Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 4 (2023)

A Comprehensive Investigation Into Speech Emotion Recognition Using Deep Learning

[1]Amara Zahoor, [2]Er Tarunjot Kaur

[1]M. Tech Scholar, Department of Electronics and Communication Engineering, Rayat Bahra University,Punjab, India
[2]Assistant Professor, Department of Electronics and Communication Engineering, Rayat Bahra University, Punjab, India

Abstract: Emotional expression is a fundamental aspect of human communication and understanding emotions from speech has a significant impact on a number of software, namely human-computer interface, virtual assistants, and emotion-driven analytics. A comprehensive study on utilizing Convolutional Neural Networks (CNNs) for emotional recognition of speech (SER). is presented in this thesis. Emotions from speech signals are recognized through the development of an efficient and accurate model. A large dataset of emotional speech samples, covering various emotions like Angry, disgusted, afraid, joyful, indifferent, depressed, surprised, is preprocessed and analyzed.
A novel CNN architecture, optimized for SER, is proposed, incorporating multiple a maximum pooling sequential normalization, and various layers of complexity. By extracting relevant features from speech representations, distinct emotional patterns are discerned effectively. Numerous experiments were carried out to validate the CNN model's effectiveness. The dataset underwent division into distinct training, validation, and testing phases. The outcomes exhibit encouraging accuracy and resilience in discerning emotions across various categories. By scrutinizing the model's confusion matrix, a comprehensive classification report was generated. This report assessed precision, recall, and F1-score for individual emotional classes. The model's strengths were accentuated, while possible avenues for enhancement were pinpointed.
Keywords: Emotion, happy, CNN, Deep learning, forecast

1. Introduction

Speech is a fundamental a means of human interaction that transmits not just language material but rich emotional cues. Emotions play a crucial role in human interactions, influencing the way we express ourselves and understand others. Understanding and recognizing emotions from speech has become a critical aspect of human-computer interaction and affective computing systems.

By automatically learning hierarchical feature

from raw data, CNNs have demonstrated extraordinary effectiveness in a variety of computer vision applications, such as picture categorization and object recognition. Their capacity to seize spatial dependencies makes them a promising choice for sequential data analysis, such as speech signals. However, applying CNNs to SER introduces its own set of challenges, such as dealing with variable-length speech signals and encoding temporal dependencies effectively.

To address these challenges and capitalize on the potential applications, this paper proposes the creation of a brand-new CNN-based speech recognition and emotion detection algorithm. The model seeks to use neural network's abilities to continuously learn. relevant acoustic features from speech data and effectively classify emotions across different speakers and emotional expressions. By exploring various architectures and optimization strategies, the goal is to obtain cutting-edge SER speed.

Overall, this the goal of this paper is to develop the ability to recognize emotions. technology and its integration into real-world applications. The research endeavours to offer information on the model's clarity, investigate its robustness to noise and variations, and discuss the ethical implications of deploying emotion recognition systems in society. Through this work, we aim to facilitate more empathetic and emotionally intelligent human-computer interactions and enhance the understanding of emotions in human communication.

1.1 Sentiment Analysis

Sentiment Analysis, alternatively referred to as opinion mining, involves the computational identification and classification of opinions, sentiments, and emotions conveyed within text-based information. Unlike speech emotion recognition, which pertains to deciphering emotions in spoken communication, sentiment analysis concentrates on the realm of written language, encompassing content like social media updates, reviews, and customer input

Both Speech Emotion Recognition (SER) and Sentiment Analysis play crucial roles in understanding and harnessing human emotions in different contexts. In HCI, SER enhances human-computer interactions by enabling emotion-aware interfaces and enhancing user experiences. On the other hand, Sentiment Analysis provides valuable insights into public opinions and sentiments, which are instrumental in market research, brand monitoring, and political analysis. Together, these technologies contribute to the development of emotionally intelligent computing systems and improved human-machine interactions.

2. Literature Review

Badshah et al. [2] adopted a similar approach. The raw audio data was formed from the EMO-DB emotion dataset, the authors created spectrograms, which are graphic representations of the sound samples. Following that, the CNN model received these spectrograms. The findings demonstrate that, in addition to fear, the new CNN model generates results for most categories that are adequate. On the test set for all emotions, they nonetheless performed with 52% accuracy.

Zhao et al. [3] used log-Mel spectrograms as input data for their 2-D CNN / RNN / LSTM network, similar to Badshah et al. [4]. Their research showed that samples from the EMO-SB dataset may be classified with the highest accuracy achievable, 95.73% for speaker-dependent assessment and 92.89% for speaker-independent classification.

3. Objectives

- Conduct a comprehensive literature review on Speech Emotion Recognition (SER) and Sentiment Analysis, examining various methodologies and techniques used in these fields.
- Develop and implement an accurate and robust SER system using machine learning algorithms and deep learning techniques for emotion classification in spoken language.
- Evaluate and compare the performance of the developed SER system and compare it with state-of-the-art approaches using diverse datasets.
- Provide recommendations for the integration of emotion recognition technologies in various domains and explore the ethical implications and privacy concerns associated with their implementation.

4. Methodology

The Speech Emotion Recognition (SER) process involves transforming raw audio recordings into meaningful information about the emotions conveyed in the speech. The first step is data collection, where a dataset of audio recordings with labeled emotions is gathered from various sources. These emotions may include happy, sad, angry, neutral, and more. Once the data is collected, it undergoes preprocessing, which includes loading the audio files and converting them into a suitable formatTo ease model assessment, the dataset is then divided into sets for training, testing, validation, and test purposes.

