Journal of Advanced Research in Instrumentation and Control Engineering
Volume 10, Issue 2 - 2023, Pg. No. 13-20
Peer Reviewed Journal

Review Article

Speech Emotion Recognition Using 2D CNN LSTM

Abhishek B Udnur[1], Prathmesh M Babel[2], Akash P Gaikwad[3], Tushar S Satpute[4]

[1]Research Scholar, Department of Computer Science Engineering, Sharad Institute of Technology College of Engineering.
[2,3,4]Research Scholar, Department of Artificial Intelligence and Data Science, Sharad Institute of Technology College of Engineering.

I N F O

Corresponding Author:
Abhishek B Udnur. Research Scholar, Department of Computer Science Engineering, Sharad Institute of Technology College of Engineering.
E-mail Id:
abhishek.udnur1@gmail.com
Orcid Id:
https://orcid.org/0009-0009-0327-6882
How to cite this article:
Udnur AB, Babel PM, Gaikwad AP, et al. Speech Emotion Recognition Using 2DCNN-LSTM. *J Adv Res Instru Control Engi* 2023; 10(2): 13-20.

Date of Submission: 2023-10-31
Date of Acceptance: 2023-11-20

A B S T R A C T

This study focuses on Speech Emotion Recognition (SER) in the digital era, addressing mental well-being amidst extensive technology usage. SER is a critical tool impacting healthcare, entertainment, education, and more. The research explores diverse deep learning-based techniques for emotion detection in speech. However, the challenge of understanding abstract features in deep neural networks, a "black box" issue, persists. The study underscores SER's importance in comprehending digital human behavior and its potential in designing supportive media architectures. The dataset used, Ravdess Dataset, is described in detail. The implementation covers essential preprocessing steps using the librosa library, including data exploration and feature extraction like Mel spectrogram calculations, Fast Fourier Transform (FFT), Hamming window application, and Mel Frequency Cepstral Coefficients (MFCCs). It explains data augmentation, model building using a CNN-LSTM architecture, and model evaluation, achieving a high accuracy of 94.02% in emotion recognition. Deployment aspects discuss utilizing the trained model for emotion detection, emphasizing practical application through a Flask web framework. The discussion highlights the success of CNN-LSTM networks in extracting emotional information from speech signals. Techniques to combat overfitting and enhance model generalization are explored. The conclusion stresses the ongoing pursuit of higher accuracy in SER, suggesting avenues for future research, including novel network architectures and feature merging methods. The study provides a comprehensive insight into SER techniques and their potential in addressing mental well-being in the digital age.

Keywords: Emotions, Speech, CNN, LSTM, Librosa, Mel Frequency Cepstrum

Introduction

In the contemporary world, individuals tend to overlook their mental well-being due to the perpetual use of technology and the internet. Consequently, Speech Emotion Recognition (SER) has gained widespread popularity. This study aims to explore the diverse techniques and strategies employed in detecting emotions in speech while ensuring that no plagiarism occurs. The research conducted here is based on the analysis of existing literature and advancements in Deep Learning technology. Deep neural networks are typical "black box"[1] approaches because it is extremely difficult to understand how the final output is arrived at. The interpretability of how the highly abstracted features are learned by deep neural networks (DNNs) is poor. However deep neural networks perform dramatically better than traditional approaches in some experiments as roughly shown in figure 1. Researchers are finding that SER has a profound impact on various sectors, such as healthcare, entertainment, education, market research, security, and forensics. The insights gained from studying human behavior and activity patterns on the internet can be used to design media architecture that aids users in overcoming tough periods. This important work has the potential to benefit numerous individuals .

In an era characterized by an increasing reliance on virtual interactions and a decline in physical engagement, the establishment of meaningful emotional connections has become less likely. Consequently, a growing number of individuals are experiencing mental stress and loneliness within the digital realm .

