**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

**Content**

[1 Introduction 3](#_Toc3931677)

[1.1 Document purpose 3](#_Toc3931678)

[1.2 Background 3](#_Toc3931679)

[1.3 References 3](#_Toc3931680)

[2. Business architecture 4](#_Toc3931681)

[3. Module design 5](#_Toc3931682)

[3.1 System structure design 5](#_Toc3931683)

[3.2 Functional design description 5](#_Toc3931684)

[3.2.1 User APP module 5](#_Toc3931685)

[3.2.2 Management system module 8](#_Toc3931686)

[3.2.3 Face recognition algorithm module 10](#_Toc3931687)

[4 support conditions 16](#_Toc3931688)

[4.1 Hardware Support 16](#_Toc3931689)

[4.2 Software Support 16](#_Toc3931690)

[5. Schedule 17](#_Toc3931691)

[5.1 Overall schedule 17](#_Toc3931692)

[5.2 Learning schedule 17](#_Toc3931693)

[5.3 Development schedule 17](#_Toc3931694)

[6. Risk assessment and solution 19](#_Toc3931695)

# 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：

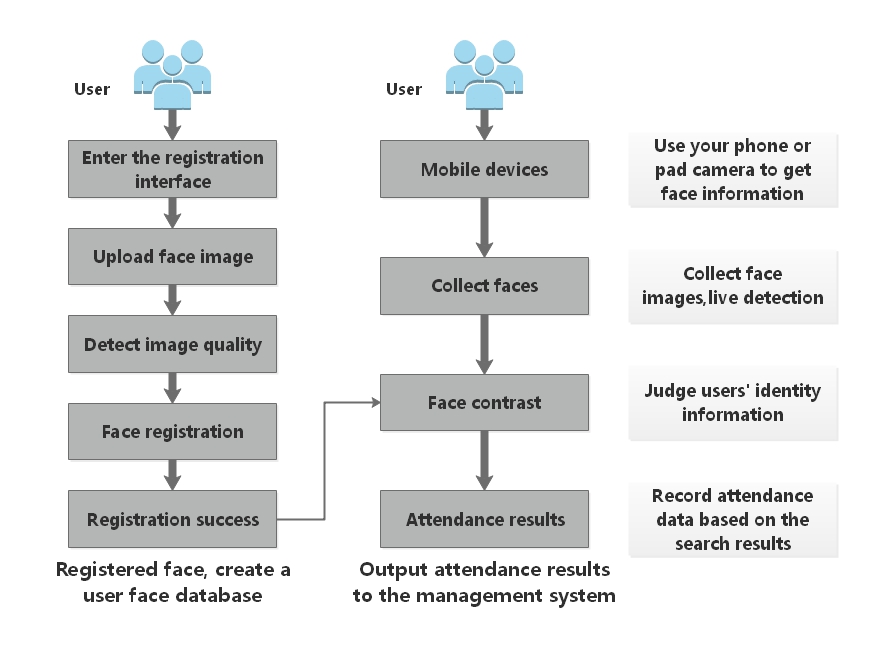
|  |  |  |
| --- | --- | --- |
| Name | Position | Duties |
| WanHongda | Member | Face recognition algorithm design, API writing and APP Implementation |
| ZhangJiaqing | Member | Information Management Interface and APP Design |
| ZhangWei | Leader | Server design and writing, Information Management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

## 1.3 References

[1].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:



