International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2024: 7.101

Design and Implementation of a Facial Expression Recognition and Emotion Analysis System Based on Deep Learning

Xinyu Xia[1], Bangying Wu[2], Kan Ji[3]

[1]School of Computer and Software Engineering, Southwest Petroleum University, Chengdu, China

[2, 3]Network and Information Center, Southwest Petroleum University, Chengdu, China

Abstract: ***In response to the growing mental health concerns faced by individuals under high-pressure environments in modern society, as well as the limitations of effectively monitoring emotion states through traditional methods, this paper leverages deep learning techniques based on a modified VGG model to design and implement a facial expression recognition and emotion analysis system that can be transplanted to mobile platforms. This system has four functional modules including login & registration, emotion center, facial expression recognition & conversion, and personal center. In our study, the publicly available dataset FER-2013 was used to train the proposed model for recognizing and classifying various user emotions, covering a total of 7 emotional categories: anger, surprise, disgust, fear, happy, neutral, and sadness. In the system, facial expression data is collected through two methods: real-time photo capture and image import from album. Based on the acquired images, the system performs facial expression recognition and provides feedback on the detected emotional state to users. The results indicate that the proposed system not only provides users with convenient, real-time emotion detection via mobile devices, but also maintains good performance under low computational resource consumption.***

Keywords: Mental Health; Deep learning; VGG; Facial expression recognition; Emotion analysis

1. Introduction

Facial expressions are the most direct and prominent way of expressing emotions, and effective recognition has important theoretical significance and practical value [1-2]. In recent years, with the increasing emphasis on mental health issues in society, a large number of studies on mental health problems have emerged. Facial expression recognition as one of the personal emotion analysis methods becomes the most essential research direction. Against the background of rapid development in the field of facial expression recognition, many related applications have emerged, including but not limited to smart classroom teaching evaluation system [3]. classroom atmosphere evaluation system [4] and so on. At present, facial expression recognition algorithms can be divided into two categories: manual feature extraction based on traditional methods and deep learning-based methods. Early facial expression recognition was achieved through traditional manual feature extraction methods, mainly including optical flow-based methods and local binary pattern-based method[5]. The Pfister team recognizes facial expressions by describing dynamic textures[5]. Huang et al. proposed facial expression recognition based on spatiotemporal complete local quantization mode[6], which improved the performance of facial expression recognition. Although the traditional manual feature extraction methods are able to recognize facial expressions with a certain degree of accuracy, still suffer from problems such as low efficiency, high complexity, and low recognition performance. With the continuous improvement of computer computing power, deep learning, especially convolutional neural networks (CNN), has made significant progress in the field of facial expression recognition. CNN extracts facial features through local receptive fields, which enables models to effectively recognize facial expression features[7]. At the same time, with the continuous improvement and upgrading of expression recognition models, the algorithms for expression recognition are also constantly improving, such as facial expression recognition algorithms based on dual stream features and attention mechanisms[7] and expression recognition methods based on gradient Gabor histogram features[7], which have achieved a certain degree of improvement in the accuracy of facial expression recognition.

However, there are still some urgent problems need to be solved in the field of facial expression recognition. Firstly, even with the series of studies on facial expression recognition and emotion analysis mentioned above, most of them are computer programs[7], lacking mobile programs that can provide users with convenient, fast, real-time and accurate facial expression recognition and emotion analysis detection to help users detect and analyze emotions at any time; Secondly, although the above models have made significant progress in facial expression recognition tasks, these models often require higher hardware configurations for application deployment in order to improve recognition accuracy and continuously deepen and widen the network structure[7]. Thirdly, most current systems often only provide one type of recognition result and cannot intuitively present the proportion of emotions conveyed through facial expressions.

The VGG network structure is very simple, and its performance can be improved by continuously deepening the network structure to achieve accurate differentiation and recognition of image expressions in multi scene and multi noise environments[7]. Based on the advantages of easy scalability and strong adaptive ability of VGG deep neural network[7], our paper proposes a deep learning-based expression recognition and emotion analysis system that is easy to install and deploy. The system uses data normalization, data enhancement and other techniques to improve the accuracy of expression recognition and emotion analysis results to a certain extent, providing users with an emotion self-testing channel, and thus providing more reliable emotion analysis results for psychological counseling institutions to help them quickly understand the emotional state of patients and provide a basis for formulating treatment plans. In our system, the modified VGG19 model is used to detect and analyze facial expression images uploaded by users independently for helping them analyze and classify their emotions at that time. The intuitive expression recognition results and distribution chart of various emotions are fed back to users in the form of emojis, which helps users to relieve negative emotions and relieve emotional pressure in time. In terms of data preprocessing, we utilize maximum normalization to scale the pixels of the image to [0,1], and append a data augmentation module to make the proposed model more robust by performing random flipping, rotation, scaling, cropping, and small translation operations on the training set. In the modified VGG model, all convolutional layers use the same small convolution kernel, providing good generalization ability. During training, L2 regularization is applied to prevent excessive parameter weights and reduce overfitting risks. In addition, the system we proposed in this paper can be applied to mobile platforms, which makes up for the shortcomings of facial expression recognition and emotion analysis in serving the public.