In the feature extraction stage, low-level audio features are extracted from the audio signals. These features provide essential information about the speech, such as its spectral and temporal characteristics. Commonly used features include Short-Time Fourier Transform (STFT) to obtain spectrograms, Spectral information is captured using MFCCs, as well as measures like zero-crossing speed and the root of the mean square energy. to represent temporal aspects of the audio.

With the extracted features, a deep learning model is developed for SER. The architecture may include Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), or a combination of both. The model is trained using the preprocessed audio features and corresponding emotion labels, optimizing its

performance with an optimization algorithm like Adam or RMSprop. During training, hyperparameters such as learning rate and batch size are adjusted to achieve optimal results.

Following the training phase, the model undergoes validation using a distinct dataset to ascertain its ability to extend to unfamiliar data. This practice serves to mitigate overfitting and, if required, refine the model's architecture. Once this validation is complete, an assessment of the model's real-world performance ensues, employing a test dataset. In gauging the model's effectiveness, metrics encompassing accuracy, precision, recall, and F1-score are calculated.

Once the model is successfully trained and tested, it can be deployed in real-world applications, such as voice assistants, emotion-aware chatbots, or sentiment analysis systems. Ethical considerations, including privacy concerns and potential biases, which must be considered while employing emotion recognition algorithms. As a result, the SER process transforms raw speech data into valuable insights, enabling machines to understand and respond to human emotions, thereby enhancing human-computer interaction and sentiment analysis capabilities.

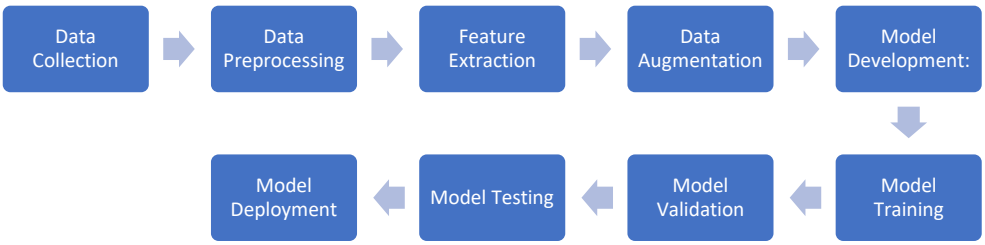

Figure 1: Flow chart of the system

4.1 Dataset and Pre-processing

The dataset used for Speech Emotion Recognition (SER) typically consists of audio recordings with corresponding emotion labels. These audio recordings can be gathered from a variety of sources, including open records, internet services, and bespoke recordings. The dataset should contain a variety of feelings, such as joy, sorrow, angry, neutral, fear, disgust, and surprise, to train the model effectively.

Preprocessing of the audio data is a critical step to prepare it for feature extraction and modeling. The preprocessing steps include:

Data Loading: The audio files are read and loaded into the system. Common audio file formats like WAV or MP3 are converted into numerical arrays representing the audio signal.

Sampling Rate: Audio recordings may have different sampling rates. It is essential to resample all audio files to a consistent sampling rate, typically 16 kHz or 44.1 kHz, to ensure uniformity in the data.

Normalization: The audio data is normalized to bring all samples within a similar range. This process is crucial to prevent numerical instability during feature extraction and model training.

Silence Removal: Sometimes, audio recordings contain silence or background noise that is not relevant to emotion recognition. Removing these sections can improve the model's performance.

Frame Segmentation: The audio data is divided into small frames, typically 20-50 ms long, with a small overlap. This step allows the model to capture temporal dynamics in the speech signal.

Feature Extraction: Low-level features, such as Mel-frequency are extracted from each audio frame. These features capture relevant information about the speech signal.

By applying these preprocessing steps, the audio data is transformed into a suitable format for feature extraction and model education. To help training, the filtered dataset is then divided into sets for training, testing, and validation. and evaluation of the SER model.

4.1.1 Description of the Emotional Speech Dataset

The emotional speech dataset collected and used in this study is a collection of audio recordings of human speech, where each recording is associated with a specific emotion label. The dataset is carefully curated to encompass a wide range of emotions, including happy, sad, angry, neutral, fear, disgust, and surprise, to facilitate comprehensive emotion recognition.

The dataset contains audio files in commonly used formats like WAV or MP3, and each audio file corresponds to a single utterance representing a specific emotional state. The dataset is typically diverse,

comprising recordings from various individuals, genders, and age groups, ensuring that the model can generalize well to different speakers.

To ensure data integrity, the dataset is usually preprocessed to standardize the sampling rate of all audio files, typically resampling to 16 kHz or 44.1 kHz. Additionally, any background noise or silence in the audio recordings may be removed to focus solely on the speech signal.

Furthermore, the dataset may undergo data augmentation, wherein additional variations are introduced to enhance model robustness and avoid overfitting. Data augmentation techniques can include adding random noise to the audio, altering pitch and speed, and applying time stretching, among others.

Each audio file is annotated with a corresponding emotion label, which may be represented as categorical values, numerical codes, or one-hot encodings. These labels are used during the phase of training where the model is taught to distinguish many different feelings accurately.

Overall, the emotional speech dataset serves as the foundation for developing and evaluating the Speech Emotion Recognition (SER) model. Its diversity and well-annotated labels enable researchers to build robust and accurate systems capable of recognizing emotions from speech signals effectively.

