

JavaScript 结课报告

题目： 人脸融合轮播图

班级： 软件 16-4 班

姓名： 艾 程

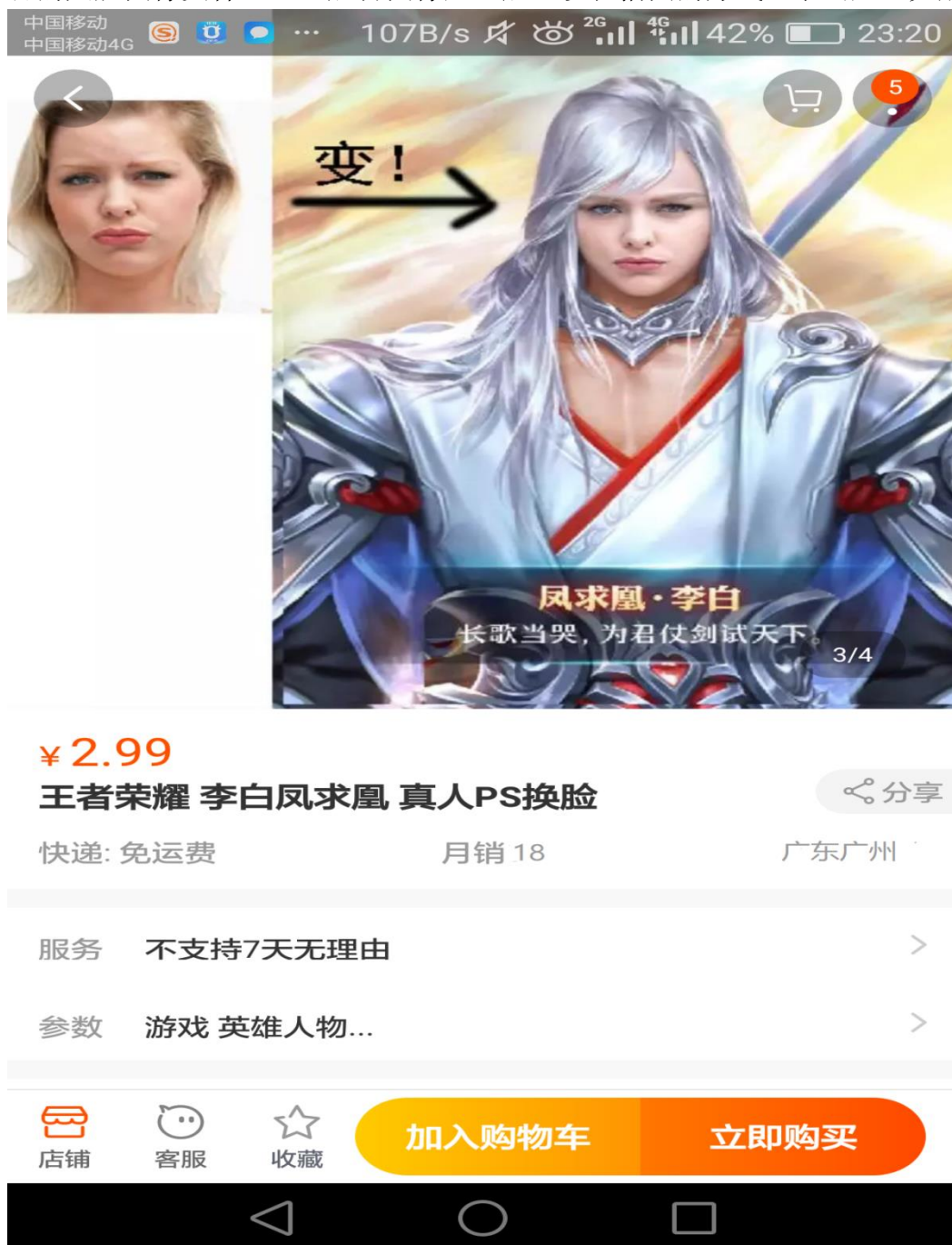
成绩：

2018 年 11 月 2 日

一、项目概述

一开始想要做这个是因为某一天，闲暇刷淘宝时，发现居然有这种商品。

这个效果其实可以直接利用网络上现成的 sdk 免费开发出来的——例如腾讯优图下的人脸融合功能，拿来赚钱岂不是美滋滋？于是我便萌生了做这个项目的想法——由用户上传需要融合的人脸图像文件，经过后台图像处理，以轮播图的方式显示出处理完成的图片。



