

# A Study on Student Attendance Analysis and Statistics Platform Based on Capture Recognition Technology

Qianchuan Ding<sup>1, #</sup>, Qun Yin<sup>2, #\*</sup>, Pengfei Duan<sup>2\*</sup>, Xianyun Zhang<sup>2</sup>, Meisu Yin<sup>3</sup>

<sup>1</sup> Family means verification institute center of Yunnan province, Kunming, Yunnan, China

<sup>2</sup> Kunming University of Science and Technology, Kunming, Yunnan, China

<sup>3</sup> Hunan University, Changsha, Hunan, China

#Co-first Authors.

\*Corresponding Author.

## Abstract:

With the development of face recognition technology and HD camera, it is possible to use face recognition to realize the classroom attendance statistics. The traditional way of classroom attendance statics needs teachers to roll call according to the list of students, but face recognition can not only save the time of class, but also lighten the burden of statistics attendance of school. This paper realizes the face recognition system of attendance analysis and statistics platform, and it needs cameras and a main computer. In software, under the development environment of VS2017, it relies on OPENCV, arcsoft face recognition SDK and My -SQL database realizes a real-time video stream face recognition system with C++ programming. The system captures the student face by the camera installed in the classroom, then using SDK extracts the face feature and compares with the feature in the database. When the compare value is over the set value, the system will output the corresponding students' information and completion time to the TXT file.

**Keywords:** Face recognition, VS2017, Arcsoft, C++.

## I. INTRODUCTION

With the coming of digital intelligent information age, schools are attaching more and more importance to information tization. The school examination situation system has been gradually improved, but the classroom teaching attendance always can only rely on the teacher's paper list to select points, but also easy to answer the situation [1, 2]. All kinds of disadvantages are also obvious, the problem of replacement is also repeatedly banned. This not only took up the class time, but also caused trouble with the statistics of the academic affairs office. At present, the attendance application is more popular for all kinds of attendance machines, for classroom attendance is too inconvenient and restrictions, but also easy to cause the situation of brush off. In recent years, face recognition has been widely used in various industries, such as the face payment, screening of suspects by the police, face detection at airports and high-speed railway stations, etc. Compared with other biometrics, such as fingerprint recognition, face recognition is difficult

to counterfeit, high stability, easy to promote, fast, is the best choice for accurate and efficient attendance [3, 4].

In order to reduce the burden of statistical attendance, a set of automatic attendance system suitable for the actual needs of schools is needed to replace manual attendance. The goal of this paper is to use the camera installed in the classroom to realize classroom attendance based on face recognition technology, and honsoft's offline face recognition SDK will be used to achieve the purpose of face recognition. The main functions of this module can be divided into three sub-modules: image acquisition and processing sub-module, face recognition sub-module and data acquisition sub-module. Son image acquisition and the processing module are set to obtain real-time video streaming through the classroom to install cameras and convert the SDK can handle frame, in face recognition module using the SDK to face detection and feature values of students in class are extracted and complete the human face, data acquisition modules are connected to the database access to relevant information of students. The basic functions of this module are: store the camera images to the designated directory, real-time video streaming face comparison, the comparison judgment for the same person, the corresponding student information and completion of the comparison time output to the TXT file under the designated directory. This not only improves school teaching efficiency, but also automates statistical attendance [5].

## II. MATERIALS AND METHODS

### 2.1. Overall Design Scheme

The student attendance analysis and statistics platform are divided into three main parts: student information self-collection terminal, portrait snapshot identification module and attendance data query WeChat small program. This paper completes the portrait snapshot identification module. This module obtains the real-time video stream through the camera installed in the classroom, and then USES honsoft's offline SDK2.0 as the face recognition engine, and connects to the student information database to obtain the stored student information to realize face recognition [5, 6]. The basic functions to be satisfied in this module are: store the video shot by the camera to the designated directory, real-time video streaming face comparison, if the comparison is successful, the comparison value exceeds the threshold corresponding student information and complete the comparison time output to the TXT file under the designated directory.

Like snap recognition module is the second part of the attendance statistics analysis platform, and the two halves have docking, specifically for the use of students of self-help information acquisition terminal information data in the database for face recognition and the recognition results output to a TXT to attendance data query WeChat small application analysis of the data [6].

### 2.2. Face Recognition

The core of portrait snapshot recognition module is facing recognition. In this paper, honsoft's offline face recognition SDK 2.0 version is used to realize face recognition. Hongrun USES neural network to

realize face recognition. There are many variations of neural networks, and in recent years, researches on these variations have been exploding, such as convolutional neural networks that are good at image recognition and long-term and short-term memory networks that are good at speech recognition. Due to the integration of correlation knowledge between adjacent pixels in the convolutional neural network, the invariance of image translation, rotation and local deformation can be obtained to a certain extent. Therefore, very ideal recognition results can be obtained [7, 8].

A total of 6 steps are required to invoke the iris SDK to realize face recognition, which is: activation of SDK, the initialization engine, face detection, face feature extraction and face comparison, and finally destruction of the engine.