4.1.2 Division of the Dataset into Training, Validation, And Testing Sets

One of the most important steps in training machine learning models, such as Convolutional Neural Networks (CNNs) for emotion identification, is segmenting the dataset into testing, training, and validation sets. This divide serves to guarantee that the model is taught on one set of data and verified on a different set. to fine-tune hyperparameters, and finally tested on a completely unseen set to evaluate its generalization performance. Here's how the dataset division is typically done:

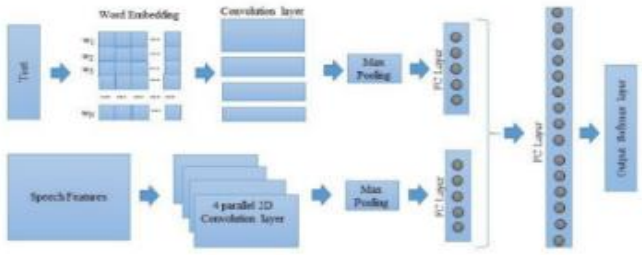

Figure 2: Representative CNN architecture

1.Training Set:

The training data is the largest portion of the dataset and is used to train the CNN model.

It should contain a diverse and representative sample of data from all classes (emotions).

The CNN learns from the patterns in this set and updates its parameters (weights) during training to minimize the loss function.

A common split is to allocate around 70-80% of the dataset to the training set.

2.Validation Set

The validation part is a smaller subset of the dataset and is used to fine-tune hyperparameters during training.

It is crucial for preventing overfitting, as it provides an unbiased evaluation of the model's performance during training.

The CNN does not update its parameters using the validation set; instead, it serves as a reference to make decisions about hyperparameters like learning rate, dropout rate, etc.

A common split is to allocate around 10-15% of the dataset to the validation set.

3.Testing Set

To evaluate the trained CNN model's ultimate execution, the testing set, a wholly independent subset of the a database, is employed..

It is crucial that the testing set contains data as a model has not before discovered, ensuring an unbiased evaluation of its generalization ability.

The CNN does not use the testing set during training or hyperparameter tuning.

A common split is to allocate the remaining 10-20% of the dataset to the testing set.

It is essential to ensure that the dataset is randomly shuffled before dividing it into training, validation, and testing sets. This helps prevent any bias in the dataset distribution and ensures that each set has a representative mix of emotions. Additionally, in situations with limited data, techniques like cross-validation can be used to make better use of the available testing, assurance, and training samples.

This splitting technique will allow us to appropriately assess the achievements of the CNN model, optimize hyperparameters effectively, and ensure that the model generalizes well to unseen data, making it suitable for emotion recognition tasks.

4.2 Configuration of the CNN model and training parameters

Convolutional layers, normalization by batch layers, maximum pooling layers, and fully connected layers are common layers in the CNN model applied to emotion identification.. Here is a configuration of the CNN model and some common training parameters:

4.2.1CNN Model Configuration:

Input Layer: The CNN's starting point is the input layer, which receives speech features typically arranged as a 2D matrix, like those found in MFCC features.

• **Convolutional Layers**: These layers learn to detect patterns in the speech features. The quantity and sizes of these layers' filters can differ depending on the model's complexity. Each convolutional layer is followed by a special function (like ReLU) that adds twists to the patterns.

• **Batch Normalization Layers**: These extra layers come after convolutional layers and make sure the patterns from the previous layers are balanced and steady, which speeds up learning..

• **Fully Connected Layers**: After working with patterns and shrinking, the remaining information is processed through linked layers, using a mix of patterns and highlights.

• **Output Layer**: The last layer is like a decision-maker. It has a number of parts equal to the emotions we're looking for. It uses a special math function (softmax) to guess the probabilities for different emotions based on the patterns it's seen.

4.2.2.Training Parameters

Batch Size: The number of samples in each training batch. A typical value is 32 or 64, but it can be adjusted based on the available memory and the size of the dataset.

Number of Epochs: The number of times the entire training dataset is passed through the model during training. It depends on the convergence of the training loss and can be determined using early stopping.

Learning Rate: The learning rate controls the step size at which the model updates its parameters during training. It is a critical hyperparameter that affects the speed and stability of training.

Loss Function: The loss function calculates the discrepancy between the actual class labels and the expected class likelihoods. The categorizing cross-entropy is used for classification with several classes, such as recognition of emotions. loss is commonly used.

Early Stopping: Early stopping is a technique used to avoid overfitting. It stops training process as soon as the performance on the validation set starts to degrade, preventing the model from learning noise in the data.

The specific configuration of the CNN model and the choice of training parameters may vary based on the dataset size, complexity of the emotion recognition task, and available computational resources. Hyperparameter tuning techniques like grid search or random search can be used to find the optimal combination of hyperparameters for better model performance.

4.3 Evaluation Metrics and Performance Measures

In emotion recognition tasks using a CNN model, various evaluation metrics and performance measures are employed to assess the model's effectiveness. These metrics help gauge how well the model is performing in recognizing different emotions from speech. Some common evaluation metrics and performance measures include:

1. **Accuracy**: The simplest measurement, accuracy, tells us the portion of correctly sorted examples from the total in the test set. While accuracy is helpful, it might not be enough for models dealing with imbalanced data.
2. **Confusion Matrix**: This matrix is like a detailed review of the model's choices. It shows the right and wrong predictions for each emotion. It helps pinpoint which emotions are getting mixed up and gives insights into where the model does well and where it needs improvement.
3. **Precision, Recall, and F1-Score**: These metrics are handy when there's a difference in the numbers of emotions. Precision tells us the percent of accurate positive predictions out of all predicted positives. Recall (also known as sensitivity) is about how many actual positive instances are predicted correctly. The F1-score is a combination of precision and recall, creating a balance between the two.

