

**2019-2020学年度第一学期**

**仲恺农业工程学院**

**信息科学与技术学院**

**《前沿技术综合实训》课程设计报告**

|  |  |
| --- | --- |
| **题目：** | 前端人脸识别登录 |
| **班级：** | 信管161 |
| **学号：** | 201610224328 |
| **姓名：** | 谢俊杰 |
| **指导老师：** | 贺超波 |
| **时间：** | 第1－2周/2019年9月2日－2019年9月15日 |
| **成绩等级：** |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **总分** | **指导老师评分** | **备注** |
| **一、报告文档** | **30** |  |  |
| 1内容基本要点 | 20 |  |  |
| 2格式 | 10 |  |  |
| **二、程序验收** | **50** |  |  |
| 1功能 | 30 |  |  |
| 2程序结构、编码规范 | 10 |  |  |
| 3界面 | 10 |  |  |
| **三、考勤** | **10** |  |  |
| **四、亮点或难点** | **10** |  |  |
| 1功能亮点 | 6 |  |  |
| 2解决难题 | 4 |  |  |
| **总评分** | |  |  |

目录

[1 背景介绍 1](#_Toc21523522)

[1.1 引言 1](#_Toc21523523)

[2 需求分析 1](#_Toc21523524)

[2.1 应用程序的功能需求分析 1](#_Toc21523525)

[2.2 百度云人脸对比模块设计与实现 2](#_Toc21523526)

[3 概要设计 4](#_Toc21523527)

[4 详细设计 5](#_Toc21523537)

[4.1两张图片对比代码 5](#_Toc21523538)

[4.2找两张图片进行比对，返回json数据 7](#_Toc21523539)

[4.3jsp页面通过ajax执行上传数据到servlet 7](#_Toc21523541)

[5 系统开发及测试 10](#_Toc21523544)

[6 总结 14](#_Toc21523556)

[参考文献 15](#_Toc21523556)

# 背景介绍

* 1. 引言

现在是2019年，人脸识别技术已经相当成熟了，百度自2017年发布人脸识别技术，已经被广泛应用，不管从现在的iphoneX掀起的面部解锁到手机应用端的各种人脸认证，这一技术已经悄然升息的方便了我们的生活，但是在web端注册登录缺很少用到刷脸登录，第一个最主要的原因可能是安全隐私方面人们对大数据时代的误解。

1. 需求分析

本章主要介绍在一个模仿腾讯QQ聊天登录聊天功能的前端项目基础上，运用现目前热门技术在该系统的应用开发工作。主要创新体现在在原有账号密码登录的基础上，利用百度云人脸对比接口，加入人脸识别登录功能。

* 1. 应用程序的功能需求分析

该软件最主要的功能就是要能识别出人脸， 首先该系统需要对通过摄像头拍照而获 取到的原始的人脸图片进行一系列处理才可进行下一步的工作， 该处理过程也称图像预 处理。预处理这个模块在整个人脸识别系统的开发过程中占有很重要的地位，只有预处 理模块做的好，才可能很好的完成后面的人脸定位和特征提取这两大关键模块。因此本 设计中所要完成的主要功能如下所述：

图像获取功能 ： 该模块主要是从摄像头拍照后进行获取图片，也可以从图片库中获取，获取后 的图片可以在软件的界面中显示出来以便进行识别。

图像预处理功能 ： 该模块主要包括图像光线补偿、图像变成灰色、高斯平滑、均衡直方图、实现 图像对比度增强、二值化变换等。

人脸定位功能 ： 该模块主要是将处理后的人脸图片进行定位，将眼睛、鼻子、嘴巴标记出来， 以便进行特征提取。

特征提取功能 ： 该模块是在定位后的人脸图片中将眼睛、鼻子、嘴巴的特征值提取出来。 识别功能 ： 该模块是将从图片中提取的特征值和后台数据库中的值进行比较来完成识别 功能。