二、项目设计及实现

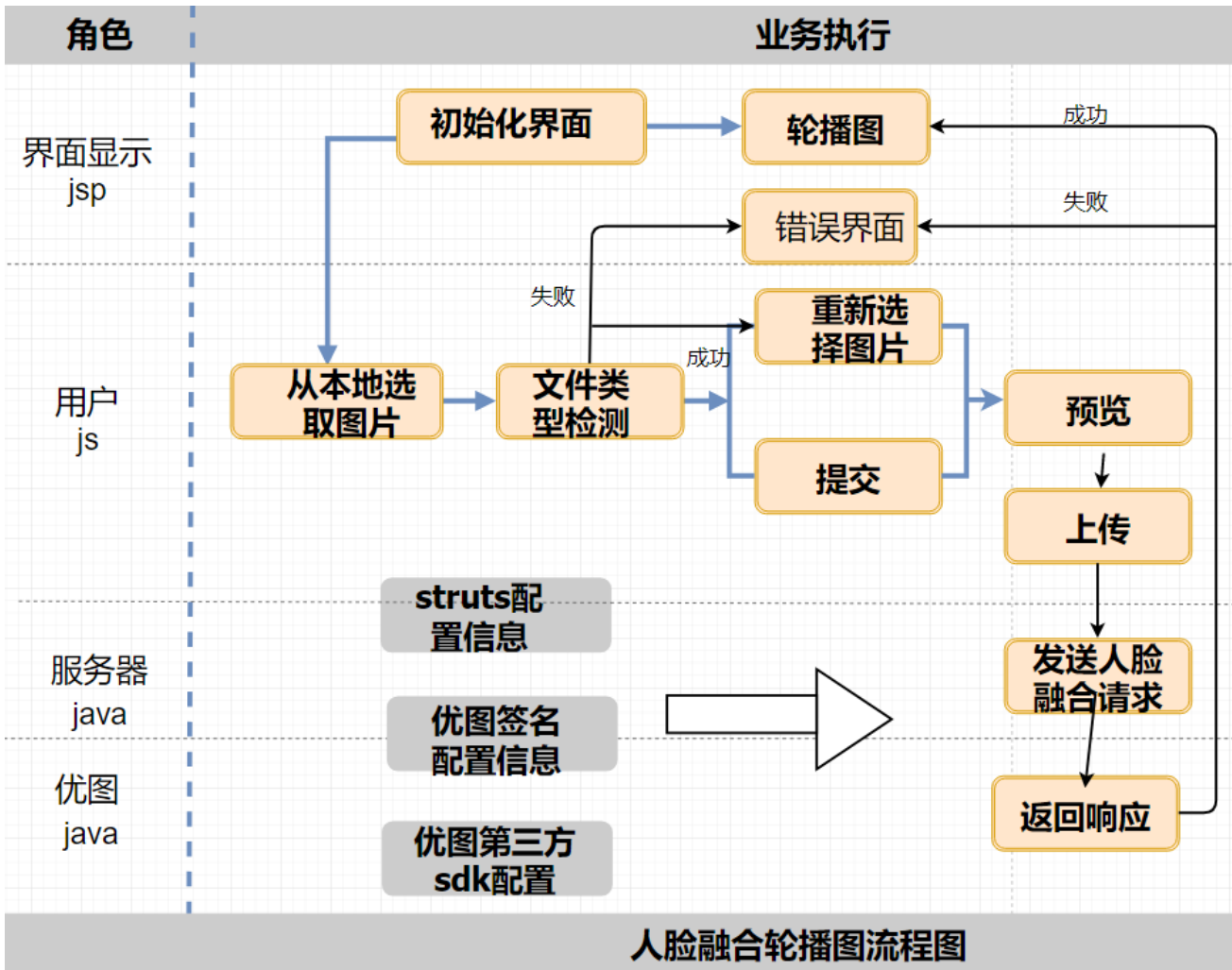
首先要取得用户选取的图片路径，上传到当前项目中。

然后采用腾讯优图提供的人脸融合功能服务进行图像处理。由于腾讯优图的人脸融合功能有多个素材模板可供开发者免费使用，因此考虑使用轮播图播放每一种素材模板的融合效果，使用轮播图主要因为节省了空间，也方便图片的归类。

上传文件的时候需要判断文件类型是否合法。轮播图，不仅可以通过两侧的左右按钮实现图片切换，还要求能够使用右下角的圆点导航实现图片切换，并且需要它有自动播放的功能，如果用户的鼠标停在轮播片上，则停止自动播放。

有关于不同语言之间的数据传递，例如 jsp 中发送的表单数据传递给 java 代码，使用刚学的 struts2 框架设置相应的 action，传递需要的数据。

流程图大概如下：



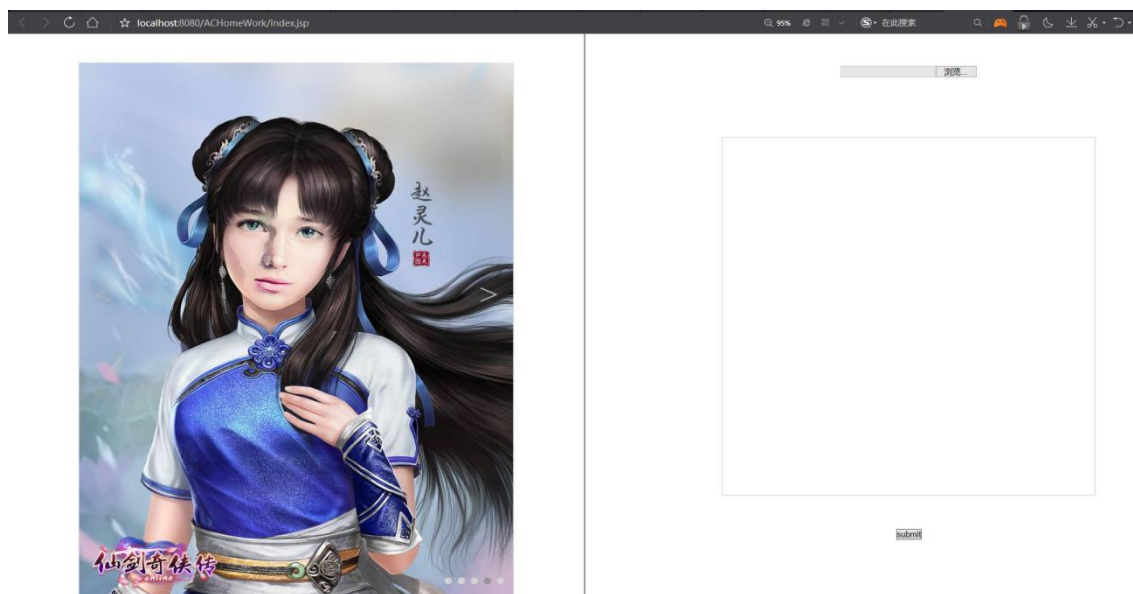
轮播图的原理与实现步骤为:

- 1、假设要显示五张 700*900 的图（例 1. jpg, 2. jpg, ..., 5. jpg），为了实现轮播效果，本算法中需要 7 张图。其中，第一张图的图片是最后一张图 5. jpg。第二张即为 1. jpg，第三张图为 2. jpg, ... 第七张图为 1. jpg。
- 2、将这七张图片拼接起来变成“大图”。
- 3、再固定一个显示框（html 中的 container），相当于一个“相机”，镜头的大小设置为 700*900，只能容下一张图。
- 4、相机镜头一开始固定在第二张图片，也就是 1. jpg 的位置，一开始显示 1. jpg。
- 5、设定相机与大图的相对的位置值——left 值，依据 left 值选定需要显示图片的位置。
- 6、用户向前向左翻页，其实是大图相对相机镜头的向右操作，所以 left 会自加 700。向后翻页操作同理，left 自减 700。
- 7、因此有，定位第一张图的 left 值为 0，第二张为-700，第三张为-1400., 第七张为-4900。

四、项目运行示意图

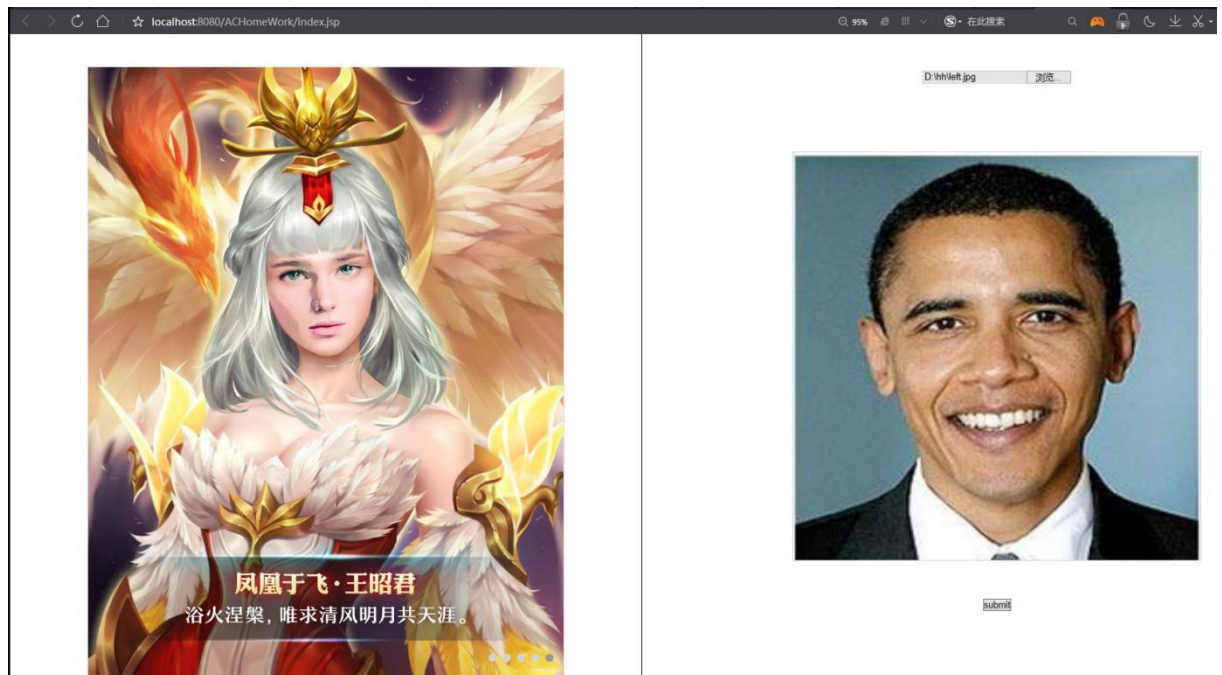
