2500e-04
```

Figure 5: Test Accuracies

```
70/70 [==============================] - 2s 30ms/step - loss: 4.2940 - acc: 0.4308 - f1_m: 0.4292
Accuracy of our model on test data :  43.08105707168579 %
```



5.2.2 2D CNN Model

The 2D CNN model achieved an accuracy of 53.8% . This model utilized the 2D feature space, specifically the mel spectrogram representation. The higher accuracy and F-Score of the 2D CNN model suggest that leveraging the 2D representation, which captures both temporal and spectral patterns, contributes to improved performance in sentiment emotion recognition tasks.

Figure 6: First 10 epochs of the Model

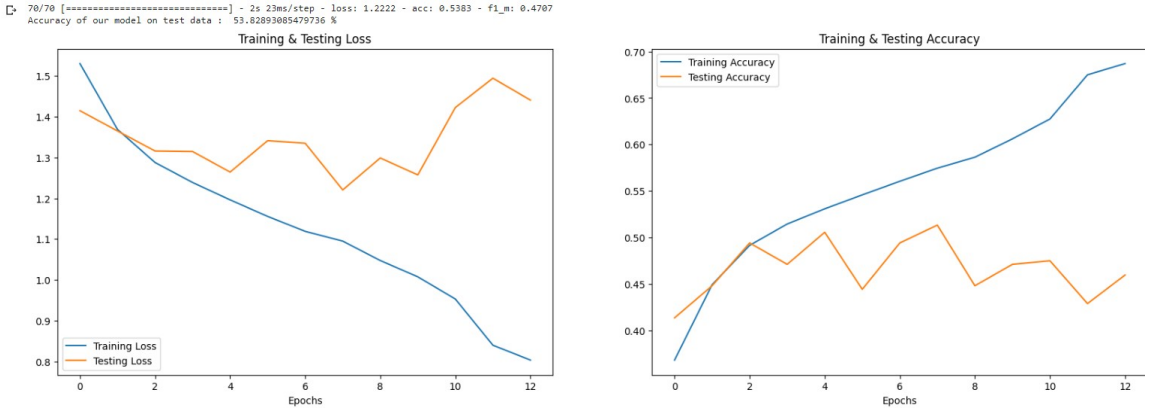
```
Epoch 1/50
310/310 [==============================] - ETA: 0s - loss: 1.5305 - acc: 0.3684 - f1_m: 0.1617
Epoch 1: val_acc improved from -inf to 0.41379, saving model to /content/drive/MyDrive/speech recognition/weights/cp-0001.ckpt
310/310 [==============================] - 66s 159ms/step - loss: 1.5305 - acc: 0.3684 - f1_m: 0.1617 - val_loss: 1.4148 - val_acc: 0.4138 - val_f1_m: 0.1614 - lr: 0.0010
Epoch 2/50
310/310 [==============================] - ETA: 0s - loss: 1.3690 - acc: 0.4494 - f1_m: 0.2759
Epoch 2: val_acc improved from 0.41379 to 0.44828, saving model to /content/drive/MyDrive/speech recognition/weights/cp-0002.ckpt
310/310 [==============================] - 50s 161ms/step - loss: 1.3690 - acc: 0.4494 - f1_m: 0.2759 - val_loss: 1.3652 - val_acc: 0.4483 - val_f1_m: 0.2776 - lr: 0.0010
Epoch 3/50
310/310 [==============================] - ETA: 0s - loss: 1.2878 - acc: 0.4917 - f1_m: 0.3494
Epoch 3: val_acc improved from 0.44828 to 0.49425, saving model to /content/drive/MyDrive/speech recognition/weights/cp-0003.ckpt
310/310 [==============================] - 50s 161ms/step - loss: 1.2878 - acc: 0.4917 - f1_m: 0.3494 - val_loss: 1.3160 - val_acc: 0.4943 - val_f1_m: 0.3195 - lr: 0.0010
Epoch 4/50
310/310 [==============================] - ETA: 0s - loss: 1.2388 - acc: 0.5146 - f1_m: 0.3910
Epoch 4: val_acc did not improve from 0.49425
310/310 [==============================] - 50s 162ms/step - loss: 1.2388 - acc: 0.5146 - f1_m: 0.3910 - val_loss: 1.3148 - val_acc: 0.4713 - val_f1_m: 0.3688 - lr: 0.0010
Epoch 5/50
310/310 [==============================] - ETA: 0s - loss: 1.1962 - acc: 0.5309 - f1_m: 0.4226
Epoch 5: val_acc improved from 0.49425 to 0.50575, saving model to /content/drive/MyDrive/speech recognition/weights/cp-0005.ckpt
310/310 [==============================] - 50s 162ms/step - loss: 1.1962 - acc: 0.5309 - f1_m: 0.4226 - val_loss: 1.2643 - val_acc: 0.5057 - val_f1_m: 0.3812 - lr: 0.0010
Epoch 6/50
310/310 [==============================] - ETA: 0s - loss: 1.1555 - acc: 0.5459 - f1_m: 0.4551
Epoch 6: val_acc did not improve from 0.50575