二 开发环境需求分析

1、硬件环境

（1）硬件配置原则 具有可靠性，可用性和安全性，具有完善的技术支持。能够满足个人学习和设计需 要。 （2）运行本软件所需的硬件资源 CPU： 800M 及以上；内存： 128M 及以上

2、软件环境

（1）系统软件配置原则 能够满足该软件的可靠性，可用性和安全性的要求 （2）系统软件配置方案 ① 配置有持续工作能力、高稳定性、高度可集成的开放式标准的操作系统，如 Windows2000，Windows NT，UNIX ，Linux 等。 ② 配备符合 ANSI/ISO 标准的高级程序设计语言处理软件。如： Visual C++ 6.0。 ③ 熟悉 C++高级程序设计语言。

3、 运行环境需求分析

（1）、硬件环境 CPU：500M 及以上；内存： 64 M 及以上。 （2）、软件环境 可以运行在微软公司近年来所出的各种操作系统。如 Windows 98、Windows Me、 Windows 2000、Windows NT 等。

* 1. 百度云人脸对比模块设计与实现

能力介绍

接口能力

两张人脸图片相似度对比：比对两张图片中人脸的相似度，并返回相似度分值；

多种图片类型：支持生活照、证件照、身份证芯片照、带网纹照四种类型的人脸对比；

活体检测控制：基于图片中的破绽分析，判断其中的人脸是否为二次翻拍（举例：如用户A用手机拍摄了一张包含人脸的图片一，用户B翻拍了图片一得到了图片二，并用图片二伪造成用户A去进行识别操作，这种情况普遍发生在金融开户、实名认证等环节。）；

质量检测控制：分析图片的中人脸的模糊度、角度、光照强度等特征，判断图片质量；

业务应用

用于比对多张图片中的人脸相似度并返回两两比对的得分，可用于判断两张脸是否是同一人的可能性大小。

典型应用场景：如人证合一验证，用户认证等，可与您现有的人脸库进行比对验证。

调用方式：请求URL数据格式。

向API服务地址使用POST发送请求，必须在URL中带上参数access\_token，可通过后台的API Key和Secret Key生成，具体方式请参考“Access Token获取”。