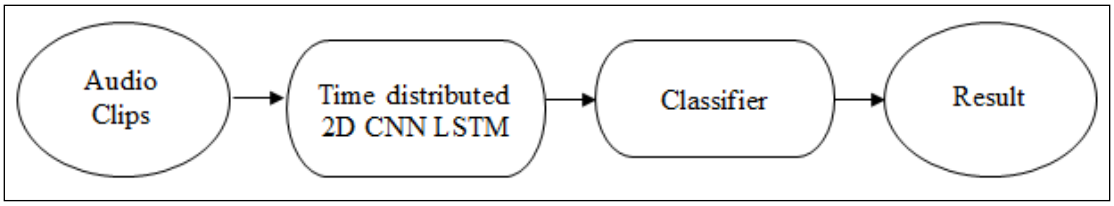

Figure 1

Employing voice analysis technology allows for the detection of emotional states, enabling the implementation of targeted measures to address these psychological challenges. The information presented here is a synthesis of various perspectives and findings from reputable sources, aiming to provide a holistic view of SER techniques. By understanding the evolving landscape of SER and the methods used to detect emotions, researchers can enhance their posture.

Dataset

The audio files used for training were sourced from www.kaggle.com. This dataset is known as the Ravdess Dataset and contains 1440 individual audio samples. These recordings were made by 24 different actors, each participating in 60 trials. Of the 24 actors, 12 were male and 12 were female.Overview of dataset is given in figure 2.

Each audio file has a unique file name describing modality, vocal channel, emotion, emotional intensity, statement, repetition, and actor.

Modality has three types 01, 02, and 03 which are for full-AV, video-only, and audio-only respectively.

Vocal channels have two types 01 and 02 which are for speech and song respectively. Emotion has eight types (01 = neutral, 02 = calm, 03 = happy, 04 = sad, 05 = angry, 06 = fearful, 07 = disgust, 08 = surprised) Emotional intensity has two types 01 and 02 for normal and strong respectively.

Two statements are there 01 for "Kids are talking by the door" and 02 for " Dogs are sitting by the door". Two repetitions are their 01 and 02 for the first and second respectively. Actors ranging from 01 to 24. (odd for males and even for females).

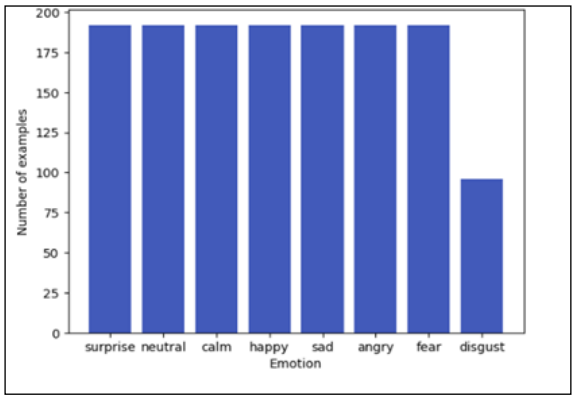

Figure 2

Implementation Progress and Outcome.

Preprocessing

Explore Data

After acquiring audio files, we can delve into the data using powerful Python libraries. One such library is librosa, which is specifically designed for processing audio and music. With librosa, we can perform a variety of tasks such as loading audio, extracting features, processing signals, and visualizing data as shown in figure 3. This makes it an essential tool for anyone working with audio data in fields such as research, machine learning, music information retrieval, etc.

Feature Extraction

To begin, we need to extract signals from the data. To accomplish this, we utilize Librosa, which converts 3 seconds of audio starting from an offset of 0.5 seconds and uses the specified sample rate. By stacking the NumPy arrays that represent audio along the 0th axis, a 2D array is created. Each row of this array corresponds to a signal, meaning the audio from a particular dataset.

