Speech emotion recognition based on CNN+LSTM Model

Peiyan Zheng
Maynooth
International Engineering Collage,
Fuzhou,China
PEIYAN.ZHENG.2022@MUMAIL.IE

Hongming Chen
Maynooth
International Engineering Collage,
Fuzhou,Chinay
HONGMING.CHEN.2022@MUMAIL.IE

Shujie Xu
Maynooth
International Engineering Collage,
Fuzhou,Chinay
SHUJIE.XU.2022@MUMAIL.IE

Chongzheng Lin
Maynooth
International Engineering Collage,
Fuzhou,China
CHONGZHENG.LIN..2022@MUMAIL.IE

Zhenxiang Sun
Maynooth
International Engineering Collage,
Fuzhou,China
ZHENXIANG.SUN.2022@MUMAIL.IE

Longxuan Liao
Maynooth
International Engineering Collage,
Fuzhou,China
LONGXUAN.LIAO.2022@MUMAIL.IE

***Abstract*—The rapid development of computer technology and artificial intelligence is impacting people's daily lives, and the convenience brought by AI products is ubiquitous. People's expectations for AI products are also increasing. Language is the most common means of communication in people's daily lives, and speech is the acoustic manifestation of language, containing a wealth of emotional information. Therefore, analyzing the emotional information in speech signals and applying it to AI products is a research hotspot in speech emotion recognition. This study utilizes a CNN+LSTM neural network structure to implement Speech Emotion Recognition (SER) and perform predictions. The experimental results demonstrate that the CNN+LSTM model performs well in recognizing multi-class emotions, achieving better performance compared to some traditional models. Experimental verification shows that this method can significantly improve the recognition rate of speech emotion.**

***Keywords*—Speech Emotion Recognition, Speech Feature Extraction, CNN, LSTM.**

I. INTRODUCTION

Speech emotion recognition is currently a research hotspot. As a key technology in human-computer interaction systems, speech emotion recognition can accurately identify emotions and help machines better understand users' intentions, thereby improving the quality of human-computer interaction. As one of the main ways of daily communication, speech contains rich emotional information. Therefore, speech emotion analysis is very important. Its application value in the field of artificial intelligence is increasingly prominent. Finding feature parameters that accurately represent speech emotion states and effective models for emotion recognition have always been the main challenges in speech emotion recognition. This study utilized spectrograms as input and overcame the errors caused by traditional algorithms in extracting emotional feature vectors. The CNN+LSTM neural network structure was used to implement Speech Emotion Recognition (SER), and the CASIA Chinese emotional speech corpus was utilized for the experiments. The spectrogram was input into three CNN layers, ReLU activation layers, and max-pooling layers to extract local features from the spectrogram. Subsequently, the flatten layer was used to flatten the output of the convolutional layers, followed by training through two LSTM layers to handle time-series data. Finally, the output was fed into the fully connected layer to obtain all features, and emotion recognition was performed using the softmax function, resulting in the final emotion classification.The CNN-LSTM model was evaluated and tested using a test set. The experimental results demonstrated that our approach performed well in recognizing multi-class emotions, achieving an average recognition rate of 79% across six emotions (anger, fear, happiness, neutral, sadness, surprise).

II. TECHNICAL BACKGROUND

In the field of human-computer interaction, speech is the primary focus of emotion recognition systems, along with facial expressions and gestures. Speech is considered a powerful mode of communication for conveying intent and emotions. In recent years, many researchers have conducted extensive research on recognizing human emotions using speech information and have explored various classification methods, including neural networks, Gaussian Mixture Models (GMM), Hidden Markov Models (HMM), Maximum Likelihood Classification (MLC), kernel regression, k-Nearest Neighbors (KNN), and Support Vector Machines (SVM).

The CNN+LSTM model is a neural network architecture that combines convolutional neural networks (CNN) and long short-term memory (LSTM) networks. The CNN is used to extract features from the input data, while the LSTM is used to model the temporal dependencies in the data.

The CNN+LSTM model is commonly used in tasks that involve time-series data, such as speech recognition, natural language processing, and video analysis. In speech recognition, the CNN is used to extract features from the audio signal, and the LSTM is used to model the temporal dependencies in the speech signal. It has shown promising results in various applications, particularly in tasks that involve sequential data. It has been used in speech recognition, sentiment analysis, image captioning, and many other tasks.