图1 百度云人脸对比接口的请求参数

1. 概要设计
2. 在百度云创建应用，获取API key 及Secret Key

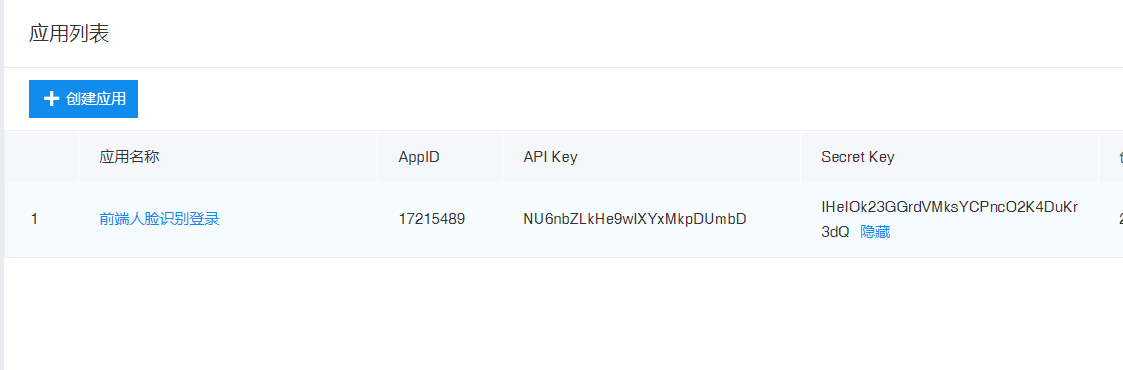


图1

1. 创建人脸库及创建用户组，在用户组内创建用户，并上传自己的人脸照片



图2

1. Eclipse添加项目，并开启服务器

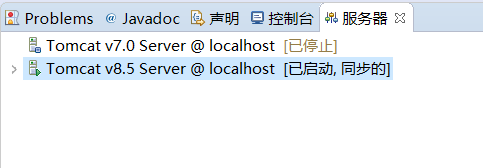


图3

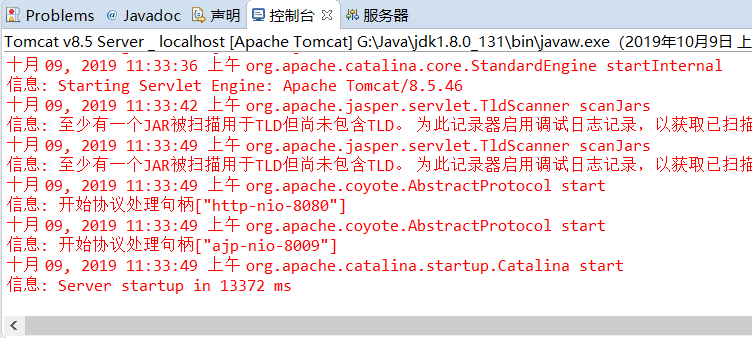


图4

四、首先进行人脸对比，将两张图片的base64数据传入进行对比，得到一些列数据，根据返回的相似度确定是否为同一人，返回相似度超过80时跳转至登录成功页面。

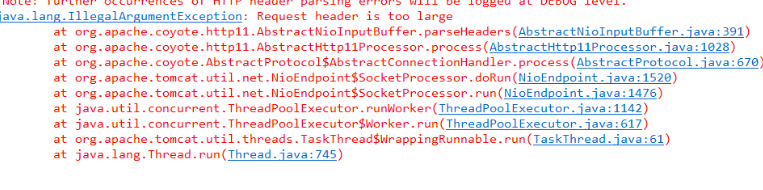


图5

1. 详细设计

4.1两张图片对比代码如下：

package faceLogin;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import utils.Base64Util;

import utils.FileUtil;

import utils.GsonUtils;

import utils.HttpUtil;

/\*\*

\* @author Administrator

\*

\*/

/\*\*

\* 人脸对比

\*/

public class faceMatch {

public static String match(String image2) {

// 请求url

String url = "https://aip.baidubce.com/rest/2.0/face/v3/match";

try {

byte[] bytes1 = FileUtil.readFileByBytes("C:\\Users\\Administrator\\Desktop\\liu8.png");

/\*byte[] bytes2 = FileUtil.readFileByBytes("C:\\Users\\Administrator\\Desktop\\liu6.png");\*/

String image1 = Base64Util.encode(bytes1);

/\*String image2 = Base64Util.encode(bytes2);\*/

List<Map<String, Object>> images = new ArrayList<>();

Map<String, Object> map1 = new HashMap<>();

map1.put("image", image1);

map1.put("image\_type", "BASE64");

map1.put("face\_type", "LIVE");

map1.put("quality\_control", "LOW");

map1.put("liveness\_control", "NORMAL");

Map<String, Object> map2 = new HashMap<>();

map2.put("image", image2);

map2.put("image\_type", "BASE64");

map2.put("face\_type", "LIVE");

map2.put("quality\_control", "LOW");

map2.put("liveness\_control", "NORMAL");

images.add(map1);

images.add(map2);

String param = GsonUtils.toJson(images);

AuthService auth = new AuthService();

String accessToken = auth.getAuth();

// 注意这里仅为了简化编码每一次请求都去获取access\_token，线上环境access\_token有过期时间， 客户端可自行缓存，过期后重新获取。

String result = HttpUtil.post(url, accessToken, "application/json", param);

String score=result.split(",")[5].split(":")[2];

return score;

} catch (Exception e) {

e.printStackTrace();

}

return null;

}

}

4.2找两张图片进行比对，会返回json数据，再将其化成字符串

新建servlet执行此java文件

faceMatch match = new faceMatch();

String score = match.match();

System.out.println(score);

4.3jsp页面通过ajax执行上传数据到servlet

这里主要思想就是调用系统摄像头，将视频流传入video标签内，就在网页上显示出了摄像头的效果，再通过canvas截图，将某一时刻的视频截为图片，

后面再将图片转为base64格式传入后端，进行一系列人脸操作

<%@ page language="java" import="java.util.\*" pageEncoding="utf-8"%>

<%

String path = request.getContextPath();