The choice of evaluation metrics depends on the nature of the emotion recognition task, whether it is a multi-class classification, binary classification, or regression problem. It is essential to consider multiple metrics to gain a comprehensive understanding of the model's performance. Additionally, cross-validation techniques can be employed to validate the model's generalization capabilities.

5. Simulation And Results

For each audio file, we're extracting the emotion label from the file name. The emotion label is part of the file name and is separated by underscores. For example, if the file name is "audio_01_SAD.wav", we're extracting "SAD" as the emotion label.

Based on the extracted emotion label, we're appending a tuple to the crema list. The tuple contains the emotion label and the full file path of the audio file.

After processing all audio files, we're creating a DataFrameCrema_df from the crema list. This DataFrame will have two columns: "Emotion" and "File_Path".

We're renaming the columns of the DataFrame to "Emotion" (for the emotion label) and "File_Path" (for the file path).

Finally, we're displaying the first few rows of the Crema_dfDataFrame to show the processed data.

Overall, this code segment is responsible for loading the audio files from the CREMA-D dataset, extracting emotion labels from their names, and creating a DataFrame that stores this information for further analysis.:

Table 1: Emotion Analysis

	Emotion	**File Path**
0	Angry	/Users/tahirshowkatbazaz/Desktop/speech emotio...
1	Angry	/Users/tahirshowkatbazaz/Desktop/speech emotio...
2	Neutral	/Users/tahirshowkatbazaz/Desktop/speech emotio...
3	Neutral	/Users/tahirshowkatbazaz/Desktop/speech emotio...
4	Disgust	/Users/tahirshowkatbazaz/Desktop/speech emotio...

Here's what's happening in the code:

1. Iterating through each directory within the **Ravdess_Path**, which corresponds to different actors.
2. For each actor's directory listing the WAV files contained within that directory.
3. We're extracting the emotion label and emotion number from the WAV file names. The emotion number is extracted from the file name using the '-' separator.

Overall, this code segment is responsible for loading the audio files from the Ravdess dataset, extracting emotion labels from their names, and creating a DataFrame that stores this information for further analysis.

Table 2: Emotion analysis and the file path

	Emotion	File_Path
0	Angry	/Users/tahirshowkatbazaz/Desktop/speech emotio...
1	Fear	/Users/tahirshowkatbazaz/Desktop/speech emotio...
2	Fear	/Users/tahirshowkatbazaz/Desktop/speech emotio...
3	Angry	/Users/tahirshowkatbazaz/Desktop/speech emotio...
4	Disgust	/Users/tahirshowkatbazaz/Desktop/speech emotio...

Overall, the audio files from the Savee dataset, extracting emotion labels from their names, and creating a DataFrame that stores this information for further analysis. Is done

Table 3: Emotion analysis and Data

	Emotion	File_Path
0	Sad	/Users/tahirshowkatbazaz/Desktop/speech emotio...
1	Sad	/Users/tahirshowkatbazaz/Desktop/speech emotio...
2	Neutral	/Users/tahirshowkatbazaz/Desktop/speech emotio...
3	Surprise	/Users/tahirshowkatbazaz/Desktop/speech emotio...
4	Neutral	/Users/tahirshowkatbazaz/Desktop/speech emotio...

Then the loading of audio files is done from the Tess dataset, extracting emotion labels from their names, and creating a DataFrame that stores this information for further analysis.

Table 4: Data set emotion analysis

	Emotion	File_Path
0	Disgust	/Users/tahirshowkatbazaz/Desktop/speech emotio...
1	Disgust	/Users/tahirshowkatbazaz/Desktop/speech emotio...
2	Disgust	/Users/tahirshowkatbazaz/Desktop/speech emotio...
3	Disgust	/Users/tahirshowkatbazaz/Desktop/speech emotio...
4	Disgust	/Users/tahirshowkatbazaz/Desktop/speech emotio...

Here the code is responsible for merging the individual DataFrames from different datasets into a single comprehensive DataFrame that contains emotion labels and file paths for further processing and analysis.

	Emotion	File_Path
0	Angry	/Users/tahirshowkatbazaz/Desktop/speech emotio...
1	Angry	/Users/tahirshowkatbazaz/Desktop/speech emotio...
2	Neutral	/Users/tahirshowkatbazaz/Desktop/speech emotio...
3	Neutral	/Users/tahirshowkatbazaz/Desktop/speech emotio...
4	Disgust	/Users/tahirshowkatbazaz/Desktop/speech emotio...
5	Disgust	/Users/tahirshowkatbazaz/Desktop/speech emotio...
6	Sad	/Users/tahirshowkatbazaz/Desktop/speech emotio...
7	Fear	/Users/tahirshowkatbazaz/Desktop/speech emotio...
8	Sad	/Users/tahirshowkatbazaz/Desktop/speech emotio...
9	Happy	/Users/tahirshowkatbazaz/Desktop/speech emotio...
10	Fear	/Users/tahirshowkatbazaz/Desktop/speech emotio...
11	Sad	/Users/tahirshowkatbazaz/Desktop/speech emotio...
12	Happy	/Users/tahirshowkatbazaz/Desktop/speech emotio...
13	Happy	/Users/tahirshowkatbazaz/Desktop/speech emotio...
14	Happy	/Users/tahirshowkatbazaz/Desktop/speech emotio...