TABLE I. REVIEW OF RESEARCH

Author	**Review of Research**
	Article content
Williams and Stevens	In 1972, they conducted one of the earliest studies on the emotional information contained in speech. They found that variations in fundamental frequency in speech signals are related to emotions.
Dellaert et al.	The pitch contour parameters in the speech signal were extracted for recognition, using three different classification algorithms: k-nearest neighbors (KNN) method, kernel regression method, and maximum likelihood Bayesian classification method. These algorithms were used to recognize four emotions in speech, achieving a maximum recognition rate of approximately 68%. [1]

Author	Review of Research
	Article content
Valery A. Petrushin	Analyzed 700 short sentences composed of five emotions (happiness, sadness, anger, fear, and neutral). Four emotional feature parameters (speech rate, formants, fundamental frequency, and energy) were extracted, and different algorithms were employed for classification and recognition. The study achieved a recognition rate of approximately 70%. [2]
Lee et al.	Divided speech emotions into two categories: positive and negative. They used two algorithms, linear discriminant classification and kernel regression, and compared their results with Petrushin's experiments. [3]
Eyben et al.	Enables the automated batch extraction of speech emotion feature parameters. [4] It includes commonly used emotional feature parameters such as amplitude, fundamental frequency, and Mel-frequency cepstral coefficients (MFCC). It has gradually gained widespread recognition in the field. [5]

Fig. 1. Review of Research

III. Design Methodology

The specific research method is as follows: (1) Signal acquisition. Based on the mechanism of speech emotion signal generation and the international classification methods for emotions, typical emotion speech databases at home and abroad were comprehensively compared to collect and prepare speech data sets with emotional labels. (2) Pre-processing. The raw speech signal undergoes pre-processing steps including sampling, quantization, pre-emphasis, framing, and windowing to transform the speech signal into a spectrogram or mel spectrogram, serving as the input for a CNN. (3) Speech emotion recognition model building. Utilizing CNN to extract spatial features from the speech spectrogram. CNN is employed to capture local features of the spectrogram, aiding in the identification of emotional states in speech. The feature sequence extracted by CNN is then input into an LSTM network. LSTM is capable of capturing the temporal characteristics of the speech signal, assisting the model in understanding emotional changes within the speech signal. A fully connected layer is appended to the output of the LSTM, followed by a softmax layer for emotion classification, predicting the emotional state of the speech signal. (4) Speech emotion recognition model training and testing. Train the CNN-LSTM model using a randomly selected 80% of the speech dataset with associated labels. Utilize the remaining 20% of the dataset for model evaluation and testing.

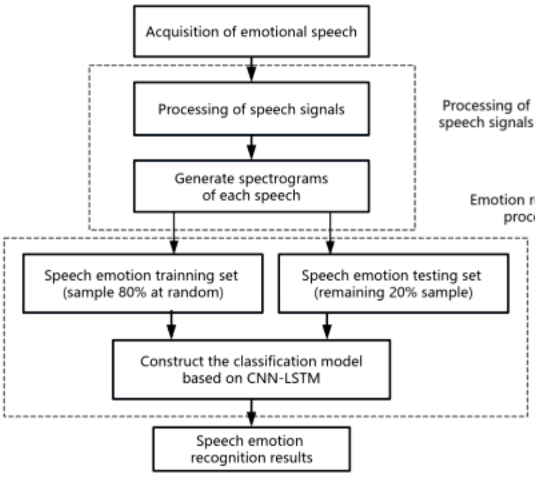

Acquisition of emotional speech
Processing of speech signals
Processing of speech signals
Generate spectrograms of each speech
Emotion recognition processing
Speech emotion trainning set (sample 80% at random)
Speech emotion testing set (remaining 20% sample)
Construct the classification model based on CNN-LSTM
Speech emotion recognition results

Fig. 2. Flowchart of design process.

IV. Datasets

We used the CASIA Chinese emotional speech database. The CASIA Chinese emotional corpus was recorded by the Institute of Automation, Chinese Academy of Sciences, and includes four professional speakers and six emotions: anger, happiness, fear, sadness, surprise, and neutrality, with 2 males and 2 females.[6] Each emotion has 200 voice signals, totaling 1200 signals. By having different people read the same text with different emotions, these data can be used to compare and analyze the acoustic and rhythmic performance under different emotional states.

V. Extraction of Emotional Feature Parameters

In the domain of speech emotion recognition, the extraction of features solely from the time domain or the frequency domain presents inherent limitations. Time-domain features lack a direct representation of the frequency characteristics of speech signals, while frequency-domain features fail to capture the temporal dynamics of speech signals. This process inevitably leads to the loss of crucial emotional information, resulting in decreased recognition rates, or it may extract redundant, extraneous information, leading to data redundancy and subsequently impacting the model's performance.