String basePath = request.getScheme() + "://"

+ request.getServerName() + ":" + request.getServerPort()

+ path + "/";

%>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>demo01</title>

<script src="http://www.jq22.com/jquery/1.11.1/jquery.min.js"></script>

<script src="js/jquery-1.9.1.min.js"></script>

<style type="text/css">

.line{

position: absolute;

top: 300px; left: -80px;

z-index: 2;

height: 15px; width: 300px;

background: linear-gradient(#33ffff, #66cccc,#99cccc); opacity: 0.7 /\* 标准的语法 \*/

/\*动画效果\*/

animation: myScan 3s infinite alternate;

-webkit-animation: myScan 3s infinite alternate;

}

@keyframes myScan{

from { top:5px; }

to { top: 300px; }

}

-webkit-@keyframes myScan{

from { top:5px; }

to { top: 600px; }

}

.box{background-image: url('image/bg2.jpg');width: 100%;height: 640px;background-size:100%, 100%;}

.video{width:250px;height:250px;margin:auto;background-image: url('image/faceBorder.png')}

</style>

</head>

<body>

<div class="box">

<div class="right">

<div class="video">

<!-- <div class="line"></div> -->

<video id="myVideo" src="" class="video" ></video>

</div>

<canvas id="myCanvas" width="600" height="400" hidden="hidden"></canvas>

<script>

var myVideo = document.getElementById('myVideo');

navigator.mediaDevices.getUserMedia({

video: true

}).then(function (mediaStream) {

myVideo.srcObject = mediaStream;

myVideo.onloadedmetadata = function () {

/\* myVideo.controls = "controls"; 不显示控件\*/

myVideo.play();

}

});

function jiance(){

var canvas = document.getElementById('myCanvas').getContext('2d');

canvas.drawImage(myVideo, 0, 0);

var imgSrc = document.getElementById("myCanvas").toDataURL(

"image/png").split("base64,")[1];

$.ajax({

type: "POST",

url:'faceMatch',

data:{

message:imgSrc

},

success:function(score){

var scoreMatch = score.split(".")[0];

if(scoreMatch>80){

window.location="loginSuccess.jsp"

}

else{

return;

}

}

})

}

setInterval("jiance()","1100"); //每隔一秒执行一次函数截图

//将图片Base64 转成文件

</script>

<script type="text/javascript" color="120,148,255" opacity='0.8' zIndex="1" count="100" src="https://files.cnblogs.com/files/lfri/canvas-nest.js"></script>

<div id="iframe">

</div>

</div>

</body>

</html>

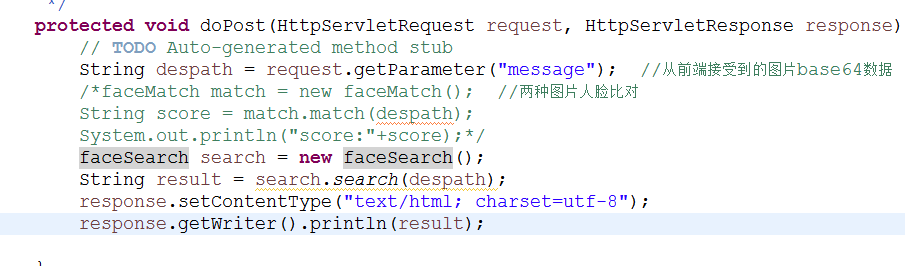
1. 系统开发及测试

我找了两张相似的图片进行比对，这里注意图片格式除了gif格式不支持，每张图片的大小不能超过2M。

返回的数据如下，result里面score:97.75291443，推荐阈值为80基本上判断为同一个人，不管调用哪种api要么相似度是90多要么相似度是10以下，很少出现中间的值。