2. System Framework Design

By conducting a comprehensive requirements analysis, two types of target users for the system were determined in our study. One is ordinary users who wish to conduct emotion self-testing through facial expression recognition and emotion analysis, and the other is psychological counseling institutions that hope to obtain more scientific and reliable emotion analysis. Based on the user needs of the two different groups mentioned above, the proposed system designs four functional modules, including login & registration, personal center, facial expression recognition & conversion, and emotion center. Among them, facial expression recognition & conversion serves as the central module, which reads image from database and fed into the recognition model we developed for facial expression analysis. Through the above analysis, the proposed system can meet the needs of different user groups.

2.1 System Architecture

The system is mainly developed for mobile applications based on the function mentioned above. Facial expression recognition & emotion analysis as the core modules of this system utilizes a modified VGG19 for expression recognition and analysis. The specific architecture is shown in Figure 1.

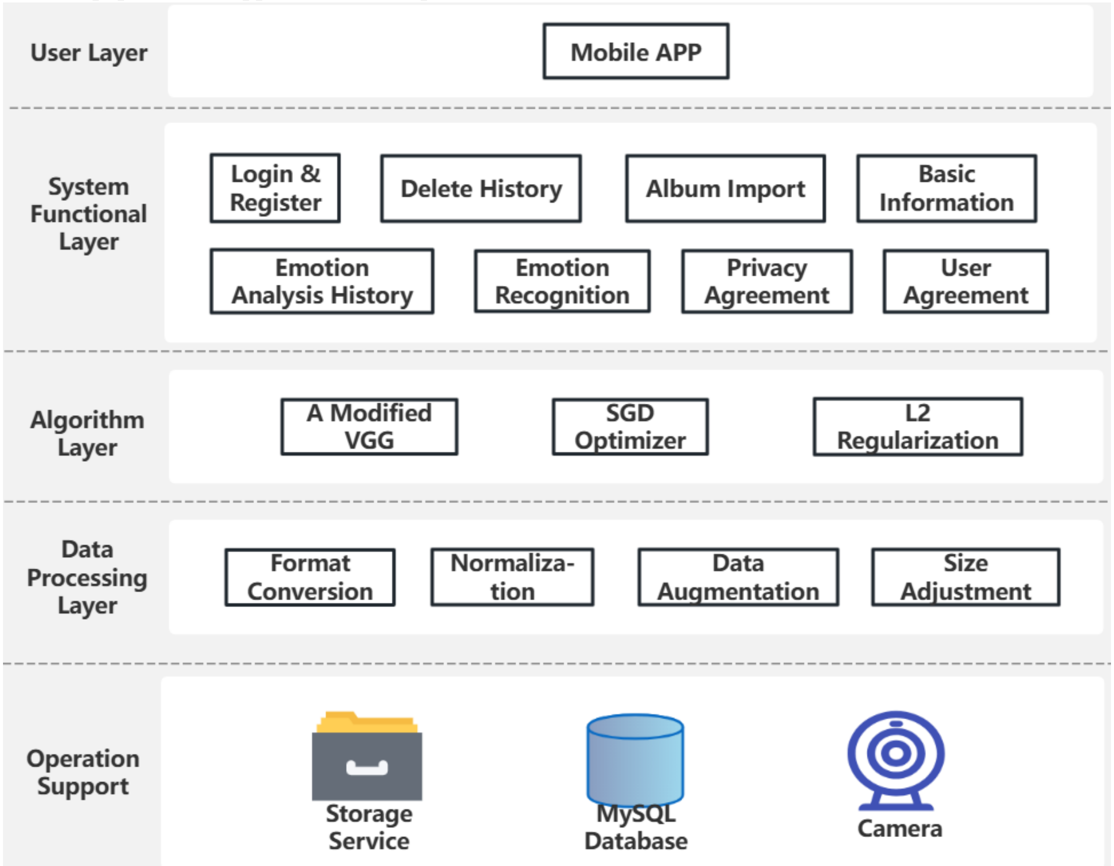

Figure 1: System Architecture

2.2 Facial Expression Recognition Model

The system's facial expression recognition & conversion module employs a modified VGG-based model for expression classification, with all 3×3 convolution kernels and 2×2 max pooling windows. Given that the input images are in grayscale, the number of input channels in the first convolutional layer is set to 1 (i.e. single channel). To further compress the spatial dimensions of feature map while preserving channel information, the proposed model introduces a Global Average Pooling (GAP) layer. The feature map is flatten into a 512-dimensional vector and passed through three fully connected layers in sequence. ReLU activation function is used between the fully connected layers, and Dropout layer is combined to prevent overfitting. The specific model design diagram is shown in Figure 3. The input of our