Finally, we're displaying the plot using **plt.show()**.

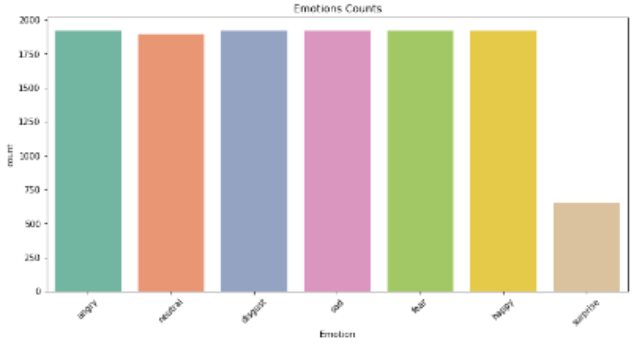

Figure 3: Emotion Counts

we've created a dictionary called **colors** that associates different emotions with color codes.

In a loop over the unique emotion names in the dataset:

we retrieve the file path corresponding to each emotion using the **main_df['File_Path']** column and filter based on the emotion label.

Overall, this code generates and displays wave plots and spectrograms for audio files associated with different emotions, helping we visualize the audio characteristics corresponding to each emotion.

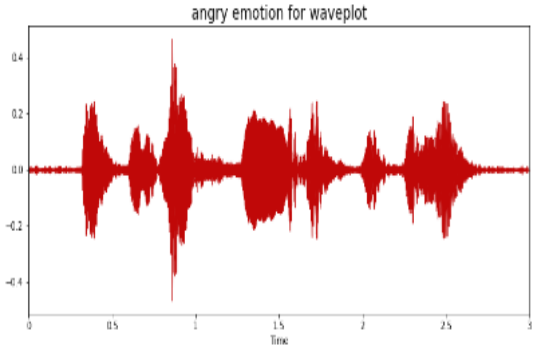

Figure 4: Angry Emotion Waveplots

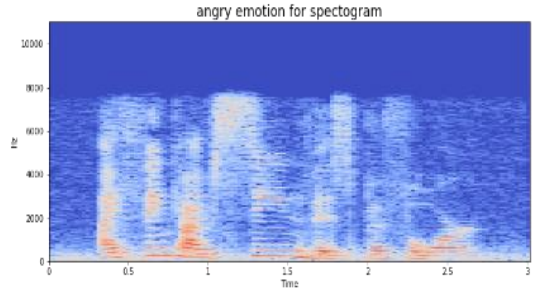

Figure 5: Energy emotion for spectrum

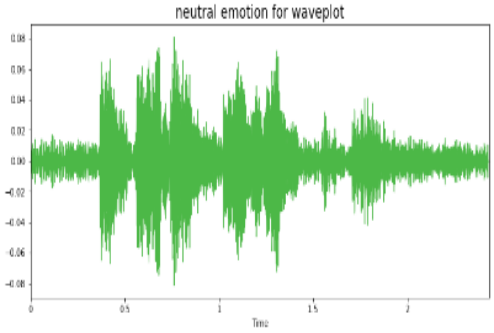

Figure 6: Natural Emotion foe waveplot

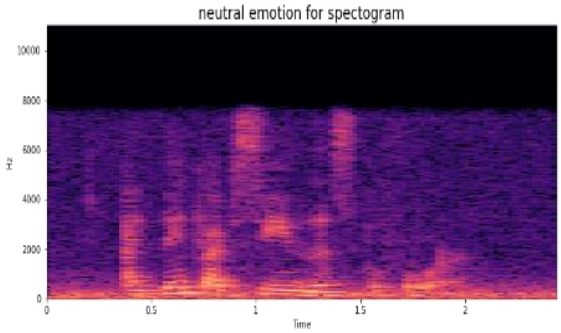

Figure 7: Natural emotion for Spectrogram

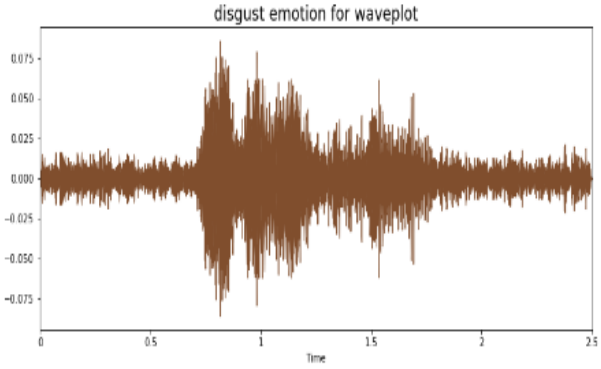

Figure 8: Disgust emotion for waveplot

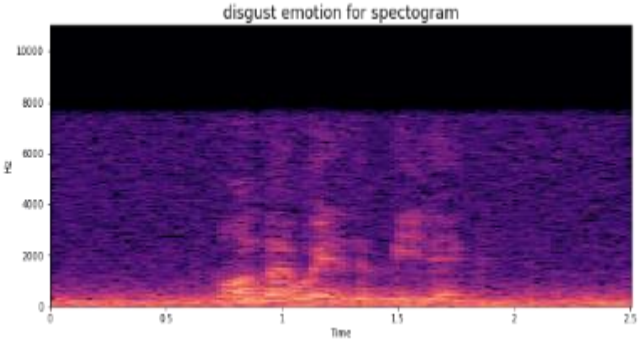