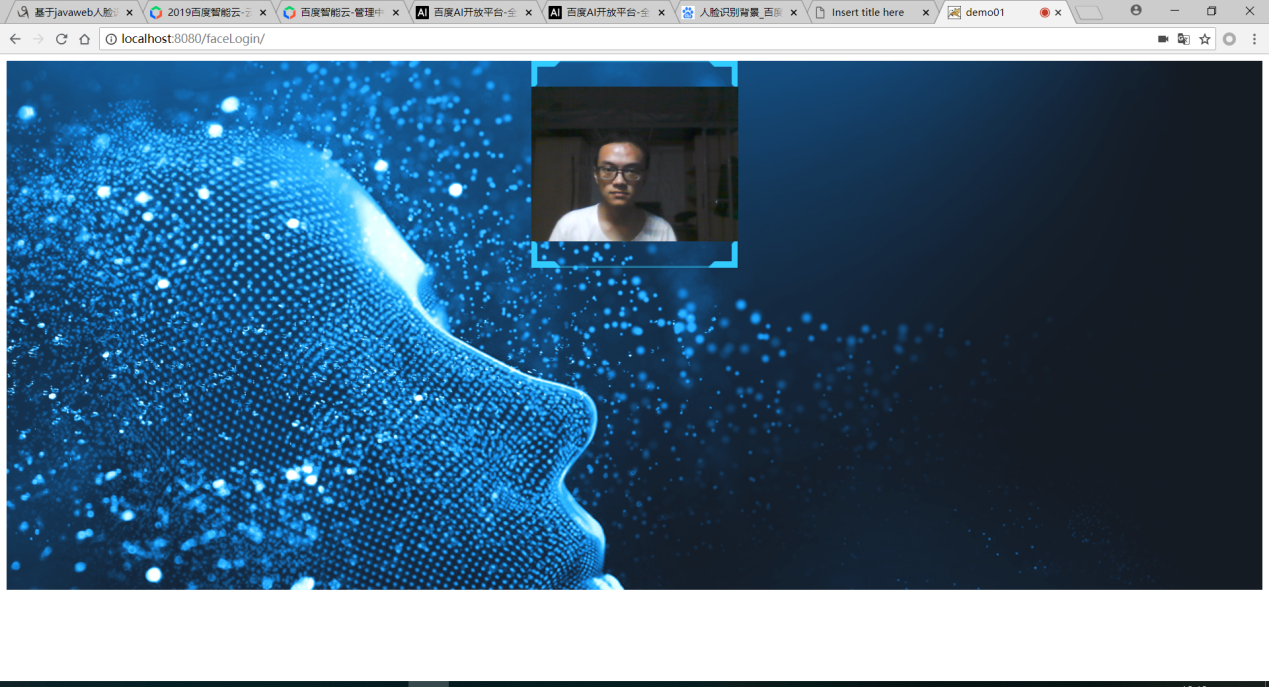
result:{"error\_code":0,"error\_msg":"SUCCESS","log\_id":304592874888025681,"timestamp":1557488802,"cached":0,"result":{"score":97.75291443,"face\_list":[{"face\_token":"8ea5f423bb1cf93b87ebce4ff2ac623b"},{"face\_token":"31c258d65509177ba99aecf28dadc496"}]}}