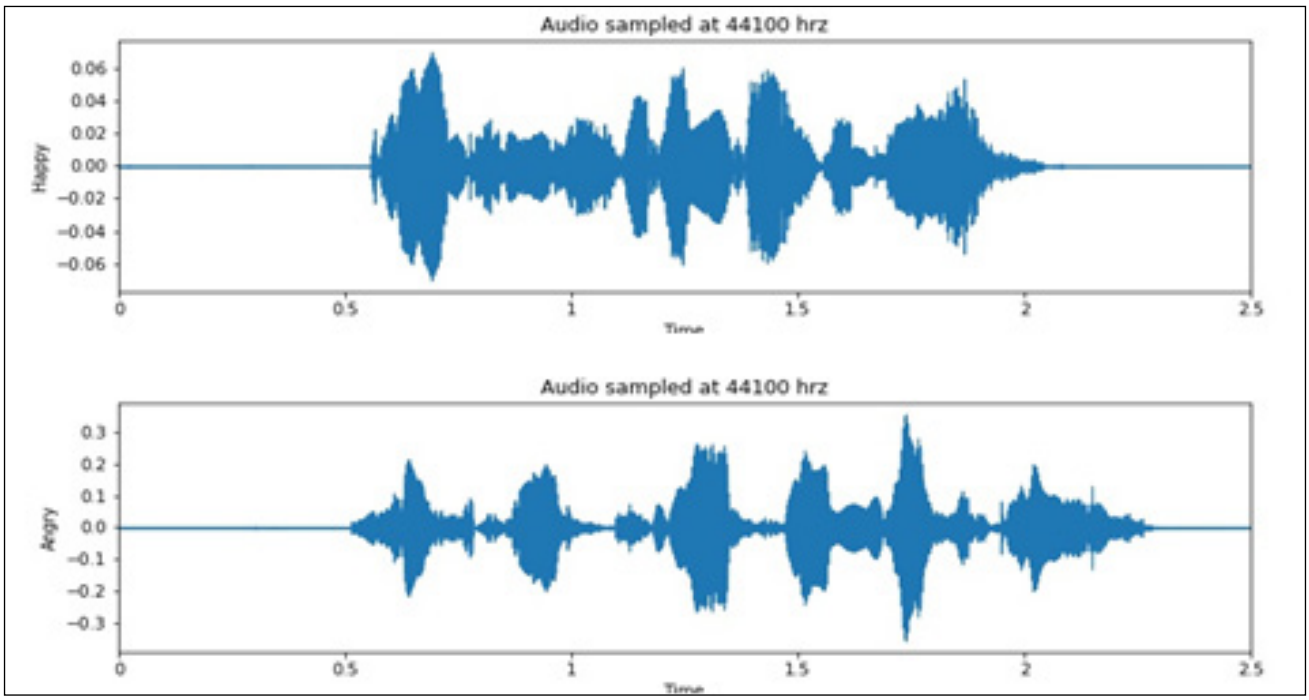

Figure 3. Wave representation of audios

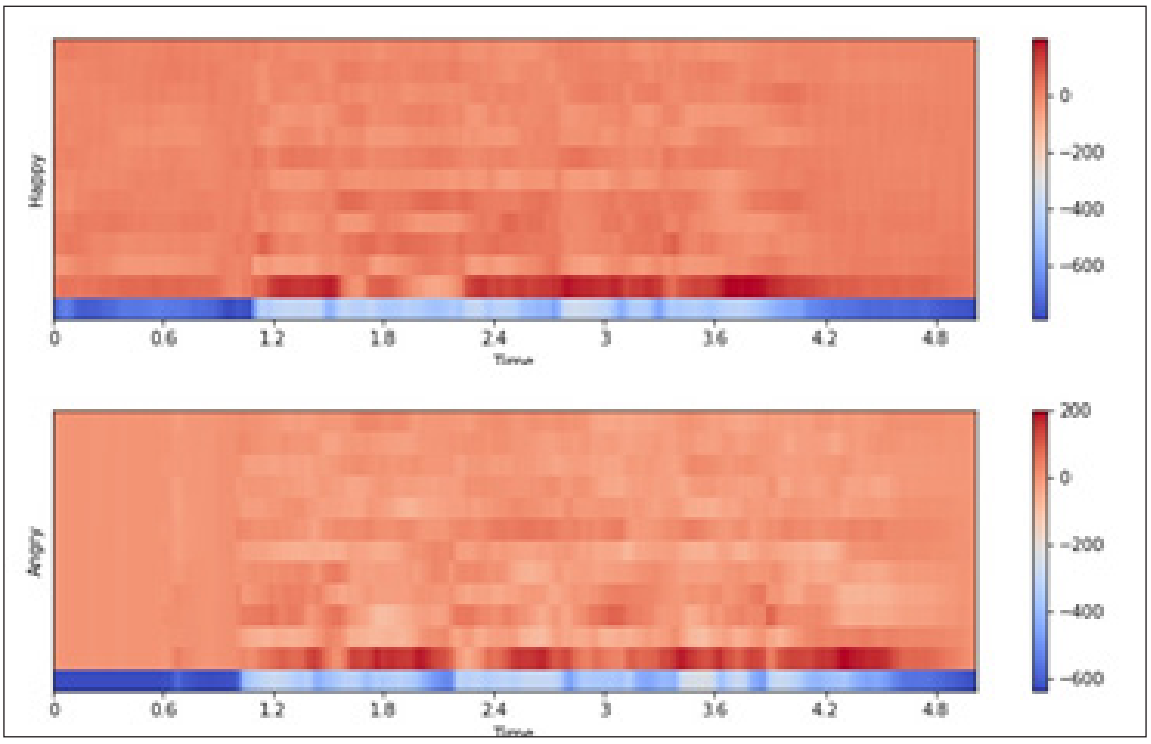

Figure 4.2D matrix of MFCC

After extracting features, they are categorized into three groups for training, testing, and validation. In order to improve the model's predictive abilities, the training set is enhanced with noise, which also increases the set's complexity and size.

In speech emotion recognition, the mel spectrogram of an audio signal is calculated using the audio signal and its corresponding sample rate.[2,3] This involves specifying parameters like the number of FFT (Fast Fourier Transform) points, window length, window type (Hamming), hop length, number of mel bands, and maximum frequency. [4,3] After generating the mel spectrogram, it is converted to decibel units using a reference value of the maximum power in the spectrogram. To process and analyze Mel spectrograms in a more manageable way, they are divided into smaller, overlapping chunks or frames. This is done by considering two important parameters: stride and window size. The window size determines the size of each chunk, while the stride or step size moves the window along the spectrogram. By stacking all the valid chunks, a NumPy array is obtained that contains all the smaller, overlapping frames extracted from the Mel spectrogram.[5] This technique is common in audio processing for segmenting data to facilitate various tasks such as feature extraction and machine learning.

The Fast Fourier Transform (FFT) is a useful algorithm that converts signals from their original domain into the frequency domain and vice versa. While the Discrete Fourier Transform (DFT) can decompose a sequence of values into components of different frequencies, it can be too slow in practice. Fortunately, the FFT is faster in practice because it computes these transformations by factorizing the DFT matrix into a product of sparse factors.

DFT is defined by the formula,

