CNN Based Facial Expression Recognition System

Yahong Zhu

A Dissertation

Presented to the University of Dublin, Trinity College

in partial fulfilment of the requirements for the degree of

Master of Science in Computer Science (Augmented and Virtual Reality)

Supervisor: Michael Manzke

August 2022

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

August 18th, 2022

Acknowledgment

First of all, I would like to thank my parents and my brother. Thank you to my family for raising me and giving me their unconditional love. Whenever I needed you, you were there for me, supporting me, encouraging me, and helping me through the uncertainties and difficulties in my life. It is because of you that I can continue to gain the courage to move forward into the future.

I would like to thank Professor Michael Manzke for teaching me and his guidance and supervision over the past year.

Finally, I would like to thank my college Trinity for educating me. I have spent the most precious year of my life here, and my college has provided me with a wonderful atmosphere to have the spirit of exploration and grow my knowledges.

Yahong Zhu

University of Dublin, Trinity College

August 2022

CNN Based Facial Expression Recognition System

Yahong Zhu, Master of Science in Computer Science
University of Dublin, Trinity College, 2022

Supervisor: Michael Manzke

Facial expression plays an important role in human communication. Recognizing facial expressions helps robots to understand human emotions better and give responses precisely. It is challenging to recognize human expression accurately since it involves multiple disciplines such as psychology, computer science, cognitive science, engineering, and sociology. Many approaches have been proposed for solving facial expression recognition problem, among those methods, convolutional neural networks (CNNs) attract major attention because of their ability to extract features automatically and extraordinary performances. This paper proposed a CNN architecture inspired by VGG-16 which achieved a good result in solving the FER problem. In addition, this paper presents the design and implementation of a FER system using this CNN model.

Summary

The paper is organized as follows: The first chapter will be a brief introduction to the topic. The second chapter will describe the significance of the topic and the research background. It will give a description of the state-of-the-art in facial expression recognition, as well as technologies used in the FER system. Then, chapter 3 will present the design of the FER system with figures and tables. Chapter 4 will show the implementation of the system. Chapter 5 will focus on the results of the experiments and analysis with data and figures. Finally, chapter 6 and chapter 7 will be the conclusion of the work in this paper and future work to conduct.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Research into facial expression recognition (FER) technology is one of the main directions in the field of human-computer interaction. Facial expressions play a key role in conveying emotions and psychological conditions in people's daily communication. The efficient recognition of human facial expressions through certain algorithms will significantly improve the efficiency and quality of human-computer interaction. In addition, with the rise of 5G technology, facial expression recognition also plays a key role in related industries such as smart healthcare and smart driving and has high social values. Current expression recognition technologies include the recognition of static and dynamic facial expressions. The main object of research in this paper is facial expression recognition based on static images.

In the field of artificial intelligence, deep learning attracts the most attention because of its widespread use in solving recognition problems. Convolutional neural networks in deep learning are commonly used in voice and image recognition, as well as natural language processing and, have achieved good results. Various CNN architectures have been continuously proposed by researchers. Using CNNs for solving facial expression recognition problems to improve the efficiency and accuracy of human-computer interaction is a challenging task that needs attention.

This paper proposes a CNN architecture that improves on VGG-16 to solve the static facial expression recognition problem. To enhance the recognition accuracy, a new training dataset is constructed by mixing the FER+ and CK+ datasets with data augmentation to address the data imbalance problem.

Furthermore, a static facial expression recognition system based on this CNN model is designed and implemented in this paper. The system contains two recognition modes which are the photo expressions recognition mode and upload recognition mode. The performance of the model and the system is tested respectively in this paper. The experiments prove that the static system yields a good result in facial expression recognition.

Chapter 2

Background

2.1 Research Significance of FER

In the highly developed field of artificial intelligence, deep learning, human-computer interaction technologies have attracted many scholars to participate in research. As an important fundamental of human-computer interaction domain, facial expression recognition is one of the most popular research topics, with high studying value and broad application potential.

In 1971, the well-known psychologist Mehrabian proposed the “7%-38%-55% rule” in his research, which states that human emotion communication is based on expressions, sounds and words [1], which states that emotional communication between humans is based on three main methods: expressions, sounds and words. Of these, facial expressions convey 55% of the information, sounds 38%, and words 7%

Studies have shown that facial expressions are more accurate and reliable than words when there is a discrepancy between the verbal and physical expressions. This shows that facial expressions are able to convey more emotional information, and an accurate recognition of facial expressions allows the computer to understand the human mind faster and more precisely, enabling a more efficient human-computer interaction process.

Facial Expression Recognition (FER) is the process of classifying images by extracting features from images of facial expressions using a computer and specific algorithms. Facial expression recognition is an interdisciplinary subject that brings together psychology, computer science, cognitive science, engineering, sociology, and other disciplines, which involves many aspects of social development.

In recent years, hardware technology has experienced continuous innovation and breakthrough in performance. GPUs are widely used in computing devices, making data processing much faster and less costly. As a result, data acquisition, storage and processing have been made much easier, and data such as images, video and text has explosively increased, bringing technology into the era of Big Data. Such an environment has led to the rise of deep learning, and a number of Convolutional Neural Network (CNN)-based models have been popping up, such as AlexNet [2], which won the ILSVRC competition in 2012, GoogLeNet [3], which won the championship in 2014, and VGGNet [4], which was the runner-up.

The convolutional neural network is capable of learning and extracting features from images in a more sufficient way, especially in extracting global information and understanding the context. It is an outstanding advantage for solving facial expression recognition problems, which helps achieve greater accuracy in recognition compared with traditional machine learning approaches, hence it is widely used in many practical scenarios.

Nowadays, facial expression recognition technology has been applied to many domains, such as:

Smart Driving For the application in smart driving, the on-board camera captures real-time images of the driving conditions of the driver, by detecting the face, analyzing the facial expression and making diagnoses. When the driver is fatigued, disturbed or drunk, the recognition system will alert the driver and automatically alarm to ensure the security, which reduces the occurrence of traffic accidents.

Human-computer Interaction The human-computer interaction technology allows users to interact with machines through languages, facial expressions, or body language, rather than using the traditional interaction tools such as mouse, keyboard and touch panel. The application of FER technology gives computers the ability to understand various human emotions and respond accordingly, enabling a more effective and friendly interaction process.

Social Network VR Applications With the rapid evolution of virtual reality (VR) technologies, social network VR applications involving human or human-like avatars have become more and more popular. Many VR programs have been released into the market, including Vtime, VRchat, AltSpaceVR, and High Fidelity VR [5]. To give users a new immersive experience, research was conducted to apply more accurate facial expression recognition and visualize them on an avatar in VR[6, Vol. 8].

2.2 State-of-the-art in FER

In 1872, biologist Darwin first mentioned the concept of facial expressions in his book *The Expression of the Emotions in Man and Animals* [7], pointing out that humans shared expressions in general and were in some ways identical to animals.

In 1971, American psychologists Paul Ekman and W.V. Friesen classified human emotions into six general categories: anger, disgust, fear, happiness, sadness, and surprise, and developed a Facial Action Coding System (FACS) based on this [8]. The system contains 44 basic action units (AU), which are the basic units of human expressions, were used to study the connection between facial muscle movements and facial expressions [8].

A facial expression recognition system is composed of three parts: face detection, feature extraction and expression classification.

Traditionally, the majority of research have presented their approaches with hand-crafted feature extraction algorithms and classifiers like MLP (Multi-layer Perceptron Model), SVM (Support Vector Machines), and k-NN (k-Nearest Neighbours) [9, p. 102]. These classifiers use handcrafted features including texture and face landmark data [9, p. 102]. In 2003, Wen [10] proposed an algorithm for feature extraction based on human appearance features. The appearance changes were extracted as a series of multi-scale and multi-orientation coefficients using Gabor wavelets, and then pre-processing the texture extraction region based on a ratio image, thus minimizing the impact of illumination and facial disparity on feature extraction. In 2007, Pantic M [11, pp. 390–395] proposed a method for locating facial points that identify facial expressions based on the geometric features of facial features and silhouettes. Feng [12] used the local binary pattern (LBP) method for texture classification and face recognition. It was capable of describing the facial appearance and representing facial expressions. Then a coarse-to-Fine expression classification was applied, which includes two-stage of classifications, resulting in better performance than other algorithms on the JAFFE database. In 2012, Wang Z [13] introduced a new approach based on sparse representation and local phase quantization (LPQ). Features are extracted using the LPQ descriptor. The test expression image was represented by a linear combination of the training expression images using the Sparse Representation-based Classification (SRC) algorithm.

In those traditional methods, facial features were manually extracted, whereas convolutional neural networks are able to learn such features automatically by their own. CNNs even performed better than human since they can detect and identify hidden pattern that the human eye could not see [14], [15].

Hence in current state-of-the-art, Khor H Q et al. proposed ELRCN (Enriched Long-term Recurrent Convolutional Network) [16] for expression recognition, where the differential matrix of optical flow and the grey-scale image are fed into a VGG-16 network, followed by the addition of a fully connected layer, and finally the LSTM algorithm is applied for classification. AlexNet [17] is a CNN architecture that won the ImageNet challenge in 2012, achieving a 15.3% error, less than the second-place 10.8%. This was the first time a dropout layer was introduced which tackled the over-fitting issue [18]. In ImageNet 2014 challenge, GoogLeNet [3], a new architecture of CNN inspired by LeNet and implemented by Google, won the classification and object recognition challenges, achieving a top-5 error rate of 6.67%. VGG-16 [4], proposed by K. Simonyan and A. Zisserman from the University of Oxford, achieves 92.7% accuracy (second place) in ImageNet 2014 challenge.

One of the largest issues of using CNN to solve the FER problem is that the complexity and the depth of the network. Overcomplicated models can increase the computational complexity and lead to over-fitting. To avoid problems above, a CNN architecture inspired by VGG-16 is proposed in this paper for giving a solution of FER, and it is trained on a fusion dataset which mix FER+ and CK+. The result reaches a high accuracy, east to train CNN model compared to VGG-16.

2.3 Technology in FER System

2.3.1 Introduction of Convolutional Neural Network

Convolutional Neural Network (CNN) is a type of artificial neural network model that introduced convolutional layer and pooling layer in the traditional network model. Compared with traditional neural network structures, CNNs have a significant reduction in computational complexity with remarkable feature extraction ability due to the weight-sharing feature. They are therefore widely used to solve image recognition and classification tasks, which have become one of the most popular topics in Computer Vision in recent years.

When training and testing images based on CNN, the images input to the network go through a series of feature learning and classification structures, with the main steps of the process as follows:

i. The image is input into the network through the input layer.
ii. The input image is convolved by the kernel of the convolutional layer to create an activation map for feature extraction
iii. The result of the convolution layer is a feature map of the original image, which is then passed to the pooling layer, where the features are down-sampled and summarized.
iv. The down-sampled feature map is sent to the fully connected layer.
v. The Softmax classifier is used to classify the result and output the recognition result.

The structure of a convolutional neural network is shown in Figure 2.1. The principles of its main structure are described in detail in the following section.

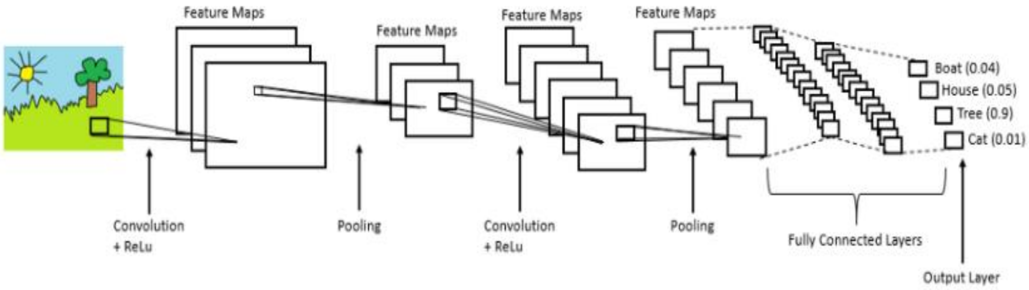


Figure 2.1: The Structure of A Convolutional Neural Network

Convolutional Layer The convolutional layer is the basic component of the CNN architecture for the extraction of image features. Feature extraction is generally performed by a combination of convolutional operators and activation functions in the CNN. The convolution operation is a particular linear operation. Each convolution layer contains multiple convolution kernels (local receptive fields) of variable sizes [19]. The entire original image is input as a tensor and processed by the convolution kernels to extract its features and create its feature map. Applying multiple convolution kernels yields a number of feature maps, thus different convolution kernels can be considered as different feature extractors, as shown in Figure 2.2.

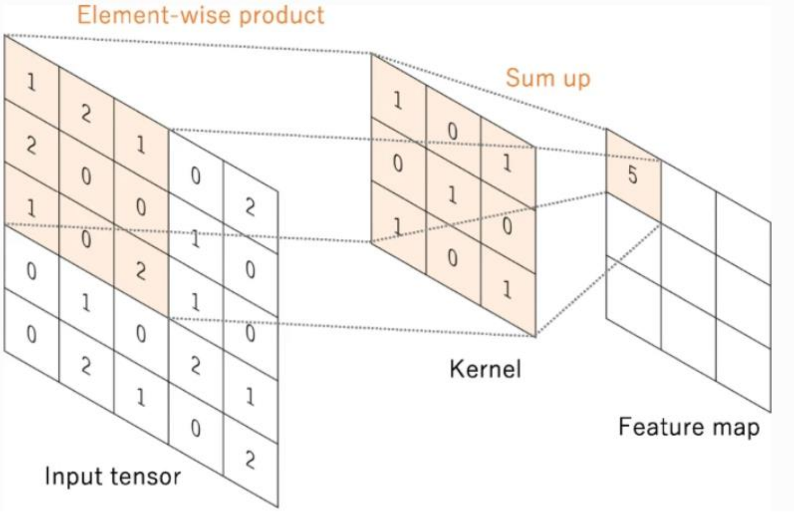


Figure 2.2: Convolution Operation

The input image is input to the convolution layer as a tensor. At each position in the tensor, the products of each element in the convolution kernel with the elements among the input tensor are calculated and summed to obtain the output value at the corresponding position in the output tensor, namely the feature map. The two essential hyperparameters that define the convolution operation are the size and number of the kernels. The size of the convolution kernel is usually 3 x 3 pixels, sometimes set to 5 x 5 pixels or 7 x 7 pixels. The number of convolution kernels is arbitrary, and it determines the depth of the output feature map.

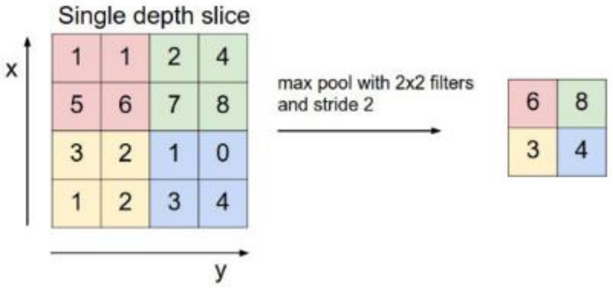
Pooling Layer Pooling, also known as down-sampling, is a layer added to CNN after the convolutional layer for feature down sampling. As the number of layers increases, the convolutional layer tends to extract more and more features, making it too computationally expensive and likely to be over-fitted. Hence, a pooling layer is added after a convolutional layer to reduce the size of the output matrix from the convolutional layer to retain as many important feature regions as possible to achieve translational invariance, i.e., a small amount of image translation will not affect the majority of the output values. The common pooling methods are Average Pooling and Max Pooling. Average Pooling calculates the average value in the neighbourhood of a location on the feature map, whereas Max Pooling (Figure 2.3) calculates the maximum value in the neighbourhood of a location to represent the feature at that location.

The most frequently used method is Max Pooling, due to the fact that pixels with higher feature values have higher activation and are therefore most likely to be acquired during down-sampling. Maximum down sampling extracts the most striking features in the feature map while reducing the computational load. The pooling layer usually uses a window size of 2*2 pixels and a stride of 2 pixels. The detailed formula for calculating the maximum in the feature map is:

$$\mathrm{y}_{\mathrm{i,j}}^{\mathrm{k}} = max_{0\le m,n\le s}\{x_{i*s+m,j*s+n}^{k}\}$$

and the formula for calculating average is:

$$\mathrm{y}_{\mathrm{i,j}}^{\mathrm{k}} = mean_{0\le m,n\le s}\{x_{i*s+m,j*s+n}^{k}\}$$



1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

6	8
3	4

Figure 2.3: Max Pooling Operation

Fully Connected Layer Fully Connected Layers (FC) is a deeply connected neural network layer, the most fundamental structure in a neural network, with fully connected attributes, i.e. each node is connected to the node in the previous layer. In a CNN model, the fully connected layer performs feature integration on the output of the pooling layer which contains distributed features, transforming the higher-level features into a 1D feature vector through a linear transformation, and then connected to the output layer, where the number of neurons is simply set to the number of categories to classify. The formula for the fully connected layer is:

$$h_{W,b}(\mathrm{x}) \int (W^{T}x + b)$$

where $f(*)$ is the activation function, W^{T} is the weight matrix, b is the bias vector and x is the input to the current fully connected layer.

VGGNet is a CNN structure proposed by the Visual Geometry Group at the University of Oxford, which achieved second place in the ImageNet competition in 2014. The network is based on AlexNet, the innovations are small convolution kernals and depth, as shown in Figure 2.4. The VGGNet is characterised by a smaller convolutional kernel compared to the traditional CNN and an enhanced depth of the network, resulting in a more significant recognition outcome. Although it came second in the competition, the network has a superior ability to extract abstract features compared to the GoogleNet that won the championship. It was therefore the preferred algorithm used in the study for feature extraction In addition to adding more layers to the network than AlexNet, the convolutional layer of VGGNet applied the same size filter, both 3×3 pixels, with a 1px padding.

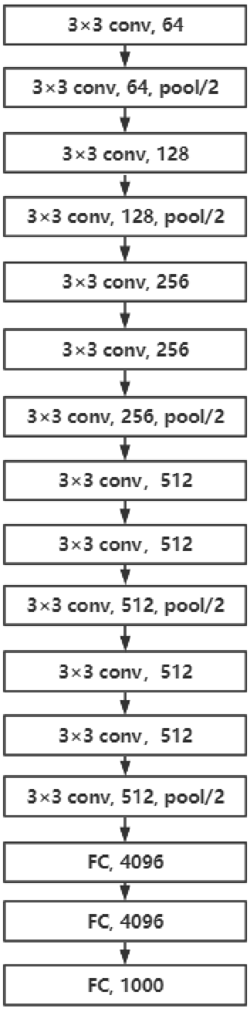


Figure 2.4: The Structure of VGG-16

Furthermore, the VGGNet has the following features:

i. The application of a 3×3 Filter captures the size of the top, bottom, left, right and central elements.

ii. Two 3×3 Filters are comparable to the effect of a 5×5 Filter, and similarly three 3×3 Filters are equivalent to 7×7. Therefore, the larger filters in the network can be replaced by several smaller filters.

iii. Multiple 3×3 Filters are more effective than larger size filters, improving the capacity of decisional function.

iv. Multiple 3×3 filters have more concise parameters compared to one larger filter.

The network has two structures, VGG-16 and VGG-19, depending on the depth. Although this network has improved the recognition performance to a greater extent than the basic AlexNet, there are still some limitations. Further improvements to the network model based on VGGNet are needed to improve the recognition results. The limitations of the network are as follows: first, training VGGNet will consume a large amount of computational resources due to the increase in network depth and heavy fully connected layer parameters, which requires larger memory resources to complete the training, thus the training time increases significantly. It has been found that even if all fully connected layers are removed, the influence on the performance of the deeper network is slight [20], which significantly reduces the complexity of the model. This paper therefore improves the VGGNet, based on the core idea of the VGGNet - the small convolutional kernel, and proposes a network structure that is suitable for the expression recognition problem. The network structure simplifies VGGNet and further improves the speed of training the model, while the loss rate is lower and the recognition accuracy is higher, so that it can meet the requirements of a recognition applications.