测试浏览器：QQ 浏览器、微软 Edge

- 1、输入访问地址：<http://localhost:8080/ACHomeWork/index.jsp>，如图 1 所示。



（图 1 显示界面）

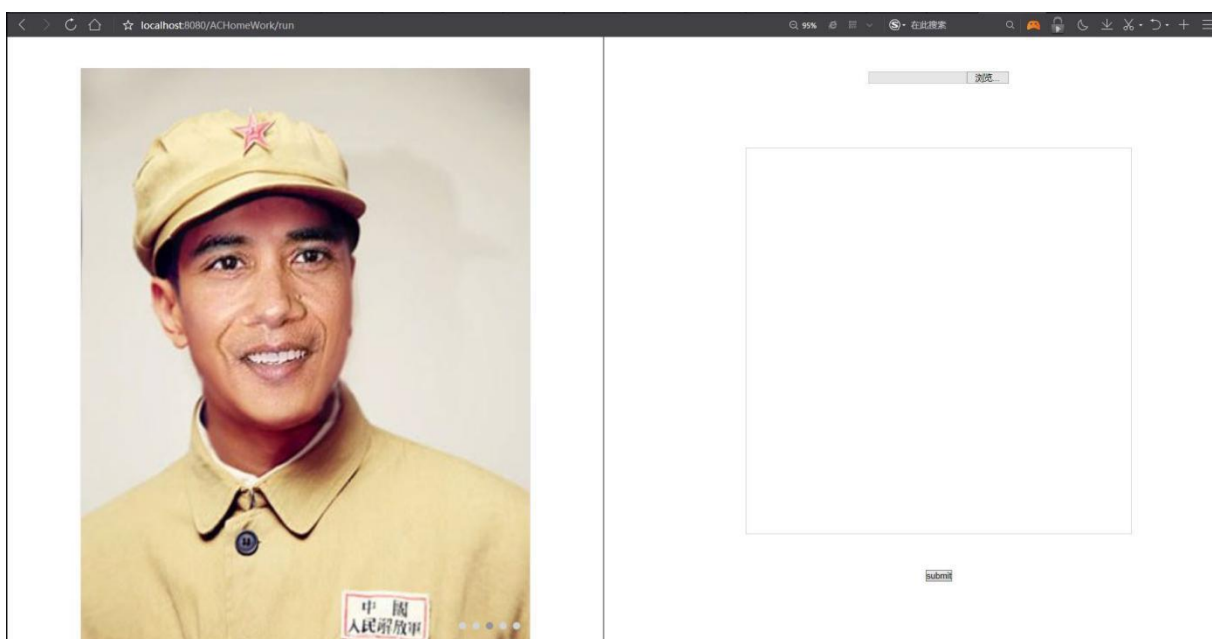
- 2、点击“浏览”按钮，在本地选择图片路径后，预览图如图二所示。



（注：此时由于轮播图的自动播放功能，会自动播放不同的图片）

（图 2 预览选中的图片）

3、点击“提交”按钮，上传需要融合人脸的图片后。结果如图三，显示已经成功融合的人脸效果。



（图 3 上传预览图成功后）

4、移动鼠标到轮播图处，左右切换按钮显示出来，可以通过左右切换按钮和小圆点导航

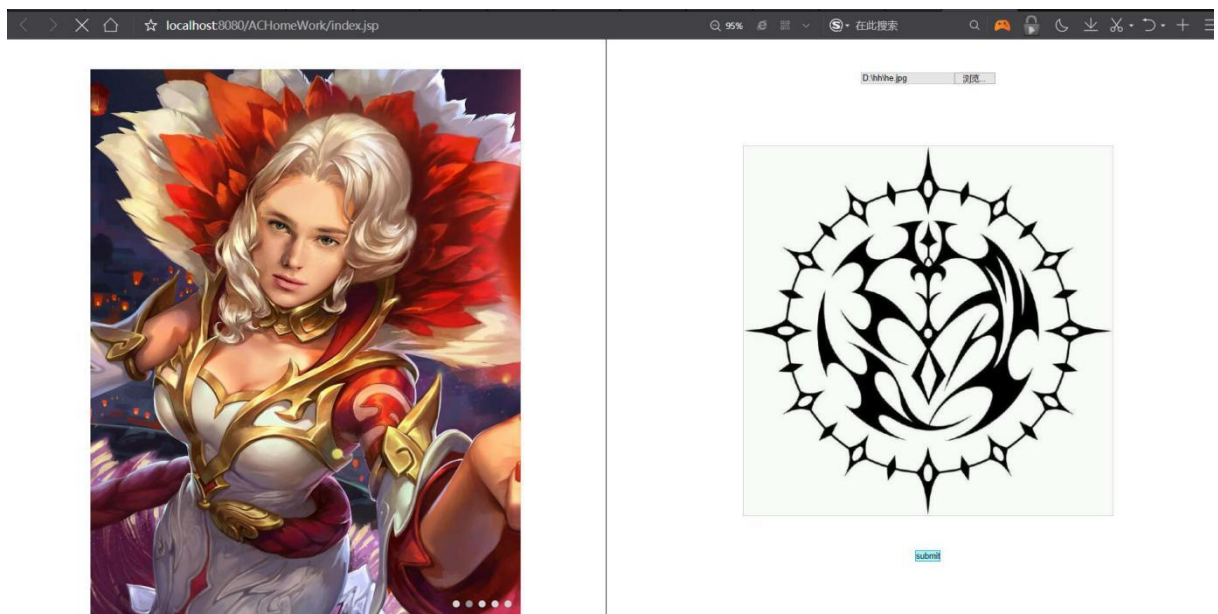
实现切换图片的操作。如果鼠标离开轮播图则自动播放。如图四。





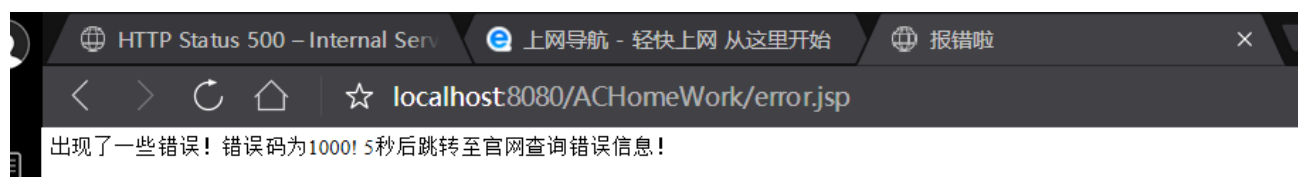
（图 4 轮播图中成功进行人脸融合的图片）

5、如果上传的图片没有人脸（无法进行人脸融合的图片），如图五。



(图 5 上传错误图片)