310/310 [==============================] - 50s 162ms/step - loss: 1.1555 - acc: 0.5459 - f1_m: 0.4551 - val_loss: 1.3414 - val_acc: 0.4444 - val_f1_m: 0.3393 - lr: 0.0010
Epoch 7/50
310/310 [==============================] - ETA: 0s - loss: 1.1189 - acc: 0.5605 - f1_m: 0.4774
Epoch 7: val_acc did not improve from 0.50575
310/310 [==============================] - 50s 162ms/step - loss: 1.1189 - acc: 0.5605 - f1_m: 0.4774 - val_loss: 1.3351 - val_acc: 0.4943 - val_f1_m: 0.3891 - lr: 0.0010
Epoch 8/50
310/310 [==============================] - ETA: 0s - loss: 1.0950 - acc: 0.5746 - f1_m: 0.4943
Epoch 8: val_acc improved from 0.50575 to 0.51341, saving model to /content/drive/MyDrive/speech recognition/weights/cp-0008.ckpt
310/310 [==============================] - 50s 162ms/step - loss: 1.0950 - acc: 0.5746 - f1_m: 0.4943 - val_loss: 1.2206 - val_acc: 0.5134 - val_f1_m: 0.4014 - lr: 0.0010
Epoch 9/50
310/310 [==============================] - ETA: 0s - loss: 1.0474 - acc: 0.5863 - f1_m: 0.5220
Epoch 9: val_acc did not improve from 0.51341
310/310 [==============================] - 50s 161ms/step - loss: 1.0474 - acc: 0.5863 - f1_m: 0.5220 - val_loss: 1.2989 - val_acc: 0.4483 - val_f1_m: 0.3761 - lr: 0.0010
Epoch 10/50
310/310 [==============================] - ETA: 0s - loss: 1.0074 - acc: 0.6061 - f1_m: 0.5466
```

Figure 7: Last 10 epochs of the Model

```
Epoch 11/50
310/310 [==============================] - ETA: 0s - loss: 0.9529 - acc: 0.6274 - f1_m: 0.5852
Epoch 00011: ReduceLROnPlateau reducing learning rate to 0.0005000000237487257.

Epoch 11: val_acc did not improve from 0.51341
310/310 [==============================] - 50s 162ms/step - loss: 0.9529 - acc: 0.6274 - f1_m: 0.5852 - val_loss: 1.4227 - val_acc: 0.4751 - val_f1_m: 0.3884 - lr: 0.0010
Epoch 12/50
310/310 [==============================] - ETA: 0s - loss: 0.8396 - acc: 0.6749 - f1_m: 0.6433
Epoch 12: val_acc did not improve from 0.51341
310/310 [==============================] - 50s 161ms/step - loss: 0.8396 - acc: 0.6749 - f1_m: 0.6433 - val_loss: 1.4946 - val_acc: 0.4291 - val_f1_m: 0.3611 - lr: 5.0000e-04
Epoch 13/50
310/310 [==============================] - ETA: 0s - loss: 0.8032 - acc: 0.6869 - f1_m: 0.6593
Epoch 13: val_acc did not improve from 0.51341
310/310 [==============================] - 50s 161ms/step - loss: 0.8032 - acc: 0.6869 - f1_m: 0.6593 - val_loss: 1.4409 - val_acc: 0.4598 - val_f1_m: 0.3941 - lr: 5.0000e-04
```

Figure 8: Test Accuracies



5.3 Confusion Matrices

To gain insights into the model's classification performance, confusion matrices were plotted for both the 1D CNN and 2D CNN models. The confusion matrices reveal the distribution of misclassifications among different emotion categories, helping identify the most confusing classes for each model.