2.3.2 Introduction of Facial Expression Recognition

The process of facial expression recognition is relatively similar to face recognition, and many of the studies on facial expression recognition are based on face recognition. A facial expression recognition system needs to include three major steps: face detection, expression feature extraction and classification recognition. Depending on the training data it can be divided into static images recognition and dynamic sequences recognition. In the first stage, face detection is performed by various detection algorithms to target the specific location of the face in the image.

Then, in the second stage, the detected face image is used to extract features, the accuracy of which determines the accuracy of the classification. In the third stage, the facial expressions are classified, and the results are based on the selected classifier.

2.4 Facial Expression Dataset

The majority of FER databases are built on 2D static images or 2D video sequences. However, some databases involve 3D images. Since most 2D databases only contain face images, a FER system built on a 2D approach may have the disadvantage of handling changing positions. A 3D method has the potential to solve the pose variation problem. Most FER databases include the six main emotions (anger, disgust, fear, happiness, sadness, and surprise), as well as a neutral expression. Some FER databases are built in controlled situations (such as a laboratory with controlled illumination), whilst others are created in wild environments [21]. This chapter introduces three popular 2D FER databases used in different systems.

JAFFE The Japanese Female Facial Expression database [22] is a dataset constructed by researchers Michael Lyons, Shigelu Akamatsu at the ATR laboratory in Japan. A multi-orientation multi-resolution set of Gabor filters which are topographically ordered and aligned approximately with the face are used in to code the expression images. The specific feedback from human observers on the face expressions was checked against the similarity space characterized by the encoding accordingly. They suggest that the Gabor feature representations of facial images can be used as inputs to the classifiers for facial expression classification. The construction process involved 10 Japanese women making three to four expression samples for each of the six basic expressions and a neutral status, namely happy, sad, surprise, angry, disgusting, fear and neutral. The experiment yielded 219 face expression data images, which were evaluated by 92 Japanese students using Euclidean distances to calculate semantic similarity and to classify the expressions. An example of a basic expression in the JAFFE database is shown in Figure 2.5.

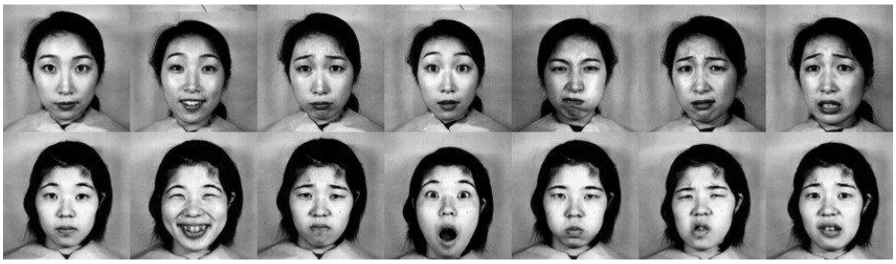


Figure 2.5: Facial Expression Samples From JAFFE Database

The Extended Cohn-Kanade(CK+) The Extended Cohn-Kanade (CK+) [23] database is a laboratory-controlled facial expression database that has been widely used in facial expression recognition. The dataset expands on the original CK database with 97 college students, with a 3:7 ratio of males to females. 81% of these subjects were European American, 13% were African American, and 6% were Asian and Latino. Participants were instructed to make 23 expression samples, including individual action units (e.g., AU12, i.e., tilt-drawn lip corners) and combinations of action units (e.g., AU1+AU2, i.e., raised inner and outer eyebrows). These 23 expression samples contain expressions of the six basic emotions (happy, surprise, anger, fear, disgust and sadness). Each image sequence begins with the natural state and ends with the peak of that expression. In addition, the last image of each sequence was coded, and the sequences were scanned in grey scale at a resolution of 640 x 490 pixels with 8-bit precision, it being noted that the camera took the picture from the front and there was a small change in the position of the participant's head. Figure 2.6 shows the image of the CK+ dataset for the six emotions.

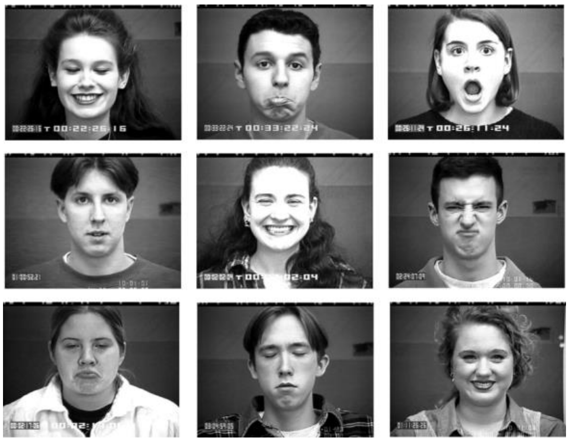


Figure 2.6: Facial Images From CK+ Database

FER2013 and FER+ The FER2013 [24] dataset is a dataset provided by the Kaggle website for its FER challenge. The dataset was collected by web crawlers, and its human eye recognition accuracy is around 65%. There are 28,708 facial expression images in the training set, 3589 facial expression images in the public test and 3589 facial expression images in the private test, each of which are grey-scale images with a fixed size of 48×48 pixels. The corresponding labels for each expression are as follows: 0 for Anger, 1 for Disgust, 2 for Fear, 3 for Happy, 4 for Sadness, 5 for Surprised and 6 for Neutral. Since the FER2013 dataset has more noisy images, FER+ is the new dataset that was built based on a 10-class voting by the participants for the images in the FER2013 dataset, providing a series of new label contents and labeling new categories - Contempt, Unknown and non-face images. According to the voting content, the maximum voting method was adopted to remove the interference of noisy images and improve the data quality of the original dataset. The voting labels were saved under CSV files and used for classification. Figure 2.7 shows a few examples of the FER2013 compared to the FER+ dataset labels.

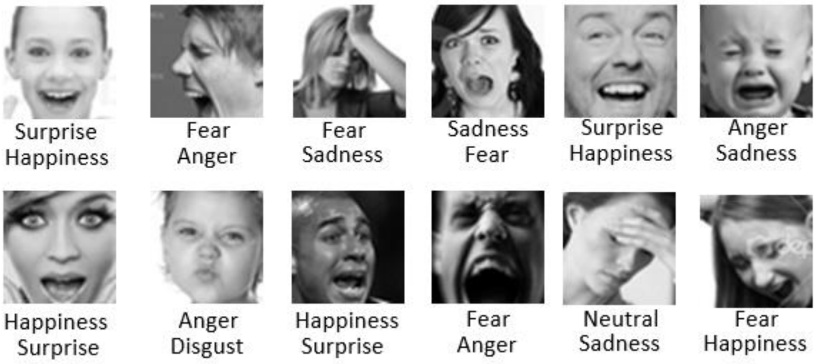


Figure 2.7: Example of FER+ Compared to FER2013

2.5 Data Pre-processing and Enhancement

Data Pre-processing Before training, pre-processing the data in the dataset is a key step, which mainly includes downsizing, data cleaning and data enhancement. For image pre-processing, it is crucial to address the effects of lighting factors on the image. Examples of lighting influences include overexposure, underexposure and shadows. These problems can cause a certain degree of destruction to the texture features of a grey-scale image. A series of image pre-processing operations are therefore needed to minimize the effect of illumination, noise interference and grey-scale unevenness.

Whitening is one of the most important methods in the pre-processing and is aimed to reduce the complexity of the data. After the whitening process, the correlation between the features of the input data is decreased and the variance of the feature remains. There are two types of whitening process, PCA whitening and ZCA whitening. The ZCA whitening adds a rotational transformation to the PCA whitening and has a higher similarity to the original input data, with the definition shown as follow:

$$x_{ZCAwhite} = Ux_{PCAwhite}$$

ZCA whitening uses the PCA algorithm to remove the correlation of data features and rotates on the result of PCA to finally obtain the outcome. PCA whitening differs from ZCA whitening in the following ways.

i. In the first case, the variance of each dimensional feature of the data must be maintained at 1. ZCA whitening only needs to ensure that the variances are equal.

ii. PCA whitening can achieve the dimension reduction as well as remove the correlation between data features, while the latter is mainly used to remove the correlation between features.

iii. ZCA whitening has a higher degree of similarity to the original input data than the PCA.

Data Augmentation In the FER problem, the quality of the dataset is critical to the performance of the model recognition. Data augmentation is an effective solution when sufficient data cannot be obtained. Data augmentation is the process of maximizing the value of existing data without adding additional data. Typical methods of data augmentation include cropping, rotation, flip, and scaling. Cropping is the process of scaling a randomly selected area of the image back to its original size. Rotation is the random rotation of an image by an angle within a specified range. Flip is to flip the original image horizontally or vertically to obtain a mirror image. Scaling is to scale down or scale up the original image by a certain percentage. In this operation, if the original image is reduced, the empty space created is filled with neighbouring pixel values. Image translation allows the original image to be placed in an arbitrary position so that the model can be trained to consider all corners of the image.

Due to the limited amount of data in the CK+ and FER+ datasets used for the experiments, the images in the dataset will be processed using data augmentation, reduce overfitting and improve the robustness of the model.

2.6 Web Development Technology

MySQL MySQL[25] is a relational database management system (RDBMS) developed by the Swedish company MySQL AB, part of the Oracle family. MySQL is one of the best RDBMS applications for web applications. It increases the speed and flexibility of data storage by keeping it in separate tables as opposed to keeping it all in one large repository. The SQL language used by MySQL is the most common standardized language used to access a database and is suitable for the development of small to medium sized websites due to its Lightweight, high speed, low cost, and open-source features.

HTML、CSS、JavaScript[26]–[28] HTML, known as Hyper Text Markup Language, consists of a series of tags that allow documents on the web to be formatted into a logical entity. CSS refers to Cascading Style Sheets, which is used to statically decorate web pages, but can also be used with various scripting languages to dynamically format elements of web pages. JavaScript is a web scripting language that has been used widely in the development of web applications and is often used to add a variety of dynamic features to web pages to provide users with a more smooth and aesthetically pleasurable using experience.

Vue Vue.js[29] is a progressive framework for JavaScript to build user interfaces, a system that uses a concise template syntax to render data into the document object model (DOM). In contrast to other large-scale frameworks, Vue is designed to be applied for the bottom-up programming. Its core library focuses exclusively on the viewing layer, making it easy to get started and to integrate with external libraries or existing projects. It is also capable of driving complex single page applications, when collaborates with a modern tool chain and a variety of supporting libraries.

Flask Flask[30], [31] is a web application framework written in Python, known as a micro framework, which is more flexible, lightweight, secure and easy to use than other frameworks of similar type. Therefore, it is possible to implement a feature-rich small to medium sized website or web service in a short time using Flask. In addition, Flask is highly customizable because of its powerful plugin library, allowing users to add features according to their own requirements, keeping the core functionality simple while achieving rich functionalities.

Development Environment PyCharm is a Python IDE built by JetBrains with a full set of tools to help users improve their productivity when developing in Python, such as debugging, syntax highlighting, project management, code skipping, intelligent hints, auto-completion, unit testing, and version control. In addition, the IDE offers a number of advanced features for supporting professional web development in a variety of back-end frameworks.

WebStorm is a web front-end development tool from JetBrains. It is the same source as IntelliJ IDEA and inherits the functionality of the powerful JavaScript in IntelliJ IDEA. It supports Node.js and various popular frameworks such as React, Angular, Vue.js, Meteor, etc.

Chapter 3

System Design

3.1 Pipeline

The pipeline of this system contains two sections (Figure 3.1). The first section is the front-end which interacts with user on the website. The second section consists of the back-end which is the server of the system.

The login page allows user to sign up and sign in the account with validation. The home page is shown after the login and allows user to choose a mode to recognize facial expressions. There are two recognition modes in this system. In the first mode, the user will open the camera and take an image with human faces which will be uploaded to the server to assist with facial recognition later on. In the second mode, the user will be able to upload a image directly to the server to recognize facial expressions. The server then detects the face in the image and trims it into a 48*48 image for facial image preprocessing. Then, the processed facial image will be delivered to the CNN model for image recognition. The result is returned to the browser if the expression was successfully identified and then stored in the database.

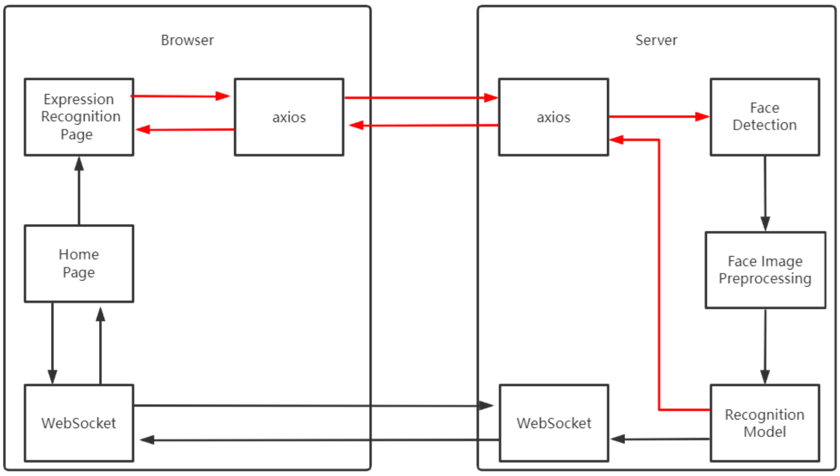


Figure 3.5: Pipeline

Use Case Diagram

The system is designed to classify and recognize static facial expressions. The key functional modules are face detection and expression recognition using the detection results. In addition, this system also needs to set the relevant permissions for different users to access the recognition images and modify the recognition records. After the above requirements analysis, the functional requirements of the system are as follows:

- User information storage: registration and login features and user account management.
- Image upload: including uploading the user's local image and invoking the camera on the device to take a photo.
- Face detection: face detection is performed on the input image, and then the facial image is cropped and saved as a 48*48 size image to remove the background interference part of the image and improve the overall system recognition performance. Meanwhile, detected faces are marked on the image with red boxes and returned to the browser for display.
- Expression classification: The cropped facial image will be input to CNN to carry out the task of expression classification.
- Recognition record storage: Store the time when the expression recognition record is created and the classification result in the database for users to access.

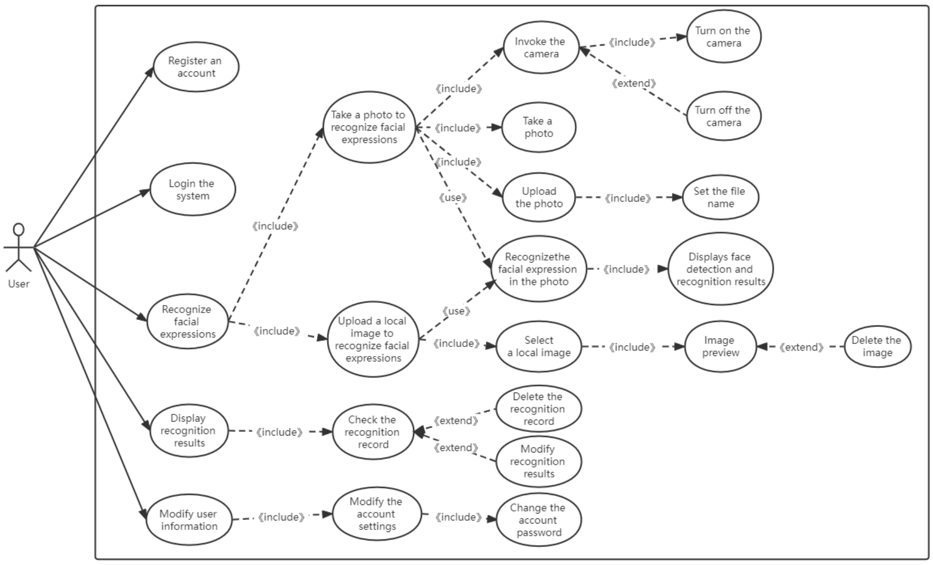
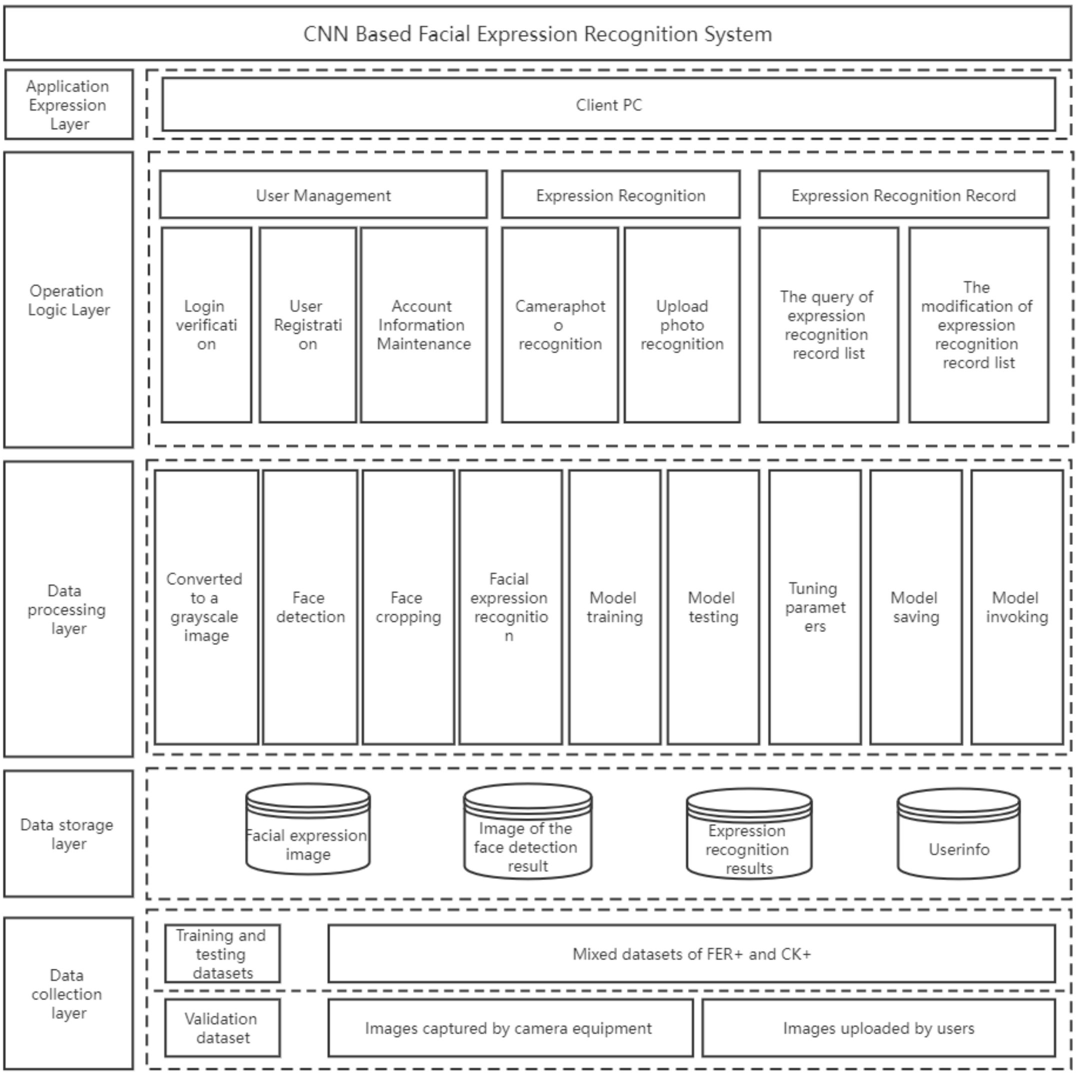

User
Register an account
Login the system
Recognize facial expressions
Display recognition results
Modify user information
Take a photo to recognize facial expressions
Upload a local image to recognize facial expressions
Check the recognition record
Modify the account settings
Invoke the camera
Turn on the camera
Turn off the camera
Take a photo
Upload the photo
Set the file name
Recognizethe facial expression in the photo
Displays face detection and recognition results
Select a local image
Image preview
Delete the image
Delete the recognition record
Modify recognition results
Change the account password
《include》
《extend》
《use》

Figure 3.6: System Use Case Diagram

3.2 System Architecture Diagram

The facial expression recognition system consists of the following modules: facial expression recognition module, recognition records module, and user account management module.