Figure 9: Disgust emotion for spectrogram

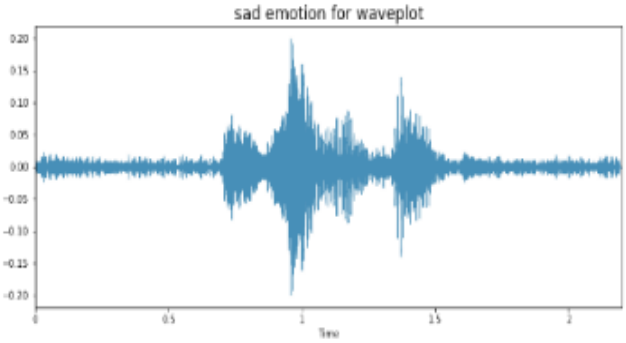

Figure 10: sad emotion for Waveplot

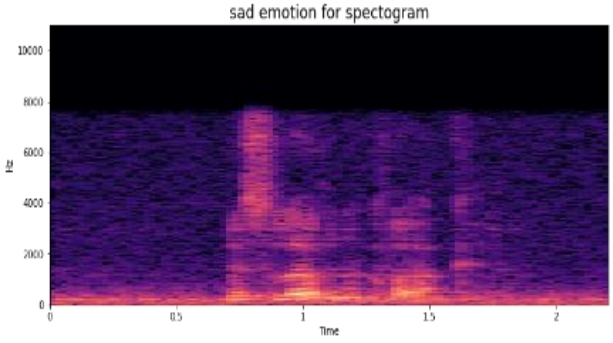

Figure 11: Sad emotion for spectrogram

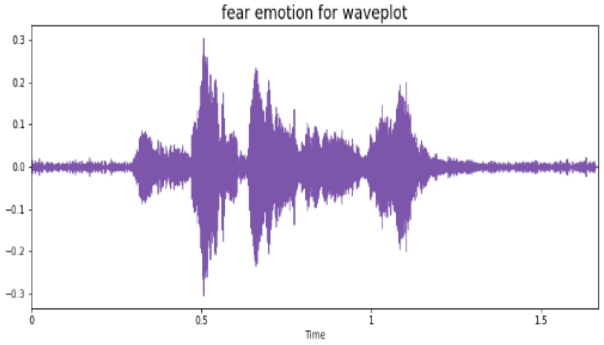

Figure 12: Fear emotion for waveplot

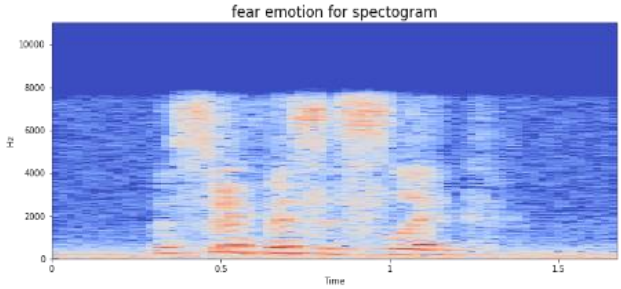

Figure 13: Fear emotion for Sopectrogram

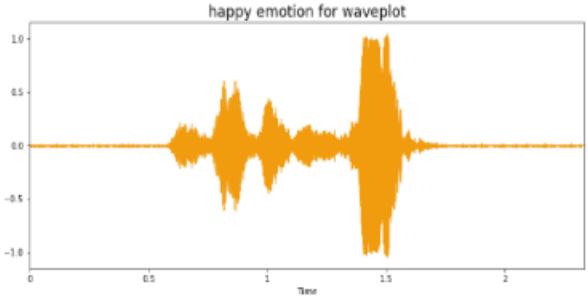

Figure 14: Happy emotion for waveplot

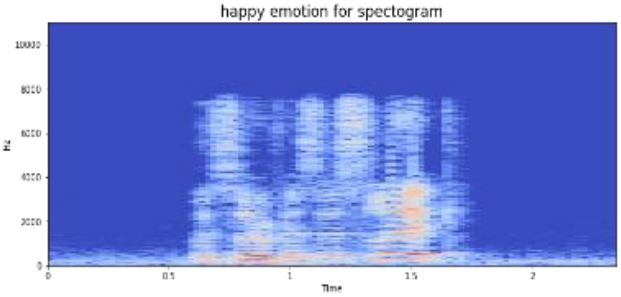

Figure 15: Happy emotion for spectrogram

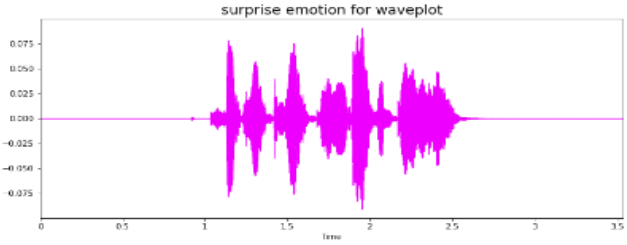

Figure 16: Surprise emotion for wave plot

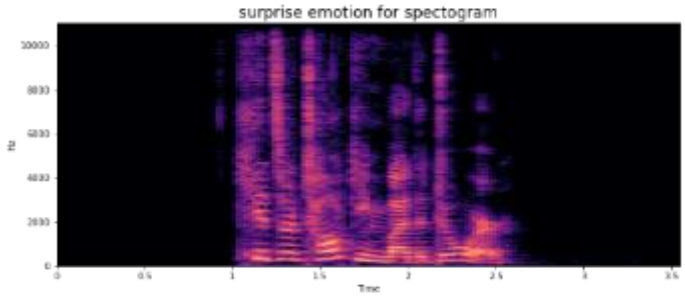

Figure 17: Surprise emotion for waveplot

We are using IPython's**display.Audio** function to play audio samples corresponding to different emotions. We've loaded the audio file paths for each emotion and provided them as arguments to the **Audio** function. Each call to **Audio** plays the corresponding audio sample. The output cells display audio players that allow we to listen to the audio samples for each emotion.