Conversely, spectrograms possess the combined advantages of both time-domain and frequency-domain features. They represent the temporal evolution of speech spectra, nearly preserving the emotional information contained within the speech signal. In the realm of audio and speech signal processing, it is imperative to transform signals into their corresponding spectrograms, utilizing the data on the spectrogram as the signal's features. The spectrogram's horizontal axis represents time, the vertical axis represents frequency, and the color intensity represents the energy distribution of frequency components at a given time. Darker colors indicate higher spectral energy, while lighter colors indicate lower spectral energy.

The transformation of speech signals into spectrograms as input can greatly assist CNN-LSTM-based speech emotion recognition models in effectively extracting speech signal features, thereby adapting to the characteristics of deep learning models and ultimately enhancing the accuracy and robustness of emotion recognition.

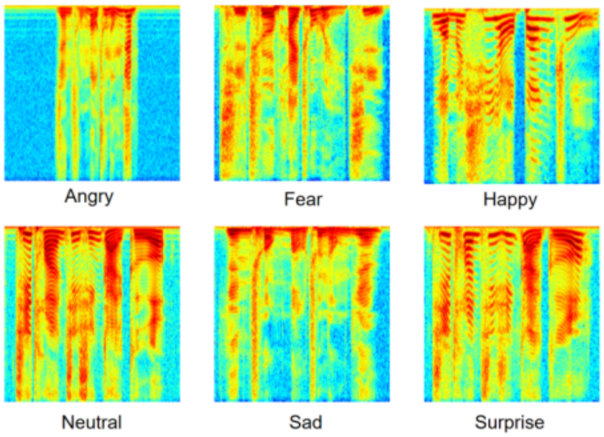


Fig. 3. Spectrogram of six different emotional states

Generating a Mel spectrogram is an important step in speech signal processing, and the general process includes preprocessing, framing, windowing, Fourier transformation, Mel filterbank, logarithmic transformation, normalization, and generating the Mel spectrogram. Firstly, the original audio signal is preprocessed, including denoising and removing silent segments, to prepare the signal data. Subsequently, the audio signal is divided into short-time windows, and windowing is applied to each window of the signal, typically using Hamming windows, Hanning windows, or similar window functions. Next, fast Fourier transformation (FFT) is performed on each windowed signal to convert the time-domain signal into the frequency-domain signal. Then, the frequency-domain signal is passed through a set of Mel filters to filter the signal at Mel frequencies, typically using 20-40 triangular filters. The energy after filtering is subjected to a logarithmic transformation to obtain the log Mel spectrogram, which is then normalized, usually using mean and variance normalization. Finally, the normalized log Mel spectrogram is plotted as an image to generate the Mel spectrogram. This series of steps provides an important foundation for subsequent tasks such as speech feature extraction and speech recognition. The flow chart of spectrogram generation is shown in the Figure 4.

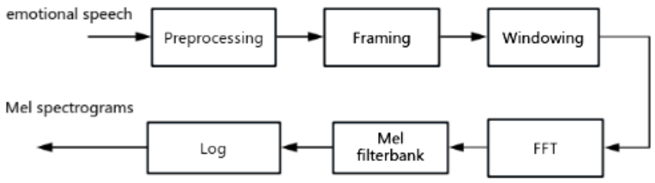


Fig. 4. The process of Mel spectrogram generation.

VI. Model construction methods introduction and testing

A. CNN

The Convolutional Neural Network (CNN) is a type of feedforward neural network composed of input layers, convolutional layers, pooling layers, fully connected layers, and output layers. In 1995, B. Lo et al. [7] and in 1998, Y. Lecun et al. [8] gradually improved the architecture of neural networks by incorporating convolutional layers and pooling layers, leading to the modern CNN. A basic CNN includes convolutional layers to extract local information through convolution and activation functions as features, pooling layers to downsample the values obtained from the convolutional layers, and finally fully connected layers to the target output. CNN are widely used in image recognition, pattern classification, object detection, face recognition, time series data, and more. Using CNN, spatial and local features of speech spectrograms can be extracted to help identify emotional states in speech. The CNN network structure diagram is shown in the Figure 5.

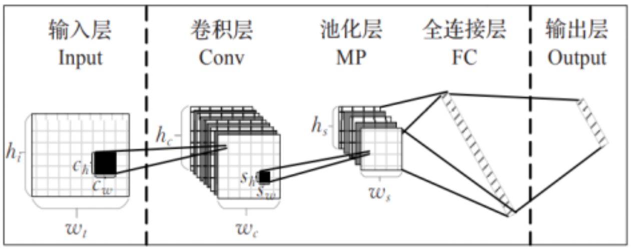


Fig. 5. CNN network structure diagram

B. LSTM

Long Short-Term Memory (LSTM) is a variation of recurrent neural networks (RNNs). It was developed to address the issue of vanishing gradients that traditional RNNs encounter during backpropagation through time, which can cause the network to get stuck in local optima and make it difficult to learn long-range dependencies between nodes. Hochreiter et al. [9] proposed the use of LSTM units, which have a special structure that allows the network to learn relationships between inputs that are separated by long time intervals. LSTM is effective in processing and predicting sequences of data that are temporally related. In many applications, such as speech recognition, LSTM has demonstrated higher accuracy compared to traditional RNNs. The LSTM structure diagram is shown in the Figure 6.

