Attendance system based on face recognition

Detailed development plan

Project Name: Attendance system based on face recognition

Project Unit: Group 9

Project Time: 2019.2.28 -2019.5.10

Version Number: 2.0

Content

1. Introduction

1.1 Document purpose

In order to standardize the module design of the software, we provide a detailed description of the software functions, input and output, algorithm flow logic and design documentation. This document will also provide reference for the later stages of programming and software testing.

1.2 Background

(1) Software project name：FaceRecognitionAttendance-FRA system

(2) Project participant：

Name	**Position**	**Duties**
WanHongda	Member	Face recognition algorithm design, API writing and APP Implementation
ZhangJiaqing	Member	Information Management Interface and APP Design
ZhangWei	Leader	Server design and writing, Information Management writing and database construction

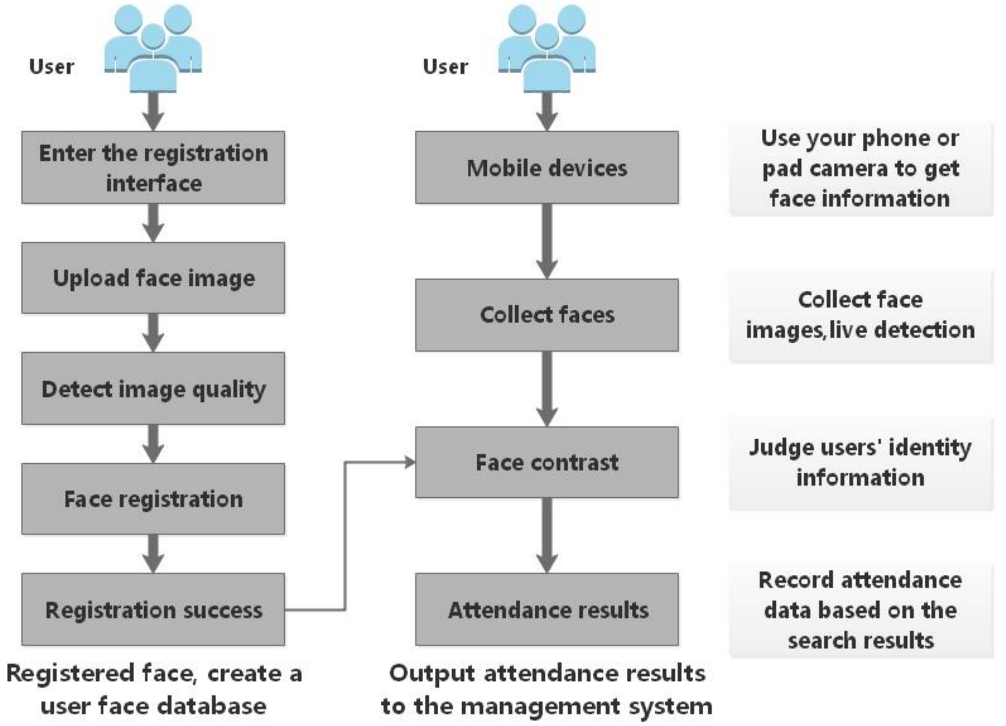
1.3 References

[1].T. Ojala, M. Pietikäinen, and D. Harwood (1996), "A Comparative Study of Texture Measures with Classification Based on Feature Distributions", Pattern Recognition, vol. 29, pp. 51-59.

[2].Ding C , Choi J , Tao D , et al. Multi-Directional Multi-Level Dual-Cross Patterns for Robust Face Recognition[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2014, 38(3):518-531.

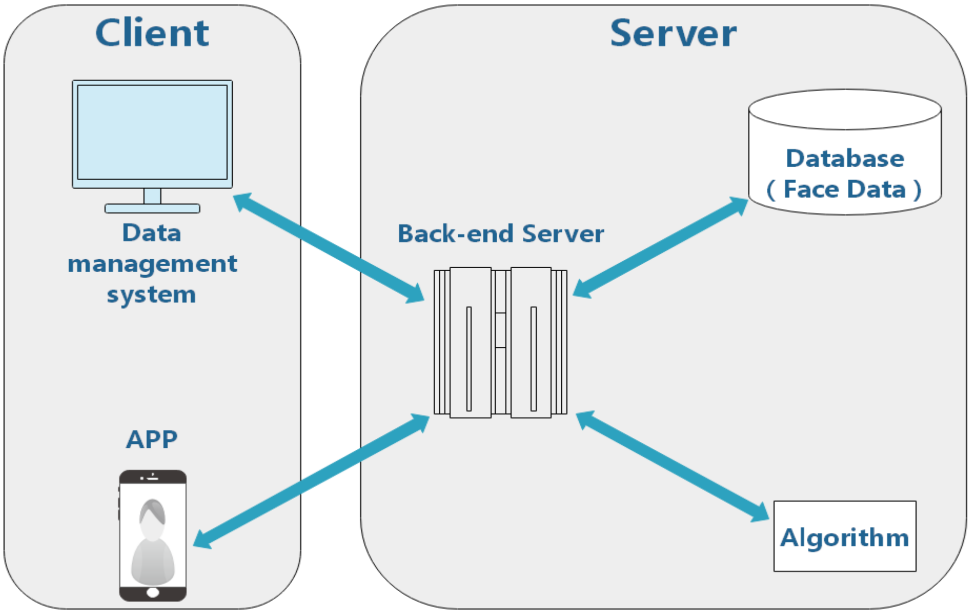
2. Business architecture

The business architecture of software design is as follows:


User
Enter the registration interface
Upload face image
Detect image quality
Face registration
Registration success
Registered face, create a user face database
User
Mobile devices
Collect faces
Face contrast
Attendance results
Output attendance results to the management system
Use your phone or pad camera to get face information
Collect face images,live detection
Judge users' identity information
Record attendance data based on the search results

3. Module design

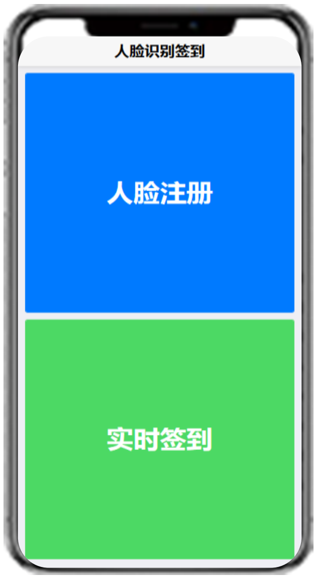
3.1 System structure design



3.2 Functional design description

3.2.1 User APP module

3.2.1.1 Design diagram



3.2.1.2 Functional description

The main function of this module is to provide users with a friendly check-in and information registration guide. For the check-in function, the camera module of the terminal where the APP is located needs to be called to capture the image of the user's face and transmit it to the server; for the registration function, the user is provided with a form for filling in the basic information and collecting the image of the user's face. Transfer to server-side storage.

3.2.1.3 Input data

(1) User check in

Get an image containing the user's face, the image format must be set to one of (jpg, jpeg, pgm or png). And the current time of the APP terminal Checkin_time (String type).

(2) User registration

In addition to getting an image containing the user's face, you also need the basic information of the following users:

Variable name	Type	Description
Name	String	User's real name
Gender	String	User's gender
Group	String	User's organization or team

3.2.1.4 Output Data

(1) User check in

Field	Description
UID	The ID assigned automatically by the server to the current user (cannot be changed once the assignment is successful)
Name	User's name
Gender	User's gender
Group	The organization or team the user is in
Checkin_time	The time of the user checked in at the APP terminal

(2) User registration

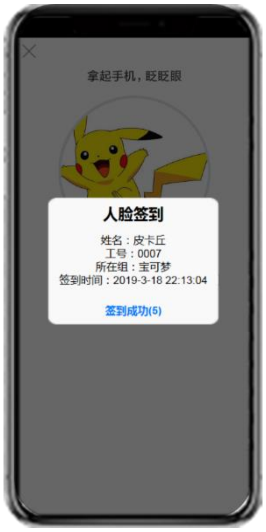
Field	Description
Code	Registration result: 0 means failure, 1 means success.

3.2.1.5 Detail design

In order to provide users with a friendly interface, it is necessary to optimize the display interface of some APPs.

(1) User check in

The software needs to provide a collection window for the user to guide the user to provide facial key information better and more convenient. At the same time, the check-in result will be returned to the user in the form of a pop-up window, as shown in the figure:

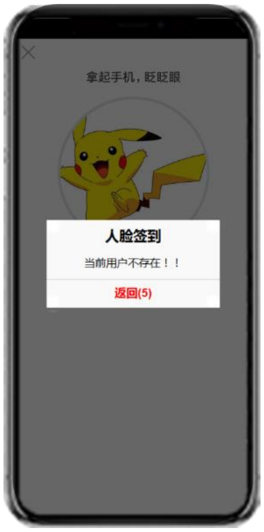


拿起手机，眨眨眼

人脸签到

姓名：皮卡丘
工号：0007
所在组：宝可梦
签到时间：2019-3-18 22:13:04

签到成功(5)



拿起手机，眨眨眼

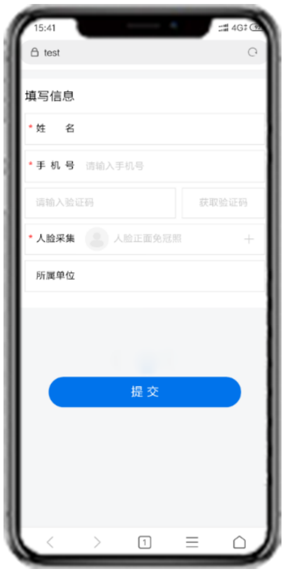
人脸签到

当前用户不存在！！

返回(5)