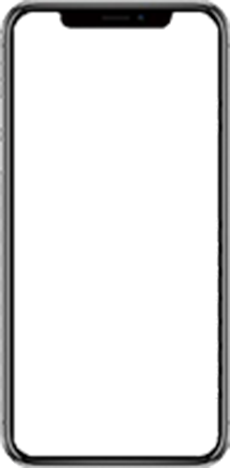
# 3. Module design

## C:\Users\Administrator\Desktop\软件项目管理ppt\结构.png3.1 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.

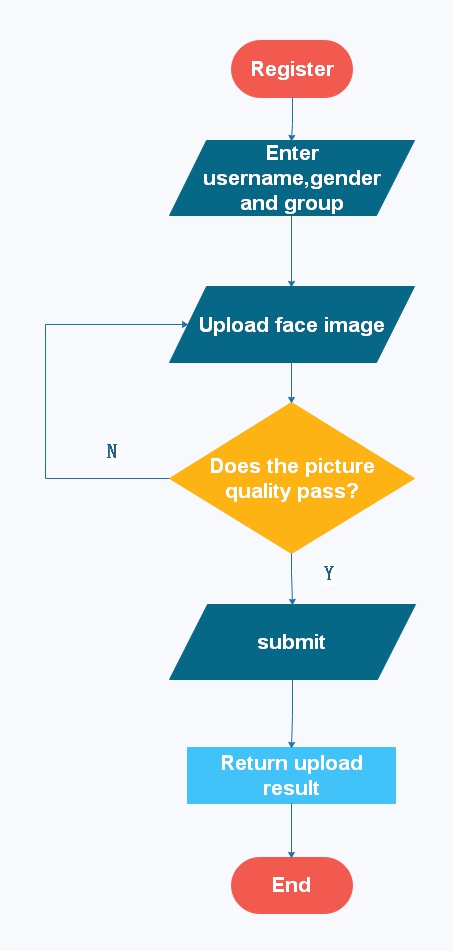
(1) User check in

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:

(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:

3.2.1.6 Flow chart

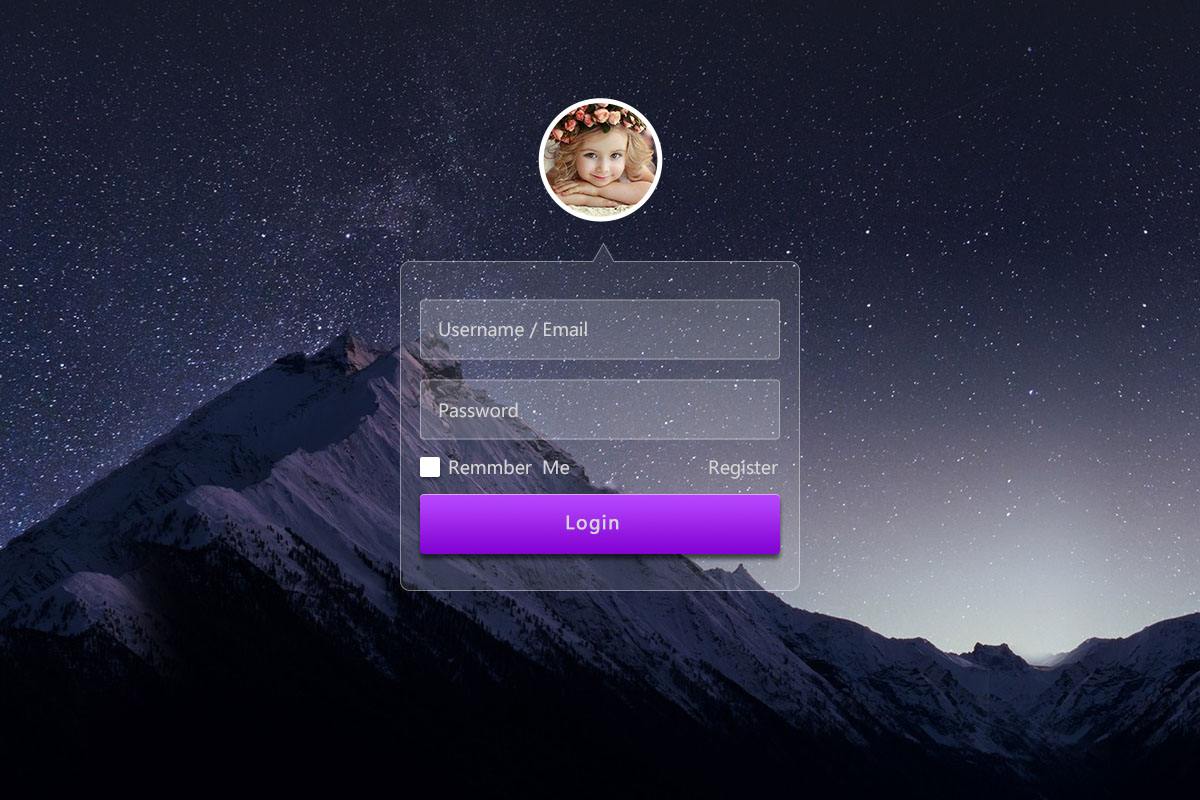
(1) User registration

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

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.