To activate the SDK, you only need to call the `ASFActivation()` function. The `appd` and `SDKKey` required by the interface can be obtained when applying for the SDK on the official website, and the network activation can be successful when the SDK is used for the first time. The `ASFInitEngine()` function is called to initialize the engine [9]. Then call the `ASFDetectFaces()` function for face detection, call the `ASFFaceFeatureExtract()` function to extract the face features, and call the `ASFFaceFeatureCompare()` function for face alignment. Finally, call the `unInitEngine()` function to destroy the engine.

#### 2.2.1. Activation of SDK

To activate the SDK, simply passes the `AppId` and `SDKKey` obtained from the official website into the function `ASFActivation()`. If this function succeeds, it returns `MOK` or `MERR_ASF_BASE_ALREADY_ACTIVATED`, otherwise failure codes are returned. Here is the key code to activate the SDK. Here is the key code to activate the SDK.

```
'Activate SDK

Procedure My_Active(MPChar ID, MPCharKEY)

Begin

res ← ASFActivation(ID, KEY);

'Call the ASFActivation() function to activate 'SDK

IF (MOK != res and

MERR_ASF_BASE_ALREADY_ACTIVATED!=

Res) Then

Printf ("ALActivation fail: %d\n", res);

'Output the return result when activation of 'SDK fails

ELSE

Printf ("ALActivation success: %d\n", res);

'Output the returned result when the SDK is

'activated successfully

END IF

End
```

### 2.2.2. Initialization engine

After the SDK is successfully activated, the engine is initialized. The initialization engine uses the `ASFInitEngine()` function. If this function is successful, it will return MOK, otherwise it will return failure codes. The function has six parameters, which is the detection mode (VIDEO mode / IMAGE mode), angle priority value of face detection, minimum face size, maximum number of faces to be detected [1-50], user selection of function combination to be detected, and initialization of returned engine handle [10].

This research is to realize face recognition in real-time video stream, so choose VIDEO mode for the detection mode and fill in `ASF_DETECT_MODE_VIDEO` for parameters. Then it is the angle priority value of the detection face. This angle refers to the angle in the two-dimensional plane, not the angle to distinguish the front face and the side face in three-dimensional space. The value reflects the face direction. 0 degrees is positive, 90 degree is 90 degrees in plane, and 180 degree is 180 degrees. Therefore,

the angle priority value is ASF\_OP\_0\_ONLY can detect 0 degrees. Next is the minimum face size, which represents the proportion of the face size relative to the long edge of the image. The effective value range of VIDEO mode is [2, 16], and that of IMAGE mode is [2, 32]. In most cases, the recommended value is 16. This paper uses the recommended value of 16. Then it is the maximum number of faces to be detected. The range is [1-50], select 5 first. The fifth parameter is the combination of functions to be detected. Because this paper only uses face detection and face recognition of SDK, ASF\_FACE\_DETECT | ASF\_FACERECOGNITION is selected as the function. Finally, initialize the returned engine handle and define a variable of MHandle type. The key code for initializing the engine is shown below [10].

```
'Initialize engine
Procedure My_Init()
Begin
    res←ASFInitEngine(ASF_DETECT_MODE_VIDEO,
        ASF_OP_0_ONLY, 16, 5,
        mask, &handle);
'Call the ASFInitEngine() function to initialize
'the engine
'MHandle handle = NULL;
'MInt32 mask = ASF_FACE_DETECT | 'ASF_FACERECOGNITION;
IF res != MOK Then
    printf("ALInitEngine fail: %d\n", res);
'Output the return result when the 'initialization engine fails
ELSE
    printf("ALInitEngine sucess: %d\n", res);
'Output the return result when the 'initialization engine is successful
END IF
End
```

### 2.2.3. Face detection

Face detection can locate the position and Angle of a face in a photo. For face detection through iris SDK2.0, you need only call ASFDetectFaces().

‘Face detection

Dim detectedFaces1←{ 0 } As ASF\_MultiFaceInfo;

‘Define an ASF\_ The type of face detected by ‘multi face information

res ← ASFDetectFaces(handle,

offInput.i32Width, offInput.i32Height,

offInput.u32PixelFormat,

offInput.ppu8Plane[0], &detectedFaces1);

‘Call the ASFDetectFaces() function to realize ‘face detection in real-time video stream.

‘Input parameter engine, image width, image ‘height, color space format, image data and ‘detected face information into the function in ‘turn.

The handle is the engine handle when the SDK is initialized, followed by the width, height, color space format and image data of the current frame, and finally saves detected face information to the structural body variable detectedFaces1 of ASF\_MultiFaceInfo type provided by SDK. OffInput provides structural variables of type ASVLOFFSCREEN for the SDK [11].

Because the student database stores the storage path of the student portrait, face detection should be carried out on the portrait images in the designated directory. The same is true for image face detection and video streaming face detection.

### 2.2.4. Face Feature value extraction

Face extraction feature is a process of transforming a face image into a series of fixed length values. This string of numbers is called a face feature, and it has the ability to represent that face. It is the most time-consuming part of face recognition to extract the feature value of the face after locating the position of the face through face detection. To implement face feature value extraction, you need to use the function ASFFaceFeatureExtract():

‘Single face feature extraction

Dim feature1  $\leftarrow$  { 0 } As ASF\_FaceFeature;

‘Define an ASF\_ The face feature type structure ‘is used to store the saved face feature values

res  $\leftarrow$  ASFFaceFeatureExtract(handle,

offInput.i32Width, offInput.i32Height,

offInput.u32PixelFormat,

offInput.ppu8Plane[0],

&SingleDetectedFaces1, &feature1);

‘The fASFFaceFeatureExtract() function is called ‘to extract the face feature value.