(2) User registration

In addition to providing the user with a face collection window, the software will provide a form for the user after the face data is successfully collected, to help the user fill in the registration information, and after the user clicks the submit button, according to whether the relevant information is successfully stored in the server database Whether the user successfully registered, and then pop-up window gives user feedback, as shown:



15:41 4G

test

填写信息

* 姓　名

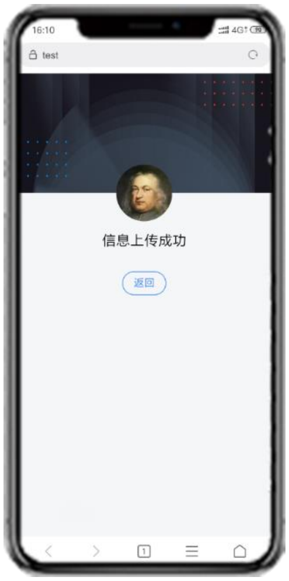
* 手 机 号　请输入手机号

请输入验证码　获取验证码

* 人脸采集　人脸正面免冠照 +

所属单位

提 交



16:10 4G

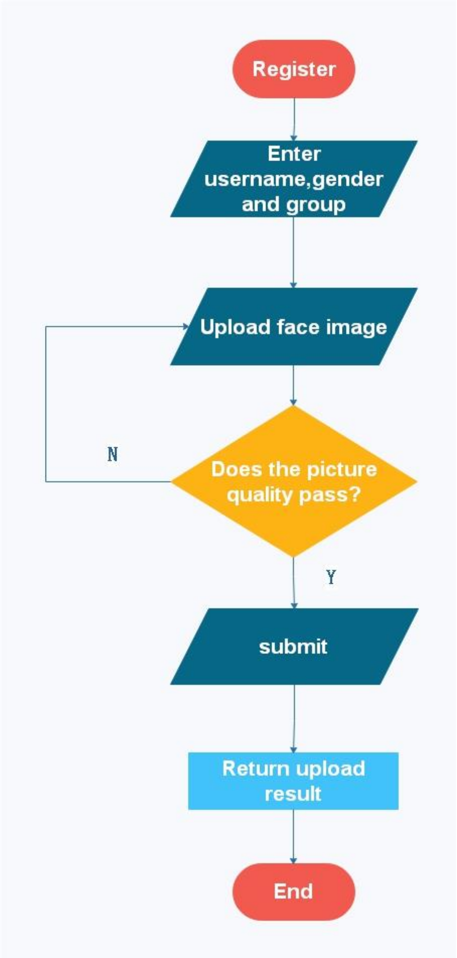
test

信息上传成功

返回

3.2.1.6 Flow chart

(1) User registration


Register
Enter username,gender and group
Upload face image
Does the picture quality pass?
N
Y
submit
Return upload result
End

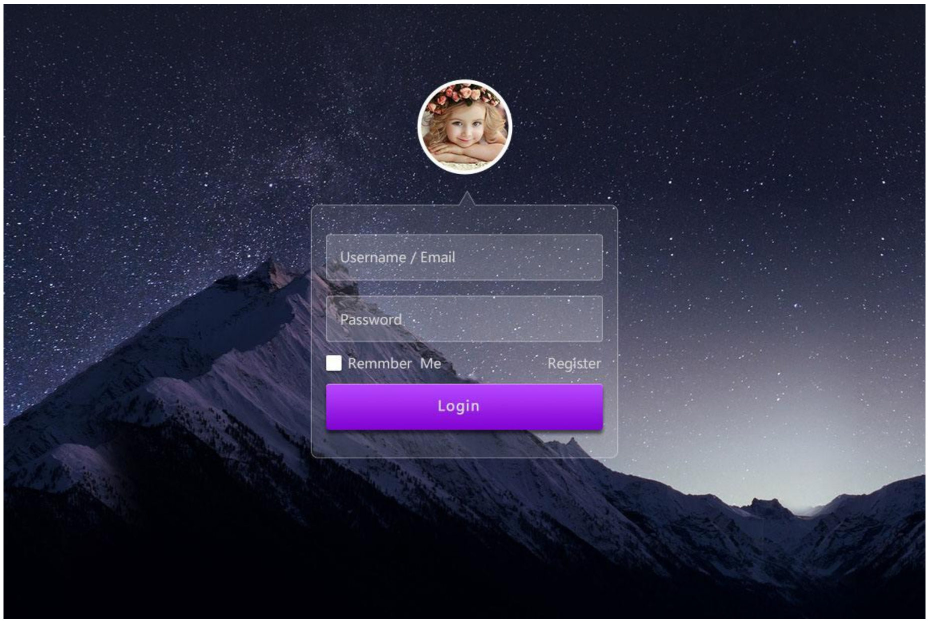
3.2.1.7 Limitation factor

All functions in the APP need to be used under normal network conditions, and the time required for information verification is related to the network speed.