According to the use case diagram (Figure 3.2), the system architecture was designed as shown in Figure 3.3. From the diagram, it can be seen that the system is divided into five layers, from the data collection layer on the bottom to the application expression layer on the top. From the front-end browser to the data service, the layers are closely linked to each other to provide services for the system operation.



CNN Based Facial Expression Recognition System
Application Expression Layer
Client PC
Operation Logic Layer
User Management
Expression Recognition
Expression Recognition Record
Login verificati on
User Registrati on
Account Information Maintenance
Cameraphot o recognition
Upload photo recognition
The query of expression recognition record list
The modification of expression recognition record list
Data processing layer
Converted to a grayscale image
Face detection
Face cropping
Facial expression recognitio n
Model training
Model testing
Tuning paramet ers
Model saving
Model invoking
Data storage layer
Facial expression image
Image of the face detection result
Expression recognition results
Userinfo
Data collection layer
Training and testing datasets
Mixed datasets of FER+ and CK+
Validation dataset
Images captured by camera equipment
Images uploaded by users

Figure 3.7 : System Architecture Diagram

Based on the system architectural design, a further technical architectural design for each corresponding layer of the structure is shown in Figure 3.4.

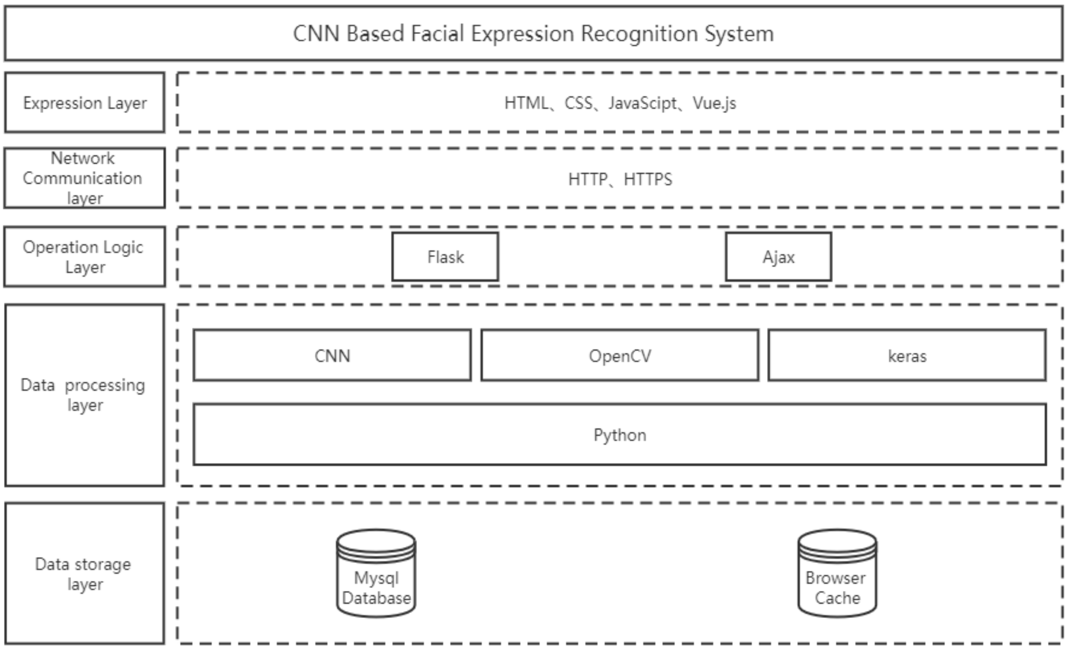


Figure 3.8: System Technical Architecture Diagram

The expression layer of the system uses the Vue.js framework and HTML, CSS, and JavaScript for development, and HTTP and HTTPS protocols for network communication. The operation logic layer is based on Flask, a web services framework using Python as the development language, and Ajax for front and back-end communications. The data processing layer is developed in Python and uses the neural network framework Keras. The CNN model is used for facial expression recognition. Haar feature based face detection from OpenCV is applied to detect faces in images. Image pre-process is implemented by OpenCV as well. The data storage layer is built using a Mysql database to store recognition data as well as user data, and a browser cache to store user login status data.

The system development will be carried out in Pycharm 2019.3.5 and Webstorm 2019.3.3, a 64-bit professional version for Windows. The front-end is developed in the Webstorm development tool and the back-end server is coded in Pycharm, using a separate front-end and back-end development approach to build the system.

3.3 Database Design

Database design is the basis of the user management, recognition records module. A well-designed database of fields and structures is essential for data storage. By analyzing the functional requirements of the system and the system architecture, three tables were designed in Mysql database for this system, respectively user information, file information and facial expression recognition records. The corresponding designs are presented as follows:

Field Name	Type	Length	Primary Key	Empty	Description
userid	int	100	YES	NO	Auto-increment
username	varchar	255	NO	NO	Username
password	varchar	255	NO	NO	Password

Table 3.1: User Information

Field Name	Type	Length	Primary Key	Empty	Description
fileid	int	100	YES	NO	The id of the uploading images, auto-increment
username	varchar	255	NO	NO	Username
filename	varchar	255	NO	NO	The name of an image
uploadpath	varchar	255	NO	NO	The storage path of the image
resultpath	varchar	255	NO	YES	The storage path of a classification result
discernpath	varchar	255	NO	YES	The storage path of a cropped facial image

Table 3.2: File Information

Field Name	Type	Length	Primary Key	Empty	Description
recid	int	100	YES	NO	The id of a record, auto-increment
username	varchar	255	NO	NO	Username
ferresult	varchar	255	NO	YES	Facial expression classification result
fertime	varchar	255	NO	NO	Date of a record
filename	varchar	255	NO	NO	The name of an image
imgsrc	medium-blob	16M	NO	NO	The binary format source of a classification image

Table 3.3: Facial expression recognition records

Figure 3.5 shows the class diagram of the system, Figure 3.6 illustrates the entity diagram of the system, which includes the user entity diagram, the recognition image entity diagram, and the recognition record entity diagram. Figure 3.7 shows the entity relationship diagram, which specifies the relationships between each of the entities [32] .

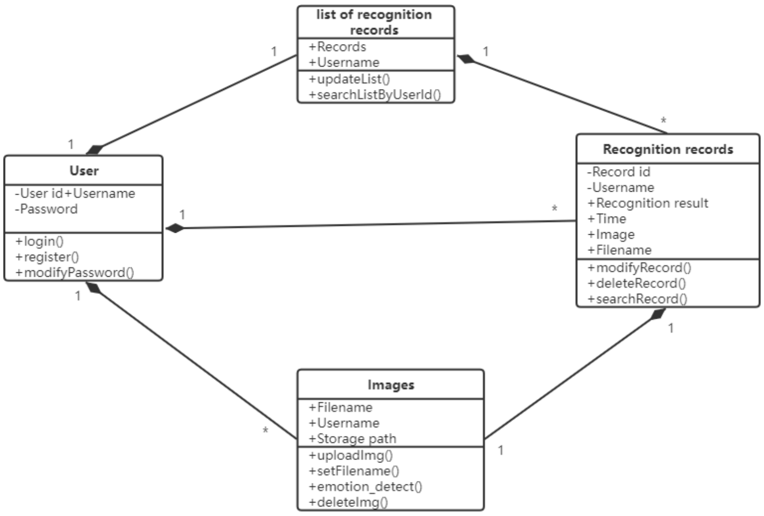


Figure 3.5: System Class Diagram

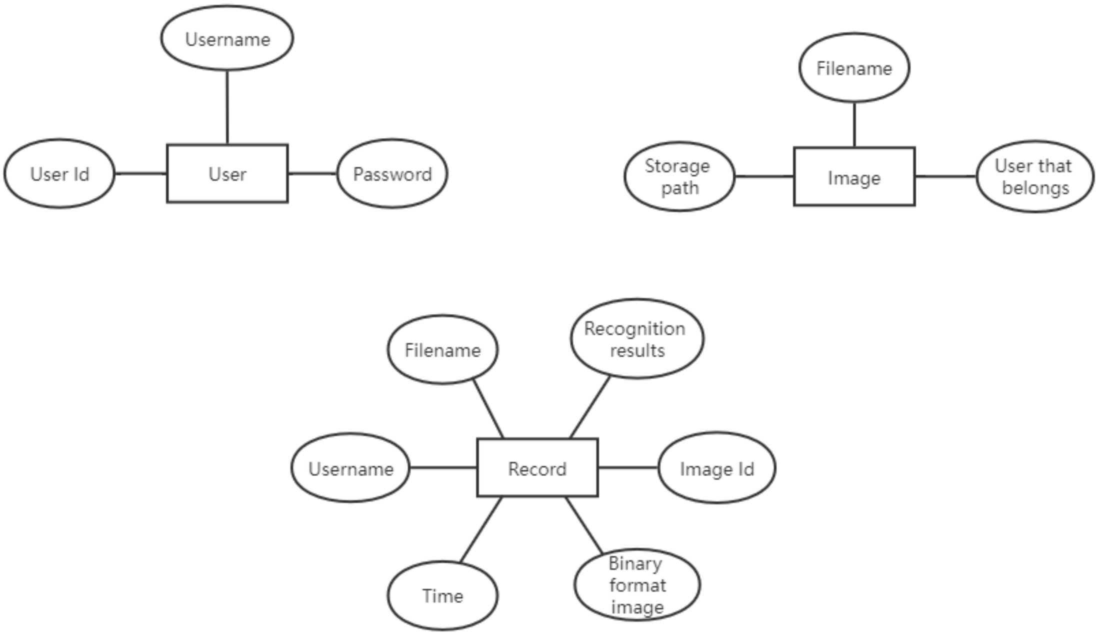


Figure 3.6: System Entity Diagram

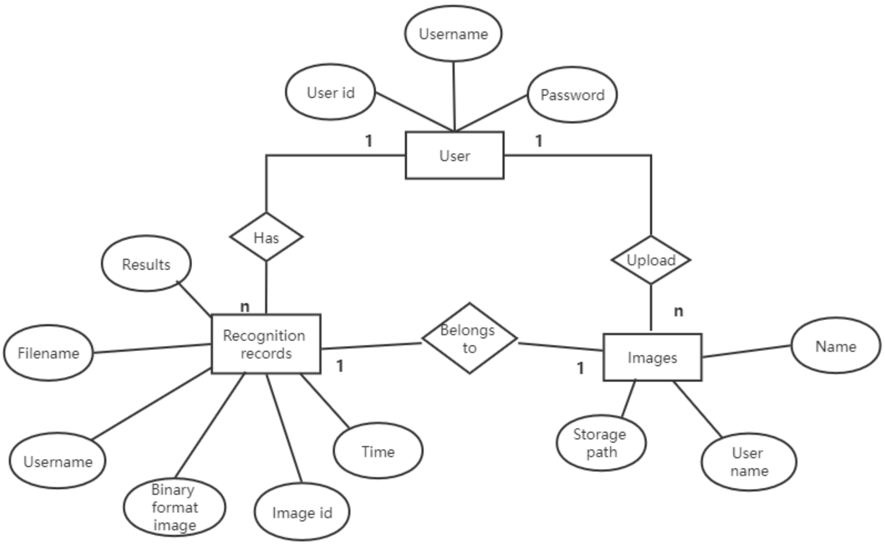


Figure 3.7: Entity Relationship Diagram

3.4 System Module Design

This section describes three system module designs through a number of relative sequence diagrams and activity diagrams. A sequence diagram represents interactions between components in a time sequence in the software engineering discipline [33, pp. 102–106]. It is a dynamic UML diagram that depicts the message exchanges between different objects.

An activity diagram is one of the UML (unified modeling language) behavioral diagrams as well. A use case, a scenario, or the specific logic of a business rule can be efficiently captured by an activity diagram, which is appropriate for business process modeling [33, pp. 96–98].

3.4.1 User Account Management Module

This module is designed to complete the functionalities of sign-in, sign-up, and modification of the user's password. As shown in Figure 3.8, it visualizes the sequence diagram of the user registration in this module. Figure 3.9.7 shows the sequence diagram of a user's password modification. After logging into the system, users can then go to the account settings page to change the password for their current account. Figure 5.8 illustrates the activity diagram of user account settings.

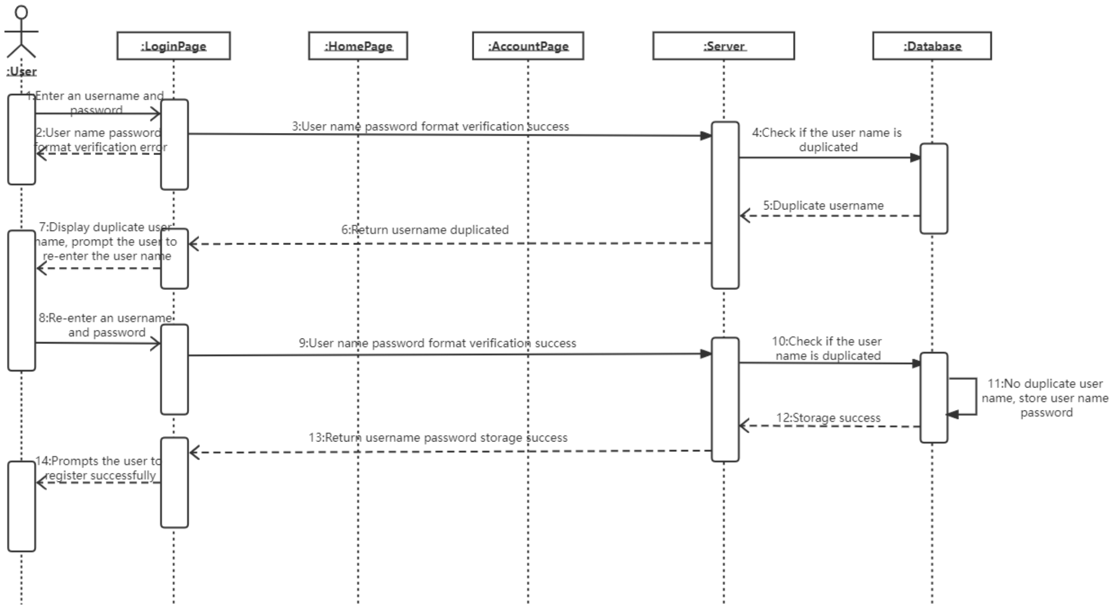

:User
:LoginPage
:HomePage
:AccountPage
:Server
:Database
1:Enter an username and password
2:User name password format verification error
3:User name password format verification success
4:Check if the user name is duplicated
5:Duplicate username
6:Return username duplicated
7:Display duplicate user name, prompt the user to re-enter the user name
8:Re-enter an username and password
9:User name password format verification success
10:Check if the user name is duplicated
11:No duplicate user name, store user name password
12:Storage success
13:Return username password storage success
14:Prompts the user to register successfully

Figure 3.8: Sequnce Diagram of User Registration

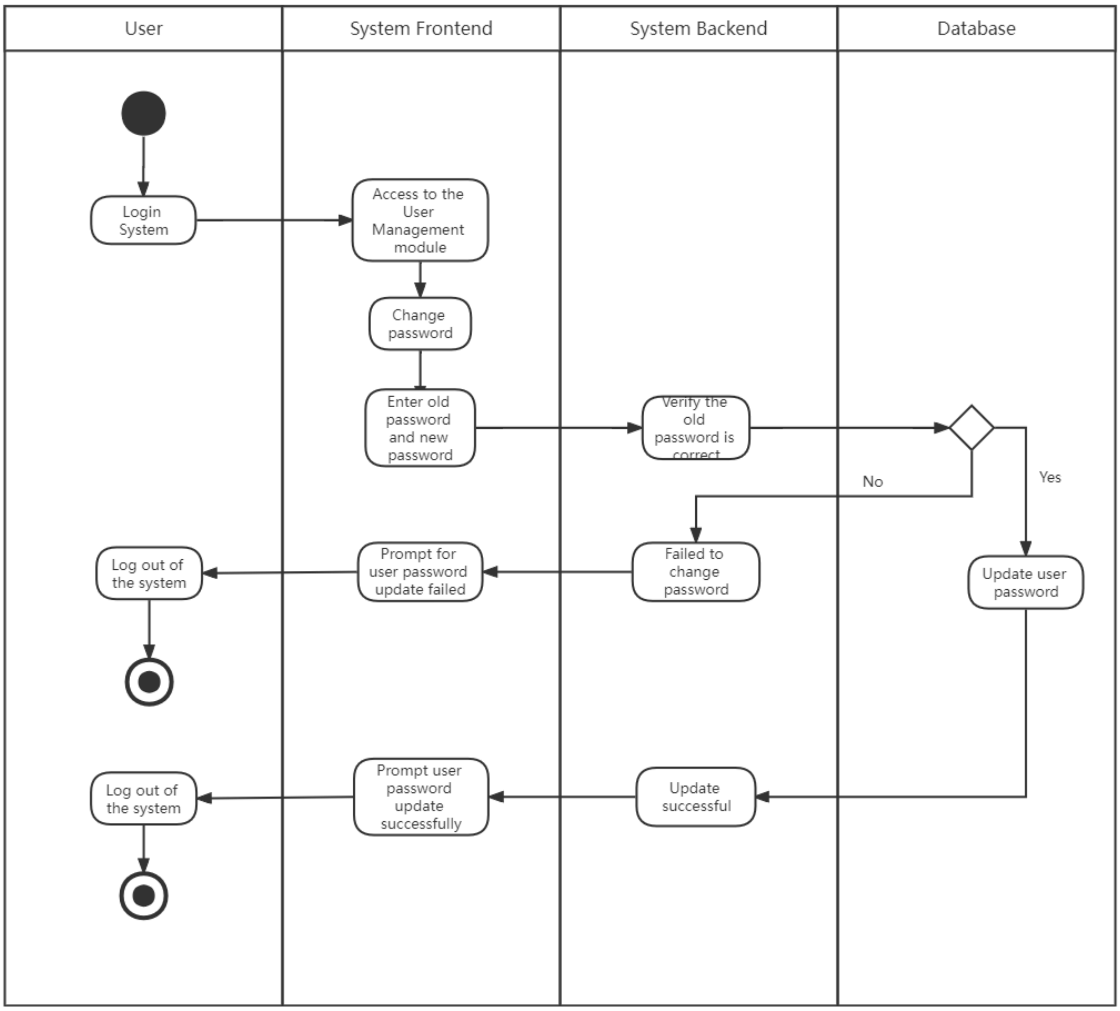

User
System Frontend
System Backend
Database
Login System
Access to the User Management module
Change password
Enter old password and new password
Verify the old password is correct
No
Yes
Log out of the system
Prompt for user password update failed
Failed to change password
Update user password
Log out of the system
Prompt user password update successfully
Update successful

Figure 3.9: Activity Diagram of Modifying Password

3.4.2 Facial Expression Recognition Module

After a user enters the system, the expression recognition module will be accessible. This module contains two parts, one for recognition by taking photos from the camera device and one for recognition by uploading images from the user. Figure 3.10 presents the sequence diagram of the facial expression recognition module. Figure 3.11 shows the activity diagram of this module, presenting the specific flow of the facial expression recognition functionality.

