**Facial Expression Recognition**

***1.Abstract***

In this study, we built a model that could efficiently analyze and recognize facial expressions. This study includes data pre-processing, data visualization, model prediction, which mainly focus on CNN. We also incorporates our model into a facial expression recognition system with a GUI. In this system, we mainly realize two functions: real-time facial expression recognition and image recognition.

***2.Introduction***

Facial expression recognition is the process of detecting human emotions from facial expressions. The human brain is able to recognize facial emotions automatically in practice, but software has now been developed that can recognize facial emotions as well. It is “Artificial Intelligence” that detects and studies different facial expressions to utilize them. This technology has been maturing and has a great variety of applications in practice.

Recognition of facial emotions has widespread consequences for society and business. For instance, facial expression recognition technology would play a significant role in security checks or maintaining social order under the controls of governmental organizations to prevent terrorism. In fact, in terms of business, companies have also been taking advantage of facial expression recognition, and consequently either optimiza their product or design derive products of facial emotion recognitions. Disney utilized facial recognition technology to figure out the emotional responses(timespot of laughing) from the audience during the release of Toy Story 5, which would be in favor of having better feedback and analysis of their movies. In addition, there are companies that use AI with facial emotion recognition technology as HR assistants, which is helpful in determining whether the job candidate is keeping honest during the interview and interested in the position by evaluating facial expressions. Besides providing feedback, facial expression recognition itself is a decent selling point as well. Apple released a feature on the iPhone called Animoji, where iPhones could capture facial expressions from customers and simulate emoji to mimic facial expressions as different animal roles.

Therefore, it is worth studying facial expression recognition and exploring the model performance. In this project, we will aim to detect, recognize and analyze facial expressions from images and real-time videos and study the performance of a CNN framework on facial expression recognition issues.

***3.Dataset***

Kaggle FER2013 is a dataset for facial expression analyzing competition. It includes 28709 training samples, 3859 validation samples(public) and 3859 testing samples(private). There are 35887 data samples in total, which can be categorized by 7 labels: anger, detest, horror, delight, sadness, astonishment and neutral. The resolution ratio of each image is 48×48.

Due to the origin of this dataset, this dataset can get around 65% artificial accuracy because there are some linear and non-linear rotations and some blocking objects like hand and hair in each image.

In the raw dataset, we have three columns: the label, 48 \* 48 pixel gray scales representing the image and the usage (train or test) in each row. The labels are represented by the column “emotion” ranging from 0 to 6 representing “anger”, “disgust”, “fear”, “happy”, “neutral”, “sad” and “surprised” separately.

4.***Data Pre-processing and Visualization***

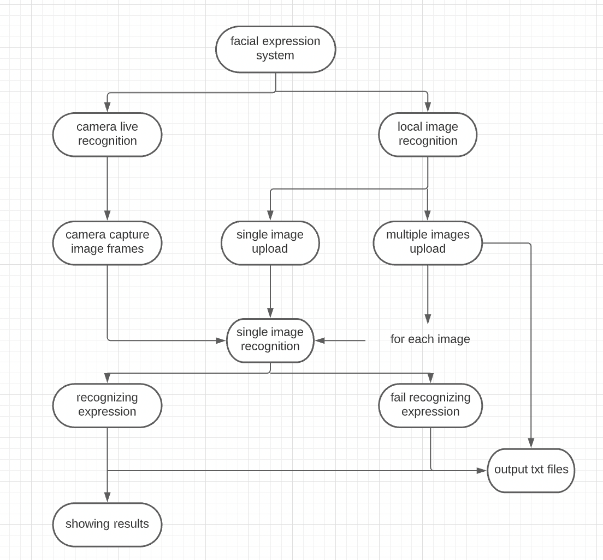
In this project, we use numpy and pandas to preprocess our raw data. After reading the raw csv file, we use a np.array to store each image and use pd.get\_dummies to store labels. The dataset can be visualized as Figure 1. We can see that this is a visualized  dataset, with each label as column names and each column containing the corresponding expression images.