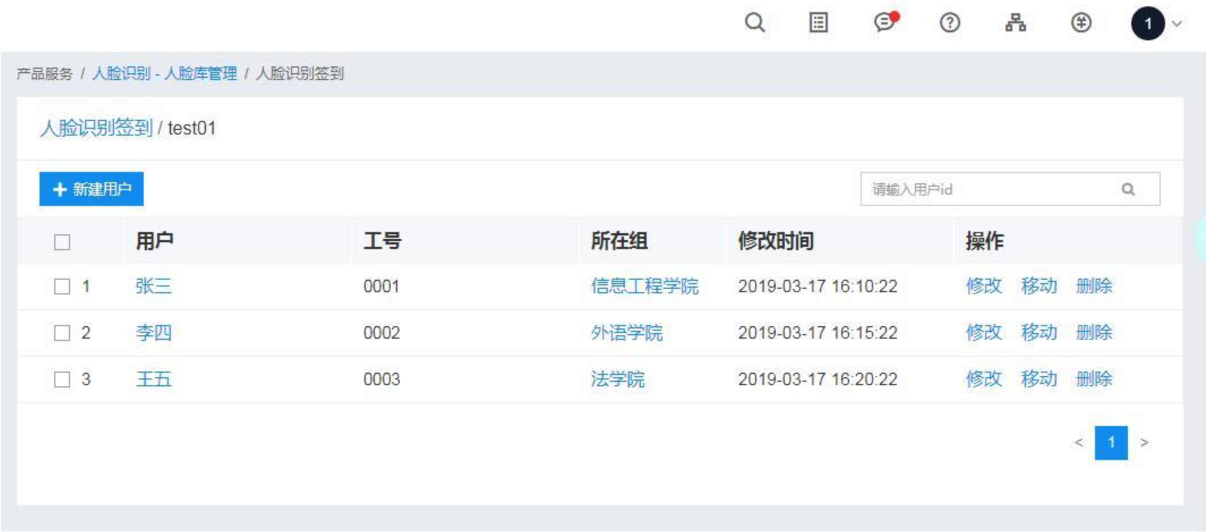
3.2.2 Management system module

3.2.2.1 Design diagram

(1) Login module



(2) Management module



□	用户	工号	所在组	修改时间	操作
□ 1	张三	0001	信息工程学院	2019-03-17 16:10:22	修改 移动 删除
□ 2	李四	0002	外语学院	2019-03-17 16:15:22	修改 移动 删除
□ 3	王五	0003	法学院	2019-03-17 16:20:22	修改 移动 删除

3.2.2.2 Functional description

The system management module provides administrators with a visual and friendly interface for administrator management. This module can be divided into two sub-modules, namely the login module and the management module.

3.2.2.3 Input data

(1) Login module

Variable name	Type	Description
Admin_name	String	Administrator's username (Required)
Password	String	Password (Required)

(2) Management module

Variable name	Type	Description
-	-	-

3.2.2.4 Output Data

(1) Login module

After the registration module obtains the padding information, if the verification succeeds, it directly jumps to the management module interface, and if it fails, it gives the corresponding popup prompt, so the module does not have output data.

(2) Management module

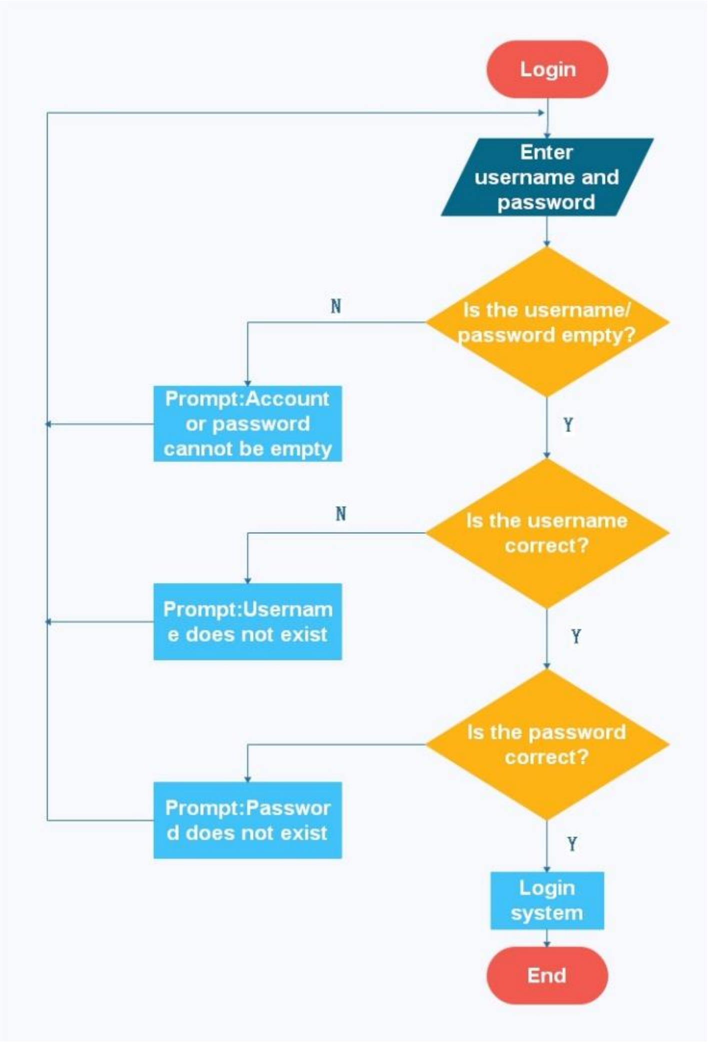
Variable name	Type	Description
UID	String	-
Name	String	User's real name
Gender	int	User's gender
Group	String	User's organization or team
Checkin_time	Date	-