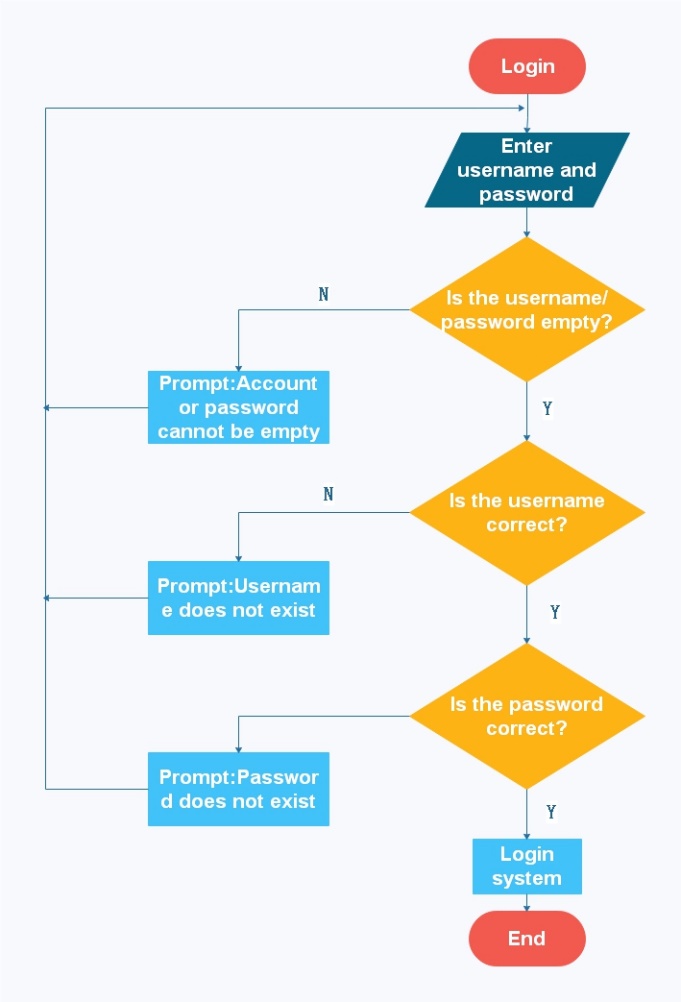
(2) Management module

|  |  |  |
| --- | --- | --- |
| 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：

### 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 DCP algorithm used in this module comes from the papers of Ding C , Choi J , Tao D , et al.[1]. 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 512-dimensional feature vectors from the pre-processed face images, and the module stores the feature vectors and the automatically assigned tags in the local xml file. When the face features are compared, the module program reads the feature vectors stored in the xml file 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 | Mat | User's face image |

(2) Face contrast

|  |  |  |
| --- | --- | --- |
| Variable name | Type | Description |
| Feature | Mat | The facial features of the user to be identified extracted by the algorithm |

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 |
| Label | int | The identified face owner's label, where -1 means the detected face does not exist in the database |

3.2.3.5 Interface

**(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** | **No** | **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 |
| **Rin** | **No** | **int** | **1** | **The inner radius of the characteristic sampling point range (No special case is not recommended)** |
| **Rex** | **No** | **int** | **4** | **The outer radius of the characteristic sampling point range (No special case is not recommended)** |
| **Grid\_x** | **No** | **int** | **8** | **The number of horizontally segmented regions of the feature image (Modifying to a smaller value can result in higher recognition speed, but the accuracy will decrease)** |
| **Grid\_y** | **No** | **int** | **8** | **The number of vertical segmentation areas of the feature image (Modifying to a smaller value can lead to higher recognition speed, but the accuracy 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.**