‘The parameters in this function are engine, ‘image width, image height, color space format, ‘image data, single face position and angle ‘information, and face features.

Input parameters are the engine handle, width, height, color space format and image data of the current frame, and then input face information obtained in face detection to get face eigenvalue of ASF\_FaceFeature type.

Similarly, it is necessary to extract the feature values of the avatars stored in the student information database. Saving these feature values into memory can effectively increase the face recognition speed of the module.

#### 2.2.5. Face comparison

Because the SDK after initialization of the engine will advance the good use of memory, so the second face detection function and facial feature value extraction function called output result will overwrite the output for the first time as a result, the application layer defined by the pointer to the same block of memory, the data is the same, will lead to the comparison result is 1 [11]. Therefore, before comparison, deep copy of data is made on the first call of face detection function and face eigenvalue extraction function interface, and then comparison is made with the copied data [12].

To achieve the face comparison, you need to pass the copied video current frame face feature value and picture face feature value into the function ASFFaceFeatureCompare(), and then you will get a comparison result between 0 and 1.

‘Single face feature comparison

Dim confidenceLevel As MFloat ;

‘Define a MFloat type variable to save the ‘comparison results

res ← ASFFaceFeatureCompare(handle,

    &copyfeature1, &feature2,

    &confidenceLevel);

‘Call the ASFFaceFeatureCompare() function to compare faces.

‘Handle is the engine

‘Copyfeature1 face feature to be compared 1

‘Feature 2 to be compared face feature 2

‘Confidence level is the result of face comparison

Finally, a threshold value is set. If the threshold value is exceeded, the same person will be judged and the corresponding information will be output to TXT.

### III. IMPLEMENTATION OF SUBMODULES

In addition to face recognition using SDK, this module also needs to use OPENCV for image processing, connect to the database to obtain relevant student information, and obtain the time to complete the face comparison, which brings to image acquisition and processing sub-module and data acquisition sub-module respectively0.

#### 3.1. Sub-module of Image Acquisition and Processing

Image acquisition and processing sub-module completes the operation of camera and image processing in portrait capture recognition module. When using iris SDK to realize face recognition of real-time video stream, continuous frame image data should be passed to the interface. Because all the data of image frames need to be passed into the interface for processing, the current frame of the video stream needs to be extracted and converted into the format supported by SDK before SDK related interfaces are called [12, 13].

The sub-module has the following functions: collecting face recognition image, saving surveillance video, extracting the current frame of video stream and converting frame format. The following operations need to be completed: open and close the camera, save the camera images to the specified directory, and



convert the video stream into frame images that can be processed by iris SDK. In addition, in order to visualize the face detection process, the rectangular coordinates of the face in honsarong face recognition results are used to process the video stream and draw a box on the face part [14].

The first is to turn on the camera to get image data. In OpenCV, the video reading operation is completed through the VideoCapture class. Therefore, first define a videoInput of the VideoCapture class to open the camera and the frame of the Mat class to obtain images. Mat can be understood as a container for storing data. It defines an object to store the data of pictures. CAMERA\_NUM is the number of the camera. When connecting to the network, enter the corresponding IP address or number, and the camera will be opened. In order to facilitate debugging and save money, the camera of notebook computer is used in this stage. Parameter 0 represents the default camera ID. Generally, the built-in camera is the default camera of the system.

The next step is to save the video captured by the camera. To save the video or camera video to the local disk, you need to use the VideoWriter class in OpenCV. So first define a VideoWriter class videoOutput to save the video; then call the constructor to determine the storage path, the encoding format, the frame rate, frame size, and color of the file. Storage path to fill in the specified directory RecSavePath, notes that the directory in the program should use double slashes. The encoding and decoding mode FOURCC can be set to a variety of specified encoding methods, but the corresponding codec needs to exist in the host. If the parameter is set to -1, a dialog box will pop up when the code is training to select|ring to select|morning to select|ranking to select|meaning to select the encoding and decoding mode manually. FPSInPut is the frame rate, which is set to 24. The videoSize of the single frame image size to be saved can also be set freely. Here, you can use the videoInput.Get (CV\_CAP\_PROP\_FRAME\_Width) and videoInput.Get (CV\_CAP\_PROP\_FRAME\_Height) to get the size of the picture taken by the camera. After the definition of VideoWriter is completed, the camera input stream can be obtained through the frame of Mat class, and then the video can be stored as long as the videoInput >> frame; videoOutput << frame; is continuously cycled. In this process, it should be noted that OpenCV only supports avi format, and the generated video file can't be larger than 2GB, and can't add audio [14-16].

