

Facial expression classification

Zhuoyue Ren*

Lanzhou University Lanzhou, Gansu, China

*Corresponding Author Email: renzhy2018@lzu.edu.cn

Abstract. At present, emotion classification has become a hot topic in artificial intelligence pattern recognition. The facial expression recognition (FER) is indispensable for computers to understand the emotional information conveyed by expressions. In the past, using traditional features to extract and classify facial expressions has not achieved satisfactory accuracy, so the classification of facial emotions is still a challenge. The model used in the paper is an existing one - MINI_XCEPTION, the dominant framework for CNNs that extracts features from images to identify and classify seven facial emotions. The model was trained on a dataset of people's facial expressions (Kaggle). This model has a significant improvement over the previous model.

Keywords: facial expression recognition; Mini-Xception; OpenCV.

1. Introduction

Experts in machine learning believe people will react differently to different things and thus produce different expressions. They can detect the emotions they express about certain things by analyzing facial expressions, thus facilitating applications in many fields, such as intelligent control, security, medical, etc. As humans, we can distinguish humans' complex and variable emotions through our eyes. However, for a computer, determining emotions is challenging because different expressions often have only subtle differences. And due to the blurred expressions of the people in the pictures and videos, it is also difficult for computers to distinguish the emotions of human faces from the blurred background [1].

Previously, some methods of machine learning could be applied to emotion classification by scientists. Still, the performance in this area was not good because the emotions were too variable and complex. Since then, deep learning has been introduced to build multi-layer models through extensive training to effectively extract features from images for automatic classification, which has been very popular in those years. Now deep learning neural networks have features superior to other methods. The first is its powerful computational capability; the second is its extensive capabilities making it the most potent application [2]. American psychologists Ekman and Friesen, after extensive experiments and tests, defined human expressions into the following six categories: Angry, Disgust, Fear, Happiness, Sadness, and Surprise [3]. In order to distinguish them from no expressions, there is usually an additional category: Neutral, with a total of 7 basic expressions. This work classifies facial expressions into seven emotions. This database is from the 2013 Kaggle competition, and there is a certain amount of error as this database was mostly downloaded from a web crawler. The main framework of face expression recognition is divided into three steps: image pre-processing, face detection and expression classification.

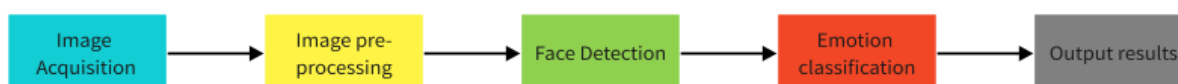


Figure 1. Emotion recognition steps

2. Face Detection

Every human being has a face and it is the most important feature of a person's appearance. With the development of technology, face recognition has become the most promising means of biological identity verification, and at the same time, face recognition technology has gradually matured and

gained popularity. In this paper, face recognition technology is the first one that needs to be used, and also the key to realize whether the expression can be successfully classified or not. The steps of face detection are as follows, first detect whether there is a human face in the input image or video, and if so, return the position, size and pose of the face.

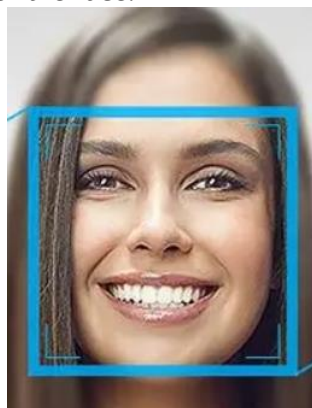


Figure 2. Face detection schematic

In this paper, after acquiring face images, detection is performed using a cascade classifier. In OpenCV, there is a very efficient method for detecting faces, the Haar cascade classifier, which is based on the following ideas.

- (1) Detection window is used to capture features related to images or videos
- (2) Determine whether there is a face in that window by the classifier.
- (3) Return the coordinates of the window with the face.
- (4) End Detection.

2.1. Haar characteristics

Haar features [4] are also known as Viola-Jones recognizers. Haar eigenvalues reflect the grayscale variation of the image. In our human faces, for example, the eyes and mouth are darker than their surroundings, and given this, we can use rectangles of different colors and sizes for the representation of facial features.