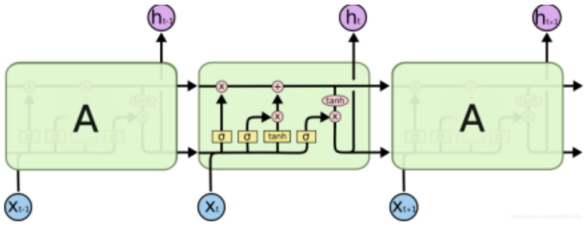


Fig. 6. LSTM structure diagram.

C. CNN+LSTM

In order to more comprehensively train the extracted emotional information from spectrograms, we used the currently popular speech recognition model, a multi-convolutional neural network based on CNN_LSTM. This approach leverages the combination of convolutional neural networks (CNN) and long short-term memory (LSTM) networks to extract features from spectrograms and capture temporal dependencies in the data. CNN are used to extract spatial features from the spectrogram, while the LSTM layer enables the model to learn temporal patterns and dependencies in the speech data.

The spectrogram is first input into three CNN layers, ReLU activation layers to extract image features. The extracted features are pooled by the maximum pooling layer to reduce the data dimension and retain the main feature information. Subsequently, the flatten layer is used to flatten the output of the convolutional layers, followed by training through two LSTM layers to handle time-series data. Finally, the output is fed into the fully connected layer to obtain all features, and emotion recognition is performed using the softmax function, resulting in the final emotion classification. The neural network structure based on CNN_LSTM is designed as shown in the Figure7. The network parameters of the CNN layer are shown in Figure 8.

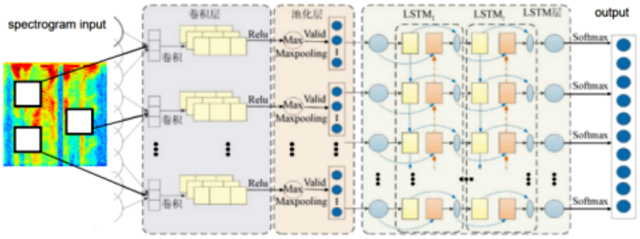
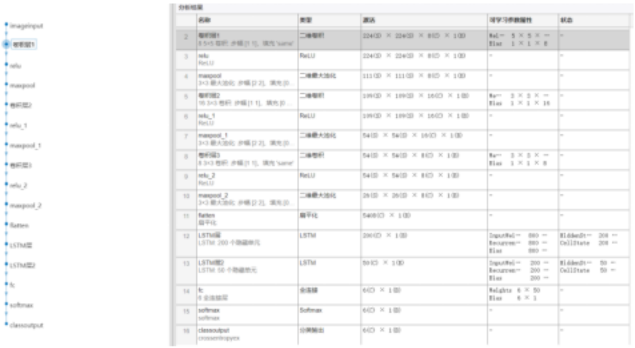


Fig. 7. CNN-LSTM model flowchart.



分析结果

	名称	类型	激活	可学习参数属性	状态
2	卷积层1 8 5×5 卷积: 步幅 [1 1], 填充 'same'	二维卷积	224(S) × 224(S) × 8(C) × 1(B)	Wei⋯ 5 × 5 × ⋯ Bias 1 × 1 × 8	-
3	relu ReLU	ReLU	224(S) × 224(S) × 8(C) × 1(B)	-	-
4	maxpool 3×3 最大池化: 步幅 [2 2], 填充 [0 ...	二维最大池化	111(S) × 111(S) × 8(C) × 1(B)	-	-
5	卷积层2 16 3×3 卷积: 步幅 [1 1], 填充 [0 ...	二维卷积	109(S) × 109(S) × 16(C) × 1(B)	We⋯ 3 × 3 × ⋯ Bias 1 × 1 × 16	-
6	relu_1 ReLU	ReLU	109(S) × 109(S) × 16(C) × 1(B)	-	-
7	maxpool_1 3×3 最大池化: 步幅 [2 2], 填充 [0 ...	二维最大池化	54(S) × 54(S) × 16(C) × 1(B)	-	-
8	卷积层3 8 3×3 卷积: 步幅 [1 1], 填充 'same'	二维卷积	54(S) × 54(S) × 8(C) × 1(B)	We⋯ 3 × 3 × ⋯ Bias 1 × 1 × 8	-
9	relu_2 ReLU	ReLU	54(S) × 54(S) × 8(C) × 1(B)	-	-
10	maxpool_2 3×3 最大池化: 步幅 [2 2], 填充 [0 ...	二维最大池化	26(S) × 26(S) × 8(C) × 1(B)	-	-
11	flatten 扁平化	扁平化	5408(C) × 1(B)	-	-
12	LSTM层 LSTM: 200 个隐藏单元	LSTM	200(C) × 1(B)	InputWei⋯ 800 ⋯ Recurren⋯ 800 ⋯ Bias 800 ⋯	HiddenSt⋯ 200 ⋯ CellState 200 ⋯
13	LSTM层2 LSTM: 50 个隐藏单元	LSTM	50(C) × 1(B)	InputWei⋯ 200 ⋯ Recurren⋯ 200 ⋯ Bias 200 ⋯	HiddenSt⋯ 50 ⋯ CellState 50 ⋯
14	fc 6 全连接层	全连接	6(C) × 1(B)	Weights 6 × 50 Bias 6 × 1	-
15	softmax softmax	Softmax	6(C) × 1(B)	-	-
16	classoutput crossentropyex	分类输出	6(C) × 1(B)	-	-