$$X_k = \sum_{n=0}^{N-1} x_n \, e^{-i2\pi kn/N} \quad k = 0, \ldots, N-1,$$

where is a primitive root of 1.

By far the most commonly used FFT is the Cooley–Tukey algorithm. This is a divide-and-conquer algorithm that recursively breaks down a DFT of any composite size into many smaller DFTs.

$$H(n) = \alpha + (1.0 - \alpha)\, cos[(\frac{2\pi}{N})n]$$

The Hamming window is an extension of the Hann window in the sense that it is a raised cosine window of the form

$$H(\theta) = 0.54 + 0.46\, cos\left[\left(\frac{2\pi}{N}\right)n\right]$$

The common approximation to this value of α is 0.54, for which the window is called the Hamming window and is of the form

$$H(\theta) = 0.54 + 0.46\, cos\left[\left(\frac{2\pi}{N}\right)n\right]$$

The Mel scale is a perceptual scale of pitches that approximates the way the human ear responds to different frequencies of sound. It is a non-linear scale commonly used in audio processing applications, particularly in the fields of speech and audio signal processing. The scale is named after the word "melody" and is intended to replicate the way humans perceive variations in pitch.

The formula for approximating the conversion from Hz to Mel scale is,

$$Mel(f) = 2595\, log_{10}\left(1 + \frac{f}{100}\right)$$

The Mel Frequency Cepstral Coefficients (MFCCs) [5] are a series of audio signal processing steps that mimic how the human ear perceives sound frequencies. Firstly, a pre-emphasis filter enhances high-frequency components and improves the signal-to-noise ratio. The audio signal is then divided into overlapping frames and a window function, such as the Hamming window, is applied to each frame.

Next, a Fourier transform converts each frame from the time domain to the frequency domain, providing a representation of the signal's frequency components. Mel filters, spaced according to the Mel scale, are applied to the magnitude spectrum obtained from the Fourier transform. These filters compute the energy in various Mel frequency bands.