The processes of Haar feature extraction are as follows:

- (1) Extracting features from images using different templates
- (2) Filtering representative features
- (3) Classify using a strong classifier

Figure 3 illustrates several templates used for Haar feature extraction. Rectangles can quantify facial features, there are white areas and black areas, they have different number of pixels, add up the white pixels and subtract the sum of black pixels, the result is the facial feature value. As shown in Figure 4, all that needs to be done here is to place the rectangle on the face and perform the calculation.

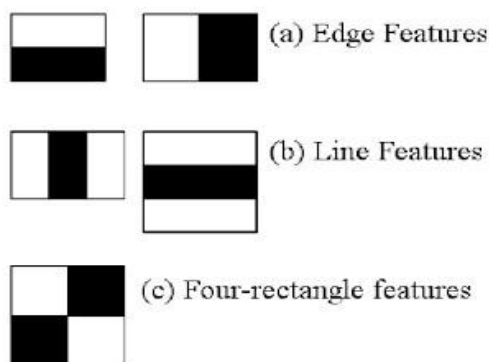


Figure 3. Feature template used for Haar features

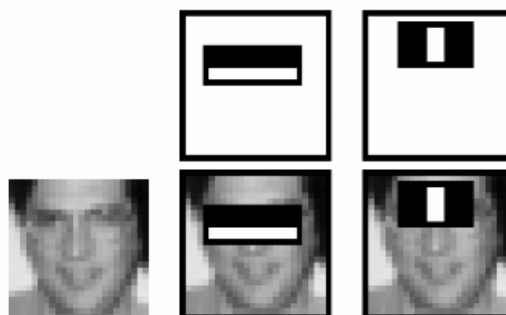


Figure 4. Demonstration of Haar feature extraction process

2.2. Cascade classifier

When face detection is required, OpenCV technology and Haar features can be used in combination, and AdaBoost algorithm needs to be used on this basis. Before using OpenCV for face detection, it is first necessary to build a classifier based on the AdaBoost algorithm. After connecting multiple classifiers in series, a stronger classifier can be formed. The face detection process based on the cascade classifier is shown in Figure 5.

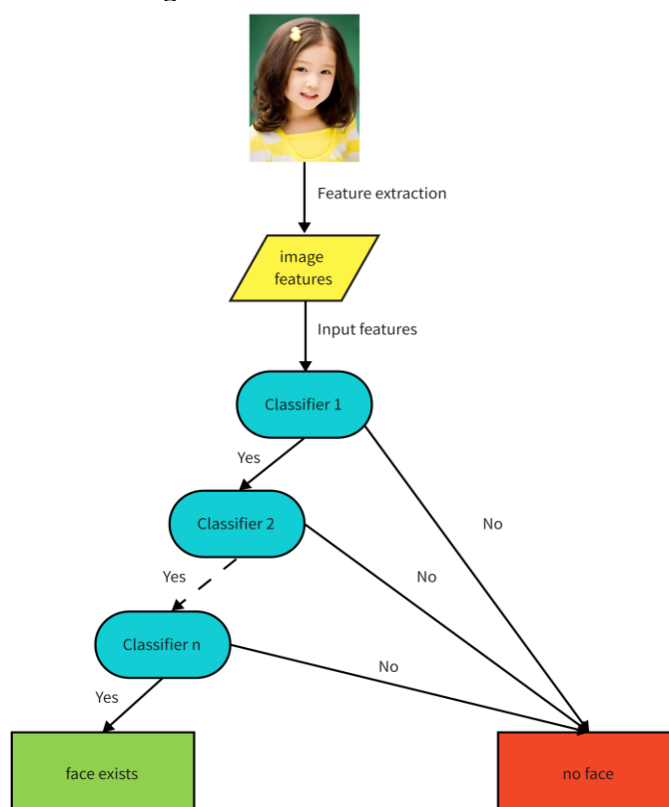


Figure 5. Face Detection Process Based on Cascaded Classifiers

3. Mini-Xception Network

3.1. Introduction to Mini-Xception Network

The traditional convolution operation is to map channel correlation and spatial correlation jointly, Szegedy et al [5] proposed the Extreme Inception module approach by assuming that the convolutional channel correlation and spatial correlation can be decoupled and mapping the two separately. In 2016 Chollet et al [6] proposed a further Inception model and used deep separable convolution [7] to replace the module in Inception, resulting in the Xception network structure. Xception is extremely similar to but slightly different from Extreme Inception in that the former first

performs regular convolution on the spatiality of each channel, and then The former performs a 1*1 point-by-point convolution for each channel. The middle part of the module is connected by residuals to reduce the training difficulty of the neural network.