Fig. 8. CNN-LSTM network structure parameters.

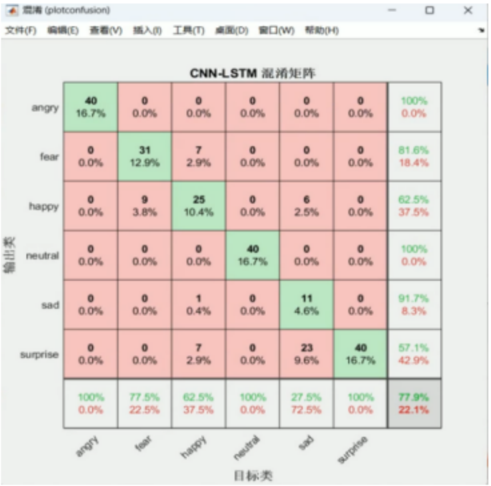
D. Testing

The datasets of spectrograms was loaded and divided into training and testing sets using an 80-20 split. The number of images in the training and testing sets were determined. The architecture of the CNN-LSTM model was constructed, comprising several layers including convolution, pooling, and LSTM layers. The network was analyzed, and training options were set, including the use of stochastic gradient descent, training on the CPU, and defining parameters such as the number of epochs and mini-batch size. The model was then trained using the training data, and the trained network was saved for future use.

The trained CNN-LSTM model was used to predict the test set, and the prediction accuracy was calculated using the mean function. The accuracy was calculated as the ratio of the number of correctly predicted samples to the total number of samples.

We also used a visualization tool called a confusion matrix to evaluate the model's performance on different emotion categories. The confusion matrix is used to show the performance of the classification model on each category. The rows of the confusion matrix represent the true categories, while the columns represent the predicted categories. The numbers in each cell represent the number of samples that were correctly predicted by the model when the true category was the row index and the predicted category was the column index.[10] By observing the confusion matrix, we can intuitively understand the model's classification accuracy on different emotion categories, thereby comprehensively evaluating the model's performance.

E. Interpretation of result



CNN-LSTM 混淆矩阵

	angry	fear	happy	neutral	sad	surprise	
angry	40 16.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
fear	0 0.0%	31 12.9%	7 2.9%	0 0.0%	0 0.0%	0 0.0%	81.6% 18.4%
happy	0 0.0%	9 3.8%	25 10.4%	0 0.0%	6 2.5%	0 0.0%	62.5% 37.5%
neutral	0 0.0%	0 0.0%	0 0.0%	40 16.7%	0 0.0%	0 0.0%	100% 0.0%
sad	0 0.0%	0 0.0%	1 0.4%	0 0.0%	11 4.6%	0 0.0%	91.7% 8.3%
surprise	0 0.0%	0 0.0%	7 2.9%	0 0.0%	23 9.6%	40 16.7%	57.1% 42.9%
	100% 0.0%	77.5% 22.5%	62.5% 37.5%	100% 0.0%	27.5% 72.5%	100% 0.0%	77.9% 22.1%

Fig. 9. Confusion matrix.

By observing the confusion matrix in Figure 9, we can see the overall precision and recall of the six different emotion categories. We found that the model's overall precision and recall are 77.9%. It shows relatively high precision and recall on the "angry" and "happy" emotions, while the performance on the "neutral" emotion is relatively low, which may be due to its similarity to other emotions. Further adjustments to the model are needed to improve its recognition ability for the corresponding emotions.