The energy in each Mel filter bank is then logarithmically computed to approximate the logarithmic perception of loudness by the human ear. A Discrete Cosine Transform (DCT) is applied to the logarithmically transformed filter bank energies, resulting in a set of cepstral coefficients. A subset of these coefficients is retained as the final MFCCs, representing critical spectral characteristics of the audio signal.

$$c_i \sum_{n=1}^{N_f} s_n \; cos\, cos\left[i(n-0.5)\left(\frac{\Pi}{N_f}\right)\right], \; i = 1, 2, \ldots\ldots, L$$

where $c_i = c_y(i)$ = MFCC coefficient, N_f is the number of triangular filters in the filter bank, Sn is the log energy output of the filter coefficient and L is the number of MFCC coefficients that we want to calculate.

These MFCCs are widely used in speech recognition[6], speaker identification, music genre classification, and other audio processing contexts, thanks to their robustness and effectiveness. One advantage of considering MFCCs as an image (as shown in figure 4) is that they offer additional information and enable transfer learning. This is a valid approach that can lead to high accuracy. Nonetheless, studies have demonstrated that statistics associated with MFCCs (or any other time or frequency domain) can also contain significant information. MEL spectrogram is calculated and used as an input for the models For the model the spectrogram is split into 7 chunks.

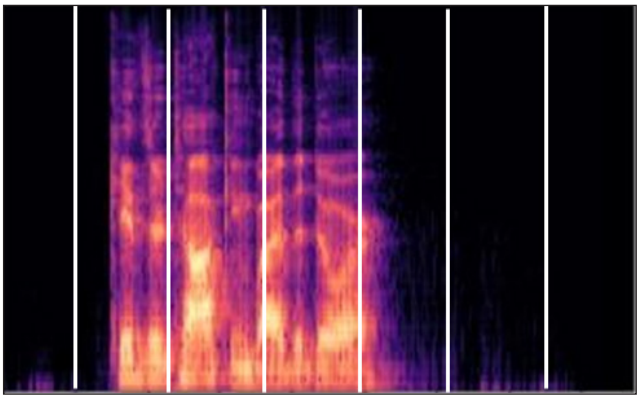

Figure 5.Sectrogram split into chunks

An example of the MEL spectrogram (split into 7 chunks) is given in figure 5.

Data Augmentation

To improve accuracy and reduce overfitting, we added Additive White Gaussian Noise to the original signal. The noise was generated with a signal-to-noise ratio (SNR) between 15 and 30, which proved highly effective in improving accuracy and addressing overfitting issues in the dataset. Prior to adding the noise, we preprocessed the datasets by scaling them using the Standard Scaler method. This technique standardized the dataset by subtracting the mean and scaling the data to have a unit variance, ensuring that the data was appropriately prepared and standardized before the noise addition step.

Model Building and Evaluation

Time Distributed Layer

The Time Distributed layer enables the application of a layer to each temporal slice of an input. Essentially, it independently applies the same layer to every time step in a sequence of data. This functionality is especially valuable when working with sequential data in tasks like Natural Language Processing (NLP), time series forecasting, and video processing. If there are more than 2 dimensions in the input data, indicating a time dimension, the data is modified by collapsing the time dimension into a single axis. This allows each time step to be treated as a separate "sample" for the relevant module. after the module computation input is reshaped back to the original dimensions.

Model Architecture

The model architecture is utilized for emotion recognition. It combines convolutional and LSTM layers to extract features efficiently and model temporal data.[4] The model comprises two main blocks, the Convolutional Block and the LSTM Block as shown in . figure 6. The Convolutional Block has multiple layers, each with batch normalization, ReLU activation, max-pooling, and dropout, organized into four blocks with varying channels and kernel sizes. This