model is a small-sized grayscale image of 48×48, and each convolutional layer uses the same size of convolution kernel. Each set of convolutional layers is followed by a max pooling layer for downsampling. Set the output node of the last fully connected layer of the model as the expression recognition categories. This strategy simplifies the traditional VGG architecture to some extent, effectively reducing its reliance on computational resources.

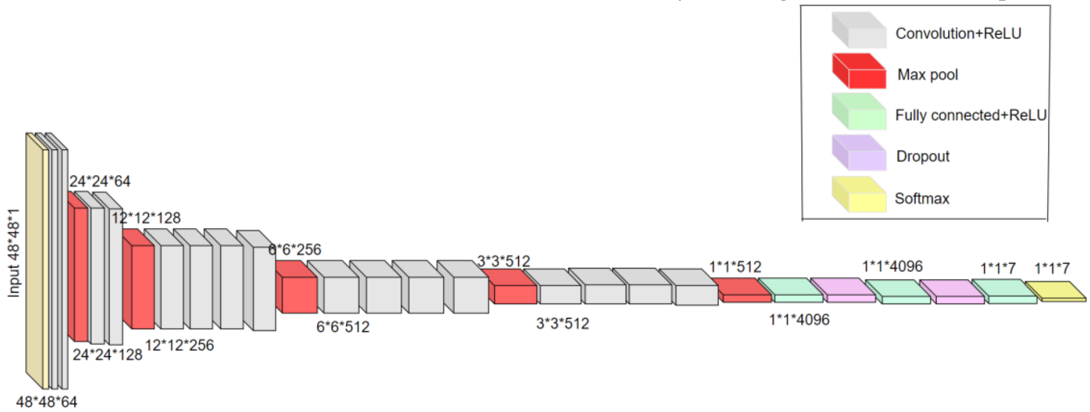

Figure 2: Facial Expression Recognition Model

2.3 System Functional Modules

The system is divided into four modules, including login and registration section, emotion center, emotion recognition and conversion, and personal center. The detailed functional structure is shown in Figure 3.

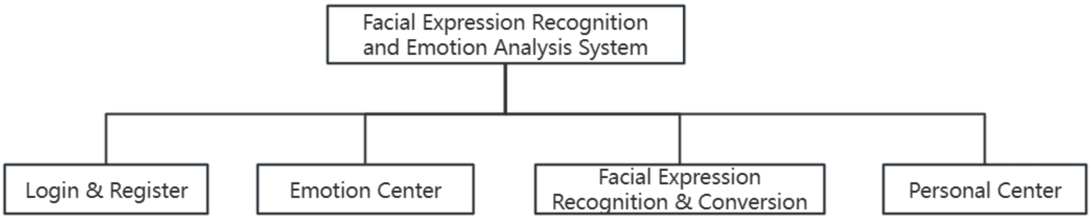

Figure 3: Facial Expression Recognition and Emotion Analysis System Modules

The login module mainly includes the display of user interface, verification of user input, detection of network connection, retrieval and verification of user information, as well as the saving of status and page redirection after successful login. The registration module is mainly responsible for creating information for new users. Users need to enter their username, password, and phone number, and submit them to the server for registration after verification. If the registration is successful, the user will be redirected to the login interface. The emotion center module is mainly responsible for displaying the historical records of user emotion analysis. Users can view and delete historical emotion analysis to help them better grasp their historical emotional states. The facial expression recognition & conversion module is the core module of this system, mainly responsible for analyzing facial expressions to detect emotions and returning the results to the user. The personal center module provides users with a convenient interface to view personal information and access relevant policies. As the first page after logging in, users can view the system privacy agreement and user agreement, as well as their personal information on this page.