Convolutional Neural Networks have many advantages, among which the ability of self-learning is the most valued by machine learning experts, but they still have problems such as long training time. In this paper, the network layers of the model can be simplified by removing the fully connected layers of the neural network. and construct a Mini-Xception network model for classification using deeply differentiable convolution instead of the traditional convolutional layer. The structure of Mini-Xception Convolutional Neural Network is shown in Figure 6.

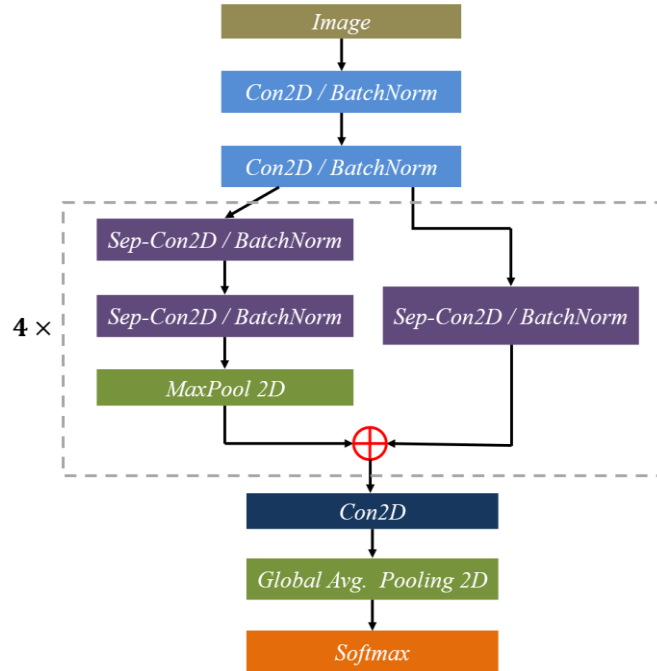


Figure 6. Mini-Xception network structure

The Mini-Xception network model is more compact and has fewer training parameters than the Xception network model, and therefore, the training time will be significantly reduced.

4. Materials and Methods

4.1. About the dataset

Using publicly available datasets saves time in collecting data on the one hand, and allows for a fairer evaluation of the model and the performance of the face expression classifier on the other hand. As shown in Figure 7, The data consists of 48x48 pixel grayscale images of faces. The test set defines the seven different expressions as different categories. In the training set, there are 28709 face images, while in the test set there are 3589 face images. Since the images in the FER2013 database are downloaded after crawling on the Internet, there is some error and the human accuracy of the FER2013 database is $65\% \pm 5\%$ [8].



Figure 7. FER2013 dataset

4.2. Pre-processing

First import the libraries and FER2013 data needed for the project. After that, the relevant variables are defined. `num_classes=7`: Classes to be processed when training the model, i.e. the number of types of emotions. `img_cols = 48`: the size of the image array fed into the neural network. `batch_size = 32`: 32 is the most appropriate value in this model. Subsequently, image enhancement is performed on these datasets. As shown in Figure 8, the `ImageDataGenerator` class can be used here to scale down the amount of data and enhance it in a way that improves image quality. The `train_datagen` variable artificially extends the dataset in the following ways:

- (1) `rotation_range`: Random rotation, 30 degrees is used here.
 - (2) `shear_range`: Shear strength (counterclockwise shear angle in degrees). Here, 0.3 is used as the shear range.
 - (3) `zoom_range`: The range of random scaling, here 0.3 is used as the zoom range.
 - (4) `width_shift_range`: Shifts a value across the entire width of the image.
 - (5) `height_shift_range`: This will shift a value over the whole height of the image.
 - (6) `horizontal_flip`: Flips the image horizontally.
 - (7) `fill_mode`: Fill pixels after changing the image with the above method
- Here, only the validation data is re-saved and no other scaling is done.

scaling, rotation, colors, noises...