这里js函数将每一秒执行一次函数截一次图并传到后台来比对，可以在控制台看到程序一直在执行，并且返回分数，当返回的分数大于80分时跳转到登录成功页面。



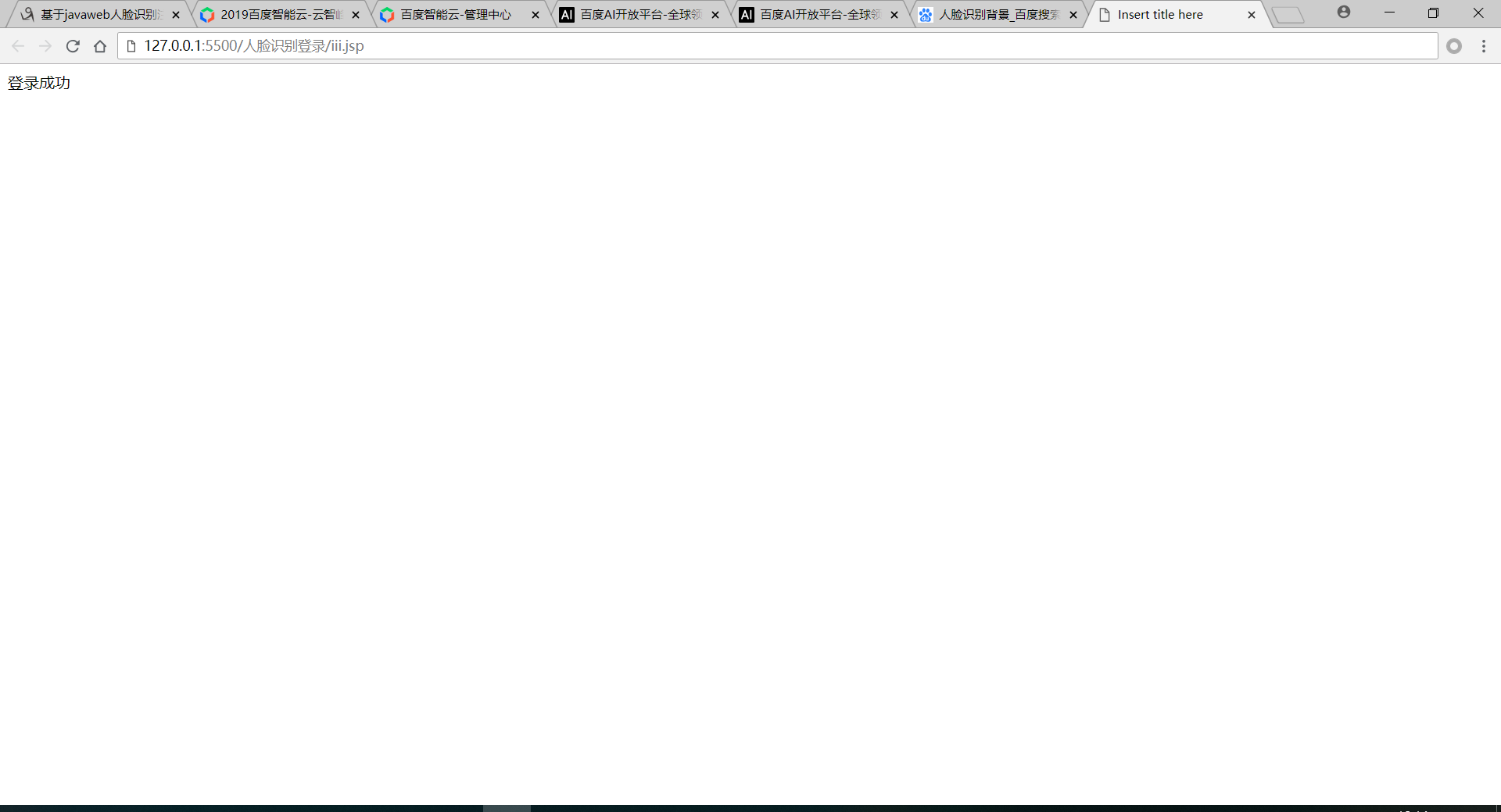


测试图片1



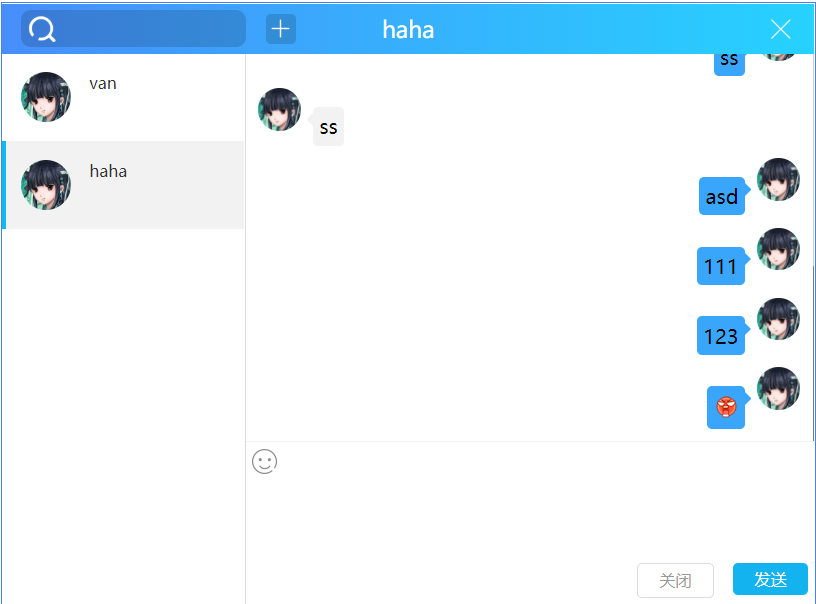
测试图片2

约一秒后，跳转至登录成功页面



测试图片3

实际的理想效果应该是登录到如下聊天页面，但是由于此前端项目使用了环信的即时通讯云接口服务，该接口内部有免费的注册登录聊天功能供开发者使用，登录成功后域名不发生改变，因此无法跳转。可以登录测试账号haha，密码123456至如下聊天页面。登录另一个账号van，密码123456，可以两者之间互相发信息与表情进行互动。



测试图片4



测试图片5

1. 总结

此次的人脸识别登录功能，通过运用百度云的人脸识别接口，比对图片中两张人脸的相似度，并返回相似度分值，超过设定的阈值即可成功登录。老师如果想测试，可在腾讯QQ聊天HTML项目文件中打开拍照.html文件。开启摄像头后点击拍照，利用截图软件(例如微信的截图功能)，截下人脸照。由于没有做人脸注册功能，上传新用户人脸照这一步骤只能手动录入。随后登录<https://console.bce.baidu.com/ai/#/ai/face/facelib/userList~appId=1215882&groupId=xie111>，百度智能云账号15814529603，密码xie5342233，在用户组中新建用户并上传人脸照片。添加成功后，在eclipse中添加faceLogin-master项目，并开启服务器，登录<http://localhost:8080/faceLogin/> 即可测试人脸识别登录功能。