3. Implementation of the System

This system is based on C/S architecture and developed using PyCharm development tool and Python programming language. The front-end interface of the system is designed using Android Studio and UI layout framework, and the back-end database is managed and maintained using MySQL. The experimental environment of the system is shown in Table 1:

Table 1: Experimental Environment

Hardware Components	Specifications
CPU	Intel i9-12900H
Memory	16G
GPU	NVIDIA RTX3060
Video Memory	6G

3.1 Dataset

Many researchers have conducted studies on facial expression classification. Tom King listed 8 basic emotions: interest, happiness, surprise, pain, fear, anger, shyness, and contempt. Izard added two emotions, disgust and guilt, on this basis. Ekman defined six basic facial expressions: surprise, fear, disgust, anger, happy, and sad[8]. Based on Ekman's facial expression research and subsequent emotion research[7], the proposed system refines facial expressions into seven categories: anger, disgust, fear, happy, sad, surprise, and neutral. The facial expression dataset used in our experiment is from the publicly available FER-2013 dataset, and the structure of the dataset is shown in Table 2.

Table 2: Examples from the FER-2013 Dataset

Index	Emotion	Pixels	Usage
0	0	70 80 82 72 58 58 60 63 54 58 60 48 89 115 121 119 115 …...	Training
1	0	151 150 147 155 148 133 111 140 170 174 182 154 153	Training
2	2	231 212 156 164 174 138 161 173 182 200 106 38 39 74	Training
……	……	……	……
35886	4	61 63 59 75 151 159 166 161 143 170 127 131 184 216 222......	Test
35887	5	7 5 8 6 7 3 2 6 5 4 4 5 7 5 5 5 6 7 7 7 10 10 8 6 8 11 12 15 7	Test

Emotion in the second column of Table 2 is the numerical category corresponding to the expression, and the expressions from 0 to 6 are shown in Table 3:

Table 3: Emotion Mapping Table

Emotion	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
number	0	1	2	3	4	5	6

The third column of Table 2, Pixels, is the pixel data of the image, with 2304 (48×48) pixel values per image. Each pixel data is separated by a space symbol, and each pixel value is between [0, 255].

3.2 Data Preprocessing

The fourth column of Table 2, Usage, represents the purpose of the sample, with two types of fields: Training and Test, used as the training and testing sets, respectively. Training data accounts for 85%, test data accounts for 15%, and detailed information of dataset distribution is shown in Table 4.

Table 4: Dataset Distribution in Our Experiment

Dataset	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
Training Set	3995	436	4097	7215	4830	3171	4965
Test Set	491	55	528	879	594	416	626
Sum	4486	491	4625	8084	5424	3587	5591

According to the statistical data analysis in Table 4, the sample distribution in this dataset is uneven, and the images of disgust emotion are significantly fewer than other emotions. In order to prevent the problem of insufficient generalization of the model during the training, we apply data augmentation techniques by randomly flipping, rotating, cropping, scaling, and small translation operations on the few emotion category data in the training set, generating more training samples to help the model better learn the features of these categories. The entire data preprocessing steps are shown in Figure 4.

Figure 4: Data Preprocessing Steps

Data preprocessing presented in Figure 4 consists 5 steps. The first step is data cleaning and loading. Cleaned facial expression data are obtained by removing outliers in the dataset, such as emotion categories labels that exceed the range or those inconsistent pixel value dimensions. The second stage is custom data transformation. For the training set, convert the original images to PIL Image format and perform random horizontal flipping and color jittering to change the brightness, contrast, and saturation of the image, increasing the diversity of the dataset and improving the model's generalization ability. Then convert the processed images into tensors, adjust the dimension order to (C, H, W), and add channel dimensions. Where C is the number of image channels, H represents the height of image, and W denotes the width of image. For the test set, the operation of data transformation is comparatively straightforward, only requiring the data to be converted into tensor format to meet the input requirements of our model.

Data segmentation as the third step, divides the dataset into training and test sets based on the labels of the Usage field in the dataset. The training set is used for the model to extract features and patterns, while the test set is employed to evaluate the final generalization ability of the model. The fourth phase involves defining a *Custom Dataset Class*, indexes samples according to the *Usage* field. It extracts the pixel information from the *Pixels* field, divides and converts the data into an integer list, and reshapes it into 48 × 48 image. Subsequently, min-max normalization is applied to scale the pixel values of the image to the range [0,1], as described by the following formula:

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

x represents the original feature value that need to be normalized. x_{min} represents the minimum value among all values of the feature in the dataset, while x_{max} represents the maximum value. x_{new} represents the new value that has been normalized.

The fifth step is to create DataLoader instances based on the partitioned dataset. When setting the DataLoader for the training set, specify a batch size of 128 and shuffle the data, so that the model is exposed to data in varying order during each training epoch, which helps improve training effectiveness. For the test set, the DataLoader is also configured with a batch size of 128, but shuffling is disabled to ensure consistency of the evaluation results.