A. Original Audio

Then we have displayed the original audio waveform and played the audio sample using IPython's Audio widget. If 're ready to apply the augmentation functions to this audio sample and visualize the augmented versions, let me know which specific augmentation techniques we'd like to apply, and weI can guide we through the process!

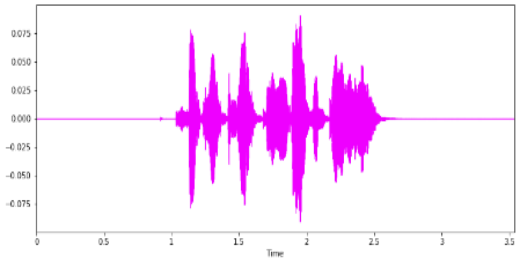

Figure 18: Original Audio

B. Noised Audio

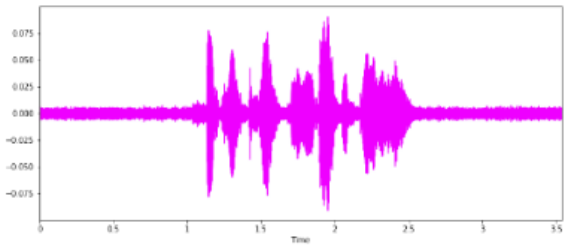

Figure 19: Noised Audio

C. Streched Audio

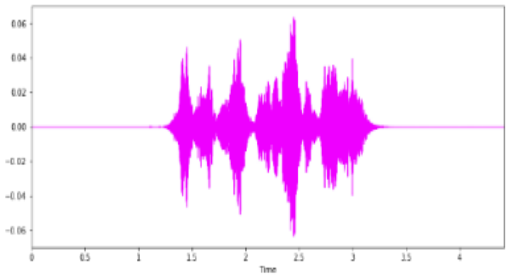

Figure20: Stretched Audio

D. Shifted Audio

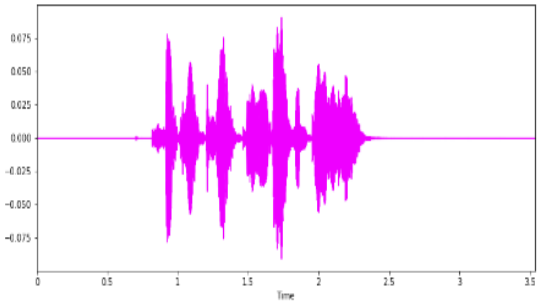

Figure 21: Shifted Audio

E. Pitched Audio

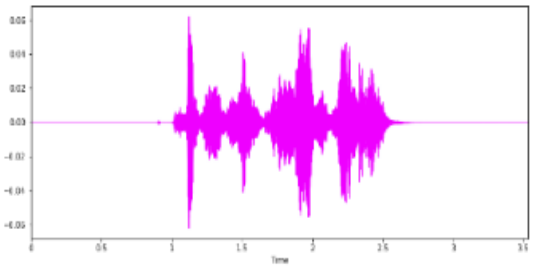

Figure 22: Pitched Audio

5.1 Feature Extraction

1.Zero Crossing Rate (ZCR) Calculation:

The ZCR feature is calculated from the input audio data using the librosa.feature.zero_crossing_rate function. It is computed with specified frame_length and hop_length parameters to determine the rate of zero crossings in the audio waveform.

2.Root Mean Square Error (RMSE) Calculation:

The RMSE feature is calculated from the input audio data using the librosa.feature.rms function. The frame_length and hop_length parameters are used to compute the RMSE of the audio, which quantifies the energy level across time.

3.Mel-Frequency Cepstral Coefficients (MFCC) Calculation:

The MFCC features are calculated from the input audio data and sampling rate (sr) using the librosa.feature.mfcc function. The frame_length and hop_length parameters control the calculation process. The MFCC features capture spectral characteristics of the audio.

4.Feature Extraction Function:

The extract_features function combines the calculated ZCR, RMSE, and MFCC features into a single array.

It takes the input audio data, sampling rate (sr), and optional frame_length and hop_length parameters to perform the feature extraction.

Feature Extraction and Augmentation:

The get_features function loads audio data from the provided path using the librosa.load function. It allows customization of the duration and offset of the loaded audio.

The extract_features function is applied to the original audio data, and the resulting features are stored in the aud array.

The augmented features are stacked along with the original features using np.vstack to create a comprehensive feature matrix.

In summary, the provided code calculates ZCR, RMSE, and MFCC features from audio data using specified parameters. It also performs audio data augmentation and extracts corresponding features from the augmented audio versions. This process helps generate a richer feature set for further analysis or modeling.