通过这次的《前沿技术综合实训》课程设计，使我大致了解了目前的热门技术，如物联网、云计算、大数据、人工智能以及区块链等方面的技术，最后选择了做人脸识别登录。由于我学习与将来工作的方向是前端开发，之前也有做过前端的一个类似于腾讯QQ聊天功能的项目，便在此项目基础上增加一个人脸识别登录功能。因为之前没有了解过人脸识别这一听起来十分“高大上”的技术，查阅之后才发现这项技术已经十分成熟，各种接口使用起来也是非常方便。但是前端能使用人脸识别接口的开发语言只有node.js，还没有学过。只能翻起以前学过的java，在网上借鉴他人Web前端+Java后台实现人脸识别登录的项目，一步一步学习模仿如何使用接口，如何调用参数。越是学习，就越发现人脸识别的算法有多么深奥复杂，也感谢百度智能云能提供如此方便，越过技术障碍与深奥算法直接向开发者提供使用接口的服务。

参考文献

[1] Web前端+Java实现人脸识别登录

<https://blog.csdn.net/youth_shouting/article/details/88804324>

[2] 【Face++】使用JS实现人脸识别Demo. <https://www.jianshu.com/p/2e187a91ec7e>

[3] 纯前端实现人脸识别-提取-合成Demo.

<https://blog.csdn.net/yanjun821126/article/details/81353840>

[4] 纯前端JS实现人脸识别眨眨眼张张嘴案例

<https://blog.csdn.net/qq_34137397/article/details/81059788>

[5] 梁永先. Java Web 程序设计慕课版[D]. 人民邮电出版社, 2016.

[6] 王立波. Java语言程序设计[M]. 电子工业出版社, 2016.