After the above data preprocessing operations, the data has been adapted to the input requirements of the model in terms of size, format, and normalization. And meanwhile, the efficiency and stability of model training have been improved, promoting the construction of a standardized, modular, and extensible data processing flow.

3.3 Model design

The model in our experiment uses 16 convolutional layers, 16 ReLU activation function layers, 5 pooling layers, 3 fully connected layers, 2 dropout layers, and the final Softmax layer. The number of channels in the first convolutional layer is set to 1 to receive the grayscale image dataset. For each convolutional layer, assuming the input size is $\mathrm{C_{in}} \times \mathrm{H_{in}} \times \mathrm{W_{in}}$, the size of the convolution kernel is $K_h \times K_w$, the step size is S, and the number of paddings is P, the output image size is:

$$H_{out} = \left\lfloor \frac{H_{in} + 2P - K_h}{S} + 1 \right\rfloor \tag{2}$$

$$W_{out} = \left\lfloor \frac{W_{in} + 2P - K_w}{S} + 1 \right\rfloor \tag{3}$$

$\mathrm{C_{in}}$ is the number of channels in the input feature map, and for the grayscale images in this dataset, $\mathrm{C_{in}}$ is set 1. $\mathrm{H_{in}}$ is the height of the input feature map, and $\mathrm{W_{in}}$ is the width of the input feature map. K_h is the height of the convolution kernel, K_w is the width of the convolution kernel, H_{out} is the height of the output feature map, and W_{out} is the width of the output feature map.

In our model, the input image size is 48×48 single channel pixels, and the convolution kernel size of the convolutional layer in the model is 3×3, with a stride and padding of 1. The number of convolutional channels doubles with each layer. The activation function after each convolutional layer is ReLU, which has the advantage of being computationally efficient, allowing the neural network to converge efficiently, having a well-defined derivative function, and is compatible with backpropagation algorithms. The function equation of ReLU is as follows:

$$\mathrm{F(x)} = \max(0, \mathrm{x}) = \begin{cases} 0, & \mathrm{x} <= 0 \\ \mathrm{x}, & \mathrm{x} > 0 \end{cases} \tag{4}$$

After completing feature extraction, the model introduces a max pooling layer to further reduce the dimensionality of the feature map. The max pooling layer calculates the maximum value of local regions in the feature map, effectively reducing the spatial size of the feature map and preserving key information of the features while reducing computational complexity.

The classifier consists of a series of fully connected layers, and its core function is to output the probability distribution of 7 categories, achieving classification prediction of input data. At the same time, to prevent overfitting during the training process, Dropout technique is used between fully connected layers to randomly drop out some neurons, improving the model's generalization ability. Suppose a neuron has an output of x, when Dropout is applied, the neuron is "turned off" with probability *p*, i.e., its output is set to 0. With probability *1-p,* the output remains unchanged but is scaled by dividing by 1-*p* to maintain the expected output of the neuron. The mathematical expression is as follow:

$$\mathrm{y} = \begin{cases} 0 & \mathrm{p} \\ \dfrac{\mathrm{x}}{1-\mathrm{p}} & 1-\mathrm{p} \end{cases} \tag{5}$$

During the training process, Dropout is randomly applied to each mini batch to prevent collaborative adaptation between neurons and reduce overfitting. In the testing phase, Dropout is disabled, instead, the trained model is used directly for prediction. During the training process of the model, cross-entropy loss function is used. For a given input sample x, the model outputs a predicted class probability distribution as follow:

$$\bar{\mathrm{y}} = (\bar{\mathrm{y}}_1, \bar{\mathrm{y}}_2, \ldots, \bar{\mathrm{y}}_C) \tag{6}$$

Among them, *C*=7 is the total number of categories for facial expression classification. Meanwhile, the true labels of the samples are represented in the form of one hot encoding as follows:

$$\mathrm{y} = (\mathrm{y}_1, \mathrm{y}_2, \ldots, \mathrm{y}_C) \tag{7}$$

Where only the element corresponding to the real category is 1, and all others are 0. The mathematical expression for the cross-entropy loss function is:

$$\mathrm{L} = -\sum_{\mathrm{i}}^{7} \mathrm{y_i} \log(\overline{\mathrm{y_i}}) \tag{8}$$

In addition, our model also sets a weight decay hyperparameter *λ=1e-4*, and adds an L2 regularization term to the loss function. The loss function is then updated to:

$$L_{regularized} = L + \frac{\lambda}{2}\sum_j \theta_j^2 \quad (9)$$

These operations are achieved by penalizing excessive weight parameters to prevent overfitting of the model, thereby improving its generalization ability.