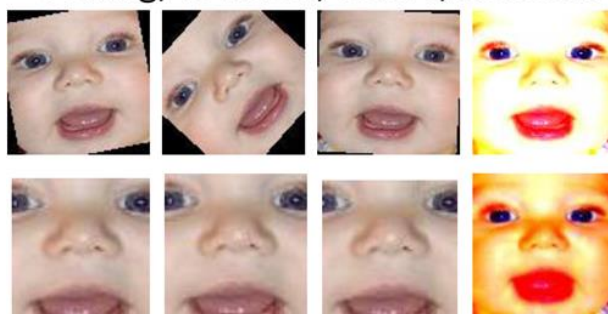


Figure 8. Data enhancement effect

4.3. Building a network model

When constructing the network model, first the expression recognition problem can be categorized as one of the classification problems, and after that the appropriate structure is selected based on the experience of constructing the classification problem model. Since the camera needs to be called for real-time expression recognition in this paper, the speed of recognition affects the user's experience. Therefore, the Mini-Xception model is constructed.

The processes of constructing the model are as follows:

- (1) Build Mini-Xception convolutional network model

- (2) Train the model using pre-processed expression data
- (3) Obtain the final model
- (4) Prediction of the pre-processed expression data using the previously trained model
- (5) The prediction results will be selected as the category with the highest post-prediction confidence
- (6) Use Softmax activation function to output 7-dimensional feature vectors representing 7 expressions

Figure 9 illustrates the process of building the model.

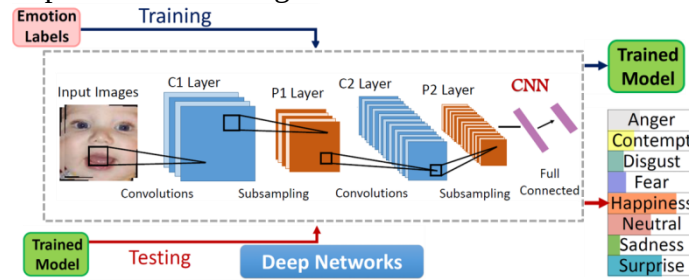


Figure 9. Model building process

5. Experimental procedure and results

5.1. Model Training

For the selection of datasets, the validation set is drawn from 20% of the test set and the training set is selected from the remaining 80% of the test set. The training parameters are set as shown in Table 1.

Table 1. Training parameter settings

Parameter	Value
Batch Size	32
Epoch	100
Classes	7

In the process of optimizing the network parameters, the Adam optimizer [9] is chosen because of the large number of parameters to be processed. the Adam optimizer has properties such as fast convergence, easy parameter tuning, invariance of the diagonal scaling of the gradient, and solution containing large-scale data and parameters. Adam's parameter update formula is

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t}} \tilde{m}_t \quad (1)$$

In the above equation, θ is the parameter representing the update, η represents the learning rate, $\tilde{m}_t$ represents the mean value of gradient first moment and $\sqrt{\hat{v}_t}$ represents second moment variance. In this paper, since expression recognition is a classification problem, it is appropriate to use the cross-entropy loss function as the loss function in terms of optimization of the model. The equation of the cross-entropy loss function is as follows.

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m \left[y^i \log(h_{\theta}(x^i)) + (1 - y^i) \log(1 - h_{\theta}(x^i)) \right] \quad (2)$$

When using cross-entropy as the loss function for Softmax classification, it is not only very suitable but also better than other loss functions in terms of results, so the classification problem of deep learning is generally using cross-entropy as the loss function.

5.2. Analysis of results

The accuracy rate is constantly rising, and the accuracy rate is constantly improving. The obtained training and loss curve are shown in Figure 10 and Figure 11. It is also more intuitive to see the different accuracy rates for each expression by using the confusion matrix in Figures 12 and 13.