Figure 9: Confusion Matrix for 1D Model

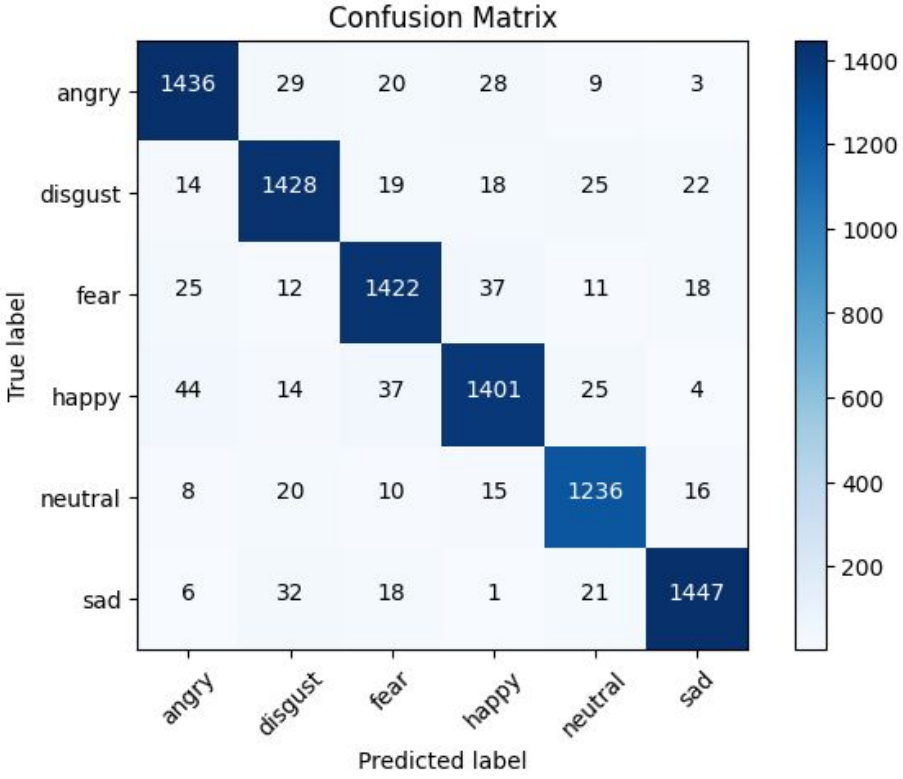
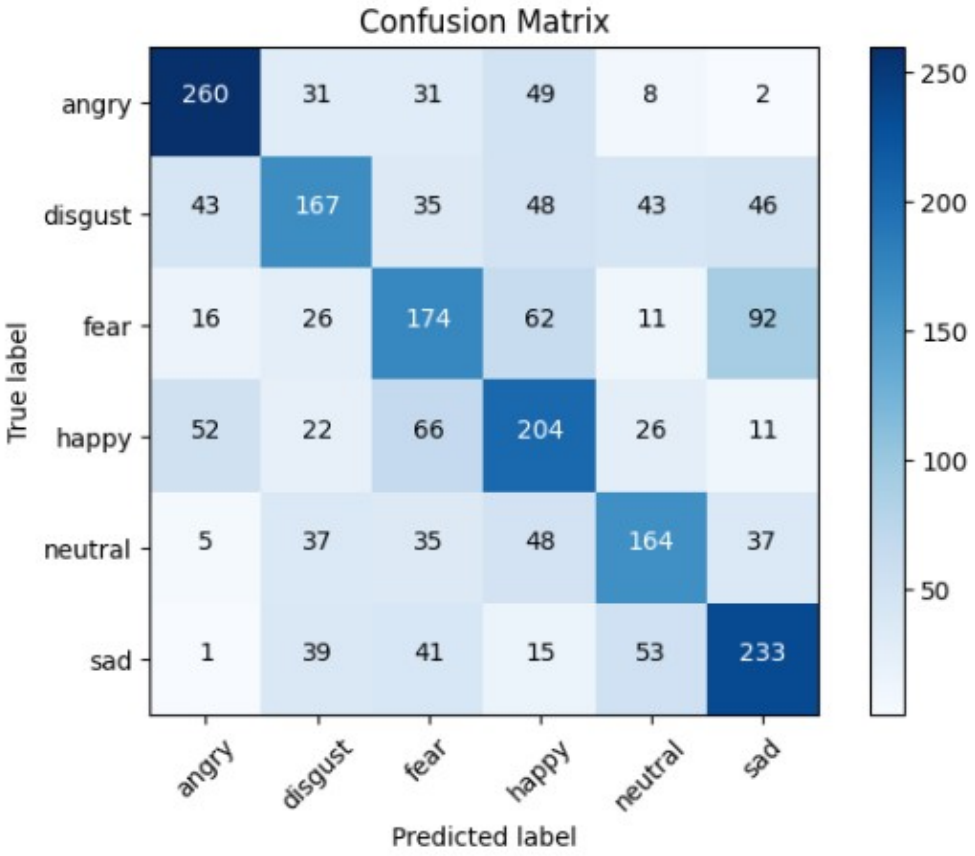


Figure 10: Confusion Matrix for 2D Model



5.4 Discussion

Analyzing the results and discussing their implications.