3.2.2.5 Detail design

The management module needs to implement basic CRUD operations and communicate with the server-side database.

3.2.2.6 Flow chart

Login module flow chart：


Login
Enter username and password
Is the username/ password empty?
N
Prompt:Account or password cannot be empty
Y
Is the username correct?
N
Prompt:Usernam e does not exist
Y
Is the password correct?
Prompt:Passwor d does not exist
Y
Login system
End

3.2.3 Face recognition algorithm module

3.2.3.1 Functional description

The main function of this module is to provide a complete set of face recognition comparison functions. This module can be divided into two sub-modules, namely the face feature extraction module and the face comparison module.

3.2.3.2 Algorithm detail design

The principle of LBP algorithm used in this module comes from the papers of T. Ojala,

M. Pietikäinen, D. Harwood[1] and DCP algorithm comes from the papers of Ding C , Choi J , Tao D , et al.[2](only choose one of the two). The algorithm has a good precision and recall rate after comparison on multiple databases. Certain modifications are applied to the software project. The algorithm extracts and generates 256-dimensional feature vectors(or 512) from the pre-processed face images, and the module stores the feature vectors and the automatically assigned tags in the local xml file(or in the database). When the face features are compared, the module program reads the feature vectors stored in the xml file(or the database) and sequentially calculates the distance between the vectors (the distance calculation algorithm uses one of the Euclidean distance, the chi-square distance or the L1 distance), and the distance is the smallest. If the threshold is less than the preset threshold, it is the face owner, and the output returns the corresponding label.

3.2.3.3 Input data

(1) Face feature extraction

Variable name	**Type**	**Description**
Face_image	cv::Mat	User's face image

(2) Face contrast

Variable name	**Type**	**Description**
Face_image	cv::Mat	Pending detection face image

3.2.3.4 Output Data

(1) Face feature extraction

Variable name	**Type**	**Description**
Feature	Mat	The facial features of the user extracted by the algorithm

(2) Face contrast

Variable name	Type	Description
Tag	int	The identified face owner's label, where -1 means the detected face does not exist in the database

3.2.3.5 Interface

Developers can create a new faceDetection by referring to the following code:

```
#include <iostream>

FaceDetection faceDetection();
```

Face recognition: Identify the face owner in the picture and return its identity tag:

```
#include <opencv2/opencv.hpp>

//Set the default recommended value for Rin/Rex/Grid_x/Grid_y
cv::Mat image_test = "test_train.jpg";
cv::Mat image_predict = "test_predict.jpg";
std::int Rin = 1;
std::int Rex = 4;
std::int Grid_x = 1;
std::int Grid_y = 1;
std::float Threshold = 0.85;

//Directly call face recognition
faceDetection.train(image_test);
std::int tag = faceDetection.predict(image_predict);

//If there are optional parameters
std::map<std::int, std::int, std::int, std::int, std::float> options;
options["Rin"] = Rin;
options["Rex"] = Rex;
options["grid_x"] = Grid_x;
options["grid_y"] = Grid_y;
options["Threshold"] = Threshold;

//Call face recognition with parameters
faceDetection.train(image_test, options);
std::int tag = faceDetection.predict(image_predict, options);
```

(1) Face feature extraction: used to extract feature information of a face in a specified image.

Face feature extraction——Request parameter details

Parameter name	**Whether it is necessary**	**Type**	**Defaults**	**Description**
Face_image	Yes	cv::Mat	Null	Image information containing the face (Data size should be less than 10M), the image format should be one of jpg, jpeg, pgm or png(Recommended jpg format)
Rin	No	std::int	1	The inner radius of the characteristic sampling point range (No special case is not recommended)
Rex	No	std::int	4	The outer radius of the characteristic sampling point range (No special case is not recommended)
Grid_x	No	std::int	1	The number of horizontal segmented regions of the feature image (Modifying to a larger value can improve recognition accuracy, but speed will decrease)
Grid_y	No	std::int	1	The number of vertical segmentation areas of the feature image (Modifying to a larger value can improve recognition accuracy, but speed will decrease)

Face feature extraction——Return parameter details

Field	**Whether it is necessary**	**Type**	**Description**
Feature	Yes	Mat	Extracted face feature vector information

(2) Face contrast: used to determine the identity of the face owner.