The next step is to extract the current frame and convert the format. Because the hongruan SDK processes the image data of consecutive frames and returns the detection results, the data of all the image frames need to be transferred to the interface for processing, including the width, height, color space format and image data of the image, and cannot be directly transferred by using the mat class. Here use the ASVLOFFSCREEN structure provided by the SDK to store image data. First of all, need to define an ASVLOFFSCREEN type offInput to save the information of the current frame of the video stream, initialize it, and set the parameters such as the width, height and format of the image. The width and height of the picture is consistent with the picture taken by the camera, and the color space format is ASVL\_PAF\_RGB24\_B8G8R8. With offInput, the image data of the current frame can be converted and stored in its variable. In this way, the current frame of the real-time video stream is converted into a format that can be processed by the SDK, and then the data can be directly used when calling the relevant interfaces of the SDK. In addition, due to the SDK's restrictions on image size, specifically, the width and

height is greater than 0, and the width is a multiple of 4. The image height of YUYV / I420 / N V21 / NV12 format is a multiple of 2, and the image height of BGR24 format is not limited. Therefore, if the size of the screen does not meet these conditions, the SDK cannot be called [16].

Finally, in order to visualize the face detection process, the video stream is processed, and the detected face parts are framed with green frames. The principle of the program is to use the information of face rectangle extracted by SDK and the corresponding function of OPENCV to process the image. After the format conversion of the image of the current frame, it is passed into ASFDetectFaces() for face detection, and the detected face information will be saved to the corresponding parameters. In one of the parameters, there is an MRECT type structure to store the face frame information, which is the coordinates of the four corners of the face rectangle. Then use this coordinate, and then use the rectangle drawing function of OpenCV to frame the face. If you want to draw a rectangle on a map of type Mat, select cv::rectangle(), or select cvRectangle() if you want to draw a rectangle on a graph of type IPLImage \*. The Mat type was set in the process of opening the camera, so select cv::rectangle() function. The parameters of the rectangle drawing function are image path, one vertex of the rectangle, another vertex on the diagonal of the rectangle, the line color (RGB), and the thickness of the lines that make up the rectangle (when negative value is taken (CV\_FILLED) filled rectangle), the type of line and the number of decimal places of coordinate points [18-20]. In order to facilitate the use, the function of the face frame is encapsulated into a function. When using, the coordinate information and frame of the face frame obtained from the SDK face detection can be transferred. The color of the box is set to (0, 255, 0), which will show green in the window [17].

### 3.2. Data Acquisition Sub-module

The sub-module of data acquisition completes the database related operation and current time acquisition in the portrait snapshot recognition module. In order to realize face recognition, the corresponding student information needs to be obtained from the database into memory, including the student's name, student number and avatar saving path. This sub-module realizes the following functions: database connection and query, as well as the current time acquisition. The operations that need to be done are: connect and query the database, read the student information into memory, and get the current time.

The first is to connect to the database, and C++ can connect to MySQL database in two ways. The first method is to use ADO connection, the second method is to use MySQL API function to connect; this design uses the second method. The initial connection to the database takes two steps. First call the mysql\_init() function initializes a MYSQL structure and then calls the mysql\_real\_connect() function connects to the MySQL server to complete the initialization of the database. The mysql\_real\_connect() function has eight parameters. The first parameter is in the mysql\_init() function initialization, the next five parameters are the host name or IP address, the user name, password, the database name, port number, and when the seventh parameter unix\_socket is null, it indicates that socket or the pipeline mechanism is not used, and the last parameter is often set to 0. Then it turns on txt to output the results. Open the txt file directly by using ofstream in the output mode. TxtSavePath can be defined as a save path in the macro

definition. Note that the path uses a double slash [18, 19].

The next step is to query the database, and take out the corresponding data from the database and save it. This is to obtain the face feature values of students before the comparison, and output the corresponding students' information after the comparison is successful. The contents that need to query the database are students' name, student number and the storage path of their avatars. First, you need to use the `mysql_query()` to query the result set. This function returns TRUE on successful execution and FALSE on error. The query result will be cached. And `TABLE_NAME` is the name of the table to be queried. After the table query is successful, you can use the `mysql_store_result()` to save the result set. Then using the `mysql_fetch_row()` retrieves the next row of the result set. Loop through the process to traverse the table. When there are no more rows to retrieve, the `mysql_fetch_row()` returns null. In this way, all the data in the result set can be extracted row by row. In this loop, every time the `mysql_fetch_row()` gets the current row database, assigns a value to column, and then points to the next row automatically. In the process of traversing the student information table, the data is extracted row by row. For the student's name and student number, a vector container is defined to store information. For the student's head image path, face detection and face feature extraction are completed directly on the head image, and then the face eigenvalues are saved in memory. In order to facilitate the path directly into the face detection function, the path is saved to the array. After getting the storage path of the head image, the correlation function of SDK is used to complete face detection and face feature value extraction. After the extraction of face eigenvalues, the face eigenvalues are stored in memory. Similar to the storage of names and student numbers, a vector is used to store them. This completes the steps of getting the necessary information from the database into memory [19, 22, 23].

Finally, after the face comparison is successful, need to output the completion time. If the string is spliced with the information obtained from the struct tm structure, if it is a single digit, it will not be supplemented with 0. For example, 20190302 needs to be output, and 201932 is obtained when using `snprintf` to splice strings. In order to unify the format, the `strftime` function is used. The format command set starting with the percent sign (%) is recognized, and the formatted output is placed in a string. There are many ways to express the exact date and time of the formatted string. You can format various time strings according to your needs. This article uses the decade year with the century part in %Y, month in decimal %m, day of month in decimal %d, hour in the 24-hour system in %H, minutes in decimal %M and seconds in decimal %S. They are case sensitive [24].