6、经过服务器审核后响应返回错误码，并且提示用户，如图六。



(图 6 出错界面 提示用户)

7、5 秒后会自动跳转到官网的错误码，提示了错误码 1000 为“无人脸”错误。如图七。



图 7 官网错误码页面

五、总结

学到的知识

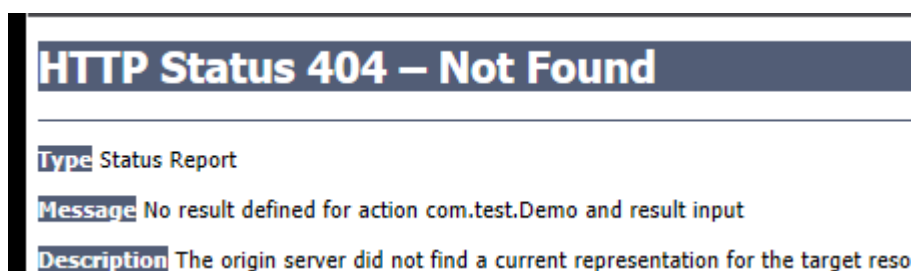
- (1) 配置与使用第三方 sdk
- (2) 正则表达式匹配文本信息检测文件类型
- (3) structs 的配置与信息传递
- (4) css 的编写

（5）轮播图的逻辑实现

经验和教训

（1）测试阶段遇到了如下报错：

```
java.io.FileNotFoundException: liBai_Test2.jpg not exist
    at com.youtu.Youtu.GetBase64FromFile(Youtu.java:145)
    at com.youtu.Youtu.FaceFuse(Youtu.java:822)
    at com.test.Demo.main(Demo.java:19)
```



```
严重：与元素类型 "package" 相关联的属性 "name" 应有左引号。at (null:6:19)
```

现象：十分迷茫，这是我保证文件没有缺失、代码没有明显问题、代码完全没有修改过的情况下，重启 tomcat 服务器以及刷新页面操作三次，居然报了三种不同错误。

原因：代码运行存在延迟。

解决：重启 eclipse, 更换浏览器运行。

经验：以后遇到这种问题，不用急着找代码的漏洞，这锅也许就是软件的问题，毕竟我们也不知道软件的底层是怎么实现的。试着重启 eclipse\tomcat，更换\刷新浏览器即可。

（2）正则表达式出错

现象：就算文件类型是其他的也能运行。

解决与经验：正则表达式匹配开始前，最好输出检查一下是否取到了正确的值。

（3）点击事件无响应

原因：eclipse 没有找到点击操作的响应对象。

解决与经验：我的语法问题，比如“onclick=takePhoto()”漏了个括号。

六、项目的改进方向

1、提高图片的渲染速度。

测试的时候发现，有时候图片显示速度很慢，甚至直接挂掉，图片显示不出来，这个时候就需

要依靠刷新界面让界面恢复正常。百度了一下，这个大概和我 css 代码的编写不严谨有关系，以后可以考虑再深入学习 css 的样式设计，对其进行改进或者使用渲染速度快的界面显示框架。

2、功能的扩展、浏览器的兼容。

其实一开始是想做，直接从网页端调取摄像头并且拍照，上传到服务器进行图像处理的。写好了 Js 代码后，测试的时候发现，QQ 浏览器、Edge、火狐浏览器——居然只有火狐浏览器可以成功调用电脑的摄像头并且拍照。我的调用摄像头并拍照的代码没能兼容得了其他浏览器，因此我舍弃了这个功能更何況。而且由于安全性问题，火狐浏览器无法取得用户上传图片的绝对路径，加大了获取图像信息的难度，这也说明了我的代码没能够做到浏览器的兼容。因此后期可以考虑改进。

3、利用开源库自己实现人脸融合功能，不用联网也能运行。

腾讯优图虽然暂时免费，但是并不开源，融合的模板都是官方给的，如果需要自定义还需要向官网申请，十分麻烦，而且图像处理时必须联网。而且项目测试阶段接近尾声的时候，我去官网查看模板的文档，发现——该功能由于算法需调整，居然下线了！虽然服务仍然可以免费使用到 11 月 30 日，可是这个功能终究不是自己的，处于被动劣势，因此可以考虑利用其它的开源库替换优图的人脸融合功能，还能解决“必须联网”这一大缺点。