**Fig. 1: FER 2013 dataset**

Then we use train\_test\_split from sklearn library to split the dataset into training set and testing set. Note that this approach will split the dataset with a random ratio.

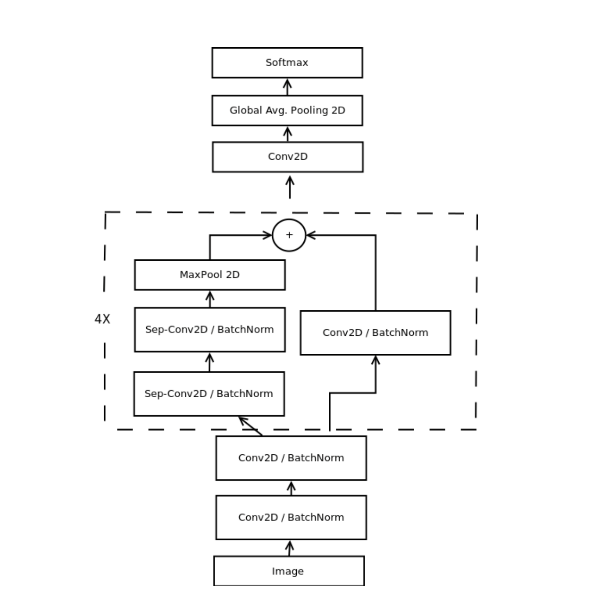
***5.Model and Methodology***

**1. Main framework:** As mentioned above, we aim to realize two main functions concerning facial expression recognition. The first one is the real-time facial recognition. When the user turns on the camera, we firstly detect the human face in the camera, and then recognize the facial expression based on our model. Our system will then show out the recognition results with a frame characterizing the human face. Specifically, we will show the probability of each facial expression label and take the highest probability as a result. The second one is the raw image facial expression. This is similar to the first one, only changed from real-time camera to raw image. The flow chart of the system is shown in Fig 2.



**Fig 2: flow chart of FER system**

**2. Supervised learning:** In this project, we use the basic approach of supervised learning to achieve our goal of facial expression recognition. What we are basically trying to do is: 1. Load the FER-2013 dataset and do preprocessing. 2. The preprocessed data is divided into two parts: feature and label. The feature is sent to the model, and the label is used as ground-truth. 3. The model receives features as input, and outputs predict through a series of operations. 4. Create a loss function Loss(MSE in this case) by taking predict and predict as variables. The function value of Loss indicates the gap between predict and ground-truth. 5. Establish the Optimizer optimizer. The goal of optimization is the Loss function, so that its value is as small as possible. The smaller the value, the higher the accuracy of Model prediction. 6. During the optimization process of Optimizer, the Model changes the weights of its own parameters according to the rules. This is a repeated loop and continues the process until the loss value stabilizes, and it is impossible to obtain a smaller value.



**Fig.3: mini\_XCEPTION**

**3.mini\_XCEPTION:** We use the main CNN framework mini-XCEPTION in this work as our model. The basic frame of the model is shown in Fig 3. In mini-XCEPTION, we need

|  |  |  |
| --- | --- | --- |
| Parameters | Model1 | Model2 |
| batch\_size(size of each batch) | 32 | 32 |
| nrows(size of data) | 1000 | 35000+(all) |
| num\_epoches | 100 | 200 |

**Table 1: model1 and model2**

to calculate the dimensional change. As shown in the following:

In the above two formulas, W2 is the width of the Feature Map after convolution; w1 is the width of the image before convolution; F is the width of the filter; P is the number of Zero Padding, and Zero Padding refers to the number of circles around the original image. , If the value of P is 1, then add 1 circle 0; S is the stride; H2 is the height of the Feature Map after convolution; H1 is the width of the image before convolution. The upper Equation and the lower Equation are essentially the same. Note that we also use BatchNormalization() in this framework, with which we can accelerate convergence and avoid overfitting, also make large learning rates possible.