### 3.3. Program Control

The features of the Image Acquisition and Processing sub-module and the Data Acquisition sub-module have been completed, but our work is not yet complete. After these two modules are initialized, need to combine the sub-modules to make them work together to complete the complete system function together. The main program flowchart is as follows Figure 1:

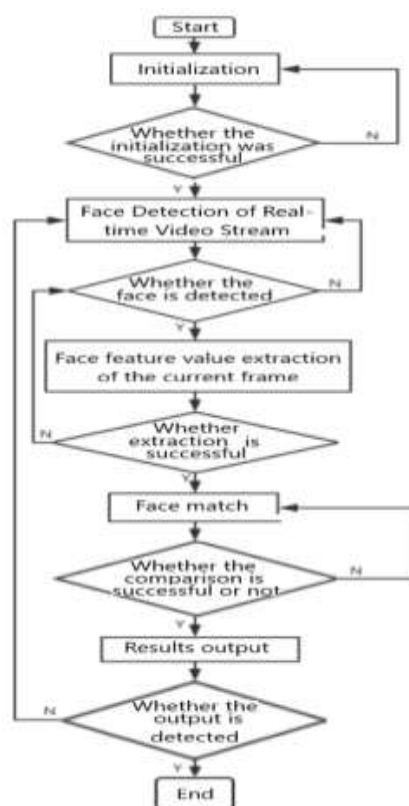


Fig 1: main program flow chart

From the flowchart, can see that the conditions for the realization of the system functions are: successful module initialization, successful face detection of real-time video stream, successful extraction of the facial feature value of the current frame, successful face comparison, and the system did not exit halfway. Each abnormal condition will make the system unable to output the correct result.

It is worth noting that Initialization includes activation of SDK, initialization of SDK engine, opening of TXT file, connection and query of database, taking the name and student number of the class into memory, opening of camera and setting the relevant parameters of saving video. Face detection of real-time video stream includes saving the camera shot image to the specified path, extracting the current frame from the video stream and converting it for SDK processing. Finally, SDK is used for face detection of the current frame. If the face is detected in the current frame, face feature extraction is carried out. After successful extraction, the face eigenvalue data in memory is traversed and compared with the face eigenvalue of the current frame for the face comparison. If the face comparison is successful and exceeds the set threshold, then output the corresponding student's name, student number and completion of the comparison time to TXT, and then start a new round of face comparison cycle that is the next frame of face detection, feature value extraction and comparison [22, 23].

Among them, the initialization of the sub-module of image acquisition and processing includes the opening of the camera, the setting of relevant parameters for saving video, and the initialization assignment

in various needed variables. Anti-initialization includes the closing of the camera and the closing of the video file.

The initialization of the data acquisition submodule includes the connection to the database and the query to the corresponding table, and saves the query results to memory. De-initialization includes freeing the resulting memory and disconnecting the database.

## IV. TEST ANALYSIS

### 4.1. Test Data

During the test, this program runs on Intel® Core™ i5-4210m CPU @2.60ghz dual-core four-thread processor, 8.00g memory, 512GB solid-state disk PLEXTOR PX512M6G+, Windows 10 professional 64-bit (10.0, version 10240) laptop, and the image acquisition device is the HD camera built-in in the notebook USB\VID\_5986&PID\_053A&MI\_00. 24 frames per second are saved by Intel IYUV codec, and the comparison is made under the condition of 1:200 face library.

Among them, the program is running CPU frequency at around 3.1 GHz, hard disk write speed is about 390 MB/s, face library data from LFW (Labeled Faces in the Wild) in the Subset of images - people with the name starting with the head of part of the students of class A and acquisition [23].

Table 1: performance analysis table

Operation	Performance analysis
Fetch the information of students in class into memory (200 people, name, student number, profile image path, face feature value extraction)	20s
Save the video	90ms
Video streaming face recognition (face detection, feature value extraction, picture frame)	100ms
Face comparison (including result output)	6ms

Test conclusion according to the table 1: it can meet the basic requirements of practical application.

### 4.2. Actual Effect

In addition, in order to visualize the SDK for the face comparison, the comparative value of each

successful comparison will be printed to the screen, and the failure information will also be displayed, as shown in the following Figure 2:

```

H:\BISHE\test\T_class3_thread_Detecti...
Face feature comparison successful : 0, 060038
Face feature comparison successful : 0, 513366
Face feature comparison successful : 0, 184679
Face feature comparison successful : 0, 165670
Face feature comparison successful : 0, 007982
Face feature comparison successful : 0, 000000
Face feature comparison successful : 0, 051276
Single Face Feature Extraction 1 fail : 81925
Face feature comparison failed : 2
Single Face Feature Extraction 1 fail : 81925
Face feature comparison failed : 2
Single Face Feature Extraction 1 fail : 81925
Face feature comparison failed : 2
Face feature comparison successful : 0, 171920
Face feature comparison successful : 0, 159725
Face feature comparison successful : 0, 433844
Face feature comparison successful : 0, 209832
Face feature comparison successful : 0, 234080
Face feature comparison successful : 0, 000143
Face feature comparison successful : 0, 016345
Face feature comparison successful : 0, 113130
Single Face Feature Extraction 1 fail : 81925
Face feature comparison failed : 2
Face feature comparison successful : 0, 236273
Face feature comparison successful : 0, 083961
Face feature comparison successful : 0, 820293
Face feature comparison successful : 0, 191007
Face feature comparison successful : 0, 048306
Face feature comparison successful : 0, 777946
  
```

Fig: 2 effect of face comparison

The image above clearly shows the comparison value after every successful comparison and the failure code for the comparative failure. (Representative reasons for this error code are available from the official website of hongsoft).