TABLE II. RESULT COMPARISON TABLE

Author	Comparison	
	Model	*Accurancy*
D.Dai et al.（2019）[10]	HMM	65.4%
S.Mao et al.[11]	RNN	65.9%
Our model	CNN-LSTM	78.9%

Fig. 10. Result comparison table.

VII. CONLUSION

This study is based on a dual CNN-LSTM model for speech emotion recognition, which effectively extracts emotional information from spectrograms by using multiple convolution kernels for multiple channels. The results show that this model has achieved excellent performance in emotion classification recognition. In the future, the dataset size and diversity can be further improved to obtain better model parameters, and how to optimize the model when recognizing more complex and similar emotions can be explored. In addition, other methods can be introduced to improve speech emotion recognition research. It is believed that with the continuous development of technology, speech emotion recognition will play a more important role in the field of artificial intelligence.

ACKNOWLEDGMENT

This project not only tests our self-learning abilities but also applies the theoretical knowledge we acquired in class and after class, enhancing our overall skills in the field of signal processing. I'd like to thank our lecturer, Dr. Chin Hong Wong and Maynooth International Engineering College, Fuzhou University. They provided me with tremendous support and encouragement. It is safe to say that we will never complete our project without them.

REFERENCES

[1] Dellaert F, Polzin T, Waibel A. Recognizing Emotion In Speech [J]. Proceedings of the Cmc, 1996, 3(2):1970-1973.

[2] Petrushin V A. Emotion recognition in speech signal: Experimental study, development, and application [C].The Proceedings of the. 2000:222-225.

[3] Lee C M, Narayanan S, Pieraccini R. Recognition of negative emotions from the speech signal[C].Automatic Speech Recognition and Understanding, 2001. ASRU '01. IEEE Workshop on. IEEE, 2001:240-243.

[4] Eyben F. Opensmile: the munich versatile and fast open-source audio feature extractor[C].ACM International Conference on Multimedia. ACM, 2010:1459-1462.

[5] Valstar M, Eyben F, Mckeown G, et al. AVEC 2011-the first international audio/visual emotion challenge[C].International Conference on Affective Computing and Intelligent Interaction. Springer-Verlag, 2011:415-424.

[6] Y. Liu et al., "A Discriminative Feature Representation Method Based on Cascaded Attention Network With Adversarial Strategy for Speech Emotion Recognition," in IEEE/ACM Transactions on Audio,

Speech, and Language Processing, vol. 31, pp. 1063-1074, 2023, doi: 10.1109/TASLP.2023.3245401.

[7] Shih-Chung B. Lo, Heang-Ping Chan, Jyh-Shyan Lin, Huai Li, Matthew T. Freedman, and Seong K.Mun. 1995. Artificial convolution neural network for medical image pattern recognition. Neural Networks, 8(7): 1201-1214.

[8] LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. In Proceedings of the IEEE, 86(11): 2278-2324. https://doi.org/10.1109/5.726791.

[9] Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. Long short-term memory. Neural computation, 9(8): 1735-178.

[10] S. Mao, D. Tao, G. Zhang, P. C. Ching, and T. Lee. Revisiting hidden markov models for speech emotion recognition[C]. in ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), May 2019, 6715–6719.

[11] S. Yoon, S. Byun, and K. Jung. Multimodal speech emotion recognition using audio and text[J].CoRR, vol. abs/1810.04635, 2018.

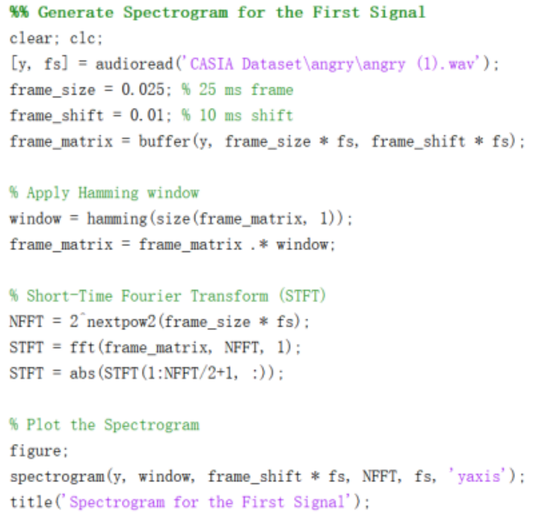
APPENDICES

Our source code and some pictures, report PPT sent in the following github link:

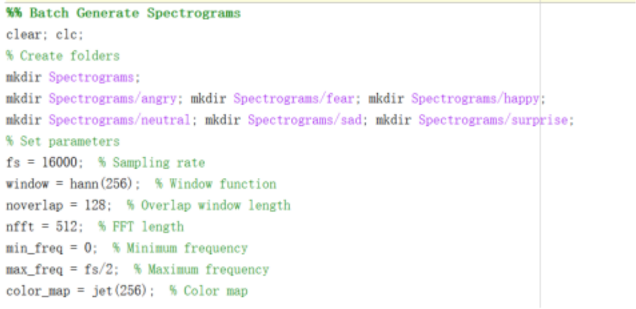
https://github.com/chm123456/Speech-Emotion-Recognition.