In terms of optimization algorithms, the stochastic gradient descent algorithm with driving force is used to update the model parameters. The standard Stochastic Gradient Descent (SGD) algorithm suffers from slow convergence and oscillations near local minima. Therefore, a momentum based stochastic gradient descent method is introduced. This method considers the information of the previous gradient when updating parameters, which can accelerate convergence and reduce oscillations. As for the training dataset $\{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\}$ which has N samples, assuming the model parameters are θ and the loss function is $L(\theta)$, stochastic gradient descent updates the parameters by computing the gradient using only one sample $x^{(i)}$ at each iteration:

$$\theta = \theta - \eta \nabla L(\theta; x^{(i)}) \quad (10)$$

η=0.01 is the learning rate, which controls the step size of parameter updates. An excessively large learning rate may cause parameter updates to overshoot the optimal solution, while a too small learning rate slows down the training. To address this, a momentum hyperparameter β=0.9 is introduced, which adjusts the current update direction by accumulating the previous gradient directions. The parameter update formula is given by:

$$v_t = \beta v_{t-1} - \eta \nabla L(\theta_t; x^{(i)}) \quad (11)$$

$$\theta_t = \theta_{t-1} + v_t \quad (12)$$

The two equations above represent the velocity vector at time t, which makes the current update direction reference the velocity at the previous time, reduces the oscillation of parameter updates, and accelerates the convergence of the function.

3.4 Model Training

In each training epoch, the model first performs forward propagation on the training set, producing predictions by passing the input data through all layers of the network. Subsequently, the predicted values are then compared with the ground truth labels using the cross-entropy loss function to compute the error between the model's output and the true results. Based on the loss value, the gradient of each layer parameter is automatically calculated through backpropagation algorithm, and the model parameters are updated using Stochastic Gradient Descent optimizer, iteratively adjusting the model to reduce prediction error. The pseudocode of the algorithm is shown in Table 5.

Table 5: Algorithm Pseudocode

```
Algorithm 1： Emotion Recognition Model
Input： Processed dataset
Output： Best precision、F1、recall、AUC and loss for training and testing
1:   Initialize cross-entropy loss function and optimizer
2:   For each epoch do:
       Set model to training mode
       For each batch in training DataLoader:
          Forward pass: compute predicted outputs by passing inputs to the model
          Backward pass: compute gradient of the loss with respect to model parameters
          Update model parameters using optimizer
3:     End for
       Load the saved model:
     Forward pass: compute predicted outputs by passing inputs to the model
           Get predicted classes and probabilities
       End for
     End for
```

4. Evaluation and Results

This section presents the evaluation methodology and experimental results of the proposed model, demonstrating its performance and effectiveness through quantitative and qualitative analyses.

4.1 Model Evaluation

During the model training process, the training accuracy and loss are continuously recorded for each epoch. Through multiple iterations, the model's performance gradually improved. The evaluation metrics used in our study include loss, precision, F1 score, and recall. As shown in Figure 5, the loss of the model's training set has converged over 300 epochs, with the minimum loss reduced to 0.048. Furthermore, precision, F1 score, and recall have also converged and approached 1.

The model exhibits good generalization ability when evaluated the test set, with a loss of 0.95 and a classification accuracy of 0.72. In addition, to more clearly demonstrate the improvements in the predictive ability of the proposed algorithm, we further investigate the impact of model parameters on the prediction performance by fine-tuning key parameters. Specifically, we chose to modify the optimizer, batch size, and convolution kernel size of the model, changing only one parameter at a time while keeping the others constant. The training results of the fine-tuned models are shown in Table 6 for comparison.