三、项目代码

1、JS 文件——GetPhoto.js

主要用于判断文件类型、获取、显示用户选取的本地图片。涉及到正则表达式匹配。

```
window.onload=function () {  
    //取得 jsp 的标签对象，为了取得 file 的路径  
    var file=document.getElementById('file');  
    //取得 jsp 的标签对象，为了显示选中 file 的图片  
    var img=document.getElementById('img');  
    var dataImg; //定义图片对象  
  
    file.onchange=function () {  
        //判断是否支持 FileReader  
        if(typeof FileReader==="undefined"){  
            //弹窗提示信息
```

```

        alert('您的浏览器不支持，最好使用 QQ 浏览器、微软 Edge 浏览器');
    }
    //读取文件
    var result=this.files[0];
    //获取文件类型
    var type=result.type;
    if(!type){
        alert('请上传图片');
    }else {
        //判断图片类型
        type=type.split('/')[1];
        console.log(type);

        //使用正则表达式匹配判断是否是 jpeg,jpg,png,bmp,gif 图片类型
        if(type.match(/^(jpg|bmp|png|jpeg|gif)$/g)){
            console.log(result);
            //声明一个 FileReader
            var reader=new FileReader();
            //以数据流的形式读取图片
            reader.readAsDataURL(result);
            //图片读取完毕后执行操作
            reader.onload=function (e) {
                //获取图片读取结果
                dataImg=this.result;
                //加载图到标签上
                img.setAttribute('src',dataImg);

            };
        }
        else {
            //弹窗提示
            alert("请选择正确的图片格式");
            //清空 input 的 value 值
            this.value="";
        }
    }
}
}
}

```

2、JS 文件——1b.js

主要用于展示轮播图所需要的代码逻辑。涉及到事件的监听。

```
let timer = null; //定时器（实现自动跳转）
```

```
let index = 0;           //索引有五个，分别定位了要显示的五张图（索引从 0~4）
//获取轮播片右下角的圆点导航对象
let dots = document.getElementsByTagName("span");
//事件的触发，表示资源已经加载完成,则执行 init 初始化操作
window.addEventListener('load', init, false);
```

//初始化能够操作的控件

```
function init() {
    //获取到 wrap（因为要设置其 left 才能控制轮播图）
    var wrap = document.querySelector(".wrap");
    var next = document.querySelector(".arrow_right"); //获取到左箭头
    var prev = document.querySelector(".arrow_left"); //获取到右箭头
    // 当鼠标点击左箭头
    next.onclick = function () {
        next_pic(wrap); //往左切图
    }
    prev.onclick = function () { // 当鼠标点击右箭头
        prev_pic(wrap); //往右切图
    }
    autoPlay(wrap); //启动定时器自动播放轮播图
    // 取到轮播片容器对象
    var container = document.querySelector(".container");
    //鼠标停在轮播片时
    container.onmouseenter = function () {
        clearInterval(timer); //取消自动播放
    }
    //鼠标离开轮播片时
    container.onmouseleave = function () {
        autoPlay(wrap); //自动播放
    }
}
```

//用户的上一页操作（图片向右移动）

```
function prev_pic(wrap) {
    index--; //索引定位到上一页
    if (index < 0) { //退到第一页再往前翻，则打开图片索引为 4 的图
        index = 4;
    }
    showCurrentDot(wrap); //显示当前图片
    var newLeft;
    //首页是从 left=-700 开始的，往前翻一页则为 0（图片宽 700）
    //往后则是-1400，所以往后四页的临界值为-2800，以此类推
```


//如果从第一页(实际显示 5.jpg)往前翻, 则回到第五页 (实际显示 4.jpg) 的 left 值
//不恒等, 判断数值和类型是否完全一致

```
if (wrap.style.left === "0px") {  
    //left 的临界值, container 中显示大于这个值则会出现空白  
    newLeft = -2800;  
} else { //如果是其他页往前翻, 则正常定位图片位置。  
    newLeft = parseInt(wrap.style.left) + 700;  
}  
wrap.style.left = newLeft + "px";    //加上单位  
}
```

//用户的下一页操作 (图片往左移动)

```
function next_pic(wrap) {  
    index++;           //索引定位到下一页  
    if (index > 4) {    //如果在最后一页往后一页。  
        index = 0;      //则索引定位回第一页  
    }  
    showCurrentDot(wrap); //显示当前图片    var newLeft;  
    //如果正在显示的是最后一页图片  
    if (wrap.style.left === "-4200px") {  
        //则回到第二页 (因为第一页和最后一页图片一样的)  
        newLeft = -1400;  
    } else {           //否则正常向左定位图片开始的位置  
        newLeft = parseInt(wrap.style.left) - 700;  
    }  
    wrap.style.left = newLeft + "px"; //加单位  
}
```

//定时器 , 实现自动播放

```
function autoPlay(wrap) {  
    timer = setInterval(function () { //表示隔两秒, 往下翻图  
        next_pic(wrap);  
    }, 2000);  
}
```

//显示当前是第几张图片

```
function showCurrentDot(wrap) {  
    for (var i = 0, len = dots.length; i < len; i++) {  
        dots[i].className = "";    //小圆点空心显示。  
    }  
    dots[index].className = "on";  //正显示图片对应的小圆点变为实心。
```

```
}
```