There are many kinds of failure in face matching, such as inconsistent versions of two face features to be compared, wrong format of face feature detection, the memory error of face feature detection, etc. In this paper, palm blocking is used to simulate the failure of extracting face feature value of the current frame of video stream. Among them, 81925 is the low confidence of the face feature detection results of SDK error code information. Because the palm is used to block the face, the confidence of the feature value extracted by SDK is low. There are other reasons such as low confidence of face feature detection results, blurred picture or incorrect face frame [24, 25].

When the comparison value exceeds 0.8, the corresponding student information will be output to TXT. For those of you who is responsible for WeChat applet, there is an additional output of the avatar path of the corresponding student. The output effect is shown in the Figure 3 below:



```

201502020104, \\WIN-01706210236\image\201502020104.bmp, 2019-04-15 17:11:47
201502020104, \\WIN-01706210236\image\201502020104.bmp, 2019-04-15 17:11:47
201502020104, \\WIN-01706210236\image\201502020104.bmp, 2019-04-15 17:11:47
201502020104, \\WIN-01706210236\image\201502020104.bmp, 2019-04-15 17:11:47
201502020104, \\WIN-01706210236\image\201502020104.bmp, 2019-04-15 17:11:47
201502020104, \\WIN-01706210236\image\201502020104.bmp, 2019-04-15 17:11:48
201502020104, \\WIN-01706210236\image\201502020104.bmp, 2019-04-15 17:11:48
201502020104, \\WIN-01706210236\image\201502020104.bmp, 2019-04-15 17:11:48
201502020104, \\WIN-01706210236\image\201502020104.bmp, 2019-04-15 17:11:48
201502020104, \\WIN-01706210236\image\201502020104.bmp, 2019-04-15 17:11:48
201502020104, \\WIN-01706210236\image\201502020104.bmp, 2019-04-15 17:11:48
201502020104, \\WIN-01706210236\image\201502020104.bmp, 2019-04-15 17:11:49
201502020104, \\WIN-01706210236\image\201502020104.bmp, 2019-04-15 17:11:49
201502020102, F:\\Download\\4.jpg, 2019-04-15 17:11:55
201502020104, \\WIN-01706210236\image\201502020104.bmp, 2019-04-15 17:11:59
201502020104, \\WIN-01706210236\image\201502020104.bmp, 2019-04-15 17:11:59
201502020102, F:\\Download\\4.jpg, 2019-04-15 17:12:04
201502020102, F:\\Download\\4.jpg, 2019-04-15 17:12:04
201502020102, F:\\Download\\4.jpg, 2019-04-15 17:12:04
201502020102, F:\\Download\\4.jpg, 2019-04-15 17:12:05
201502020102, F:\\Download\\4.jpg, 2019-04-15 17:12:23
201502020102, F:\\Download\\4.jpg, 2019-04-15 17:12:23
201502020102, F:\\Download\\4.jpg, 2019-04-15 17:12:23
201502020102, F:\\Download\\4.jpg, 2019-04-15 17:12:23
201502020102, F:\\Download\\4.jpg, 2019-04-15 17:12:24
201502020102, F:\\Download\\4.jpg, 2019-04-15 17:12:24
201502020102, F:\\Download\\4.jpg, 2019-04-15 17:12:24
201502020102, F:\\Download\\4.jpg, 2019-04-15 17:12:24
201502020102, F:\\Download\\4.jpg, 2019-04-15 17:12:24
201502020102, F:\\Download\\4.jpg, 2019-04-15 17:12:24
201502020102, F:\\Download\\4.jpg, 2019-04-15 17:12:25
201502020102, F:\\Download\\4.jpg, 2019-04-15 17:12:25

```

Fig 3: result output effect diagram

As you can see, the project can output the information of 3-4 students per second, which includes student ID, photo, time, etc. In addition, the saved video can also be played. The basic functions of a portrait snapshot recognition module have been achieved.

Then take a look at the use of computer resources in this project. will show that in two aspects: CPU usage and memory usage.

First of all, let's look at the use of the CPU. In the program preparation stage, which is the stage when each initialization is completed. The usage of the program CPU is shown in the Figure 4 below:

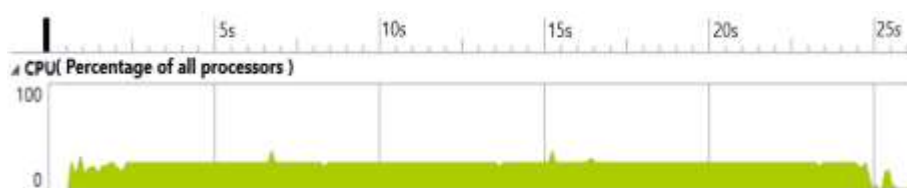


Fig 4: CPU utilization during initialization

At this stage, CPU usage stabilized at approximately 25% and peaked at 36%.