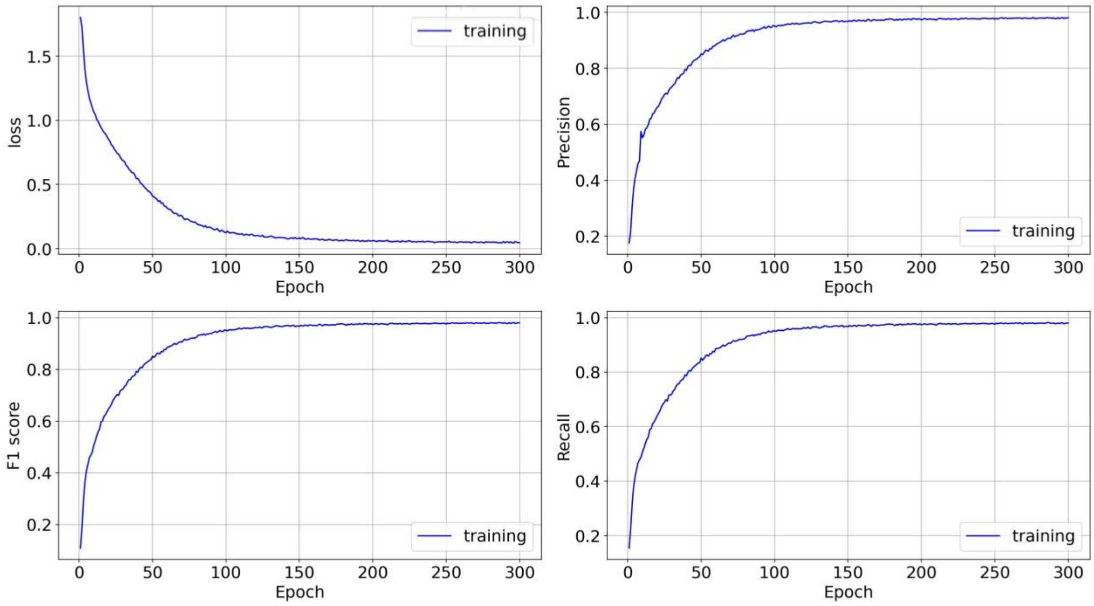

Figure 5: Model Evaluation Results

Table 6: Comparison of Model Performance after Parameter Fine-tuning

Before modification: The size of the Convolution Kernel is 3×3, Batch Size is 128, Optimizer is SGD.					
Indicators	Datasets	Before Modification	Batch Size 64	Convolutional Kernel 2×2	Optimizer Adam
Precision	Training Set	0.98	0.97	0.36	0.33
	Test Set	**0.72**	**0.71**	**0.35**	**0.03**
F1	Training Set	0.98	0.97	0.05	0.13
	Test Set	**0.70**	**0.70**	**0.05**	**0.06**
Recall	Training Set	0.98	0.97	0.14	0.14
	Test Set	**0.69**	**0.69**	**0.14**	**0.14**
AUC	Training Set	1.00	1.00	0.50	0.50
	Test Set	**0.92**	**0.92**	**0.51**	**0.51**

As shown in the Table 6 that reducing the batch size from 128 to 64 led to a training accuracy of 0.97, with other performance metrics remaining comparable to those of the original model. This indicates that, compared with the initial batch size of 128, modifying the batch size has no significant negative impact on the overall fitting ability of the model.

When the convolutional kernel of the model is adjusted to 2×2, the accuracy dropped significantly to 0.36 on the training set and 0.35 on private test set. The F1 score also decreases sharply to 0.05. The poor precision on both the training and test sets suggests that the modified model suffers from underfitting under the configuration of 2 × 2 convolutional kernel. After replacing the optimizer from SGD to Adam, the precision of the model on both the training and test sets significantly declines. The precision on the training set dropped to 0.33, and the performance on the test set was extremely poor, with prediction precision even less than 0.05. Moreover, the indicators of F1 score, recall, and AUC showed substantial deviations compared to the original model. These results indicate that using the Adam optimizer markedly degrades performance of the model we designed.

4.2 System Results

Through multiple rounds of training and optimization, the facial expression recognition and emotion analysis system implemented in this paper has achieved promising results. Based on the trained model, we conducted recognition tests on real facial images within the system.

As shown in Figure 8, when the system is provided with a facial image from outside the training set, it performs classification by evaluating and scoring the image across different expression categories. The bar chart of the classification results feedback from the system to the user shows that the category "Sad" received the highest score, nearly reaching the maximum, while all other categories scored zero, except for "Fear", which received a very low classification score. At the same time, the system presents users with intuitive feedback on the results of facial expression recognition and emotion analysis in the form of emoji icons, visually conveying a sad emotional state. The system we implemented demonstrates effective human-computer interaction capabilities.