In our model, we also use Keras to achieve image augmentation. The training of neural network requires a lot of data. The amount of data determines the height that the network model can reach. The network model is as close as possible. Near this height. Therefore, after loading the data, before data training, the data can be enhanced. To prevent the network from over-fitting too quickly, some image transformations can be done artificially, such as flipping, rotating, cutting, etc. The above operation is called data augmentation.

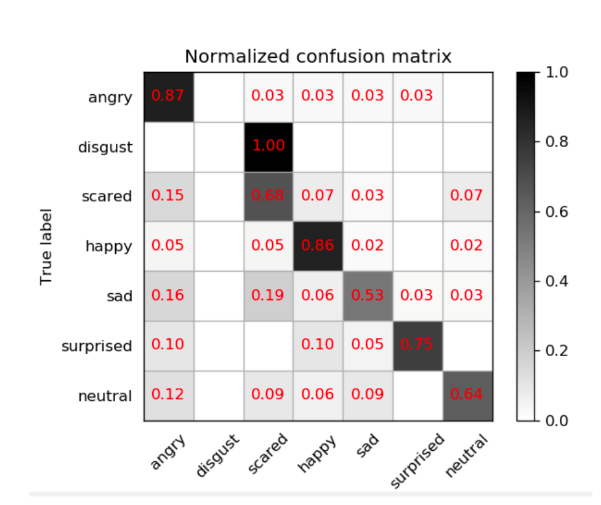
Another big advantage is to expand the amount of data in the database, making the trained network more robust.

The main steps of our model can be described as follows:1. Load the data set; 2. Establish a neural network model; 3. Train the model and save the model parameters; 4. Forecast.

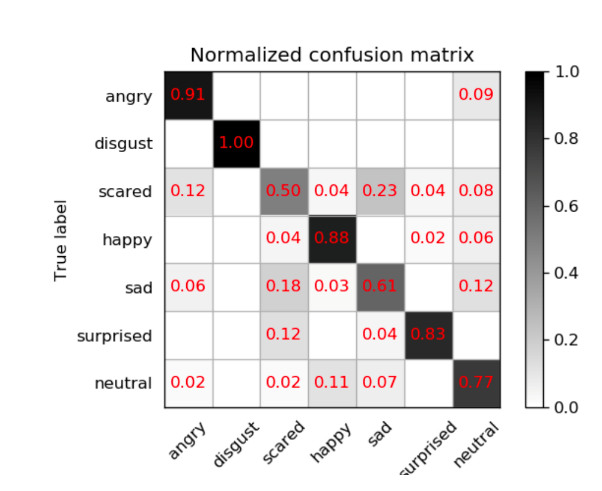
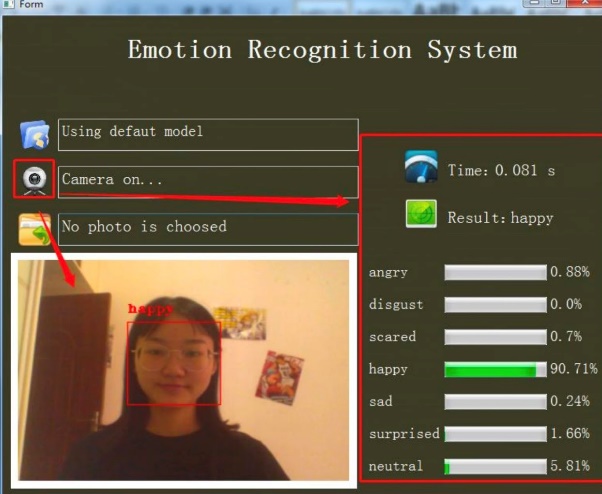
***6.Result***

Due to the scale of the dataset, we actually did two trainings and generated two models in our experiment, which is shown in Table 1.