Face contrast——Request parameter details

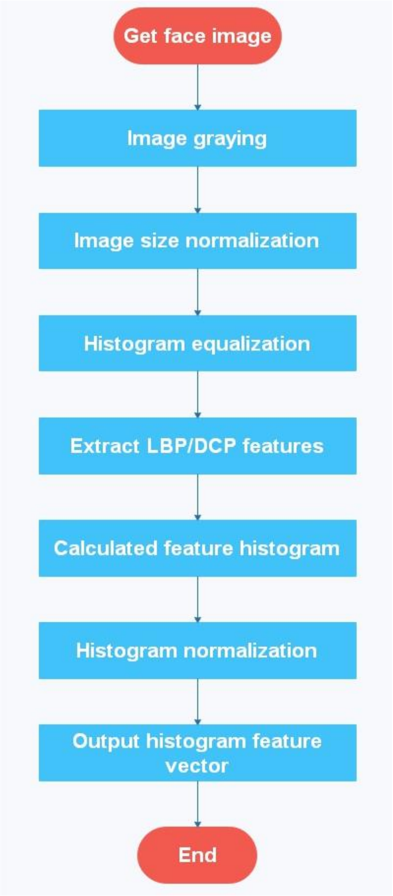
Parameter name	**Whether it is necessary**	**Type**	**Defaults**	**Description**
Face_image	Yes	cv::Mat	Null	Pending detection image(Data size should be less than 10M), the image format should be one of jpg, jpeg, pgm or png(Recommended jpg format)
Threshold	No	std::float	0.85	Threshold, either by yourself or by default

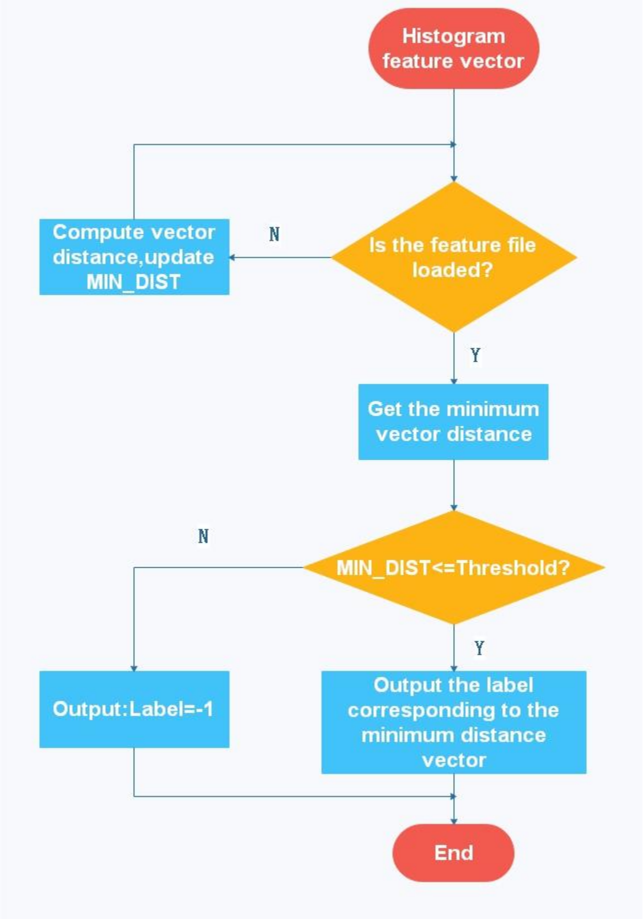
Face feature extraction——Return parameter details

Field	**Whether it is necessary**	**Type**	**Description**
Tag	Yes	int	The identified face owner's tag, where -1 means the detected face does not exist in the database

3.2.3.6 Algorithm flow

(1) Face feature extraction && Face contrast




Histogram feature vector
Is the feature file loaded?
N
Compute vector distance,update MIN_DIST
Y
Get the minimum vector distance
MIN_DIST<=Threshold?
N
Y
Output:Label=-1
Output the label corresponding to the minimum distance vector
End

3.2.3.7 Compiler Environment

This part of the algorithm is compiled and implemented in Microsoft Visual Studio 2010 and above, and requires an extension module of OpenCV3.0 and above.

4. support conditions

4.1 Hardware Support

(1) A terminal capable of running a face sign-in program requires a memory of 512 MB or more, a storage space of 4 GB or more, and a front camera that can operate normally.

(2) One server, Intel® XEON® Cascade Lake baseband/core frequency: 2.6GHz/3.5GHz, 2 core 4G, bandwidth 6Gbps, 100G hard disk storage.

(3) A terminal capable of running an information management interface needs to have more than 2G of memory, 128G or more of storage, 100M of broadband, and a LAN using TCP/IP protocol.