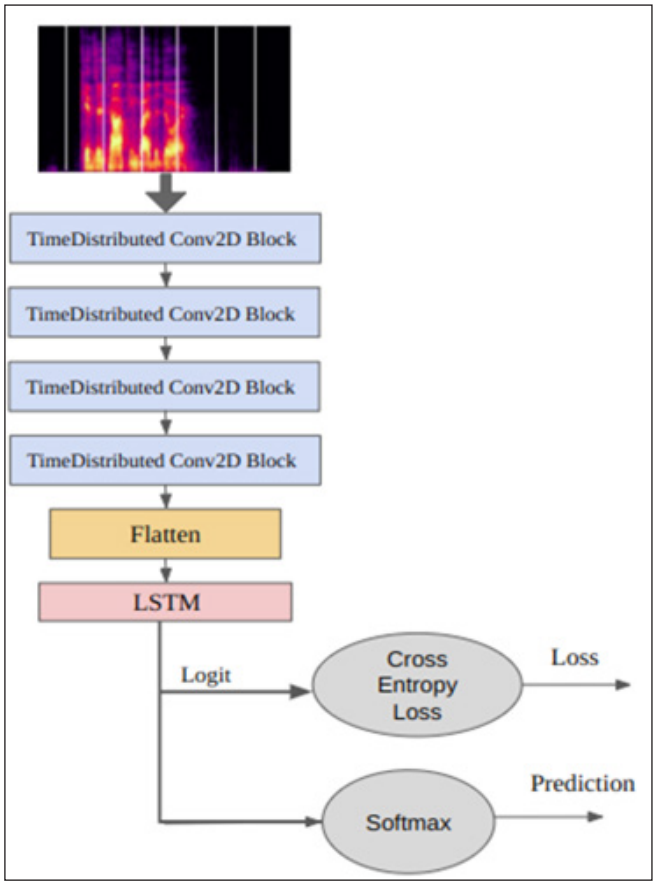

Figure 6.Model Architecture

allows the model to extract features from input data effectively. The LSTM Block uses a unidirectional LSTM layer with 128 input size and 64 hidden size, processing input sequences in batches. After convolutional feature extraction, the output is flattened and passed through the LSTM layer, which incorporates dropout for regularization [2] and captures temporal dependencies in the data. The LSTM output from the last time step is then extracted and fed through a linear softmax layer, which maps the LSTM output to the specified number of output emotions. The resulting output provides both logits and softmax probabilities, serving as predictions for the respective emotions. Overall, this model is a structured approach that efficiently combines convolutional and LSTM architectures to recognize emotions through feature extraction and temporal analysis.

The convolutional part of the model processes 4D input data representing time-distributed 2D grayscale images. Each convolutional block within this part follows a specific pattern. Initially, a 2D convolution is applied, generating 16 output channels using a 3x3 kernel while preserving the original spatial dimensions through padding. Batch normalization is then employed to normalize the convolutional output, followed by the application of a Rectified Linear Unit (ReLU) activation function for introducing non-linearity.

Subsequently, max pooling with a 2x2 kernel and stride 2 is performed to reduce the spatial dimensions of the output. To prevent overfitting, a dropout operation with a rate of 0.4 is applied for regularization. This convolutional block pattern is iteratively repeated, with the number of output channels increasing through subsequent stages (16, 32, 64, 128), detailed overview is given in table 1.

This integration of convolutional layers for spatial feature extraction and LSTM layers for temporal sequence analysis enables the model to effectively analyze spatiotemporal data or sequential patterns, making it highly suitable for a wide array of tasks necessitating this combined approach (table 2).

Tabe 1.Summary of CNN nets

Layer	Input Size (channels x h x w)	Output Size	Kernel Size	Stride
Conv2D	1x128x128	16x128x128	3x3	1
MaxPool2D	16x128x128	16x64x64	2x2	2
Conv2D	16x64x64	32x64x64	3x3	1
MaxPool2D	32x64x64	32x16x16	4x4	4
Conv2D	64x16x16	64x16x16	3x3	1
MaxPool2D	64x16x16	64x4x4	4x4	4
Conv2D	64x4x4	128x4x4	3x3	1
MaxPool2D	128x4x4	128x1x1	4x4	4

The LSTM component of the model operates on the features extracted by the convolutional layers. Initially, it receives 3D input data, representing the output from the convolutional part, with each sample containing 128 features. This input is fed into an LSTM layer configured with 128 input features and 64 hidden units (as shown in table 2), emphasizing sequence processing and capturing temporal dependencies. The LSTM layer is set to operate in a batch-first manner, facilitating efficient batch processing. For regularization and to mitigate overfitting, a dropout layer is applied to the LSTM output with a dropout rate of 0.3.Finally, the LSTM output is passed through a linear layer, transforming the input features of 64 (output from the LSTM) to an output with 8 features, tailored for the specific task at hand.