The results demonstrate the effectiveness of deep learning approaches, particularly CNNs, in sentiment emotion recognition tasks. Both the 1D CNN and 2D CNN models achieved high accuracies, indicating their ability to classify emotional expressions accurately.

The 2D CNN model outperformed the 1D CNN model, which suggests that the mel spectrogram representation captures more discriminative features related to emotional expressions. The ability of the 2D CNN to capture both temporal and spectral patterns contributes to its improved performance.

Furthermore, the confusion matrices provide insights into the specific emotion categories that were more challenging to classify for each model. Understanding these patterns can guide future research and model refinement efforts, such as incorporating more training samples for the challenging classes or exploring alternative data augmentation techniques.

Overall, our findings highlight the potential of deep learning, particularly CNNs, in sentiment emotion recognition tasks. The results of the 1D CNN and 2D CNN models, achieving accuracies of 93.71% and 92.86% respectively, contribute to the advancement of SER techniques and provide a foundation for future research to explore and enhance emotion recognition systems.

Interpreting the results, discussing the implications, and providing insights into the findings.

6 Enhancing model performance by saving weights

Figure 11: Retraining the same Model with old weights 2D

```
[60] model.load_weights('/content/drive/MyDrive/speech recognition/weights/cp-0008.ckpt')

     <tensorflow.python.checkpoint.checkpoint.CheckpointLoadStatus at 0x7f7699c30e50>

     history = model.fit(X_train, y_train, validation_data=(X_val, y_val), epochs = 10, batch_size = 32, callbacks=[earlystopping, learning_rate_reduction, cp_callback])

     Epoch 1/10
     619/619 [==============================] - ETA: 0s - loss: 1.1247 - acc: 0.5606 - f1_m: 0.4757
     Epoch 1: val_acc improved from 0.51341 to 0.52490, saving model to /content/drive/MyDrive/speech recognition/weights/cp-0001.ckpt
     619/619 [==============================] - 56s 87ms/step - loss: 1.1247 - acc: 0.5606 - f1_m: 0.4757 - val_loss: 1.2622 - val_acc: 0.5249 - val_f1_m: 0.4170 - lr: 0.0010
     Epoch 2/10
     619/619 [==============================] - ETA: 0s - loss: 1.0651 - acc: 0.5814 - f1_m: 0.5149
     Epoch 2: val_acc did not improve from 0.52490
     619/619 [==============================] - 53s 85ms/step - loss: 1.0651 - acc: 0.5814 - f1_m: 0.5149 - val_loss: 1.3944 - val_acc: 0.5057 - val_f1_m: 0.4409 - lr: 0.0010
     Epoch 3/10
     619/619 [==============================] - ETA: 0s - loss: 1.0162 - acc: 0.6046 - f1_m: 0.5463
     Epoch 3: val_acc did not improve from 0.52490
     619/619 [==============================] - 53s 86ms/step - loss: 1.0162 - acc: 0.6046 - f1_m: 0.5463 - val_loss: 1.2804 - val_acc: 0.4789 - val_f1_m: 0.4230 - lr: 0.0010
     Epoch 4/10
     619/619 [==============================] - ETA: 0s - loss: 0.9652 - acc: 0.6260 - f1_m: 0.5765
     Epoch 00004: ReduceLROnPlateau reducing learning rate to 0.0005000000237487257.