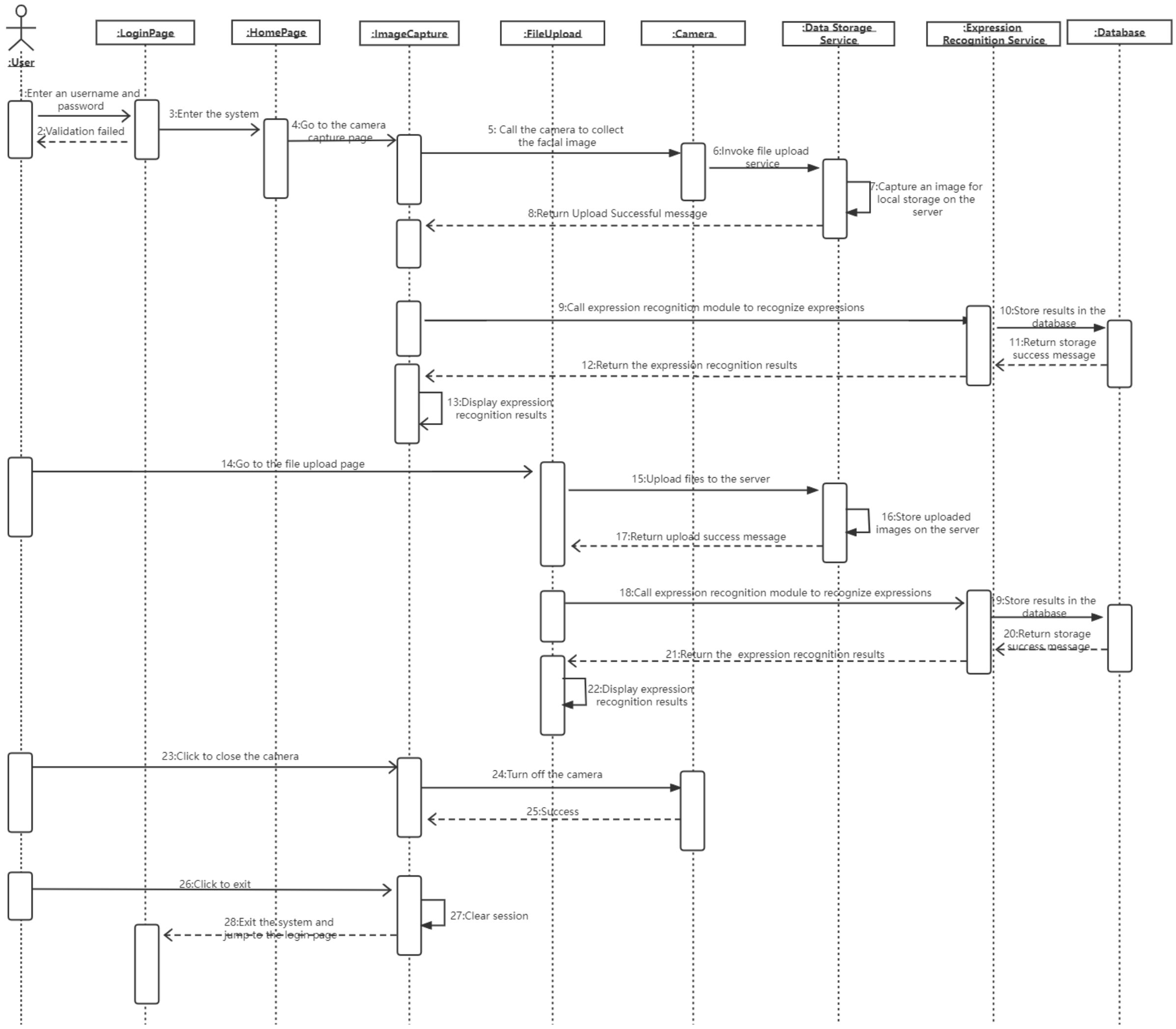

:User
:LoginPage
:HomePage
:ImageCapture
:FileUpload
:Camera
:Data Storage Service
:Expression Recognition Service
:Database
1:Enter an username and password
2:Validation failed
3:Enter the system
4:Go to the camera capture page
5: Call the camera to collect the facial image
6:Invoke file upload service
7:Capture an image for local storage on the server
8:Return Upload Successful message
9:Call expression recognition module to recognize expressions
10:Store results in the database
11:Return storage success message
12:Return the expression recognition results
13:Display expression recognition results
14:Go to the file upload page
15:Upload files to the server
16:Store uploaded images on the server
17:Return upload success message
18:Call expression recognition module to recognize expressions
19:Store results in the database
20:Return storage success message
21:Return the expression recognition results
22:Display expression recognition results
23:Click to close the camera
24:Turn off the camera
25:Success
26:Click to exit
27:Clear session
28:Exit the system and jump to the login page

Figure 3.10: Sequence Diagram of Facial Expression Recognition

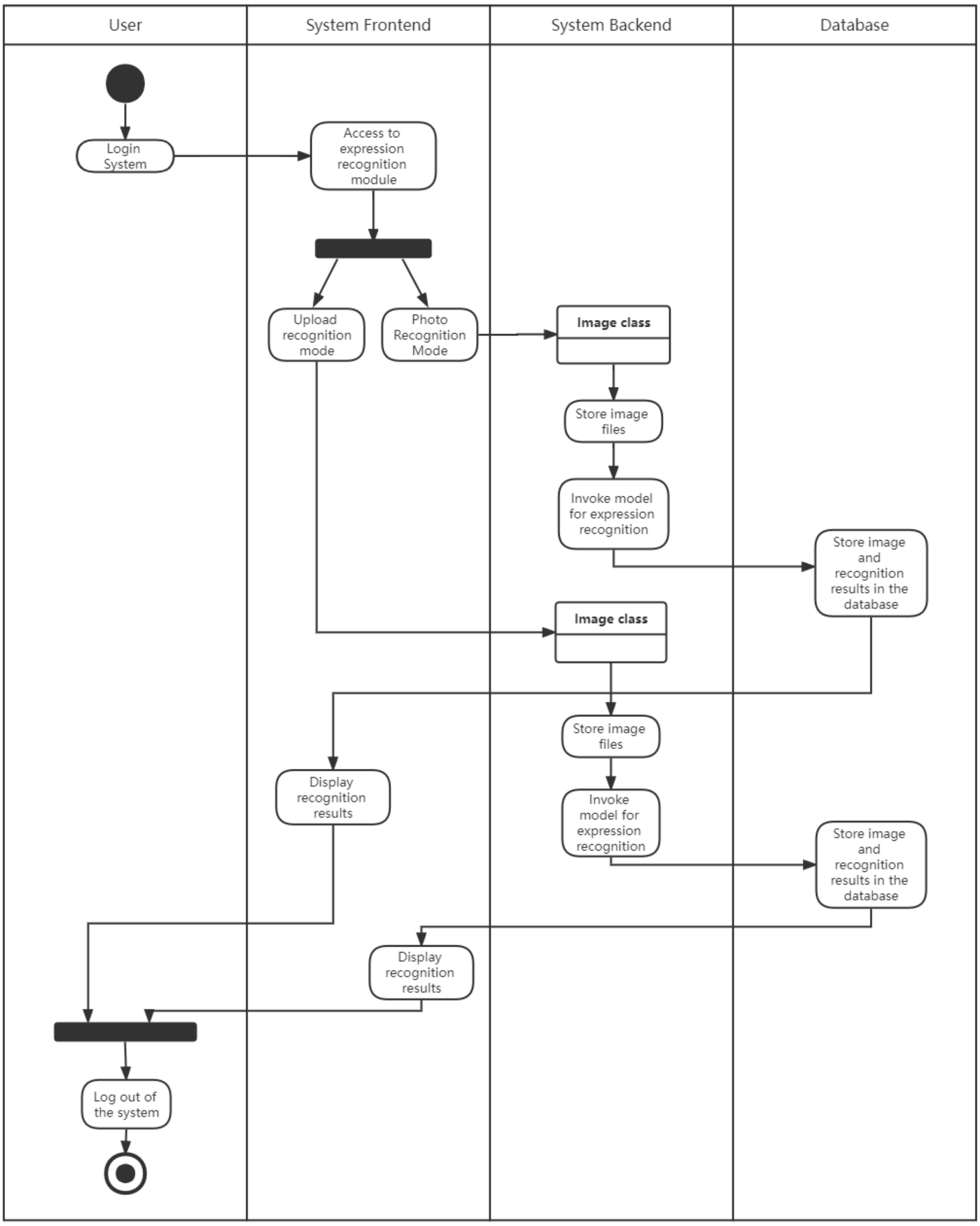

User
System Frontend
System Backend
Database
Login System
Access to expression recognition module
Upload recognition mode
Photo Recognition Mode
Image class
Store image files
Invoke model for expression recognition
Store image and recognition results in the database
Image class
Store image files
Invoke model for expression recognition
Store image and recognition results in the database
Display recognition results
Display recognition results
Log out of the system

Figure 3.11: Activity Diagram of Facial Expression Recognition

3.4.3 Recognition Records Module

Users are able to view and edit previous recognition records by accessing the recognition records page within the statistics. Figure 3.12 illustrates the sequence diagram of searching for the expression recognition records.

Figure 3.13 gives an illustration of the sequence diagram for editing an expression recognition record. Figure 3.14 shows the activity diagram for the expression recognition record module.

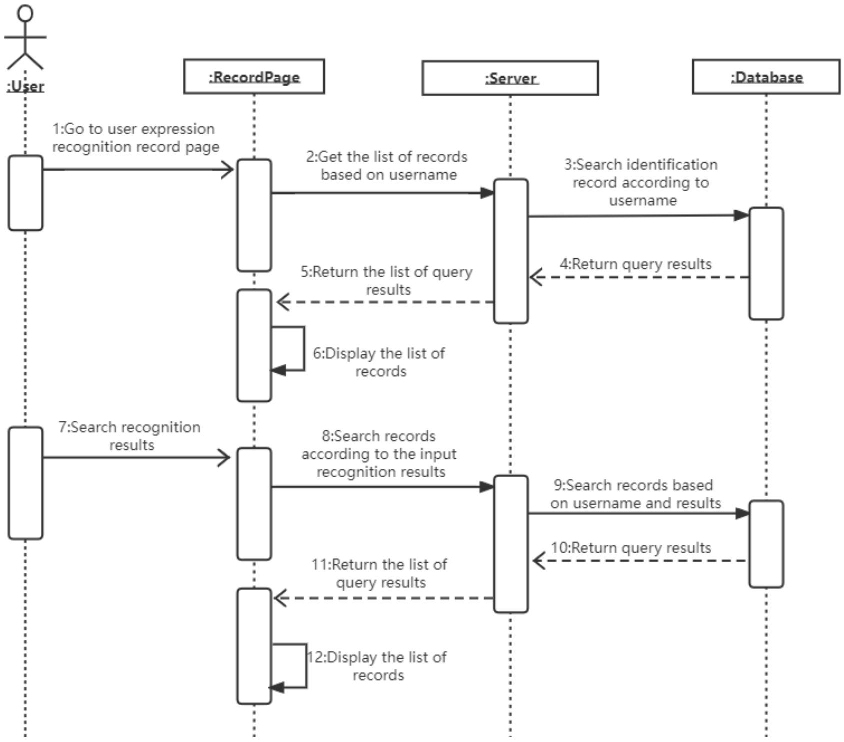

:User
:RecordPage
:Server
:Database
1:Go to user expression recognition record page
2:Get the list of records based on username
3:Search identification record according to username
4:Return query results
5:Return the list of query results
6:Display the list of records
7:Search recognition results
8:Search records according to the input recognition results
9:Search records based on username and results
10:Return query results
11:Return the list of query results
12:Display the list of records

Figure 3.12: Sequence Diagram of Searching For Recognition Records

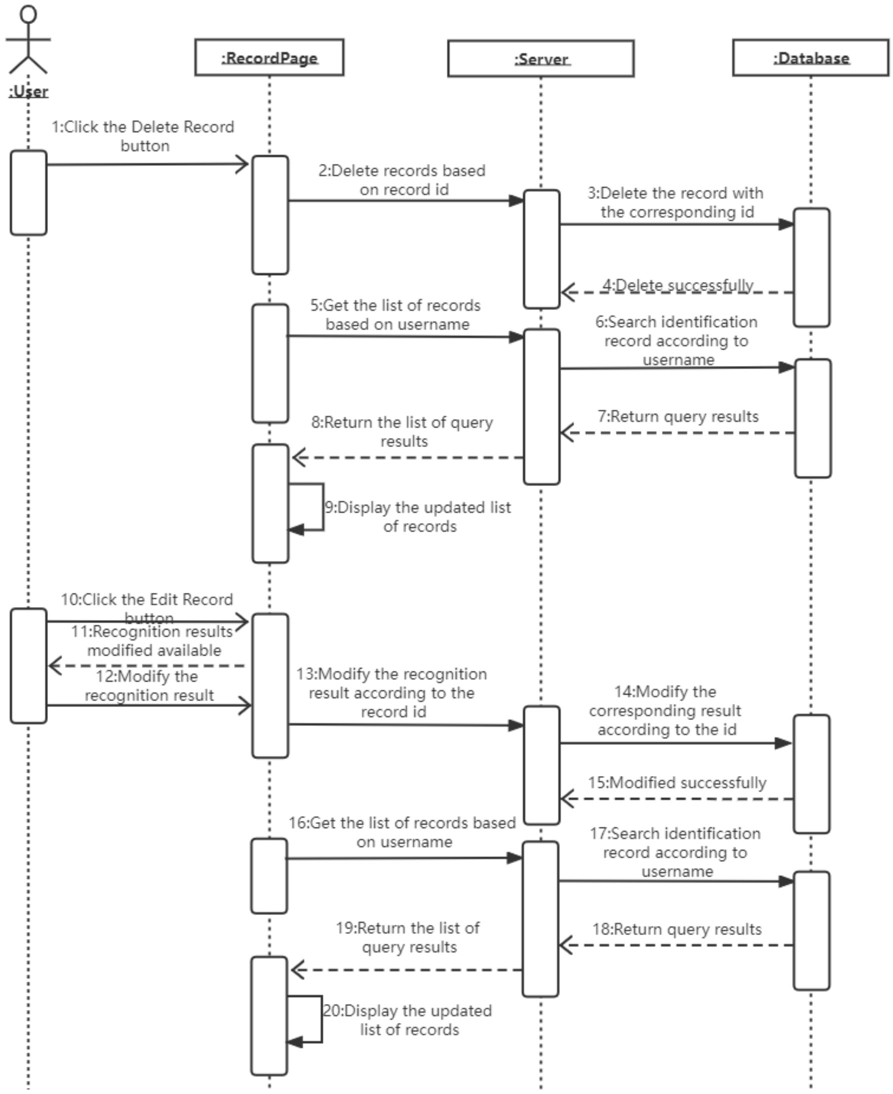

:User
:RecordPage
:Server
:Database
1:Click the Delete Record button
2:Delete records based on record id
3:Delete the record with the corresponding id
4:Delete successfully
5:Get the list of records based on username
6:Search identification record according to username
7:Return query results
8:Return the list of query results
9:Display the updated list of records
10:Click the Edit Record button
11:Recognition results modified available
12:Modify the recognition result
13:Modify the recognition result according to the record id
14:Modify the corresponding result according to the id
15:Modified successfully
16:Get the list of records based on username
17:Search identification record according to username
18:Return query results
19:Return the list of query results
20:Display the updated list of records

Figure 3.13: Sequence Diagram of Editing Recognition Records

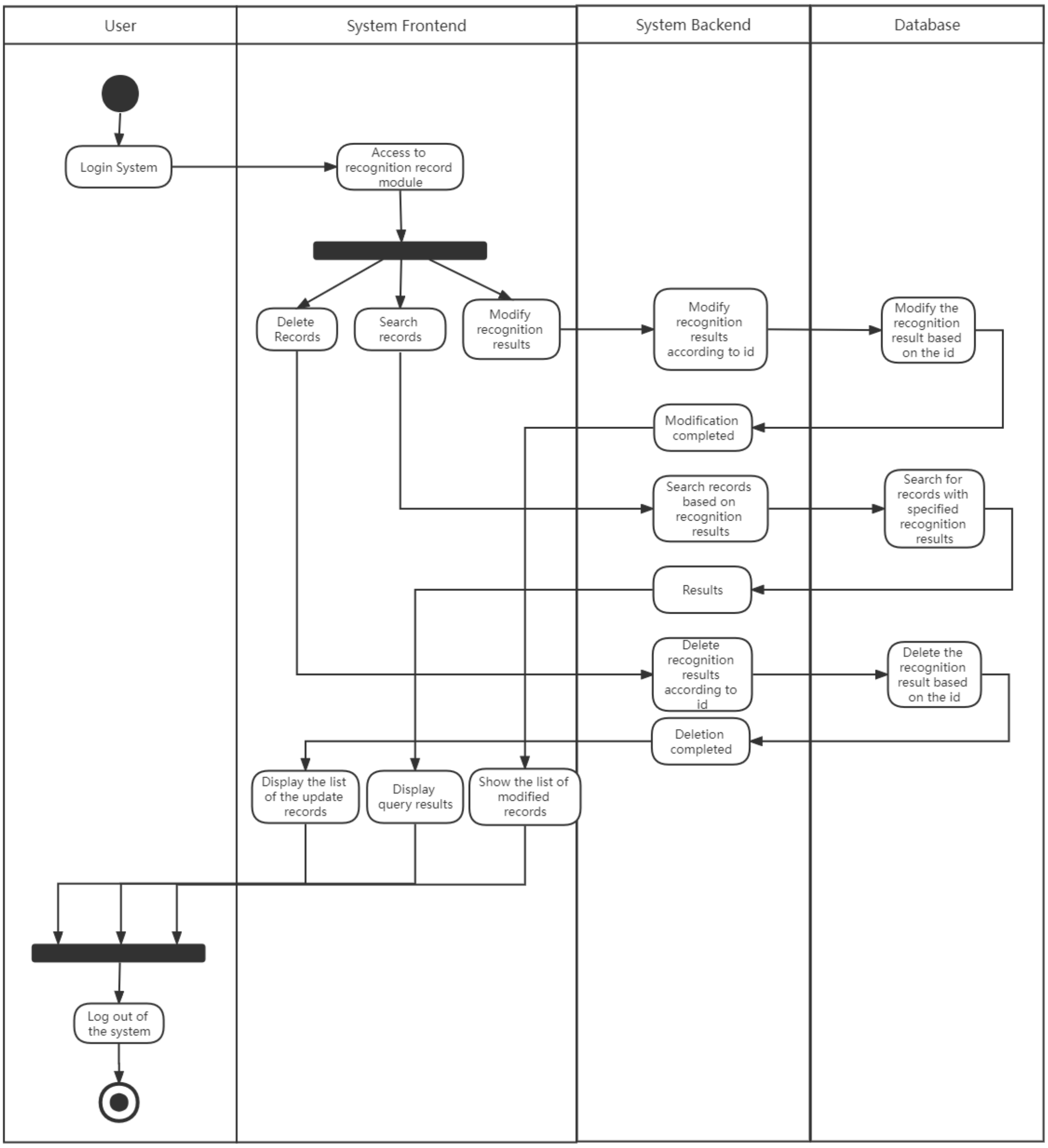

User
System Frontend
System Backend
Database
Login System
Access to recognition record module
Delete Records
Search records
Modify recognition results
Modify recognition results according to id
Modify the recognition result based on the id
Modification completed
Search records based on recognition results
Search for records with specified recognition results
Results
Delete recognition results according to id
Delete the recognition result based on the id
Deletion completed
Display the list of the update records
Display query results
Show the list of modified records
Log out of the system

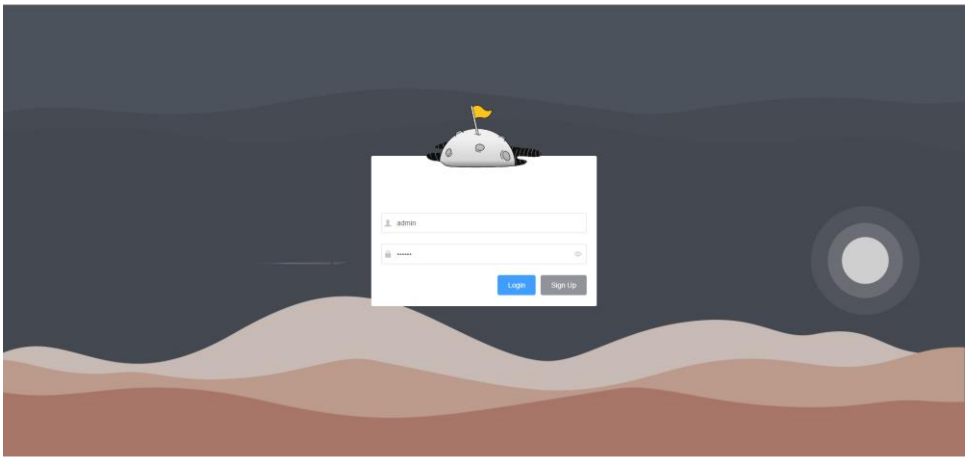
Figure 3.14: Activity Diagram of Recognition Records

Chapter 4

Implementation

4.1 User Account Management Module

The user account management module includes user login, registration, and account settings sections. The login and registration page is shown in Figure 4.1. Users can log in to the system or sign up for a new account. Each username cannot be duplicated in the system. When a user clicks on the login button, the front-end sends a request to the server, and the server connects to the database to query the password corresponding to the username. Only valid usernames and passwords can be used to log in, otherwise the login process fails.