We use confusion matrix to represent the performance of each model. Every column in the confusion matrix represents the predicted label, while each row representing the ground truth. Sum of each row represents the number of true values in the predicted label, while each column representing the number of predicted values in the true label. The confusion matrix of model 1 and model 2 are shown in Fig 4 and Fig 5.

****

**Fig. 4: confusion matrix of model 1**

****

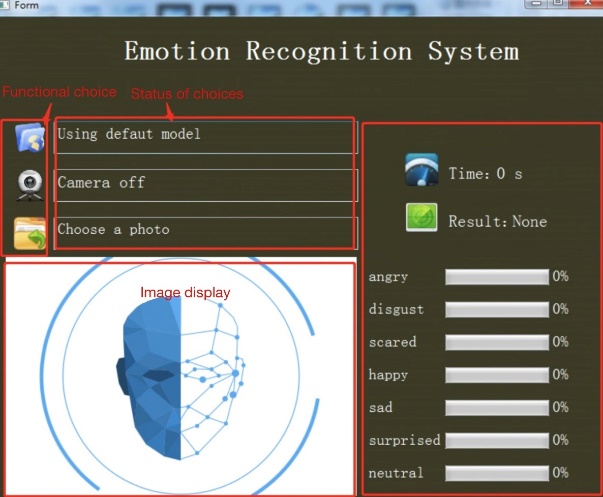
**Fig. 5: confusion matrix of model 2**

As we can see, the accuracy of model 1 is 0.62, “angry” has the best result and “disgust” the worst, this may have something to do with the dataset and batch normalization. The accuracy of model 2 is 0.79, we can see as we increase the data size, we get a better result.

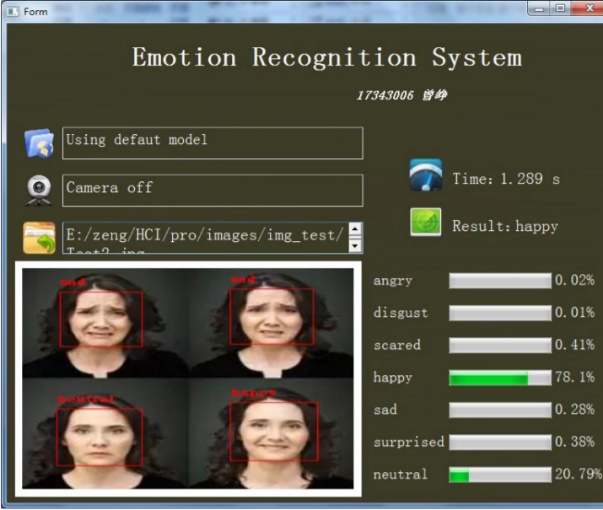
Then we can use our model to write a facial expression recognition system with a GUI. The interface of our system is shown in Fig 6. We can choose the preferred model and choose different functions in the GUI.

The results of real-time facial expression recognition are shown in Fig 7.

The results of image facial expression recognition are shown in Fig 8. In our real experiments, we didn’t encounter any false predictions within our test case scales.

****

**Fig. 6: GUI of FER system**

**Fig. 7: result of real-time recognition**

**Fig. 8: result of image recognition**

***7.Conclusion***

In this project, we used FER 2013 dataset to train a CNN model and incorporated this model into a facial expression recognition system with a GUI, which realized two functions: real-time and image facial expression recognition. We achieved the best accuracy of around 0.8 in our model. However, in our real experiments, we didn’t encounter any false predictions within our test case scales, which may be the result of dataset choice.

***8.Future Work***

Although our system shows a decent performance, there may still be parts that need to be optimized and improved and we needed to include do more experiments with respect to:

1) Due to the diversity of people, the appearance of faces, expressions and skin colors may be different, with mode variability, and the accuracy of facial recognition is not always very stable.

2) It is difficult to exclude the interference of face beard, eyes and other factors when performing feature extraction on the face again. The existence of these interference factors will reduce the accuracy of the experimental results;