In another phase of the project, the operating situation of the CPU is different. This stage is the facial recognition stage, where the faces are fully detected and the characteristics are extracted. The video streams are compared and output in real time. The following Figure 5 is the use of memory:

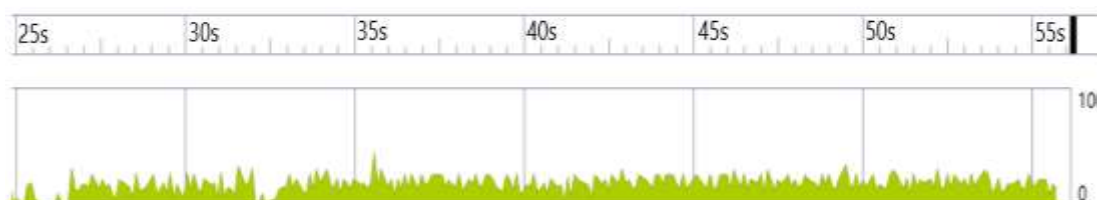


Fig 5: CPU utilization in face recognition phase

It is noticeable that the use of the CPU fluctuates slightly, mainly between 13% and 25%, with a peak value of 42%.

Then, look at the memory usage: When the names, student numbers, avatar paths and facial feature values of 200 students are extracted from the database to the memory stage, and when the camera is turned on to process the real-time video stream, the memory usage as shown in the Figure 6 below:



Fig 6: memory usage

When extracting feature values from the database in memory, the memory rose from 115M to 170M. When processing a real-time video stream, it will rise to around 230M, and then remain stable at 233M.

#### 4.3. Optimization Scheme

Here are several ways to speed up your application, such as multithreaded programming, building a face library, and replacing the SDK with a new version. In addition, hardware performance and picture quality also affect the time of SDK calls such as face detection, feature extraction and comparison [24, 25].

When comparing large amounts of data, multithreading is the best way to speed things up. In the case of 1: n (single face search scheme), multiple threads can be created for the part of face search (each thread



needs to initialize one engine), whether it is a local face search or a server face search. If the face database is too large, the face database is segmented and compared one by one (one engine is used for each piece of data), and then summarized. In addition, face eigenvalue extraction is very CPU intensive and requires a lot of time. It is better to open a single thread. For the case of  $n:n$  (multi-face search scheme), the part of face search is the same as  $1:n$ , and multi-thread search is adopted when the face database is too large. In addition, reducing the number of times you parse facial features can reduce CPU consumption [24]. However, video streaming data cannot be extracted repeatedly by assigning an ID according to the upper and lower frames (which can be achieved by Rect similarity of the rectangular box information detected by the upper and lower frames). Because the condition in this paper is limited to 1:200 face search, there is no need to use multithreading to speed up [25, 27].

Building a face library is also an option for speed. Face database is a feature database which saves the extracted face features. In order to reduce the time of saving student information into memory during program initialization, the feature value of student face is saved to face database in advance. The feature code is a binary field, which is normally provided by databases to store binary data. MYSQL USES Blob. It should be noted that if the SDK version is updated, the feature values saved by the face library should also be updated, otherwise, the problem of different versions of the feature values cannot be compared. This is also the reason why the student database chooses to save the avatar path [27, 28].

Replace the SDK with a new version. In early April 2019, the SDK was updated with version 2.1, which is faster than SDK 2.0. In addition, the new version of the Windows platform has a live detection function, can effectively prevent photos, screen remakes and other cheating attacks.

In addition, hardware performance and picture quality will also affect the SDK call time of face detection, face tracking, face feature extraction, etc. Therefore, when running the program, try to choose a high-performance host, and try to shoot clear and clear pictures in the image acquisition, can also effectively improve the speed of the program.

To sum up, this design realizes a face recognition system for real-time video stream by using honsoft's offline SDK for face recognition and combining OPENCV and MySQL database. Experimental data show that the portrait snapshot recognition module developed in this paper has achieved a certain level of functionality.

## REFERENCES

- [1] Gao Peijun, Zhao Dan, Chen Xuanang, "Multi-dimensional data modelling of video image action recognition and motion capture in deep learning framework". IET Image Processing. 2020, vol. 14 (no. 7): pp 1373-1381.
- [2] Yin Qun, Zhang Jianbo, Ji Gu, Yin Meisu, Xu Yunsheng, "A method based on the embedded control for the third order rubik's cube robot". EEA - Electrotehnica, Electronica, Automatica. 2020, vol. 68 (no. 2): pp 104-111.
- [3] Shen Junfei, Chen Yuan, "Development of energy correction based multishot snapshot spectral imaging system". Optics Communications. 2019; v 432, p 116-122.
- [4] Masoudnia Saeed, Mersa Omid, "Multi-representational learning for Offline Signature Verification using Multi-