4.2 Software Support

The required software support varies according to the different requirements of the module.

(1) OpenCV needs to control the version above 3.0.

(2) Support CX11 when configuring.

(3) Node.js software service,

(4) Apache software services,

(5) MySql database software service,

(6) Operating system of Windows 7 and above.

(7). The terminal operating system needs to select Android system, the version is between Android 5.0~8.1.

(8) Development tools: WebStorm 11, HBuilder, Xshell5, FlashFXP, Navicat Premium, Visual Studio 2010 and above version.

5. Schedule

5.1 Overall schedule

Stage	Time required
Design	1-3 weeks
Learn	4-5 weeks
Develop	6-8 weeks
Test&Debug	9-10 weeks
Release	11 weeks

5.2 Learning schedule

Learning topic	Principal	Study-time	Deliverables
Algorithm	WanHongda	2 weeks	Extract feature vectors, Recognize faces.
Front-end	ZhangJiaqing	2 weeks	Review knowledge that solves business requirements
Node.js	ZhangWei	1.5 weeks	Build servers, Business request and response
APP	ZhangWei	0.5 weeks	Android App development and packaging

5.3 Development schedule

Module	Submodule	Principal	Development time	Testing time	Test leader
Login module	Login interface	ZhangJiaqing	3 days	1 days after completion	WanHongda
	Verification request	ZhangWei		1 days after completion	ZhangJiaqing
Management interface module	Management interface	ZhangJiaqing	3 days	1 days after completion	WanHongda
	CRUD interaction	ZhangWei		1 days after completion	ZhangJiaqing
App face registration	Registration interface	WanHongda	5 days	2 days after completion	ZhangJiaqing
	Picture transfer function	ZhangWei		1 days after completion	WanHongda
	Information reception save and response	ZhangWei		1 days after completion	ZhangJiaqing
App check-in real time	Check-in interface	WanHongda	7 days	2 days after completion	ZhangWei

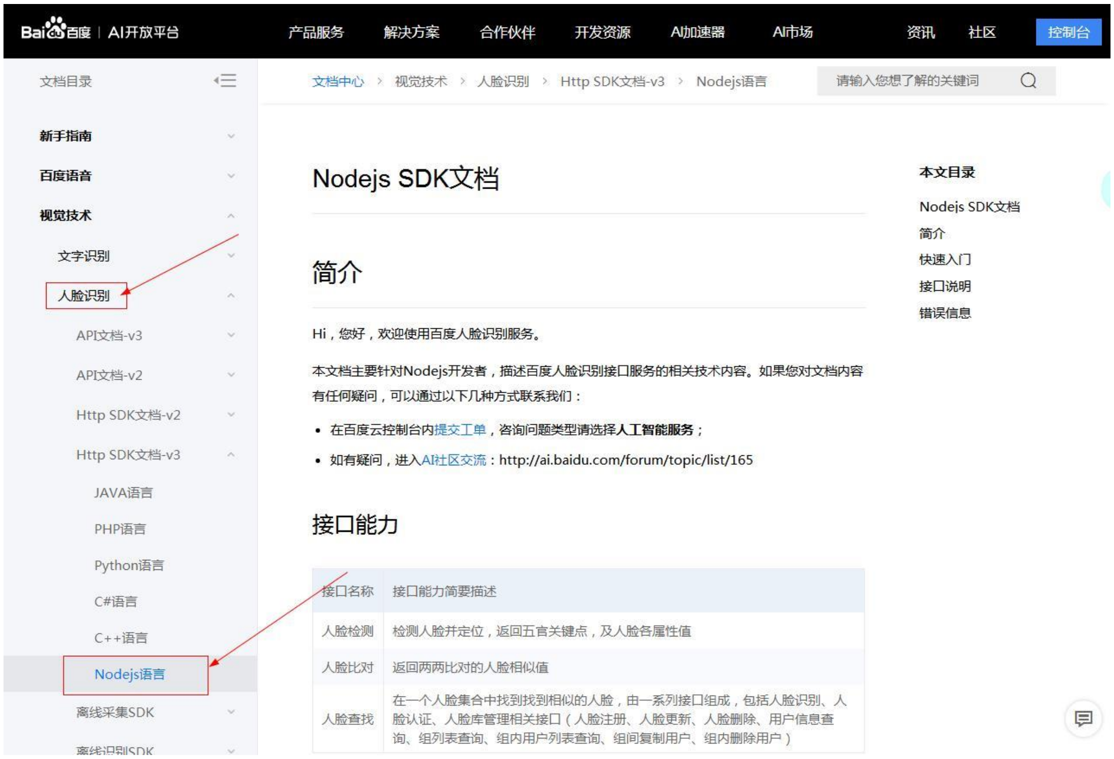
	Face capture and upload	ZhangWei			WanHongda
	Face contrast return results	ZhangWei			WanHongda
Algorithm module	Feature extraction module	WanHongda	7 days	4 days after completion	ZhangWei
	Face contrast module	WanHongda	10 days	6 days after completion	ZhangWei

6. Risk assessment and solution

Since the algorithm program is written in C++ language, there may be certain risks in the background encapsulation such as functional failure and data loss, so it is necessary to prepare a risk solution to deal with the possible risks. To this end, we have prepared two solutions PLAN A and PLAN B.