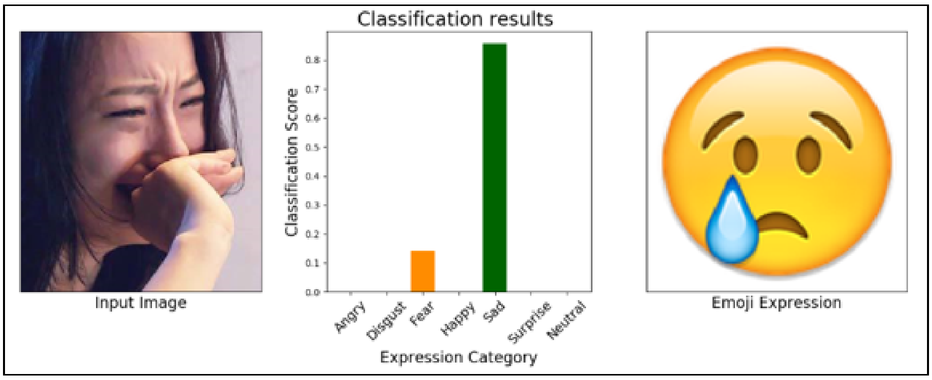

Figure 8: Expression Recognition and Emotion Analysis Results

5. Conclusion

This study designs and implements a deep learning based facial expression recognition and emotion analysis system to address the needs of mental health monitoring. By using the publicly available FER-2013 dataset and combining data augmentation, regularization, and other techniques with the modified VGG19 model, the model's generalization ability in our system is effectively improved. The system covers functional modules such as login & registration, emotional center, facial expression recognition & conversion, and personal center. It supports data acquisition through both camera capture and photo album selection, and can recognize 7 types of facial expressions. Experimental results demonstrate that the model used in our system exhibits good performance in facial expression recognition and emotion analysis, providing users with convenient emotional self-testing tool and reliable data references for psychological counseling institutions.

In future, the facial expression recognition and emotion analysis system can be further enhanced by expanding its application scenarios, such as integrating with smart wearable devices to attain real-time emotion monitoring. Or incorporating multimodal information, including voice, text and other types of information to develop more comprehensive emotion analysis capabilities. Providing more precise and holistic support for mental health services.

References

[1] Xu Feng, Zhang Junping A review of facial micro expression recognition [J]. Journal of Automation, 2017, 43 (03): 333-348
[2] Xu Linlin, Zhang Shumei, Zhao Junli Constructing a Parallel Convolutional Neural Network for Facial Expression Recognition Algorithm [J]. Chinese Journal of Image and Graphics, 2019,24 (02): 227-236
[3] Cai Yubao, Li Defeng, Lian Haigen, etc Intelligent Classroom Teaching Evaluation System Based on Facial Expression Recognition [J]. Digital Technology and Applications, 2020, 38 (10): 147-149.
[4] Ge Jike, Liu Can Design and Implementation of Classroom Atmosphere Evaluation System Based on Emotion Recognition [J]. Office Automation, 2020, 25 (17): 43-45
[5] Jiang Sheng, Zhu Jianhong Facial micro expression recognition method based on ME ResNet [J]. Computer Science, 2024, 51 (S2): 292-298
[6] PFISTER T, LI X, ZHAO G,et al. Recognising Spontaneous Facial Micro-Expressions[C]||2011International Conference on Computer Vision.IEEE, 2011:1449-1456.
[7] HUANG X H, ZHAO G Y, HONG X P,et al.Spontaneous facial micro-expression analysis using spatiotemporal completed local quantized patterns[J]. Neurocomputing,2016,175:564-578.
[8] CNKI.Word.Helper.Entity.ItalicDOSOVITSKIY A, BEYER L, KOLESNIKOV A, et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale[C]//International Conference on Learning Representations Addis Ababa: ICLR, 2021: 1-17.
[9] Hu Kun, Dong Aihua, Huang Rong Facial expression recognition based on dual stream features and attention mechanism [J/OL]. Journal of Donghua University (Natural Science Edition), 1-12 [April 18, 2025]
[10] Hu Min, Zhu Hong, Wang Xiaohua, etc Expression recognition method based on gradient Gabor histogram features [J]. Journal of Computer Aided Design and Graphics, 2013, 25 (12): 1856-1861
[11] Kong Yinghui, Qi Tiancong, Zhang Shuaitong Design of Mobile Facial Expression Recognition System Based on Deep Learning [J]. Science, Technology and Engineering, 2020, 20 (25): 10319-10326
[12] Wu Yana Research on Learning Emotion Analysis and Application Based on Facial Expression Recognition [D]. Beijing University of Technology, 2022.
[13] Li Xiaolin, Niu Haitao Feature Fusion Facial Expression Recognition Based on VGO-NET [J]. Computer Engineering and Science, 2020, 42 (3): 500-509
[14] Cheng Xuejun, Xing Xiaofei Expression recognition method using improved VGG label learning [J].Computer Engineering and Design, 2022,43 (04): 1134-1144.
[15] Ekman.Facial expression and emotion[J].American Psychologist,1993,48(4):384-392.
[16] Gao Wen, Jin Hui Analysis and Recognition of Facial Expression Images [J]. Journal of Computer Science, 1997, (09): 782-789