//圆点导航代码，点击圆点切换到不同的图片，圆点导航的圆点定位原理和轮播图类似。

```
for (var i = 0, len = dots.length; i < len; i++) {  
    (function (i) {  
        //利用选择器找到 css 中的 wrap,即小圆点  
        let wrap = document.querySelector(".wrap");  
        //设置每个小圆点的点击监听事件  
        dots[i].onclick = function () {  
            //获取到显示小圆点的索引值  
            var dis = index - i;  
            //圆点原理和轮播图一致，获取到相对镜头的 left 值即可  
            //如果显示小圆点处于最后一个，向后操作  
            if (index == 4 && parseInt(wrap.style.left) != -3500) {  
                dis = dis - 5; //则返回到第一个小圆点的索引（代表最后一张图）  
            }  
            //如果显示小圆点处于第一个，向前操作  
            if (index == 0 && parseInt(wrap.style.left) != -700) {  
                dis = 5 + dis; //则直接跳到最后一个圆点索引（代表最后一张图）  
            }  
            //设置显示小圆点的新 left 值  
            wrap.style.left = (parseInt(wrap.style.left) + dis * 700) + "px";  
            index = i; //获取显示小圆点当前的位置  
            showCurrentDot(wrap); //将显示小圆点变成实心显示  
        }  
    })(i);  
}
```

3、显示页面——index.jsp

包括 css 代码和常规的 html 标签、jsp 代码，是主要展示界面。

```
<!DOCTYPE html>  
<html>  
<head>  
    <meta charset="UTF-8">  
    <title>FaceWrap</title>  
    <link rel="stylesheet" href="style.css">  
    <!-- 引入 Js 文件 -->  
<script src="js/lb.js"></script>  
<script src="js/GetPhoto.js"></script>  
<style type="text/css">  
/* 重置 css */  
* {  
    margin:0;
```

```
padding:0;
/* background-color: orange; */
}
a{
text-decoration: none;
}
/*
```

让图片只在 container 中，所以需要限定其宽度和高度
box-shadow 是图片边缘阴影
使用 overflow:hidden 将其余的图片隐藏起来
希望 wrap 相对于 container 左右移动，所以设置为 relative

```
*/
.container {
position: relative;
width: 700px;
height: 900px;
/* margin:100px auto 0 auto; */
margin-top:50px;
box-shadow: 0 0 5px rgb(180, 182, 180);
overflow: hidden;
}
/*
```

设置 wrap 是绝对定位的(就可以通过控制 Left 和 Right 来控制图片的移动了)
z-index:1;为了对后面将要放置的 buttons 作为参考
因为共有七张图片，所以 width 为 7*700=4900px（每张图片我们设置为 700X900

```
*/
.container .wrap {
position: absolute;
width: 4900px;
height: 900px;
z-index: 1;
}
/* 把图片设置位左浮动，并限定其大小 */
```

```
.container .wrap img {
float: left;
width: 700px;
height: 900px;
}
/* 显示次序的 buttons（小圆点）
```

设置 z-index:2;以保证 buttons 是在图片的上面的

```
*/
.container .buttons {
position: absolute;
right: 5px;
```

```

    bottom:30px;
    width: 120px;
    height: 10px;
    z-index: 2;
}
/*将 buttons 下面的 span 做一个简单的修饰, 并且给和图片对应的 span 设置一个 on 类,  */
.container .buttons span {
    margin-left: 5px;
    display: inline-block;
    width: 12px;
    height: 12px;
    border-radius: 50%;
    background-color: rgb(211, 212, 211);
    text-align: center;
    color:white;
    cursor: pointer;
}
.container .buttons span.on{
    background-color: rgb(151, 150, 150);
}
/*左右切换的箭头,做简单的修饰
    因为这里使用实体来表示左右箭头, 所以设置 font-size 才能改变其大小,
    */
.container .arrow {
    position: absolute;
    top: 40%;
    color: rgb(179, 182, 179);
    padding:0px 15px;
    border-radius: 50%;
    font-size: 50px;
    z-index: 2;
    display: none;
}
.container .arrow_left {
    left: 10px;
}
.container .arrow_right {
    right: 10px;
}
.container:hover .arrow {
    display: block;
}
.container .arrow:hover {
    background-color: rgba(0,0,0,0.2);
}

```



```

}
/* 图像的宽高设置 */
.img{
    width: 600px;
    height: 600px;
    margin-top:100px;
    border: 1px solid lightgray;
}
/* 如果图片比框大，则限制成原格式大小来 */
.img>img{
    width: 100%;
    height: 100%;
}
/*轮播片父布局的边框样式 */
.div-b{ float:left;width:49%;border:1px solid #000}
/* 预览图片的样式设置 */
.right {
    margin-left: 1010px;
    height: 900px;
}
</style>
</head>
<body>
    <%
//初始化轮播图，用已有的图片实现轮播
if(request.getAttribute("pic1")==null)
    request.setAttribute("pic1","images/libai.jpg");
if(request.getAttribute("pic2")==null)
    request.setAttribute("pic2","images/zhenji.jpg");
if(request.getAttribute("pic3")==null)
    request.setAttribute("pic3","images/junzhuang.jpg");
if(request.getAttribute("pic4")==null)
    request.setAttribute("pic4","images/linger.jpg");
if(request.getAttribute("pic5")==null)
    request.setAttribute("pic5","images/wangzhaojun.jpg");

%>
<div class="div-b">
    <center>
    <div class="container" >
        <div class="wrap" style="left: -700px;">
            <img src=${pic5} alt="5"><!-- 最后一页 -->
            <img src=${pic1} alt="1"><!-- 第一页 -->
            <img src=${pic2} alt="2">    <!-- 2 -->