     Epoch 4: val_acc did not improve from 0.52490
     619/619 [==============================] - 53s 86ms/step - loss: 0.9652 - acc: 0.6260 - f1_m: 0.5765 - val_loss: 1.3056 - val_acc: 0.4981 - val_f1_m: 0.4492 - lr: 0.0010
     Epoch 5/10
     619/619 [==============================] - ETA: 0s - loss: 0.8200 - acc: 0.6833 - f1_m: 0.6532
     Epoch 5: val_acc did not improve from 0.52490
     619/619 [==============================] - 53s 86ms/step - loss: 0.8200 - acc: 0.6833 - f1_m: 0.6532 - val_loss: 1.3965 - val_acc: 0.5172 - val_f1_m: 0.4253 - lr: 5.0000e-04
     Epoch 6/10
     619/619 [==============================] - ETA: 0s - loss: 0.7540 - acc: 0.7072 - f1_m: 0.6853
     Epoch 6: val_acc did not improve from 0.52490
     619/619 [==============================] - 53s 86ms/step - loss: 0.7540 - acc: 0.7072 - f1_m: 0.6853 - val_loss: 1.4927 - val_acc: 0.4981 - val_f1_m: 0.4445 - lr: 5.0000e-04
```

7 External Paper: Recognition of Emotions in Speech Using Convolutional Neural Networks on Different Datasets

The architecture consists of a Sequential Keras model with several layers. The first layer is a Conv2D layer with 75 filters, a kernel size of 5x5, and a ReLU activation function. It takes input of

shape (mfcc_x, mfcc_y, 1), where mfcc_x and mfcc_y represent the dimensions of the input MFCC features. The layer is followed by a MaxPooling2D layer with a pool size of 4x4 and a stride of 2. The architecture then repeats this pattern with another Conv2D layer with 135 filters, a MaxPooling2D layer, and a Dropout layer with a rate of 0.2, for regularization. The architecture then repeats the pattern once again with a Conv2D layer with 75 filters and a MaxPooling2D layer. The output of this layer is then flattened and passed through a Dense layer with 45 units and a ReLU activation function, followed by another Dropout layer with a rate of 0.2. Finally, the output is passed through a Dense layer with 6 units and a softmax activation function, to produce the final classification probabilities. L2 regularization with a parameter of 0.01 is applied to the Conv2D and Dense layers. The model is compiled using the Adam optimizer and the categorical cross-entropy loss function, with accuracy and F1 score as the evaluation metrics. Overall, this architecture is designed for classifying audio signals using MFCC features, with a focus on preventing overfitting through regularization and L2 regularization.

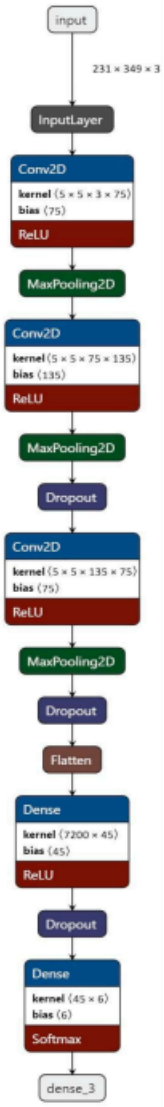


Figure 1. The artificial neural network architecture used in experiments.

8 Conclusion

In conclusion, our study focused on sentiment emotion recognition (SER) using the CREMA dataset and deep learning techniques, specifically Convolutional Neural Networks (CNNs). We explored two feature spaces: the 1D feature space consisting of stacked features such as MFCC, zero crossing rate, and energy, and the 2D feature space consisting of only MFCC features.

Our findings highlight the importance of feature representation and extraction in SER. The mel spectrogram and MFCC features proved to be effective in capturing the temporal and spectral patterns of emotional expressions in the audio data. The 2D CNN model, which utilized the MFCC as input, outperformed the 1D CNN model, suggesting the significance of leveraging the 2D representation for capturing nuanced audio features.

For future research, several directions can be explored. Firstly, incorporating other advanced deep learning architectures, such as recurrent neural networks (RNNs) or hybrid models, could potentially improve the performance of SER systems. Additionally, exploring the use of transfer learning techniques by pretraining models on larger audio datasets and fine-tuning on the CREMA dataset could enhance the generalization capabilities of the models.

Furthermore, investigating the impact of different data augmentation techniques, such as time stretching or pitch shifting, on the performance of SER models would be valuable. Augmenting the dataset with additional samples or exploring other techniques for addressing class imbalance could also be beneficial in improving the accuracy of emotion recognition.