- Loss Snapshot Ensemble of CNNs". Expert Systems with Applications. 2019; v 133, p 317-330.
- [5] Yin Qun, Zhang Jianbo, Wang Xinkai, Liu Si, "The video intelligent car based on wireless sensor". Cluster Computing. 2019, vol. 22: pp 5135-5150.
- [6] Balazia Michal, Sojka Petr, "Gait recognition from motion capture data". ACM Transactions on Multimedia Computing, Communications and Applications. 2018, vol.14 (no. 1).
- [7] Chiu Chih-Cheng, Liu Tung-Kuan, Lu Weu-Ting, Chen Wen-Ping, Chou Jyh-Horng, "A micro-control capture images technology for the finger vein recognition based on adaptive image segmentation". Microsystem Technologies. 2018, vol. 24 (no. 10): pp 4165-4178.
- [8] Oguchi Kimio, Akimoto Keita, "Feasible human recognition by using low-cost markerless motion capture". 2018 9th IEEE Control and System Graduate Research Colloquium. 2018, pp 229-233.
- [9] Yin Qun, Zhang Jianbo, Chen Lang, Gu Ji, "Design of an A/D data acquisition device based on USB". EEA - Electrotehnica, Electronica, Automatica. 2018, vol. 66 (no. 3): pp 106-113.
- [10] Yin Qun, Zhang Jianbo, Xu Yunsheng, Gu Ji, "An automatic flower planting system using analog-to-digital conversion, soil moisture and temperature acquisition". IPPTA: Quarterly Journal of Indian Pulp and Paper Technical Association. 2018, vol. 30 (no. 2): pp 291-296.
- [11] Balazia Michal, Plataniotis Konstantinos N, "Human gait recognition from motion capture data in signature poses". IET Biometrics. 2017, vol. 6 (no. 2): pp 129-137.
- [12] Jianbo Zhang, Qun Yin, Liu Si, Lang Chen, "Wireless car control system based on single chip microcomputer ". EEA - Electrotehnica, Electronica, Automatica. 2017, vol. 65 (no. 3): pp 147-155.
- [13] Gao Jun, Zhou Chang, "Toward continuous pattern detection over evolving large graph with snapshot isolation". VLDB Journal. 2016; v 25, n 2, p 269-290.
- [14] Chen Chunhui, Zhang Qun, "2D single snapshot imaging using MIMO radar based on SR". Electronics Letters. 2016; v 52, n 23, p 1946-1948.
- [15] Maione Bryan D, Luo David, "Spatially heterodyned snapshot imaging spectrometer". Applied Optics. 2016; v 55, n 31, p 8667-8675.
- [16] Zhang, Shuxing, Zhang, Jianbo, Yin Qun, Liu Si, "Intelligent pulse test instrument based on wireless network ". Sensor Letters. 2016, vol. 14 (no. 9): pp 896-905.
- [17] Jianb Zhang, Qun Yin, "Design of the smart bike odometer". EEA - Electrotehnica, Electronica, Automatica. 2015, vol. 63 (no. 3): pp 91-97.
- [18] M. BurecãandI. Lucaciu, "Oscillography Based Analysis of the Arc Stability in Manual Welding with Coated Electrode in Pulsed Current". Advanced Materials Research. 2015, vol. 1128: pp 209-216.
- [19] Jianbo Zhang, Qun Yin, "Design of public regional emergency communication system based on RS485 ". EEA - Electrotehnica, Electronica, Automatica. 2014, vol. 62 (no. 3): pp 126-134.
- [20] Yin Qun, Zhang Jianbo, "Design of Power Line Carrier Communication System based on FSK-KQ330 Module". Electrotehnica Electronica Automatica. 2014, vol. 62 (no. 3): pp 135-142.
- [21] Yin Qun, Zhang Jianbo, "Improving the teaching effectiveness of an SCM course". World Transactions on Engineering and Technology Education. 2014, vol. 12 (no. 2): pp 271-275.
- [22] Qun Yin, Jianbo Zhang, "Design of power line carrier communication system based on FSK-KQ330 module". EEA - Electrotehnica, Electronica, Automatica. 2014, vol. 62 (no. 3): pp 135-142.
- [23] Zhu Feng, Zhang Qun, "Single snapshot sparse imaging method for MIMO high-resolution eadar with OFD-LFM signal". Yi Qi Yi Biao Xue Bao/Chinese Journal of Scientific Instrument. 2013; v 34, n 5, p 1076-1085.
- [24] Hong Sheng, Wan Xian-Rong, "An angle estimation method for multi-targets in bistatic MIMO radar with single snapshot". Dianzi Yu Xinxi Xuebao/Journal of Electronics and Information Technology. 2013; v 35, n 5, p 1149-1155.
- [25] Yin Qun, Zhang Jianbo, "Design of control platform systems based on object-oriented". In International Review

on Computers and Software. 2012; vol. 7(no. 1): p. 438-442.

- [26] Yin Qun, Zhang Jianbo, "Design of control platform systems based on object-oriented". International Review on Computers and Software. 2012, vol. 7 (no. 1): pp 438-442.
- [27] Wong Fai, Chao Sam, "Cyclops - Snapshot translation system based on mobile device". Journal of Software. 2011; v 6, n 9, p 1664-1671.
- [28] Sheng Ping, Ni Dongwei, "Design and realization of dynamic target tracking and automatic close-up snapshot". Dongnan Daxue Xuebao (Ziran Kexue Ban)/Journal of Southeast University (Natural Science Edition). 2011; v 41, n 5, p 967-972.