```

```

        <img src=${pic3} alt="3">      <!-- 3 -->
        <img src=${pic4} alt="4">      <!-- 4 -->
        <img src=${pic5} alt="5">      <!-- 最后一页（用于轮播实现显示最后一页的图片）
-->

        <img src=${pic1} alt="1"><!-- 第一页的图（用于轮播实现第一页的图片） -->
    </div>
    <div class="buttons">    <!-- 小圆点 -->
        <span class="on"></span>
        <span></span>
        <span></span>
        <span></span>
        <span></span>
    </div>
    <!-- 左右点击 -->
    <a href="javascript:void (0);" rel="external nofollow" class="arrow arrow_left"><</a>
    <a href="javascript:void (0);" rel="external nofollow" class="arrow arrow_right">></a>
</div>
</center>
</div>
<!-- 预览图的提交、显示 -->
<div class="right">
<center><br><br><br>
<form action="run" method="post">
    <input type="file" id="file" name="path" margin-top="100px">
    <p class="img" >
        <img src="" name="ipath" alt="" id="img">
    </p><br><br><br>
    <input type="submit" value="submit">
</form>
</center>
</div>
</body>
</html>

```

4、jsp 报错提醒界面——error.jsp

如果出现人脸融合失败，则提示报错号码并且跳转到官网查询错误信息

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

```

```

<title>报错啦</title>
</head>
<body>
    出现了一些错误，错误码为${returnPath}!
    5 秒后跳转至官网查询错误信息！
    <%
        response.setHeader("refresh","5;url=https://cloud.tencent.com/document/product/670/14357");
    %>
</body>
</html>

```

5、java 文件——Demo. java

主要用于向优图发送人脸融合效果的请求实现，获取优图的响应结果。并且返回响应的 url 给 index.jsp 预览。

```

public class Demo {
    private String path=""; //用户上传的资源路径
    private String errorid=""; //返回报错码
    private String pic1="",pic2="",pic3="",pic4="",pic5=""; //轮播图的五张图路径

    //优图的配置信息
    public static final String APP_ID = "10116240";
    public static final String SECRET_ID = "AKIDFNGtJTRyKQHsCFdaqZ4KNAhy2sVZ55v";
    public static final String SECRET_KEY = "kHafW9YjhRxFKAmAKGySikYQ4UgdMsAD";
    public static final String USER_ID = "850161565";
    public String main() {
        try {
            //调用人脸融合功能的代码
            Youtu faceYoutu = new Youtu(APP_ID, SECRET_ID,
SECRET_KEY,Youtu.API_YOUTU_MY_END_POINT,USER_ID);
            JSONObject response;
            response = faceYoutu.FaceFuse(getPath(),"cf_wzry_libai"); //李白
            pic1=(String) response.get("img_url");
            response = faceYoutu.FaceFuse(getPath(),"cf_wzry_zhenjiImage"); //甄姬
            pic2=(String) response.get("img_url");
            response =
faceYoutu.FaceFuse(getPath(),"hezuo_junzhuangzhao_1948m_20170919140951"); //军装
            pic3=(String) response.get("img_url");

```

```

        response = faceYoutu.FaceFuse(getPath(),"cf_xianjian_linger");    //赵灵儿
        pic4=(String) response.get("img_url");
        response = faceYoutu.FaceFuse(getPath(),"cf_wzry_wangzhaojun");    //王昭君
        pic5=(String) response.get("img_url");
        errorid=(String)response.get("ret");
        System.out.println(response ); //后台查看响应
        return "success";
    } catch( Exception e) {
        e.printStackTrace();
    }

    return "input";
}

```

//剩下的 get\set 方法是为了结合 structs 使用的。

```

    public String getPath() {
        return path;
    }

    public void setPath(String path) {
        this.path = path;
    }

    public String getPic2() {
        return pic2;
    }

    public void setPic2(String pic2) {
        this.pic2 = pic2;
    }

    public String getPic1() {
        return pic1;
    }

    public void setPic1(String pic1) {
        this.pic1 = pic1;
    }

    public String getPic3() {
        return pic3;
    }

```



```

    }
    public void setPic3(String pic3) {
        this.pic3 = pic3;
    }

    public String getPic4() {
        return pic4;
    }
    public void setPic4(String pic4) {
        this.pic4 = pic4;
    }
    public String getPic5() {
        return pic5;
    }
    public void setPic5(String pic5) {
        this.pic5 = pic5;
    }
    public String getErrorid() {
        return errorid;
    }
    public void setErrorid(String errorid) {
        this.errorid = errorid;
    }
}

```

6、配置信息——struts.xml

主要用于实现前台 jsp 页面与后台 java 代码的跳转与信息交互操作。

当 java 代码正常运行，则跳回 index.jsp 显示页面，否则跳转到 error.jsp 报错页。

```

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE struts PUBLIC
    "-//Apache Software Foundation//DTD Struts Configuration 2.3//EN"
    "http://struts.apache.org/dtds/struts-2.3.dtd">
<struts>
    <package name="default" extends="struts-default">
        <action name="run" class="com.test.Demo" method="main" >

```

```
        <result name="success">/index.jsp</result>
        <result name="input">/error.jsp</result>
    </action>
</package>
</struts>
```

7、java 包——com.youtu、com.youtu.sign

主要用于处理具体的 json 数据发送以及官方文档中必要的签名配置等信息的传递，和 js 关系不大，因此省略其中代码。