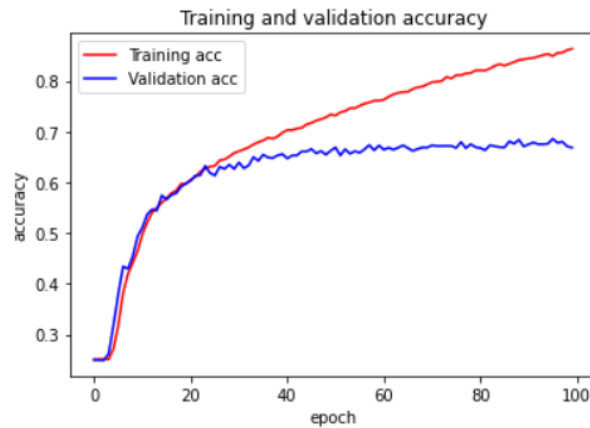


Figure 10. Training and validation accuracy



Figure 11. Training and validation loss

The confusion matrix is used below to show the final result.

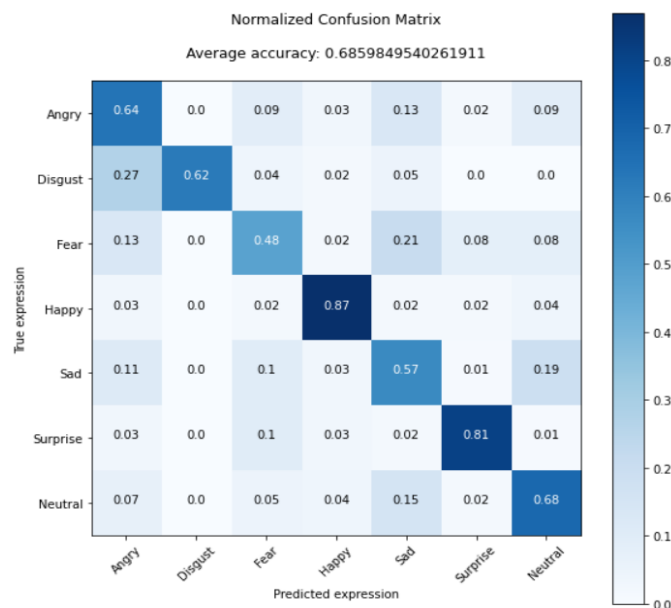


Figure 12. Normalized Confusion Matrix

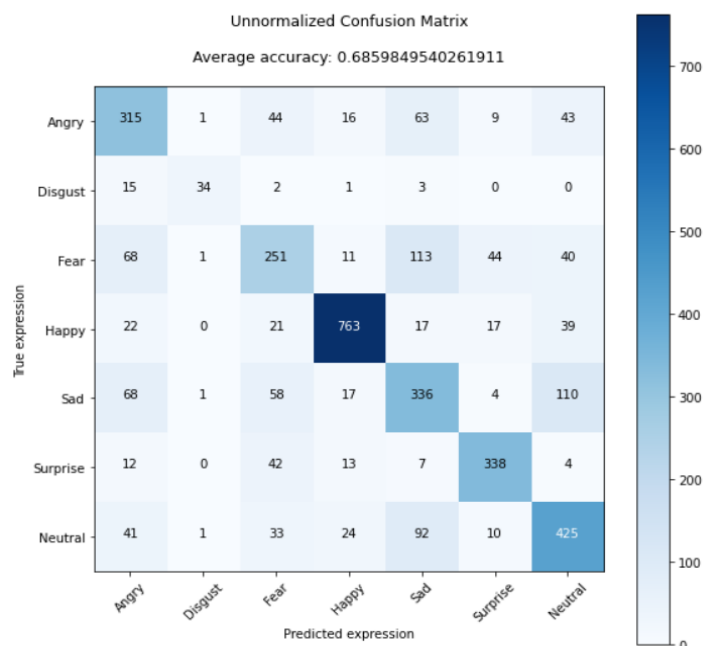


Figure 13. Unnormalized Confusion Matrix

It can be seen from the results shown in the confusion matrix that after training, the accuracy of the 7 expressions in the test set is 64%, 62%, 48%, 87%, 57%, 81% and 68% in order, and finally the accuracy of the Mini-Xception model can be calculated as 68.59% in the whole test set. The system was found to be the most accurate in classifying angry and surprised expressions, while the classification of sad and scared expression pictures was not predicted well enough. The reason may be that the number of sad and scared expressions is relatively small compared to the other categories. It was also found that the system had problems classifying some expressions, e.g., some people would have downwardly furrowed eyebrows and open mouths on their faces, but the system would recognize the expressions that were originally sad and classify them as happy expressions, because the five senses of happy and sad expressions changed in a similar way. The results of this classification show that human expressions are very complex and some subtle changes can affect the results, so the model needs to be modified to improve the accuracy.

5.3. Application Testing