Model Evaluation

Upon training the provided architecture, the model demonstrated a high level of effectiveness, achieving an impressive accuracy of 94.02%. To comprehensively evaluate its performance, various metrics including a confusion matrix and an accuracy graph were utilized. The confusion matrix, shown below, offers a detailed breakdown of the model's predictions compared to the ground truth across different classes.

Additionally, the training and validation accuracies were plotted on a graph to visualize the model's learning progress during training which is shown in figure 7.

Table 2.Summary of LSTM Nets

Layer	Input Size	Output Size
LSTM	128	64
Dropout	64	64
Linear	64	8

Tabe 3.Confusion Matrix

Sur	56	1	0	0	0	0	0	1
Neul	0	28	1	0	0	0	0	0
Calm	0	2	49	0	7	0	0	0
Hap	0	1	0	57	0	0	0	0
Sad	0	0	0	2	56	0	0	0
Ang	0	0	0	0	0	58	0	0
Fea	1	0	0	0	6	2	49	0
Disg	0	0	0	0	0	2	0	56
	Sur	**Neu**	**Calm**	**Hap**	**Sad**	**Ang**	**Fea**	**Disg**

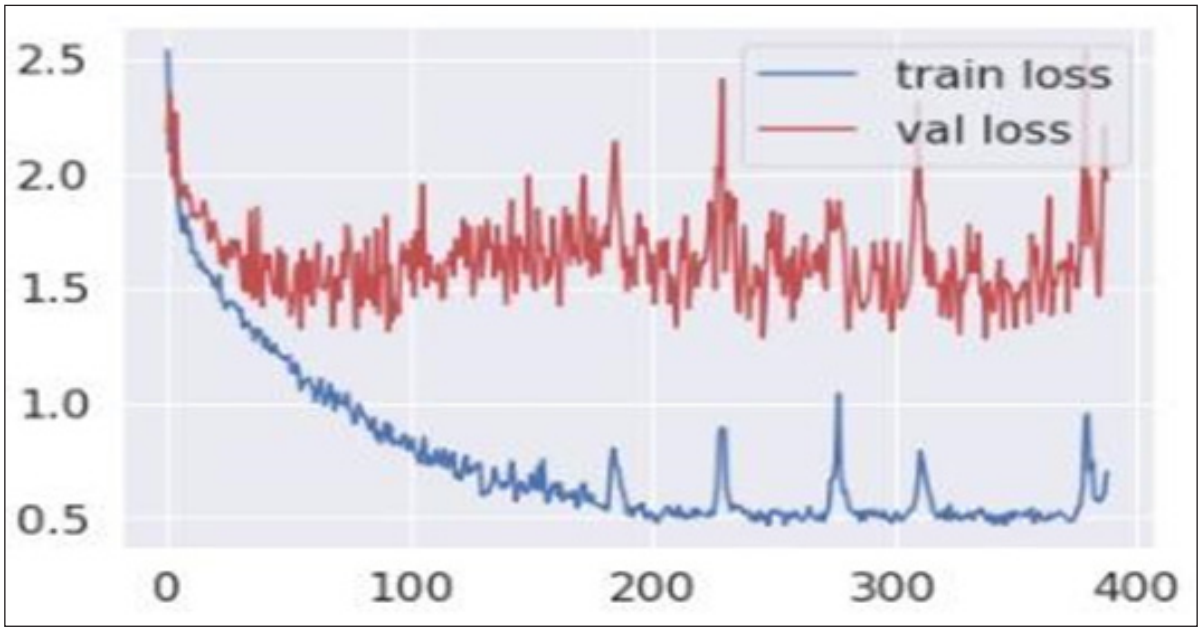

Table 7.Training and Validation Loss

Various statistical operations are performed to estimate the performance of a model from which some of them are given here.

Deployment

One way to utilize the trained model for detecting or classifying emotions is by saving it for deployment on various websites and software. For instance, we have implemented a solution using Python's Flask web framework. This solution allows users to upload audio files and receive a prediction based on the model's analysis.

Discussion

The work focuses on developing Stacked Time-distributed 2D CNN-LSTM networks, which comprise four CNN and one LSTM layer. These networks are designed to identify emotional features from speeches, which are time-varying signals that require sophisticated analysis to reflect their properties. The combination of CNN and LSTM layers is leveraged to recognize the emotional state of the speaker .