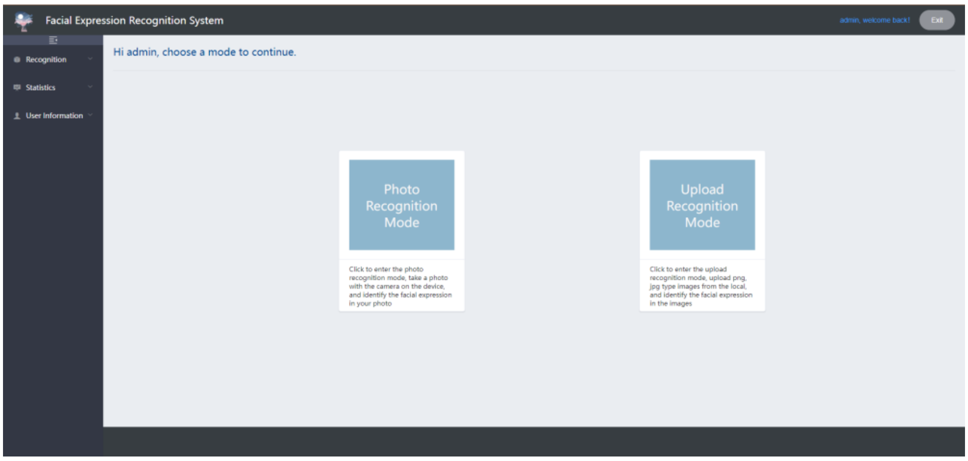
admin

••••••

Figure 4.1: Example of Login and Registration Page

When a user signs up for a new account, a username and a password should be entered. Firstly, the length of the username and password are checked by the front-end to ensure the security. Then, after the user clicks the sign-up button, the front-end sends a request to the server, then the server compares the username in the database, only a unique username can be registered successfully, otherwise, the registration process fails.

After successfully logging in, the home page pops up. Clicking the exit button on the page will exit the system and redirect the user to the login and registration page. An example of the home page is shown in Figure 4.2.



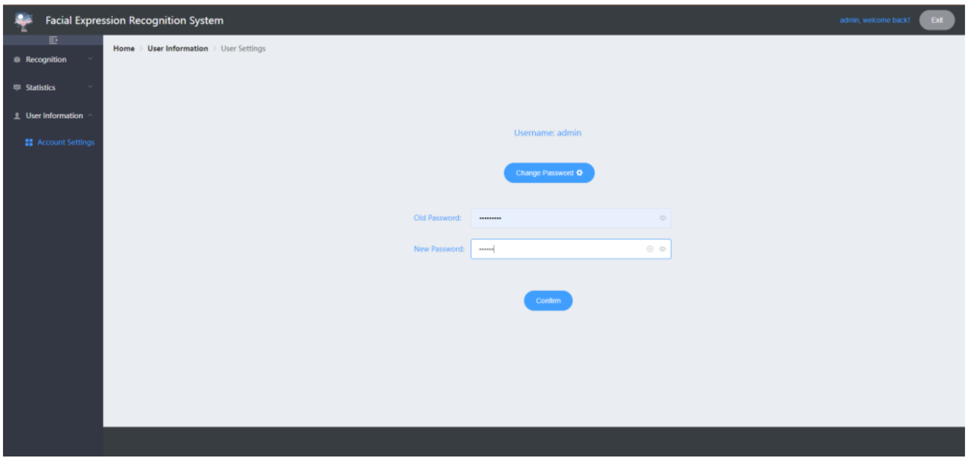
Hi admin, choose a mode to continue.

Click to enter the photo recognition mode, take a photo with the camera on the device, and identify the facial expression in your photo

Click to enter the upload recognition mode, upload png, jpg type images from the local, and identify the facial expression in the images

Figure 4.2: Example of Home Page

The user account settings page operates as shown in Figure 4.3. After the user has successfully registered and logged into the system, the account settings page is accessible under user information bar. Once the user inputs the old password and the new password, the front-end sends a request to the back-end, which connects to the database and verifies the username and the corresponding old password, then updates the password after successful verification, if not the server returns a message with a password verification failure status code to the front-end. The front-end receives the status code and alerts the user that the old password was entered incorrectly and needs to be re-entered. After the password is successfully changed, the status code returned by the server is 200. Once the front-end receives the status code 200, a message that indicates the password has been successfully changed is displayed.



Username: admin

Old Password: ••••••••••

New Password: ••••••

Figure 4.3: Example of Account Settings Page

4.2 Facial Expression Recognition Module

The facial expression recognition module consists of two components: invoking the camera device to take photos for expression recognition and uploading local images for recognition.

The main process of invoking the camera device to take photos for the facial expression recognition module is as follows.

Image Acquisition The user opens the camera page under the recognition bar, then the device camera is automatically invoked by the browser. The photo taken is displayed on the screen to preview. It allows multiple taking before uploading the image to the server. After the user uploads the image, the browser sends the image in Base64 format to the server for processing. The server returns a message which indicates whether the data is well-received.

Face Detection The server detects the face from the original image and crops the facial area from the image after capturing the facial range.

Image Pre-processing The cropped facial image is re-sized to 48*48 pixels and stored in the server in order to be suitable for the input layer of the neural network. Then, the 48*48 facial image is inputted into the trained CNN model for expression classification.

Model Recognition The CNN model identifies the facial expression from the input images, generates the recognition result, and stores it in the database.

Output Result The server returns the image with the detected face in Base64 format and the expression classification result to the browser. Then, it presents the image with a detected face which is marked by a red box, and the recognition result to the user.

Figure 4.4 depicts the main functional flow of the recognition system.

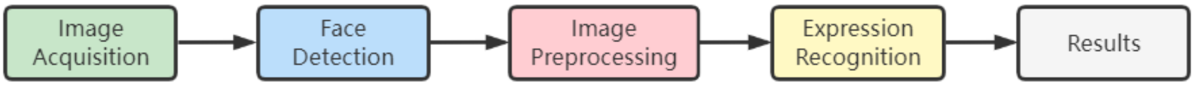
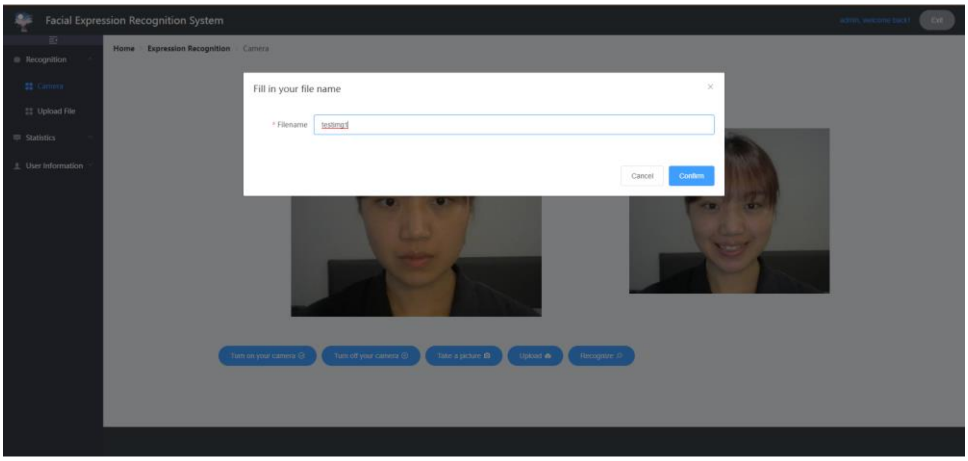


Figure 4.4: An Example of The Camera Page

After the user has successfully logged into the system, click on the recognition bar to enter the camera page. The user can click the button in this module to turn on and off the camera. By default, the camera is turned on. The browser will display the preview effect on the right side of the page once the user has adjusted the position and clicked the camera button to capture a photo.

After taking the photo, the user clicks the Upload button to set a name to the photo, an example of the page is shown in Figure 4.5. Click confirm button to upload the image to the server.



Fill in your file name

* Filename testimg1

Figure 4.5: An Example of Uploading a Photo

During this process, the front-end invokes the port */uploadList* to transmit the image data in Base64 format and the name of the image to the server in the form of *FormData* using the POST method.

The back-end server receives and stores the image locally through the *upload_file_list* function and stores the username with the image name and storage path in the database. The example of the page is shown in Figure 4.6.

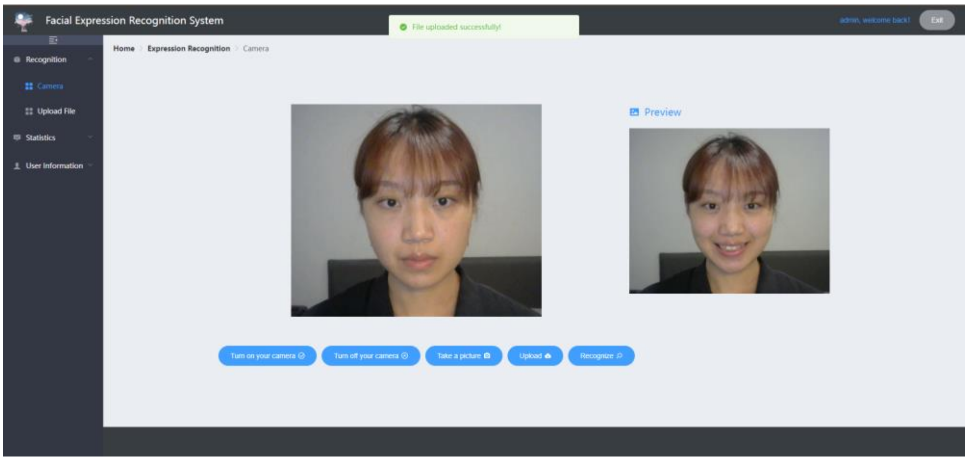


Figure 4.6: An Example of Uploading Successful

After uploading the image and clicking on the recognition button, the front-end sends a POST request to the server using the /detect port and sends the file name of the recognized image. After accepting the data, the server invokes the *detect* function to read the corresponding image from the local site, then invokes the *face_detect_fun* method to apply face detection to the image, drawing the face region with a red box. Then, save the detected image and the extracted facial image to the local site. The *motion_detect* function is then invoked to use the CNN model to make a classification of the detected face images, returning the classification result and the image in Base64 format with the face region drawn.

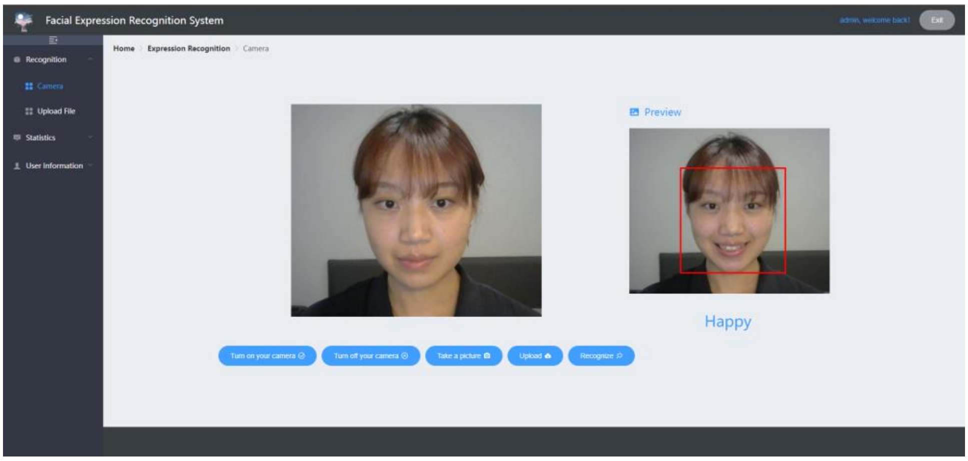


Figure 4.7: An Example of An Recognition Result

The front-end receives the data and displays the recognition result and the face detection result on the screen. Figure 4.7 shows an recognition result of an uploaded image.

The example of the local file upload recognition is shown in Figure 4.8. Once the local image is selected by clicking on the select image button, the image preview area is displayed on the page. Clicking on the upload button to invoke the *uploadImg* function, the front-end then sends a request to the back-end port */admin/uploadList* to upload the image to the server. To delete the currently selected image, click on the delete button. After the image is uploaded, click on the recognize button to request the backend/detect port to detect the face and invoke the CNN model to identify the expression. After the recognition is successful, the server returns the recognition result and the face detection result image to the front-end, which will be displayed on the screen later on.

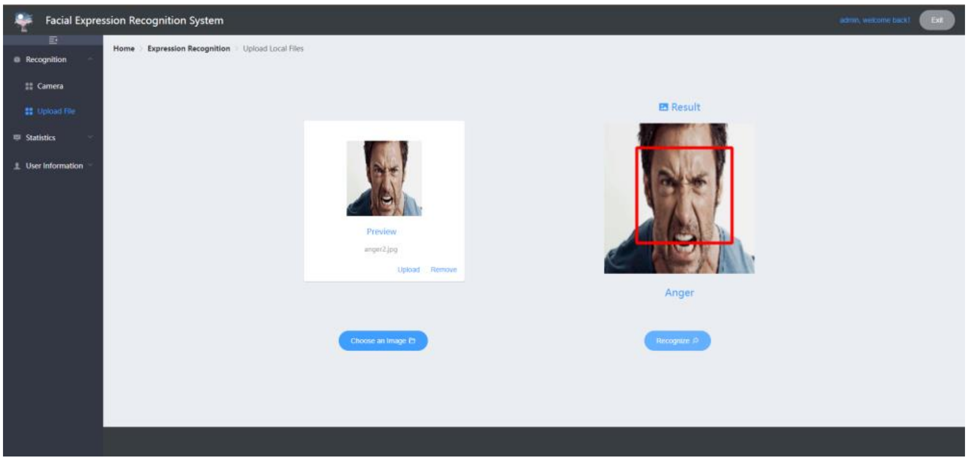
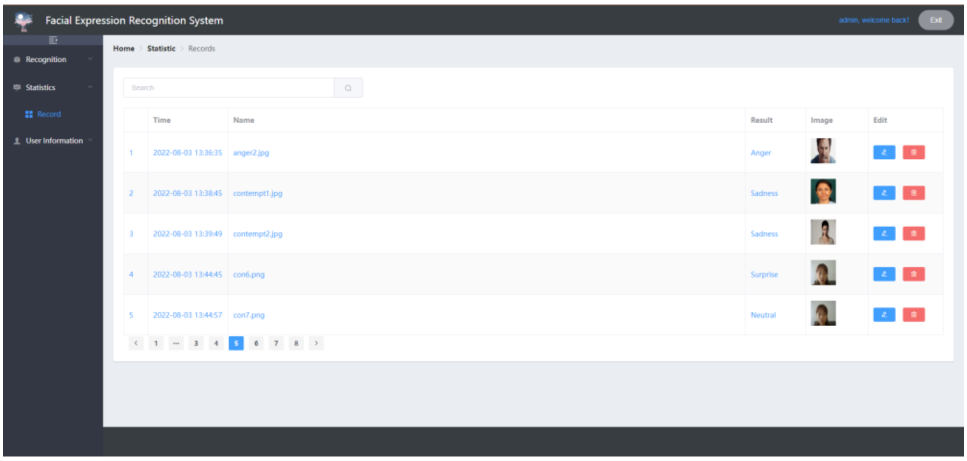


Figure 4.8: An Example of Uploading Local Image To The Server

4.3 Recognition Records Module

The Expression Recognition Records module shows the records of the user's recognition results. This module includes the functionality of deleting records, modifying recognition results, and querying records. Figure 4.9 shows the recognition records page in operation.



	Time	Name	Result	Image	Edit
1	2022-08-03 13:36:35	anger2.jpg	Anger		
2	2022-08-03 13:38:45	contempt1.jpg	Sadness		
3	2022-08-03 13:39:49	contempt2.jpg	Sadness		
4	2022-08-03 13:44:45	con6.png	Surprise		
5	2022-08-03 13:44:57	con7.png	Neutral		

Figure 4.9: An Example of The Recognition Records Page

When the user clicks the delete button, the *removeById* function is invoked. The front-end sends a POST request to the back-end port */recordList/delete* with an id of the record, and then the server deletes the corresponding record in the database, returning a status code of 200 for a successful deletion and 0 for a failed deletion. The front-end delivers a different message on the screen based on the status code returned and invokes the *getUserList* method again after a successful deletion to get the updated record list of users.

Click on the edit button allows the user to edit the recognition result. The front-end sends the record id and updates the result data to the back-end. After receiving the data, the back-end will modify the corresponding recognition result in the database according to the id and return the status code 200 if the modification is successful, or 0 if the modification is unsuccessful.

After the user has typed keywords in the query bar, the front-end sends a POST request to the back-end port */recordList/search* with the data of the keyword and the username. The back-end receives the data, connects to the database, and queries the records for the specified keywords, then returns a list of query results. The front-end receives the results and displays them as a list on the screen.

Chapter 5

Results

This chapter presented the structure of the convolutional neural network designed for training. The training environment and configuration were firstly described, followed by the structure of the hybrid dataset used. Then, the pre-processing and enhancement of the training data were introduced as well. The CNN structure and relative parameter settings were then presented. The performance of the trained CNN was evaluated by plots. The accuracy of the CNN proposed was compared against VGG-16. The ideal algorithm should be able to recognize facial expression with above 85% accuracy and with the least amount of processing time possible. Finally, the confusion matrix that reflected the accuracy of each classification was shown, and the implications of the training curves as well as the existing problems of the network were analyzed.

5.1 Experimental Environment

The configuration of the experiment was as follows: The experiment was conducted under Keras from Tensorflow which is a deep learning framework. Central Processing Unit (CPU): Intel Corei7-7700HQ with 2.8GHz and 32GB of RAM; Graphic Processing Unit (GPU): NVIDIA GeForce RTX 2080 Ti with 11GB.

5.2 Training Dataset

In order to diversify the angles of facial expressions and to achieve the goal that the FER system could obtain relatively superior recognition performance in realistic background environmental conditions, the data in the FER+ database and the CK+ database were mixed and sorted in this experiment. Since CK+ contains a number of sequential sets of images, the last three images of each expression sequence were collected to represent the sequence. In this experiment, this hybrid dataset was pre-processed and then used to train a convolutional neural network model.

The background information in raw expression images, such as those produced by photos with camera devices, may contain distracting information when used to do the classification in CNN model. For face expression recognition, in addition to environmental distractions, the human ear, and hair, etc. were also considered background information. The background information would increase the complexity and decrease the accuracy of the classification. It was thus necessary to remove the areas in the image that provide no expression information. In this section, the original images in the dataset were cropped and only the facial part of the images which contained informative details of the expression were retained, resulting in a facial expression dataset of size 48*48 pixels images. Figure 5.1 shows the processed face expression dataset images.