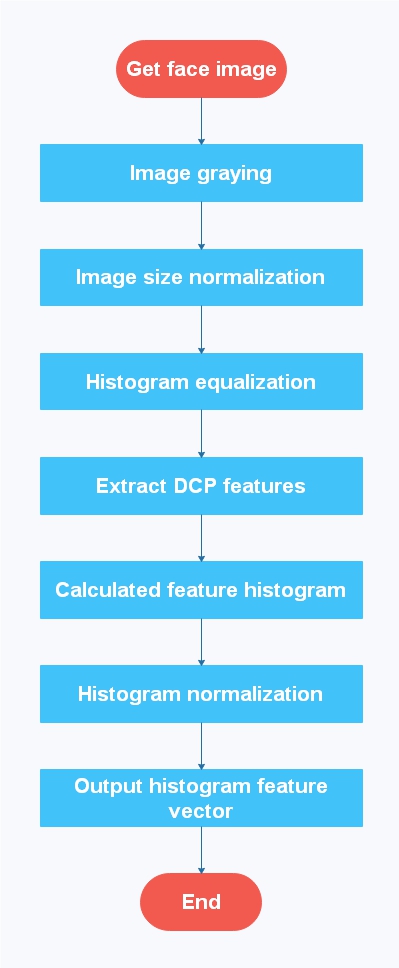
**Face contras——Request parameter details**

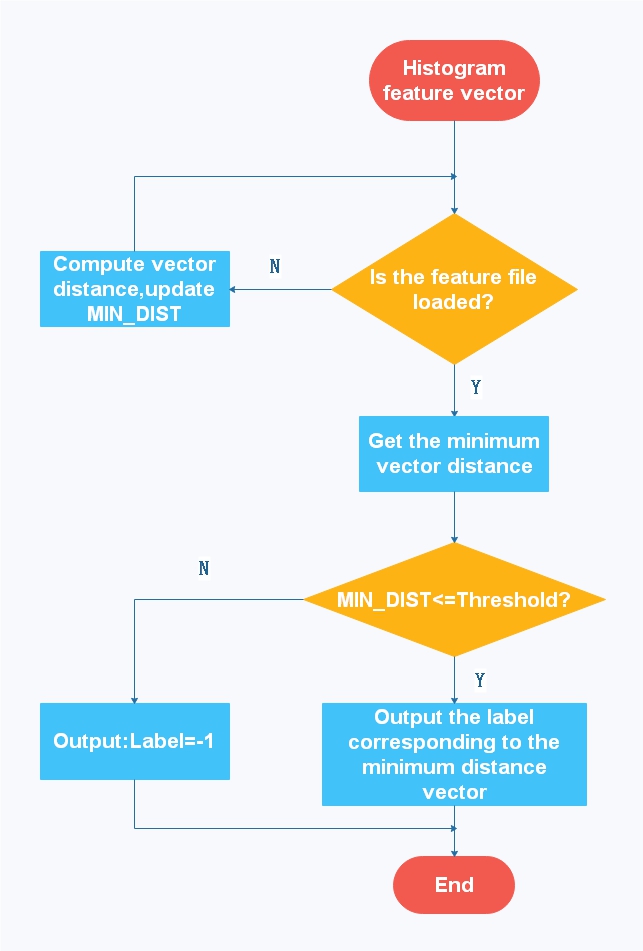
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter name** | **Whether it is necessary** | **Type** | **Defaults** | **Description** |
| **Feature** | **Yes** | **Mat** | **Null** | Extracted face feature vector information (Must be 1xN-dimensional Mat() type data) |
| **Threshold** | **No** | **float** | **0.85** | **Threshold, either by yourself or by default** |

**Face feature extraction——Return parameter details**

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

3.2.3.6 Algorithm flow

(1) Face feature extraction

(2) Face contrast

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.

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>.