Screenshots of some important code::

Generate Spectrogram:

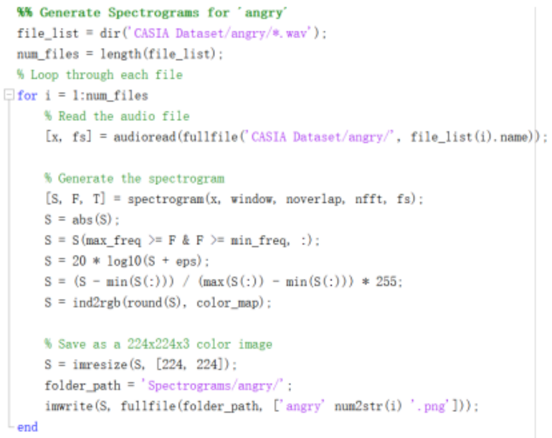


```
%% Generate Spectrogram for the First Signal
clear; clc;
[y, fs] = audioread('CASIA Dataset\angry\angry (1).wav');
frame_size = 0.025; % 25 ms frame
frame_shift = 0.01; % 10 ms shift
frame_matrix = buffer(y, frame_size * fs, frame_shift * fs);

% Apply Hamming window
window = hamming(size(frame_matrix, 1));
frame_matrix = frame_matrix .* window;

% Short-Time Fourier Transform (STFT)
NFFT = 2^nextpow2(frame_size * fs);
STFT = fft(frame_matrix, NFFT, 1);
STFT = abs(STFT(1:NFFT/2+1, :));

% Plot the Spectrogram
figure;
spectrogram(y, window, frame_shift * fs, NFFT, fs, 'yaxis');
title('Spectrogram for the First Signal');
```

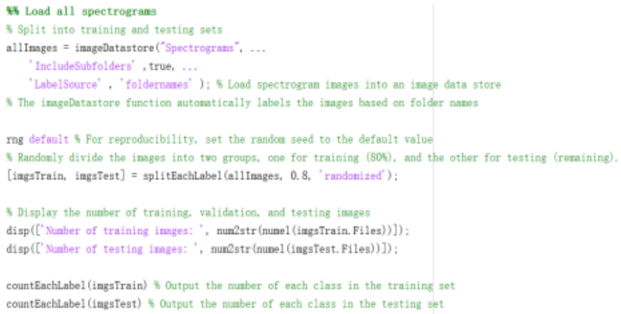


```
%% Batch Generate Spectrograms
clear; clc;
% Create folders
mkdir Spectrograms;
mkdir Spectrograms/angry; mkdir Spectrograms/fear; mkdir Spectrograms/happy;
mkdir Spectrograms/neutral; mkdir Spectrograms/sad; mkdir Spectrograms/surprise;
% Set parameters
fs = 16000;  % Sampling rate
window = hann(256);  % Window function
noverlap = 128;  % Overlap window length
nfft = 512;  % FFT length
min_freq = 0;  % Minimum frequency
max_freq = fs/2;  % Maximum frequency
color_map = jet(256);  % Color map
```

Load the spectrogram and divide the data set and test set:

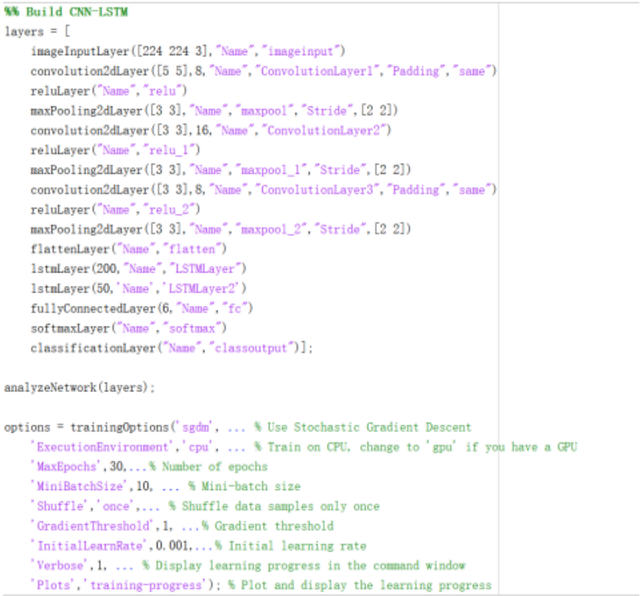


```
%% Generate Spectrograms for 'angry'
file_list = dir('CASIA Dataset/angry/*.wav');
num_files = length(file_list);
% Loop through each file
for i = 1:num_files
    % Read the audio file
    [x, fs] = audioread(fullfile('CASIA Dataset/angry/', file_list(i).name));

    % Generate the spectrogram
    [S, F, T] = spectrogram(x, window, noverlap, nfft, fs);
    S = abs(S);
    S = S(max_freq >= F & F >= min_freq, :);
    S = 20 * log10(S + eps);
    S = (S - min(S(:))) / (max(S(:)) - min(S(:))) * 255;
    S = ind2rgb(round(S), color_map);

    % Save as a 224x224x3 color image
    S = imresize(S, [224, 224]);
    folder_path = 'Spectrograms/angry/';
    imwrite(S, fullfile(folder_path, ['angry' num2str(i) '.png']));
end
```



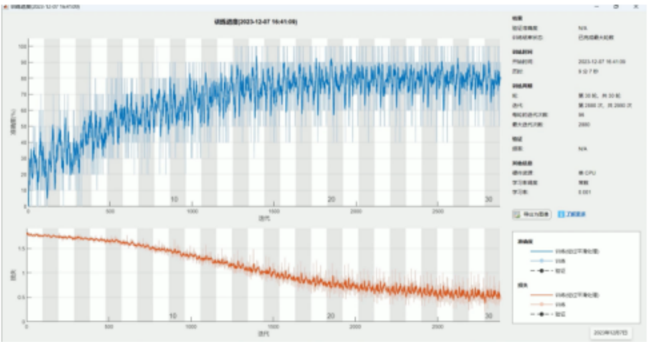
```
%% Load all spectrograms
% Split into training and testing sets
allImages = imageDatastore("Spectrograms", ...
    'IncludeSubfolders' ,true, ...
    'LabelSource' , 'foldernames' ); % Load spectrogram images into an image data store
% The imageDatastore function automatically labels the images based on folder names

rng default % For reproducibility, set the random seed to the default value
% Randomly divide the images into two groups, one for training (80%), and the other for testing (remaining).
[imgsTrain, imgsTest] = splitEachLabel(allImages, 0.8, 'randomized');

% Display the number of training, validation, and testing images
disp(['Number of training images: ', num2str(numel(imgsTrain.Files))]);
disp(['Number of testing images: ', num2str(numel(imgsTest.Files))]);

countEachLabel(imgsTrain) % Output the number of each class in the training set
countEachLabel(imgsTest) % Output the number of each class in the testing set
```

Build CNN-LSTM:



```
%% Build CNN-LSTM
layers = [
    imageInputLayer([224 224 3],"Name","imageinput")
    convolution2dLayer([5 5],8,"Name","ConvolutionLayer1","Padding","same")
    reluLayer("Name","relu")
    maxPooling2dLayer([3 3],"Name","maxpool","Stride",[2 2])
    convolution2dLayer([3 3],16,"Name","ConvolutionLayer2")
    reluLayer("Name","relu_1")
    maxPooling2dLayer([3 3],"Name","maxpool_1","Stride",[2 2])
    convolution2dLayer([3 3],8,"Name","ConvolutionLayer3","Padding","same")
    reluLayer("Name","relu_2")
    maxPooling2dLayer([3 3],"Name","maxpool_2","Stride",[2 2])
    flattenLayer("Name","flatten")
    lstmLayer(200,"Name","LSTMLayer")
    lstmLayer(50,'Name','LSTMLayer2')
    fullyConnectedLayer(6,"Name","fc")
    softmaxLayer("Name","softmax")
    classificationLayer("Name","classoutput")];

analyzeNetwork(layers);

options = trainingOptions('sgdm', ... % Use Stochastic Gradient Descent
    'ExecutionEnvironment','cpu', ... % Train on CPU, change to 'gpu' if you have a GPU
    'MaxEpochs',30,...% Number of epochs
    'MiniBatchSize',10, ... % Mini-batch size
    'Shuffle','once',... % Shuffle data samples only once
    'GradientThreshold',1, ...% Gradient threshold
    'InitialLearnRate',0.001,...% Initial learning rate
    'Verbose',1, ... % Display learning progress in the command window
    'Plots','training-progress'); % Plot and display the learning progress
```

The process of training:


100
90
80
70
60
50
40
30
20
10
0
0
500
1000
1500
2000
2500
10
20
30
1.5
1
0.5
0
N/A
0.001