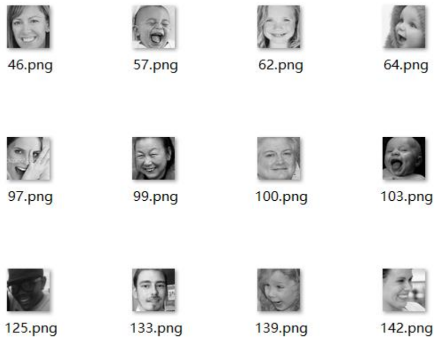
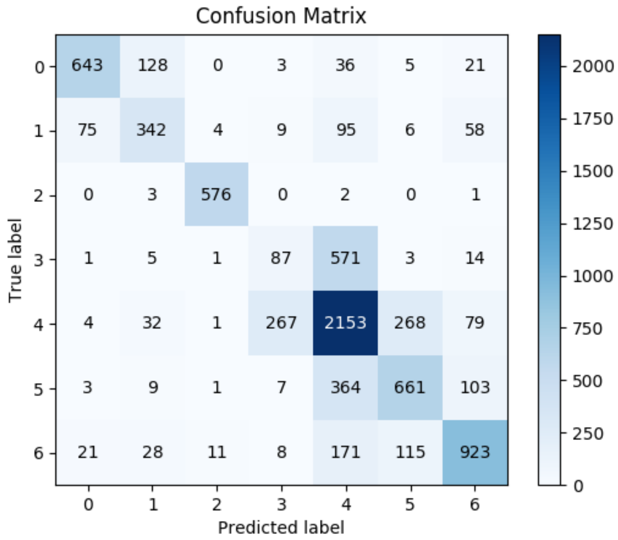


Figure 5.1: Examples of the Training Dataset Images

To avoid the low training accuracy and over-fitting problems caused by insufficient training data, the images were augmented in this experiment. Considering the special characteristics of facial images, the results of the recognition would be influenced by the vertical flip of the images, therefore, only the horizontal flip of the images was applied. In addition, ZCA whitening was applied to the images to reduce the redundant information, so that all the features in the images had equal variance. In the combined dataset, the number of samples for the happy and neutral classifications was high, reaching over 9000 and 10000, while the number of samples for the disgusting and fear was low, both less than 1000. The proportion of images for each category was very different and imbalanced. This extreme uneven distribution of data would make the model somewhat "inert", i.e., most of the images would be classified into the category which has the largest amount of data. in the recognition and to cause over-fitting.



	0	1	2	3	4	5	6
0	643	128	0	3	36	5	21
1	75	342	4	9	95	6	58
2	0	3	576	0	2	0	1
3	1	5	1	87	571	3	14
4	4	32	1	267	2153	268	79
5	3	9	1	7	364	661	103
6	21	28	11	8	171	115	923

Figure 5.2: The Confusion Matrix of The Improved CNN Model Training On The Unbalanced Dataset

Figure 5.2 presents the confusion matrix of the improved model which was trained on the unbalanced dataset. As can be seen from the figure, the image with a true label of 3 had 571 data classified to label 4 which contained a larger quantity of data, resulting a poor classification accuracy on the fourth category.

To prevent this issue, the categories with fewer data were upsampled, i.e., the proportion of data in each category was kept in balance by using replication for the data in the categories with fewer data. On top of the hybrid dataset, the disgust and fear data were replicated and expanded to three times the size of the original data. Moreover, down-sampling was also applied to the categories with more images, i.e., some of the images were dropped from the denser categories so that the proportion of each category was kept at an equally balanced level. In addition, to further minimize the imbalance in the dataset, the *create_class_weight* function was used to calculate the weight of each classification image based on its quantity, and the weight array was assigned to the *class_weight* parameter during the training of the model to reduce the training loss caused by the imbalance. Figure 5.3 shows the proportion of data in each category in the unbalanced dataset and the balanced dataset.

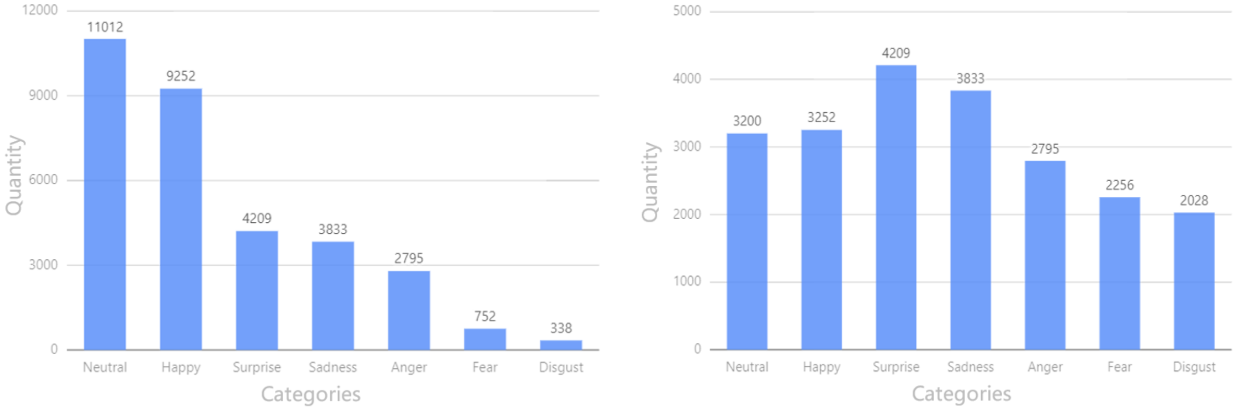


Figure 5.3: Distribution of Data Before and After Balancing

The structure of the dataset after the data balancing process is shown in Table 5.1.

Neutral	Happy	Surprise	Sadness	Anger	Fear	Disgust	Total
3200	3252	4209	3833	2795	2256	2028	21573

Table 5.1: Structure of The Dataset After Balancing

5.3 Hyper-parameter Settings

This paper used Adam as the optimization algorithm of the loss function in the network, which required less tuning [An Optimizer for Stochastic Gradient Descent]. The training set images were divided into 58 batches, each with 260 images, for batch training. The input size of the network is 48*48 pixels, and the initial learning rate was set to 0.0001. During the training process, the learning rate gradually decreased at a decay rate of 10^{-6} as the training iterations increased, and the maximum number of iterations (Epoch) was 300.

5.4 Convolutional Neural Network Structure

The network structure was designed on the basis of VGG-16, which had been improved to provide higher recognition accuracy for solving the FER problem and faster training speed, thus reaching the requirements of practical application.

The structure of VGG-16 is described thoroughly in Chapter 2. This section presented the results of training the dataset that has been constructed in this paper using VGG-16 and analyzed the results which leading to how the neural network was designed in this paper. The training plots with accuracy and loss are shown in Figure 5.4.

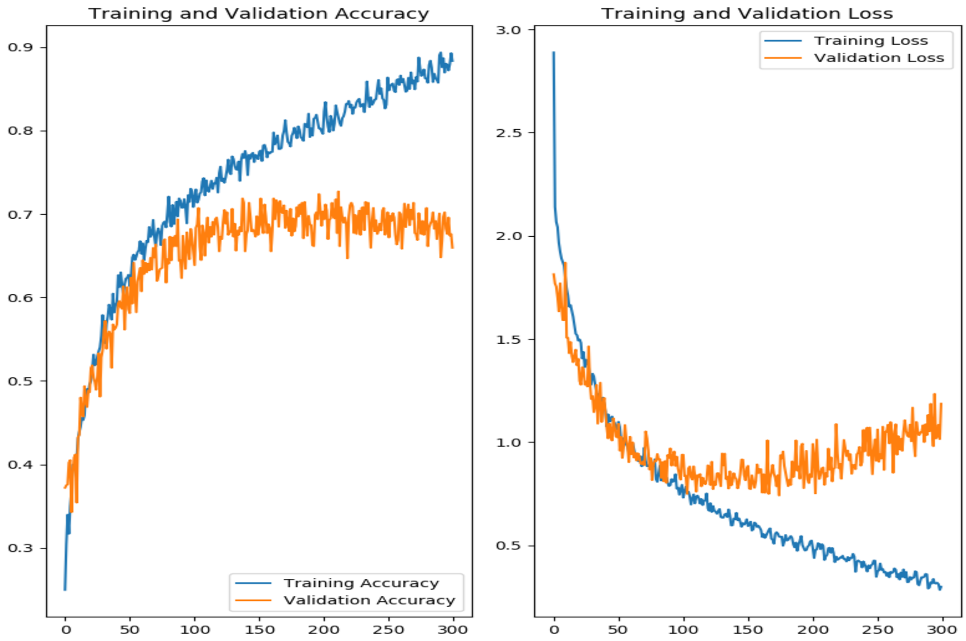


Figure 5.4: Accuracy and Loss Curves of VGG-16

While using the VGG-16 to train the data, the model had achieved an accuracy of 89% on the training set, yet merely 72% on the testing set, with a gap of nearly 20% between them, implying that the model had a relatively strong capability of classification on the training set, inside of the testing set. As a result, the performance of the model outside the training data set is not as good as required for practical applications.

It could be noted from the loss curves that the loss of the model in the training set was continuously dropping until it was below 0.5, whereas the loss in the test set failed to drop

below 0.5, and instead showed an increasing trend after 150 epochs, reaching over 1.0 at 300 epochs. This indicates that the model was over-fitting, which showed that the complexity of the model was too high as compared to the FER problem to be solved.

Therefore, this model was improved on the VGG-16 structure. Due to the abundant parameters of the fully connected layer in VGG-16, the parameters of the fully connected layer were adjusted to 256 and 512, and the activation function of the last layer was changed to softmax which was used for the classification problem.

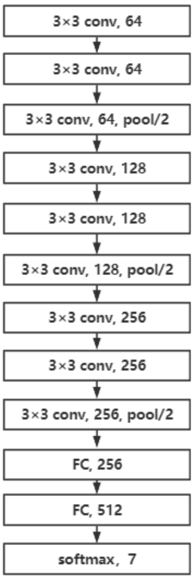


Figure 5.5: The Structure of The Improved CNN Model

Therefore, the number of model parameters was downsized, saving the memory usage during the training and increasing the processing speed. In addition, the layers were adjusted from 13 in VGG-16 to 9, and the numbers of convolutional filters were reduced from 64, 128, 256, and 512 to 64, 128, and 256, so that the complexity of the model could be optimized, and over-fitting would be avoided. Furthermore, 25% dropout and 2*2 pixels sized max-pooling layers were used to partially remove the extracted features in feature maps. The structure of the improved CNN is shown in Figure 5.5 and Table 5.2.

The CNN model was designed with 9 convolutional layers, 3 pooling layers, and 3 fully connected layers. The processed images were sent into the CNN model with a size of 48*48 pixels and three channels - R, G, and B. The first three layers were convolutional layers with

64 filters in 3*3 pixels size, followed by ReLU activations. The padding is 1 pixel (same padding), and the convolution stride is fixed at 1 pixel.

In addition, Batch Normalization was also used after every convolutional layer to reduce the internal covariate shift of the network, which helped to speed up the training process.

A dropout layer was followed, randomly dropping 25% of neurons in the layer to prevent overfitting. Then, a max-pooling layer with a 2*2 pixels sized filter was added behind the first three convolutional layers for down-sampling the features.

The structure of convolutional layers 4,5, and 6 was nearly identical to that of the prior three convolutional layers. The number of filters was increased to 128 while preserving the 3*3 pixels size. These convolutional layers were followed by a dropout layer and a max-pooling layer that hold the same parameters as before.

The convolutional layer 7,8,9 raised the number of the convolutional filters to 256, whilst maintaining the other parameters. These layers were also followed by the same max-pooling layer and dropout layer.

Finally, the output from the previous max-pooling layer was flattened and then fed into three fully connected layers. The numbers of neurons in the first two FC layers were 256 and 512, with ReLU activation function and 25% dropout for each layer. The last layer was a special fully-connected layer which used a softmax activation function. This layer held 7 neurons, in order to give the probabilities for each class label.

Type	Filter	Stride	Output	Dropout
Input Layer			3*48*48	
Convolutional Layer 1	64*3*3	1	64*48*48	
Convolutional Layer 2	64*3*3	1	64*48*48	
Convolutional Layer 3	64*3*3	1	64*48*48	25%
Max-pooling 1	2*2	2	64*24*24	
Convolutional Layer 4	128*3*3	1	128*24*24	
Convolutional Layer 5	128*3*3	1	128*24*24	
Convolutional Layer 6	128*3*3	1	128*24*24	25%
Max-pooling 2	2*2	2	128*12*12	
Convolutional Layer 7	256*3*3	1	256*12*12	
Convolutional Layer 8	256*3*3	1	256*12*12	
Convolutional Layer 9	256*3*3	1	256*12*12	25%
Max-pooling 3	2*2	2	256*6*6	
Fully Connected Layer 1	256		256*1*1	25%
Fully Connected Layer 2	512		512*1*1	25%
Output Layer (Softmax)			7*1*1	

Table 5.2: The Structure of The Improved CNN Model

5.5 Results Analysis

The training result of the improved CNN model on the dataset are shown in Table 5.3.

DATA	QUANTITY	ACCURACY	LOSS
TRAIN	15101	0. 9337	0. 1808
TEST	6472	0. 87375	0. 4037

Table 5.3: The Training Result of The Improved CNN Model

The performance of the CNN model in each category is shown in the confusion matrix (Figure 5.6), where the labels 0 to 6 represent Disgust, Anger, Happy, Neutral, Surprise, Fear and Sadness.

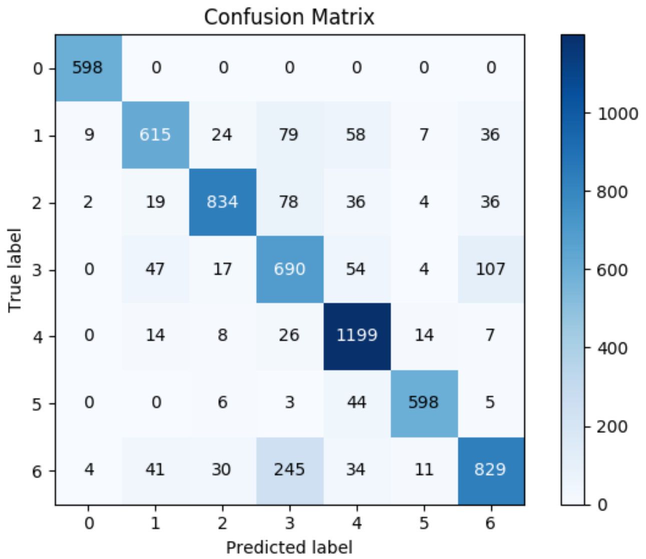


Figure 5.6: The Confusion Matrix of The Improved CNN Model Training On The Hybrid Dataset

As can be seen from the confusion matrix shown in Figure 5.6, the model performed relatively well in all categories, performing best in recognizing happy and surprise categories with high accuracy, yet with lower accuracy for some of the negative expression categories, in particular fear and sadness. The possible reasons for this problem were that facial features varied less significantly when expressing such expressions, feature extraction was not sufficient, the recognition capability of the model was still inadequate, and it was easy to be confused with other expression categories.

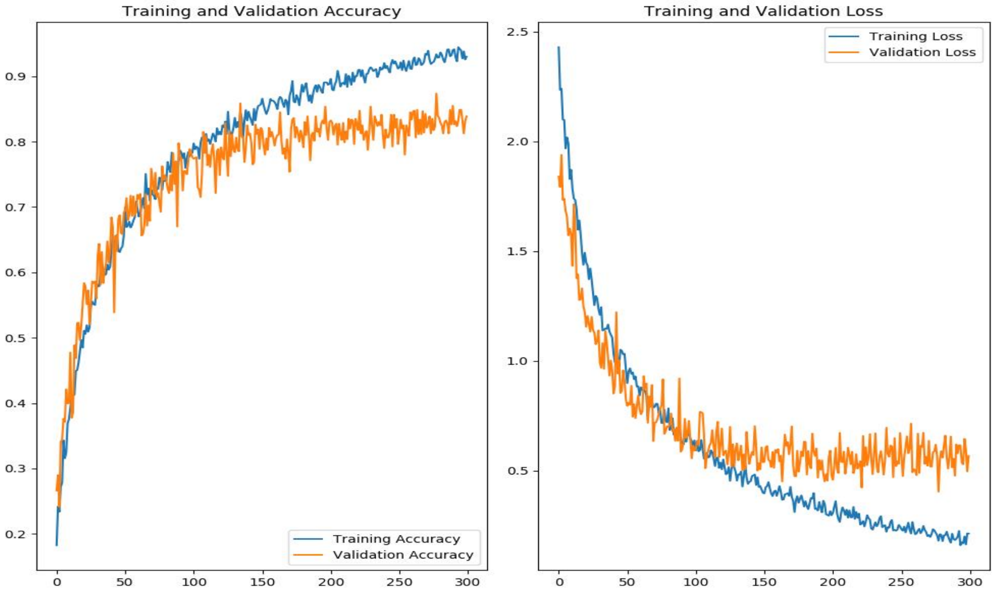


Figure 5.7: The Accuracy and Loss Curves of The Improved CNN Model On The Hybrid Dataset

Figure 5.7 shows the accuracy and loss curves of the model in the training and validation sets during the training process. It is illustrated in Figure 5.7 that the accuracy of the CNN model rose sharply between 0 and about 130 Epochs in the test set, then slow down after 150 epochs, and gradually stabilized between 150 and 300 epochs, with a final training accuracy of 87.3%. The accuracy of the training set still tended to increase after 150 epochs, till 300 epochs when the accuracy steadily raised to over approximately 90%, reaching a final accuracy of 93.3%. The loss curve shows that the loss rate of the CNN model between 0 and 150 Epochs decreased rapidly on both the training and test sets, with a slight over-fitting of the model starting at 200 epochs. Overall, the over-fitting was considered to be less severe, and acceptable. Until 300 epochs, the loss rate of the model converged to a fairly satisfying level, with a final loss rate of 0.1808 on the training set and 0.4037 on the test set. In general, the improved model resulted in better classification performance for the problem and was able to meet the basic requirements of the system for expression recognition.

5.6 System Testing

This section tested the static facial expression recognition system. The test environment was Chrome on Windows 10 with version 89.0.4389.82. After deploying the system on the testing device, the browser was opened to the website http://192.168.0.198:8080/ for testing. The test consisted of two main parts. The first part was to test the flow of the system in practice, with a usability check of every component included, ensuring that the page display matched the design in every situation. The second part was to test the accuracy of the expression recognition module. The test samples were both images from the test set used to train the neural network and photographs taken in a real environment. The results of these two tests were then compared.

Tests were carried out according to the designed system modules and the results obtained from the login and registration module are shown in Appendices 1 to 4. Table 5.4 summarizes the test items, test data and results for this module.

Testing Items	Testing Data	Testing Results
Incorrect user login information	The username was *admin,* and the password was *111111111111111*	User login failed, incorrect username or password
Duplicate username on registration	Username already existed as *admin*	The username already existed, and registration had failed
Username or password too short on registration	Username and password length less than 6	The username and password were too short
Registration successful	Unregistered username and password of suitable length	Sign-up successful

Table 5.4: Login and Registration Module Test Results

According to the above test results, the system login and registration module successfully completed the tests for various situations and generally met the requirements for operation. The following tests were carried out on the user settings section and the results are summarized in Table 5.5. Finally, the identification records module was tested, and the results are summarized in Table 5.6. The test results are shown in Appendices 5 to 8.

Testing Items	Testing Data	Testing Results
The old password was incorrect	The username was *admin,* and the old password was *111111111111111*	User login failed, wrong username or password
New password with short length	Password length less than 6	Password length too short
Modification successful	The old password was successfully verified, and the new password was the appropriate length	Modification successful

Table 5.5: Test Results of the User Settings Module

Testing Items	Testing Data	Testing Results
Displaying the identification records list	Accessed to the recognition records page	Displayed the list of user recognition records correctly
Deleting records	Deleted a record from the list	Deleted successfully
Modifying identification results	Edited an existing record	Modification successful
Searching records	Searched list by expression category	The list was displayed correctly

Table 5.6: Test Results of the Recognition Recording Module

The test results for the expression recognition module are shown in Table 5.7 and Appendix 9-16.