6.1 PLAN A

Use the SDK provided by Baidu Company to replace the algorithm part. For details on Baidu SDK, please refer to the documentation: http://ai.baidu.com/docs#/Face-Node-SDK/top.



Nodejs SDK文档

简介

Hi，您好，欢迎使用百度人脸识别服务。

本文档主要针对Nodejs开发者，描述百度人脸识别接口服务的相关技术内容。如果您对文档内容有任何疑问，可以通过以下几种方式联系我们：

- 在百度云控制台内提交工单，咨询问题类型请选择**人工智能服务**；
- 如有疑问，进入AI社区交流：http://ai.baidu.com/forum/topic/list/165

接口能力

接口名称	接口能力简要描述
人脸检测	检测人脸并定位，返回五官关键点，及人脸各属性值
人脸比对	返回两两比对的人脸相似值
人脸查找	在一个人脸集合中找到找到相似的人脸，由一系列接口组成，包括人脸识别、人脸认证、人脸库管理相关接口（人脸注册、人脸更新、人脸删除、用户信息查询、组列表查询、组内用户列表查询、组间复制用户、组内删除用户）

6.2 PLAN B

It turns out that it is realistic and feasible to write LBP or DCP algorithms using JAVA language. The algorithm program written in JAVA language can be slightly modified or even directly applied to the development of the client, so that the interface parameters of the LBP/DCP algorithm of the JAVA language can be appropriately simplified.

You can create a new FaceDetection with reference to the following code. After the initialization is complete, modify the corresponding parameters as needed. The program will return the identity tag of the face owner:

```java
public class Sample{
    //Set the output tag of the log
    //Set the default recommended value for Rin/Rex/Grid_x/Grid_y
    private static final String TAG = "Sample";
    private static int Rin = 1;
    private static int Rex = 4;
    private static int Grid_x = 1;
    private static int Grid_y = 1;
    private static float Threshold = 0.85;

    public static void main(String[] args) {
        //Optional: Reset parameter values according to the developer's needs
        //If not set, the default configuration is used.
        this.Rin = 1;
        this.Rex = 4;
        this.Grid_x = 1;
        this.Grid_y = 1;
        this.Threshold = 0.85;

        //Initialize a FaceDetection
        FaceDetection faceDetection = new FaceDetection();

        //Call interface
        String trainPath = Environment.getExternalStoragePublicDirectory()+"/test_train.jpg";
        String predictPath =Environment.getExternalStoragePublicDirectory()+"/test_predict.jpg";
        faceDetection.train(trainPath, Rin, Rex, Grid_x, Grid_y);
        int label = faceDetection.predict(predictPath, Rin, Rex, Grid_x, Grid_y, Threshold);
        Log.d(TAG，label.toString());
    }
}
```

Face feature extraction——Request parameter details

Parameter name	**Whether it is necessary**	**Type**	**Defaults**	**Description**
Face_image_path	Yes	String	Null	Target image storage path (Data size should be less than 10M), the image format should be one of jpg, jpeg, pgm or png(Recommended jpg format)
Rin	No	int	1	The inner radius of the characteristic sampling point range (This parameter is only used in the DCP algorithm and no special case is not recommended)
Rex	No	int	4	The outer radius of the characteristic sampling point range (This parameter is only used in the DCP algorithm and no special case is not recommended)
Grid_x	No	int	1	The number of horizontal segmented regions of the feature image (Modifying to a larger value can improve recognition accuracy, but speed will decrease)
Grid_y	No	int	1	The number of vertical segmentation areas of the feature image (Modifying to a larger value can improve recognition accuracy, but speed will decrease)

Face feature extraction——Return parameter details

Field	Whether it is necessary	Type	Description
Feature	Yes	String	Character vector in string format(The original vector dimension is 1xN-dimensional and all data are separated by spaces)

(2) Face contrast: used to determine the identity of the face owner.

Face contras——Request parameter details

Parameter name	Whether it is necessary	Type	Defaults	Description
Face_image _path	Yes	String	Null	Pending detection image storage path (Data size should be less than 10M), the image format should be one of jpg, jpeg, pgm or png(Recommended jpg format)
Threshold	No	float	0.85	Threshold, either by yourself or by default

Face feature extraction——Return parameter details

Field	Whether it is necessary	Type	Description
Tag	Yes	int	The identified face owner's tag, where -1 means the detected face does not exist in the database

It should be noted that to write algorithm programs in JAVA language, it is necessary to configure OpenCV for Android. The CX11 is not required to be configured, but if the customer wants better compatibility and the resulting installer volume is negligible, then it is recommended to configure.