In order to better apply to daily expression recognition scenarios and test the actual expression recognition performance, the expression recognition process under actual scenarios is designed by combining the face detection and expression recognition algorithms introduced in previous chapters, and its recognition process is shown in Figure 14.



Figure 14. Actual scene expression recognition process

Firstly, the image is acquired and pre-processed, then the face position is detected using Haar cascade classifier, the face position region is recognized by using Mini-Xception for expression recognition, and finally the result of expression recognition is displayed and output. According to the above process, this paper designs an expression recognition system interface based on PyQt5. The general functions of the system are as follows:

- (1) Recognizing based on the model file after it can be selected.
- (2) Open the camera to recognize the face expressions in the real-time screen.
- (3) Selecting a face picture and recognizing the expressions in it.

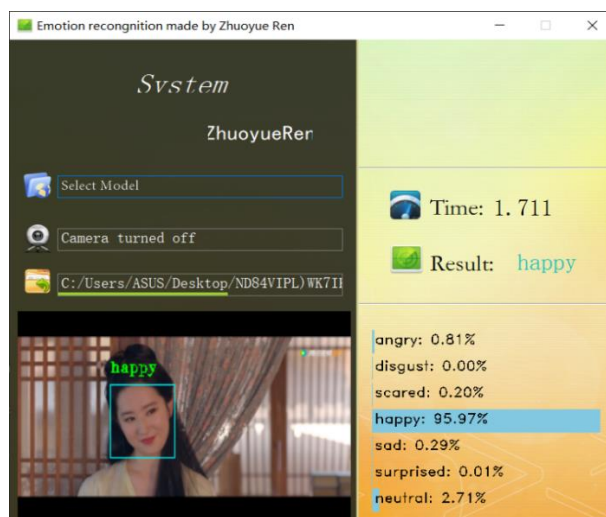


Figure 15. Emotion recognition system picture test-1

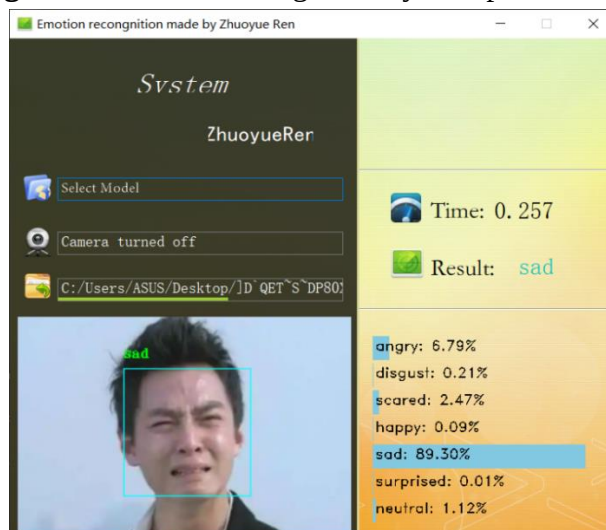


Figure 16. Emotion recognition system picture test-2

With the real-time video scene turned on, users can see the real-time results on the test page. It can be found that the system is very fast in recognizing expressions. This shows that the efficiency of the previously constructed Mini-Xception model is very high, which can significantly cut down the time spent on recognition and optimize the user experience.