The experiments have successfully attained the objective of extracting emotional information from the experimental data. However, it is crucial to investigate how to establish the causal relationship between acted emotions and audio features. The designed networks have learned many causal features from the data regarding the underlying mechanism and have achieved high accuracy in recognizing emotions during experiments. Thus, to some extent, the mechanism has been deduced from the data rather than being an

exact form of an assumed algorithm. The similar prediction performances of the designed networks during extensive experiments demonstrate their effectiveness in recognizing speech emotions.

In this experimental study, several techniques were used to reduce overfitting in deep neural networks for speech emotion recognition, specifically in CNN LSTM networks. The model was regularized, which imposed smoothness constraints by penalizing layer parameters and activities during optimization. Incorporating Batch Normalization (BN) layers allowed input features to be considered in conjunction with others within each batch, improving generalization, accelerating training, and reducing overfitting.[2] Cross-validation techniques were used to prevent overtraining and improve generalization. Model selection was used to choose the best-fitted models based on validation accuracy, ensuring superior predictive performance. Despite these measures, overfitting was not entirely eliminated, and additional efforts were needed to address this remaining challenge. The training accuracies consistently exceeded validation accuracy, indicating that the models learned some random or emotion-related features from the training data.[7]

Conclusion

This paper represents a significant step in the development of models for Speech Emotion Recognition. In an age marked by digital connections, the ability to understand and address emotional well-being through technology is of paramount importance. Our model, based on Stacked Time-distributed 2D CNN-LSTM architecture, holds the potential to be a valuable tool in various fields, including mental health support, entertainment, education, and security. The inclusion of data preprocessing techniques such as Mel spectrogram and MFCCs, coupled with data augmentation, has improved the model's accuracy and ability to detect emotions from speech. The convolutional and LSTM layers have effectively extracted features and captured temporal dependencies, contributing to the model's robust performance. As future work, further validation and testing on diverse datasets should be conducted to ensure the model's generalizability. Additionally, exploring real-time applications and integrating the model into systems aimed at enhancing emotional well-being is a promising avenue for future research. In a world where digital connections are prevalent, the development of models for Speech Emotion Recognition is an essential step towards creating meaningful emotional connections and addressing psychological challenges in the digital realm. This work has the potential to benefit a wide range of individuals and has significant implications for the advancement of technology-driven emotional support systems. In conclusion, the Stacked Time-distributed 2D CNN-LSTM model developed in this research represents a significant advancement in the field of speech emotion recognition. By leveraging a combination of convolutional and LSTM layers, this model has demonstrated its ability to effectively identify and classify emotions in speech. The high accuracy achieved during training and evaluation further supports its efficacy in recognizing diverse emotional states. The real-world deployment of this model, exemplified by the Flask-based web solution, opens doors to practical applications in various domains. It offers the potential for enhancing user experiences in fields such as mental health support, entertainment, and security, where emotion recognition plays a crucial role. While this research marks a significant milestone, there is room for future investigations to delve deeper into the underlying mechanisms of emotion recognition from audio data.

References

1. Z.C. Lipton, The Mythos of Model Interpretability, arXiv preprint arXiv: 1606.03490, 2016.
2. A. Neumaier, Solving ill-conditioned and singular linear systems: a tutorial on regularization, Siam Rev. 40 (3) (1998) 636–666.
3. L. He, M. Lech, N.C. Maddage, N.B. Allen, Study of empirical mode decomposition and spectral analysis for stress and emotion classification in natural speech, Biomed. Signal Process. Control 6 (2) (2011) 139–14
4. Zhao, J., Mao, X., & Chen, L. (2019). Speech emotion recognition using deep 1D & 2D CNN LSTM networks. Biomedical signal processing and control, 47, 312-323.
5. Mel-frequency cepstrum. (2023, September 4). In Wikipedia. https://en.wikipedia.org/wiki/Mel-frequency_cepstrum
6. Emotion recognition. (2023, October 4). In Wikipedia. https://en.wikipedia.org/wiki/Emotion_recognition
7. Z. Huang, M. Dong, Q. Mao, Y. Zhan, Speech Emotion Recognition Using CNN, ACM Multimedia, 2014, pp. 801–804.