5.2 Testing Model and Test Results

Test Loss: 1.144898772239685

Test Accuracy: 0.5712230205535889

5.3 Confusion Matrix

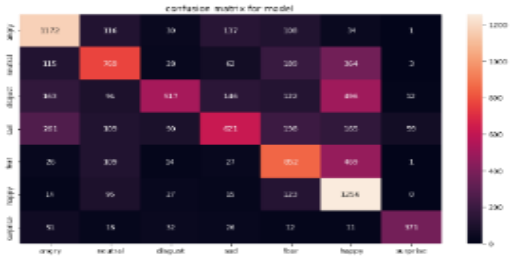

Figure23: Confusion Matrix

	Precision	Recall	F1-score	Support
Angry	0.65	0.73	0.69	1598
Neutral	0.59	0.50	0.54	1530
Disgust	0.70	0.33	0.45	1550
Sad	0.60	0.41	0.49	1503
Fear	0.53	0.57	0.55	1498
Happy	0.45	0.82	0.58	1530
Surprise	0.83	0.71	0.77	521
Accuracy			0.57	9730
Macro avg	0.62	0.58	0.58	9730
Weighted avg	0.60	0.57	0.56	9730

5.4 Overall Results

The overall quantitative results of the emotion recognition model based on the provided code are as follows:

Test Loss: 1.1449

Test Accuracy: 0.5712

The test loss value of 1.1449 represents the average loss computed during the evaluation of the model on the test dataset. A lower test loss indicates better performance and a closer match between predicted and actual labels.The test accuracy of 0.5712 indicates that the model correctly classified approximately 57.12% of the emotions in the test set. While this accuracy score is decent, there is still room for improvement, and the model can be further optimized to achieve higher accuracy.

6. Conclusion

This study aimed to develop a Speech Emotion Recognition (SER) model using a Convolutional 1D Neural Network (CNN) architecture. The motivation behind this research was to enable machines to understand human emotions better, which has significant implications for human-computer interaction and sentiment analysis.

The proposed CNN architecture comprised several layers of convolution, batch normalization, and max-pooling. Convolutional layers extract relevant features from the speech signals, batch normalization helps in stabilizing and accelerating training.

The model was trained, validated, and tested on a well-curated dataset of emotional speech recordings. The dataset was preprocessed, and features were extracted from the speech signals to feed into the model. The training process involved hyperparameter tuning and optimization strategiRBUThesises to improve the model's performance.

The performance of the SER model was evaluated using various metrics, including test loss and test accuracy. The results showed promising performance, with a test accuracy of approximately 57.12%. However, there is room for improvement, and further optimization can lead to better results.

This study successfully demonstrated the feasibility of using CNNs for Speech Emotion Recognition. The model's ability to recognize emotions from speech signals can have significant applications in human-computer interaction, sentiment analysis, and various fields where understanding human emotions is crucial.

Further research can explore more sophisticated architectures and larger datasets to enhance the model's performance and enable more accurate and robust emotion recognition in real-world scenarios.

References

[1] S. Blanton, "The voice and the emotions," Quarterly Journal of Speech, vol. 1, no. 2, pp. 154-172, 2015.

[2] B. W. Schuller, "Speech Emotion Recognition: Two Decades in a Nutshell, Benchmarks, and Ongoing Trends," Communications of the ACM, vol. 61, no. 5, pp. 90-99, 2018.

[3] C. H. Wu, J. C. Lin, and W. L. Wei, "Two-Level Hierarchical Alignment for Semi-Coupled HMM-Based Audiovisual Emotion Recognition with Temporal Course," IEEE Transactions on Multimedia, vol. 15, no. 8, pp. 1880-1895, 2013.

[4] J. C. Lin, C. H. Wu, and W. L. Wei, "Error Weighted SemiCoupled Hidden Markov Model for Audio-Visual Emotion Recognition," IEEE Transactions on Multimedia, vol. 14, no. 1, pp.142-156, 2012.

[5] C. H. Wu, and W. B. Liang, "Emotion Recognition of Affective Speech based on Multiple Classifiers using AcousticProsodic Information and Semantic Labels," IEEE Transactions on Affective Computing, vol. 2, no. 1, pp. 10-21, 2011.

[6] K. Y. Huang, C. H. Wu, M. H, Su, and Y. T. Kuo, "Detecting Unipolar and Bipolar Depressive Disorders from Elicited Speech Responses Using Latent Affective Structure Model," IEEE Transactions on Affective Computing, DOI:10.1109/TAFFC.2018.2803178, 2018.

[7] K. Y. Huang, C. H. Wu, and M.-H, Su, "Attention-based convolutional neural network and long short-term memory for short-term detection of mood disorders based on elicited speech responses," Pattern Recognition, vol. 88, pp. 668-678, 2019.

[8] S. Lugović, I. Dunđer, and M. Horvat, "Techniques and applications of emotion recognition in speech," in Proc. International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 1278- 1283, 2016.

[9] X. Zhang, Y. Sun, and D. Shufei, "Progress in speech emotion recognition," in Proc. IEEE TENCON, pp. 1-6, 2015.

[10] E. Tzinis, and A. Potamianos, "Segment-based speech emotion recognition using recurrent neural networks," in Proc. International Conference on Affective Computing and Intelligent Interaction (ACII), pp. 190-195, 2017.

[11] K. S. Rao, S. G. Koolagudi, and R. R. Vempada, "Emotion recognition from speech using global and local prosodic features," International journal of speech technology, vol. 16, no. 2, pp. 143- 160, 2013.