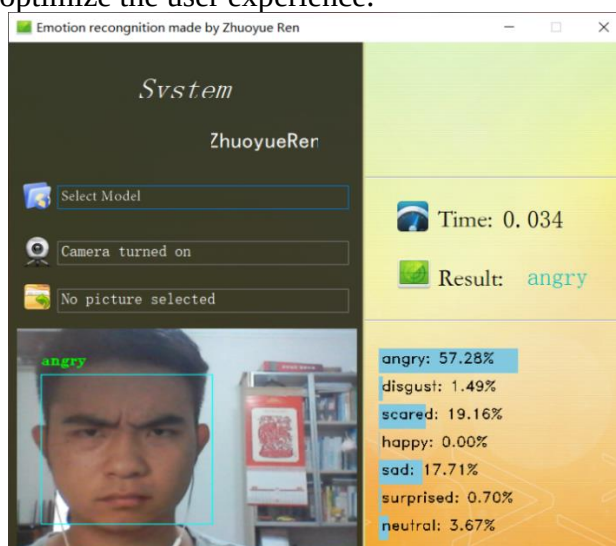


Figure 17. Expression recognition in real-time video

From the recognition results of real-time expressions, it can also be found that the situation that subtle expression changes will affect the final classification as mentioned above will also occur. For example, when the corners of the mouth are slightly raised and the teeth are missing, the expression is not recognized as happy but as normal, and the eyes need to be closed and the eyebrows need to be slightly tilted down to be correctly recognized. Also, when wearing glasses, the frames of the glasses can easily be perceived as furrowed eyebrows, so a normal expression may also be perceived as an angry expression. Therefore the model needs further improvement [10].

6. Conclusions

This paper presents an emotion recognition system. First, a data enhancement operation was performed, followed by face detection using the Harr cascade classifier. A Mini-Xception model was also constructed, simplifying the number of modules in the convolutional layers and removing the final fully connected layer, which greatly improved the recognition speed. The model was then trained so that users can add photos locally for expression recognition or by using a computer camera to accomplish expression recognition.

Also, after this research experience, there are many improvements that can be made to the model. For example, during the previous review, it was found that other researchers pre-process the images to remove the background irrelevant to feature extraction (e.g., ears), and then adjust the facial components (eyes, nose, mouth) to the same position and feed the data into the model. This improves the efficiency of recognition and increases the accuracy.

References

- [1] N. Jain, S. Kumar, A. Kumar, P. Shamsolmoali, and M. Zareapoor. Hybrid deep neural networks for face emotion recognition. *Pattern Recognition Letters*, 115: 101–106, 2018. Multimodal Fusion for Pattern Recognition. 1, 3
- [2] S. A. Hussain and A. S. A. A. Balushi. A real time face emotion classification and recognition using deep learning model. *Journal of Physics: Conference Series*, 1432(1): 012087, jan 2020. 1
- [3] P. Ekman, W. V. Friesen. Constants across cultures in the face and emotion [J]. *Journal of Personality and Social Psychology*, 1971, 17(2): 124-129.
- [4] P. Viola, M. J. Jones. Robust Real-Time Face Detection [J]. *International Journal of Computer Vision*, 2004, 57(2): 137-154.
- [5] C. Szegedy, W. Liu, Y. Q. Jia, et al. “Going deeper with convolutions.” *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.
- [6] Chollet F. “Xception: Deep learning with depthwise separable convolutions.” *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
- [7] M. Wang, B. Liu, H. Foroosh. “Design of Efficient Convolutional Layers using Single Intra-channel Convolution”, *Topological Subdivisioning and Spatial “Bottleneck” Structure [EB/OL]*. [2020-03-15].
- [8] I. J. Goodfellow, D. Erhan, C. P. Luc, et al. “Challenges in representation learning: a report on three machine learning contests [J]. *Neural Networks*”, 2015(64): 59-63.
- [9] D. P. Kingma, J. L. Ba. Adam: A method for stochastic optimization [J]. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015: 1-15.
- [10] I. Song, H. Kim, and P. B. Jeon. Deep learning for real-time robust facial expression recognition on a smartphone. In *2014 IEEE International Conference on Consumer Electronics (ICCE)*, pages 564–567, 2014. 2