Testing Items	Testing Results
Accessing to expression recognition page	Successfully invoked the camera on device
Turning off the camera	Clicked on the *Turn off The Camera* button to successfully close the camera
Turning on the camera	Click on the *Turn on The Camera* button to successfully open the camera
Taking a photo	Click on the *Take a picture* button and the photo was displayed successfully on the right side of the screen
Uploading the image	Clicked on the *Upload* button, entered the image name, confirmed, and successfully uploaded, then the successful uploaded message was displayed
Recognizing the expression	Clicked on the *Recognize* button, the recognition result and face detection image were successfully displayed on the right side of the screen

Table 5.7: Test results of the Expression Recognition Module

The test results of expression recognition by uploading the local images are presented in Appendix 17-19. The system test was run on a regular internet connection and could not be simulated and tested in an off-line situation. As can be seen from the test results, all features in this recognition module were successfully tested. When the quality of the test images was high, the image recognition effect was relatively better.

In a few cases, the recognition results were incorrect, the possible reasons for which might be:

- Incorrect face detection, which failed to correctly match the range of faces, leading to incorrect expression recognition results. The face detection algorithm needs to be further optimized.
- The face region was correctly detected, and the feature extraction result was inaccurate, so the expression recognition result was wrong. The expression recognition dataset needs to be further improved.

To address the above situation, in terms of system optimization, a face detection algorithm with a better recognition effect can be used so that its face detection effect in complex backgrounds can also meet the requirement of the application. At the same time, before recognizing expressions, expression data can be collected in advance for the recognized target to build a specialized expression library to train the model and improve the accuracy.

Chapter 6

Conclusion

With the advent of the Big Data era, the development and achievements of artificial intelligence and other fields are attracting attention, whether in business, education, healthcare, or transportation, the results of which have mature applications and dazzle in various fields. As facial expression recognition plays an irreplaceable role in human-computer interaction, studies in this field are still a key focus of scientific research, and research results and breakthroughs in expression recognition will play an important role in boosting future technological development.

Based on a summary of expression recognition algorithms and deep learning literature, this paper proposed a CNN model for human facial expression recognition and, used a B/S architecture and separated front and back ends to build a system for static human expression recognition. The main work accomplished in this paper is outlined as follows:

- By studying and summarizing the state-of-the-art in the field of expression recognition, learning and referring to their principal design ideas, summarizing the advantages and disadvantages, and combining the research needs in this paper, an optimized CNN model structure with several parameter settings improved from VGG-16 was introduced and used to train for the hybrid dataset in the experiment.
- The image data in the FER+ and CK+ datasets were mixed and then categorized to obtain seven categories of expression classification, and the imbalanced data were pre-processed to minimize the influence of data imbalance on the training accuracy of the model. It was then used for model training. The blended data set had diverse samples and helped to achieve better performance for model training.
- In this paper, a static expression recognition system was built using a B/S architecture and separated the front-end and back-end. The detailed features and system operation process were then presented. Furthermore, the reliability and the accuracy of the model were tested. The images used for testing are all taken by a camera on the computer in a realistic context, ensuring that the system had a certain application potential.

Chapter 7

Future Work

Whether from the recognition performance or system functionalities of this FER system, there are still certain limitations that are needed to be improved further. The following are the points that needed to be improved:

Environment Context While the CNN model is recognizing facial expressions, important information such as the environment context cannot be ignored [21, p. 22]. In a cheerful situation, such as a celebration party, happiness is more weighted than any other expressions, therefore even if some people are not laughing or showing visible happiness, this factor should still be taken into account.

3D FER The emergence of 3D FER datasets in recent years, such as BU-3DFE [34, pp. 211–216] and BP4D [35, pp. 692–706], has led to further progress in solving expression recognition challenges. Acquiring high-quality 3D facial images tackles illumination and pose-change issues and provides more details about the facial muscle movements driven by expressions [36, p. 14]. Moreover, a multimodal approach 2D+3D could also be considered since there are just a few 3D databases available as well as the larger storage and computational expenses due to higher dimensionality.

System Optimization As for the system optimization, this system has only been tested on Chrome. The next step is to try to test the functionalities and performance of the system on different versions of browsers and operating systems to improve its compatibility and portability. Moreover, the system should not only be run on a website, but also developed as a mobile application, so that features such as real-time recognition will be able to work better.

Future research should be continued on the above aspects in the field of expression recognition, so that facial expression recognition system can be fully applied on a day-to-day basis.

Bibliography

[1] A. Mehrabian, ‘Silent messages: Implicit communication of emotions and attitudes’, *Wadsworth,* 1971.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ‘ImageNet classification with deep convolutional neural networks’, *Commun ACM,* vol. 60, no. 6, 2017, doi: 10.1145/3065386.

[3] C. Szegedy *et al.*, ‘Going deeper with convolutions’, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* 2015, vol. 07-12-June-2015. doi: 10.1109/CVPR.2015.7298594.

[4] K. Simonyan and A. Zisserman, ‘Very deep convolutional networks for large-scale image recognition’, 2015.

[5] J. K. Patel and A. Sakadasariya, ‘Survey on virtual reality in social network’, 2018. doi: 10.1109/ICISC.2018.8399026.

[6] H. S. Cha, S. J. Choi, and C. H. Im, ‘Real-time recognition of facial expressions using facial electromyograms recorded around the eyes for social virtual reality applications’, *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.2983608.

[7] C. Darwin and F. Darwin, *The expression of the emotions in man and animals*. 2009. doi: 10.1017/CBO9780511694110.

[8] P. Ekman, ‘Facial expression and emotion’, *American Psychologist,* vol. 48, no. 4, 1993, doi: 10.1037/0003-066X.48.4.384.

[9] A. S. Vyas, H. B. Prajapati, and V. K. Dabhi, ‘Survey on Face Expression Recognition using CNN’, 2019. doi: 10.1109/ICACCS.2019.8728330.

[10] Z. Wen and T. S. Huang, ‘Capturing subtle facial motions in 3D face tracking’, in *Proceedings of the IEEE International Conference on Computer Vision*, 2003, vol. 2. doi: 10.1109/iccv.2003.1238646.

[11] M. Pantic and M. Stewart, ‘Machine Analysis of Facial Expressions’, in *Face Recognition,* 2007. doi: 10.5772/4847.

[12] X. Feng, ‘Facial expression recognition based on local binary patterns and coarse-to-fine classification’, in *Proceedings - The Fourth International Conference on Computer and Information Technology (CIT 2004),* 2004, pp. 178–183. doi: 10.1109/cit.2004.1357193.

[13] Z. Wang and Z. Ying, ‘Facial expression recognition based on local phase quantization and sparse representation’, 2012. doi: 10.1109/ICNC.2012.6234551.

[14] J. Li *et al.*, ‘Facial Expression Recognition with Faster R-CNN’, in *Procedia Computer Science*, 2017, vol. 107. doi: 10.1016/j.procs.2017.03.069.

[15] K. Liu, M. Zhang, and Z. Pan, ‘Facial Expression Recognition with CNN Ensemble’, 2016. doi: 10.1109/CW.2016.34.

[16] H. Q. Khor, J. See, R. C. W. Phan, and W. Lin, ‘Enriched long-term recurrent convolutional network for facial micro-expression recognition’, 2018. doi: 10.1109/FG.2018.00105.

[17] I. Krizhevsk, Alex Sutskever and G. E. Hinton, ‘ImageNetClassificationWith DeepConvolutionalNeural Networks’, *Advances In Neural Information Processing Systems,* 2012.

[18] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, ‘Dropout: A simple way to prevent neural networks from overfitting’, *Journal of Machine Learning Research*, vol. 15, 2014.

[19] A. Agrawal and N. Mittal, ‘Using CNN for facial expression recognition: a study of the effects of kernel size and number of filters on accuracy’, *Visual Computer*, vol. 36, no. 2, 2020, doi: 10.1007/s00371-019-01630-9.

[20] S. H. S. Basha, S. R. Dubey, V. Pulabaigari, and S. Mukherjee, ‘Impact of fully connected layers on performance of convolutional neural networks for image classification’, *Neurocomputing*, vol. 378, 2020, doi: 10.1016/j.neucom.2019.10.008.

[21] D. Canedo and A. J. R. Neves, ‘Facial expression recognition using computer vision: A systematic review’, *Applied Sciences (Switzerland)*, vol. 9, no. 21. MDPI AG, Nov. 01, 2019. doi: 10.3390/app9214678.

[22] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, ‘Coding facial expressions with Gabor wavelets’, 1998. doi: 10.1109/AFGR.1998.670949.

[23] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, ‘The extended Cohn-Kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression’, 2010. doi: 10.1109/CVPRW.2010.5543262.

[24] Research Prediction Competition, ‘Challenges in Representation Learning: Facial Expression Recognition Challenge’, *Kaggle*, 2013.

[25] MySQL, ‘MySQL Documentation’, *MySQL documentation*. 2018.

[26] A. Sharma, ‘Introduction to HTML (Hyper Text Markup Language) - A Review Paper’, *International Journal of Science and Research (IJSR)*, vol. 7, no. 5, 2018.

[27] D. Megida, ‘What is JavaScript? A Definition of the JS Programming Language’, *freeCodeCamp*, 2021.

[28] W3C, ‘HTML &amp; CSS - W3C’, *2016*, 2016.

[29] S. bin Uzayr, N. Cloud, and T. Ambler, ‘Vue.js’, in *JavaScript Frameworks for Modern Web Development*, 2019. doi: 10.1007/978-1-4842-4995-6_14.

[30] G. Mainland, G. Morrisett, and M. Welsh, ‘Flask’, *ACM SIGPLAN Notices*, vol. 43, no. 9, 2008, doi: 10.1145/1411203.1411251.

[31] S. Campbell, ‘Flask vs Django: What’s the Difference Between Flask & Django?’, *Guru99*, 2021.

[32] P. Chen, ‘Entity-Relationship Modeling: Historical Events, Future Trends, and Lessons Learned’, in *Software Pioneers*, 2002. doi: 10.1007/978-3-642-59412-0_17.

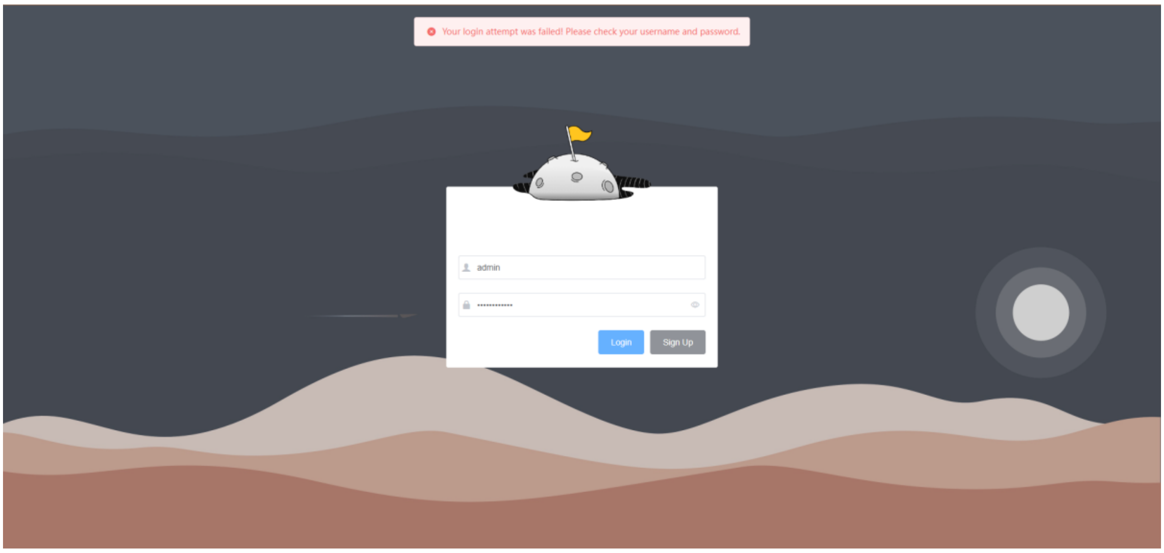
[33] James. Rumbaugh, Ivar. Jacobson, and Grady. Booch, *The unified modeling language reference manual*. Addison-Wesley, 1999.

[34] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato, ‘A 3D facial expression database for facial behavior research’, in *FGR 2006: Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition*, 2006, vol. 2006. doi: 10.1109/FGR.2006.6.

[35] X. Zhang *et al.*, ‘BP4D-Spontaneous: A high-resolution spontaneous 3D dynamic facial expression database’, *Image and Vision Computing*, vol. 32, no. 10, 2014, doi: 10.1016/j.imavis.2014.06.002.

[36] F. Nonis, N. Dagnes, F. Marcolin, and E. Vezzetti, ‘3D approaches and challenges in facial expression recognition algorithms-A literature review’, *Applied Sciences (Switzerland)*, vol. 9, no. 18. MDPI AG, Sep. 01, 2019. doi: 10.3390/app9183904.

Appendix – FER System Test

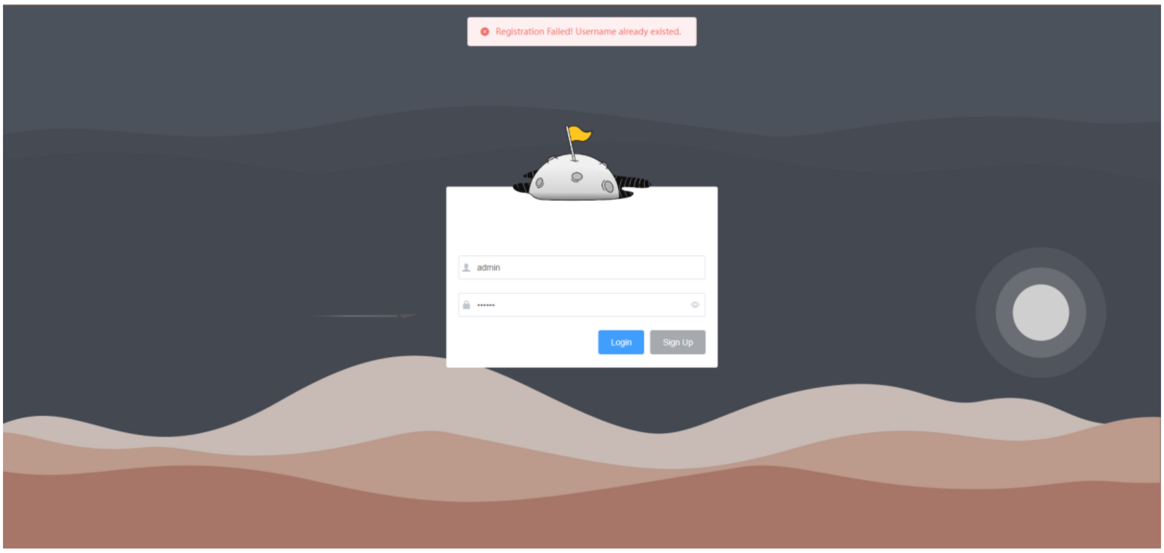


Your login attempt was failed! Please check your username and password.

admin

•••••••••••••

Figure 1: Login Failed

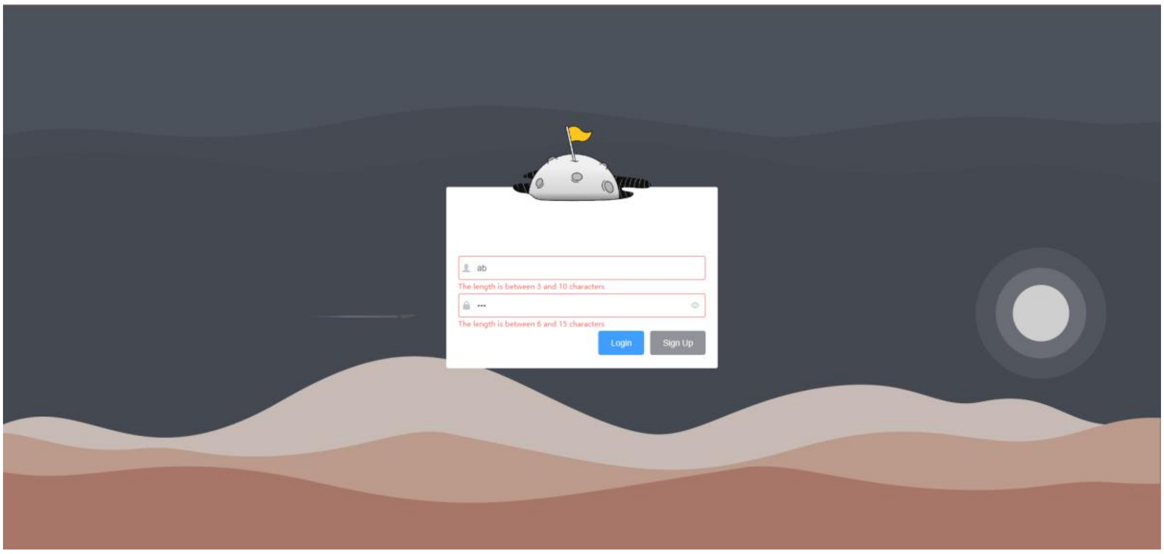


Registration Failed! Username already existed.

admin

••••••

Figure 2: Sign-up Failed



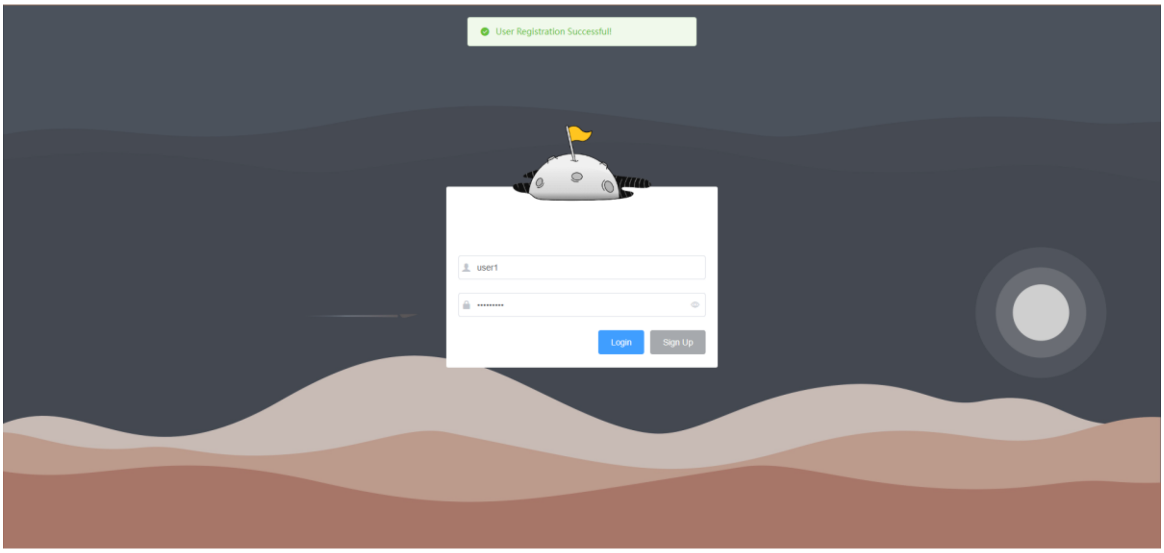
ab

The length is between 3 and 10 characters

•••

The length is between 6 and 15 characters

Figure 3: Username And Password Length Too Short

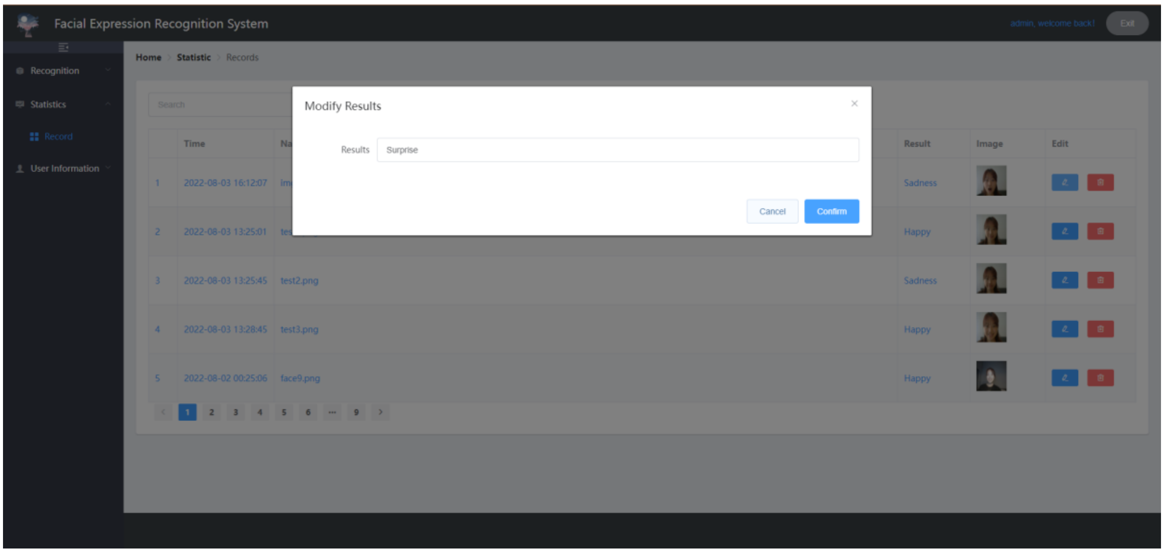


User Registration Successful!

user1

••••••••••

Figure 4: Sign-up Successful



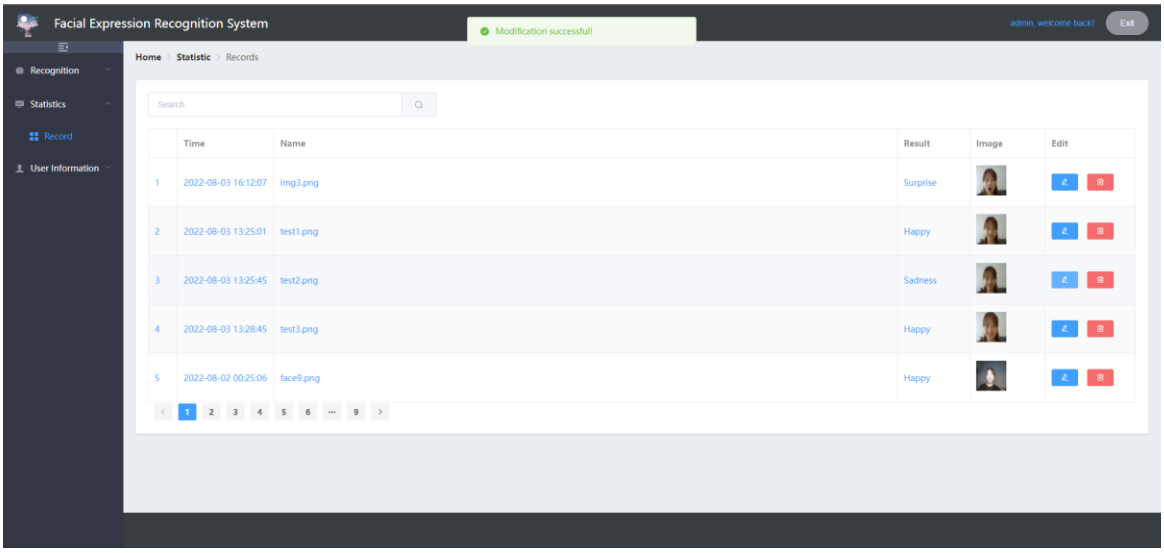
Modify Results

Results: Surprise

	Time	Na		Result	Image	Edit
1	2022-08-03 16:12:07	im		Sadness		
2	2022-08-03 13:25:01	tes		Happy		
3	2022-08-03 13:25:45	test2.png		Sadness		
4	2022-08-03 13:28:45	test3.png		Happy		
5	2022-08-02 00:25:06	face9.png		Happy		

< 1 2 3 4 5 6 ··· 9 >

Figure 5: Edit Recognition Result

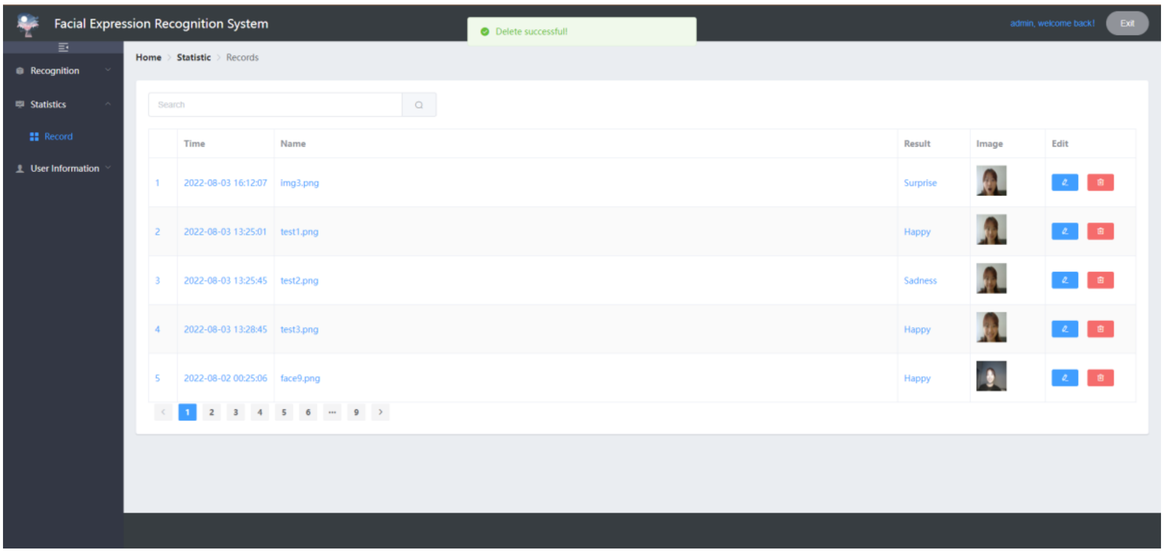


Modification successful!

	Time	Name	Result	Image	Edit
1	2022-08-03 16:12:07	img3.png	Surprise		
2	2022-08-03 13:25:01	test1.png	Happy		
3	2022-08-03 13:25:45	test2.png	Sadness		
4	2022-08-03 13:28:45	test3.png	Happy		
5	2022-08-02 00:25:06	face9.png	Happy		

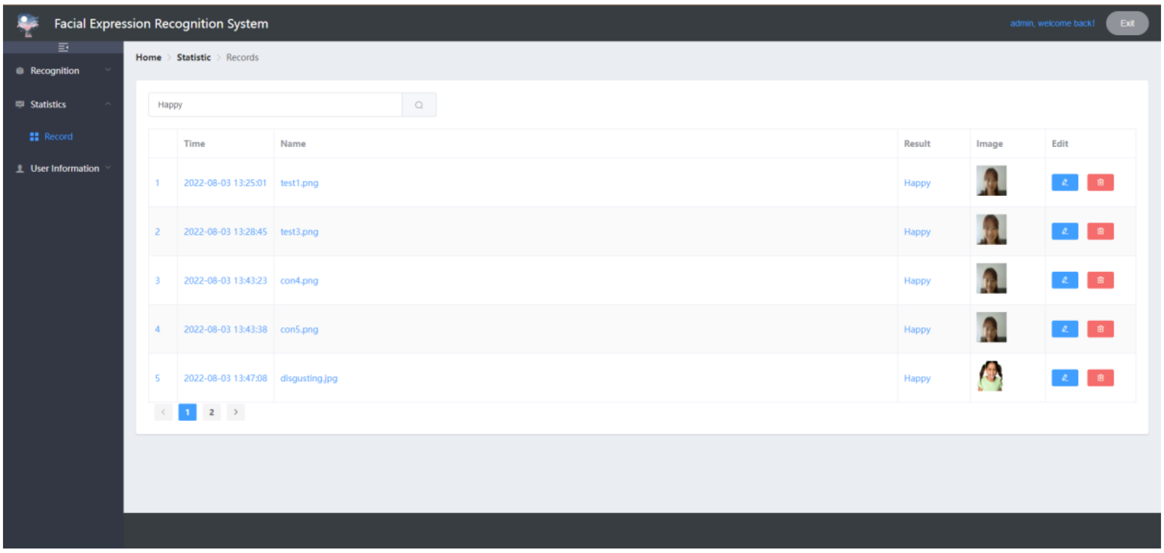
< 1 2 3 4 5 6 ··· 9 >

Figure 6: Modify Recognition Result Successful



	Time	Name	Result	Image	Edit
1	2022-08-03 16:12:07	img3.png	Surprise		
2	2022-08-03 13:25:01	test1.png	Happy		
3	2022-08-03 13:25:45	test2.png	Sadness		
4	2022-08-03 13:28:45	test3.png	Happy		
5	2022-08-02 00:25:06	face9.png	Happy		

Figure 7: Delete Recognition Result Successful



	Time	Name	Result	Image	Edit
1	2022-08-03 13:25:01	test1.png	Happy		
2	2022-08-03 13:28:45	test3.png	Happy		
3	2022-08-03 13:43:23	con4.png	Happy		
4	2022-08-03 13:43:38	con5.png	Happy		
5	2022-08-03 13:47:08	disgusting.jpg	Happy		

Figure 8: Search Records By Category

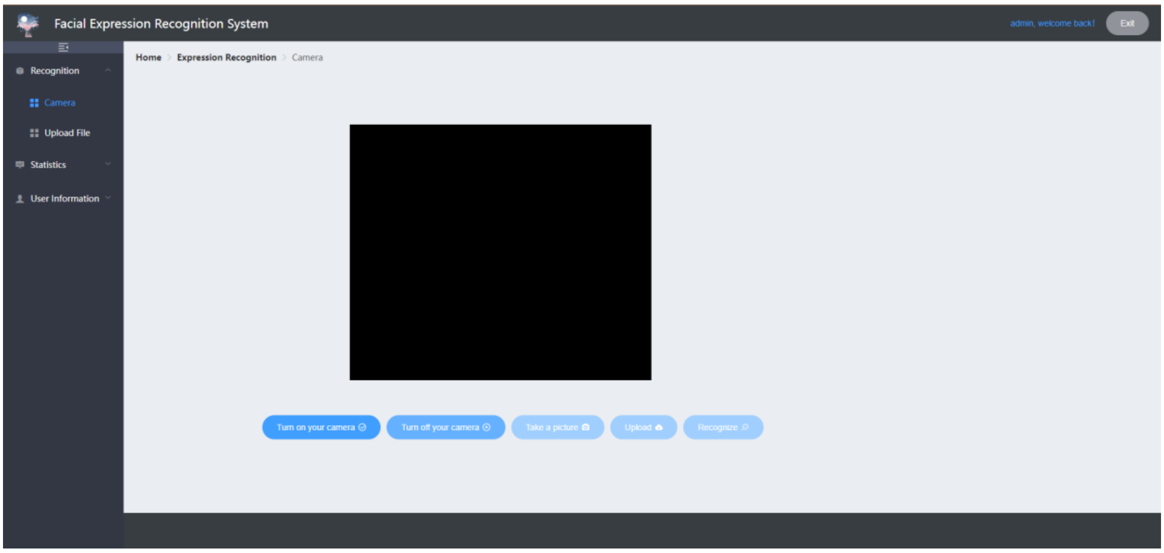


Figure 9: Turn off The Camera

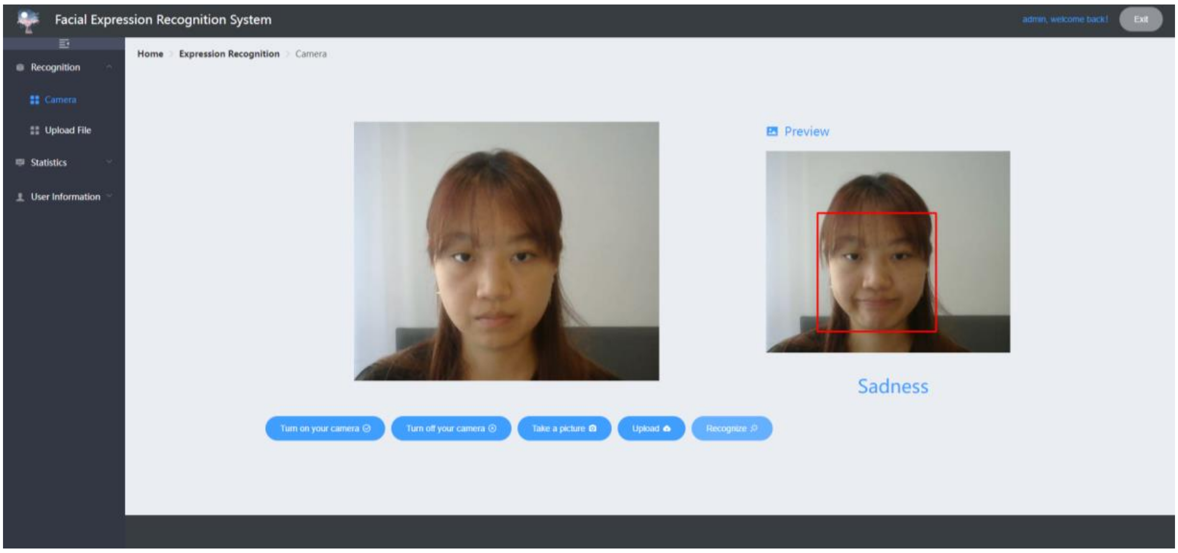


Figure 10: Recognition Result - Sadness

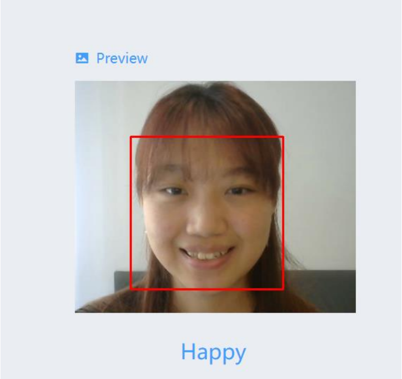


Figure 11: Recognition Result - Happy

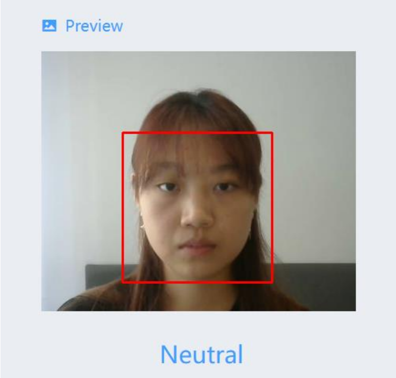


Figure 12: Recognition Result - Neutral

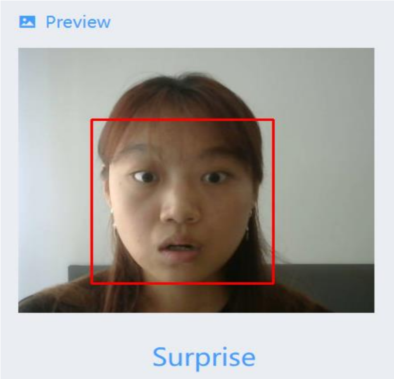


Figure 13: Recognition Result - Surprise

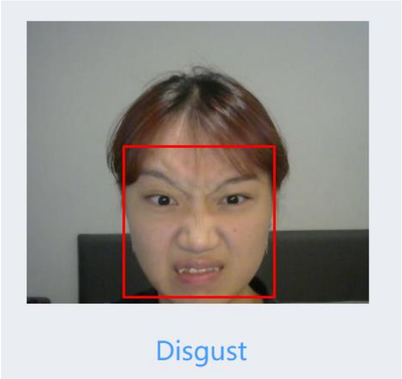


Figure 14: Recognition Result - Disgust

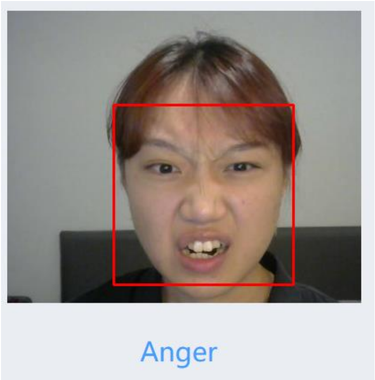


Figure 15: Recognition Result - Anger

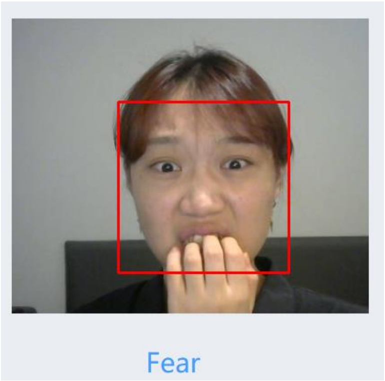


Figure 16: Recognition Result - Fear

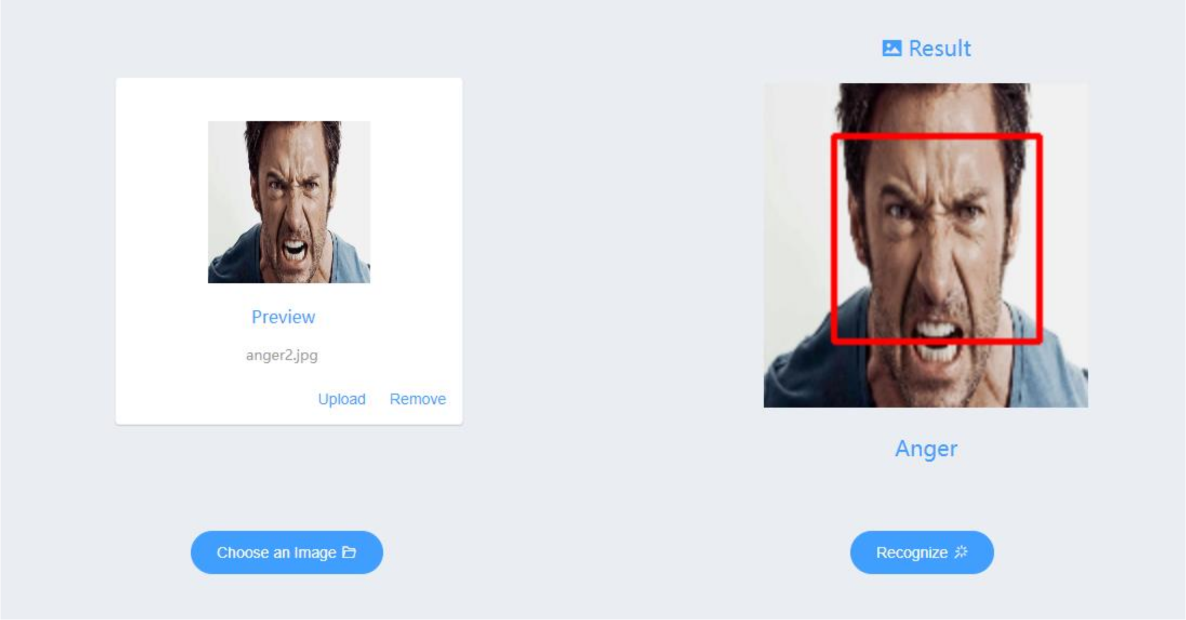


Figure 17: Recognition Result of Uploaded Image- Anger

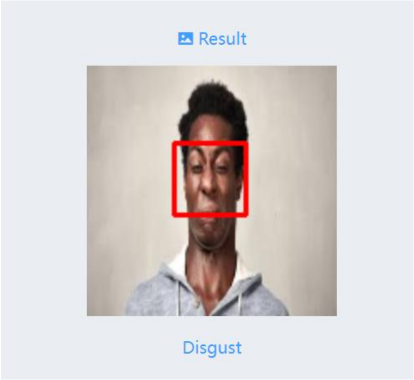


Figure 18: Recognition Result of Uploaded Image- Disgust

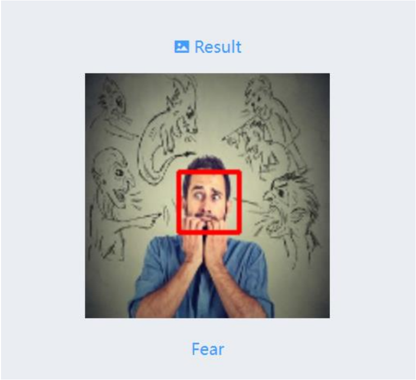


Figure 19: Recognition Result of Uploaded Image- Fear