

编码与测试报告

1. 引言

1.1 编写目的

此份软件需求分析报告描述了“情绪社区”系统的编码与测试。通常把编码和测试统称为实现。所谓编码就是把软件设计结果翻译成用某种程序设计语言书写的程序。作为软件工程过程的一个阶段，编码是对设计的进一步具体化，因此，程序的质量主要取决于软件设计的质量。但是，所选用的程序设计语言的特点及编码风格也将对程序的可靠性、可读性、可测试性和可维护性产生深远的影响。

1.2 文档约定

本文档按照下列要求和约定进行书写：

- (1) 标题分为四级，分别为宋体二号、宋体三号、宋体四号、宋体五号；
- (2) 正文字体为宋体小四；
- (3) 标题字体均为黑色加粗，正文字体均为黑色不加粗。

1.3 预期读者和阅读建议

- ❖ 用户
- ❖ 项目负责人
- ❖ 系统维护员
- ❖ 程序员

在阅读本文档前，需要对小组项目“情绪社区”的功能大致了解，读者可根据自身需求选择性阅读。

1.4 适用范围

本软件主要适用于高校在校人员，在使用方法上考虑了简便性，对操作要求不高，只要同学熟悉基本的上传视频浏览贴吧发表帖子和评论，对计算机有基本的操作规范认识和了解，就可以完成阅读。

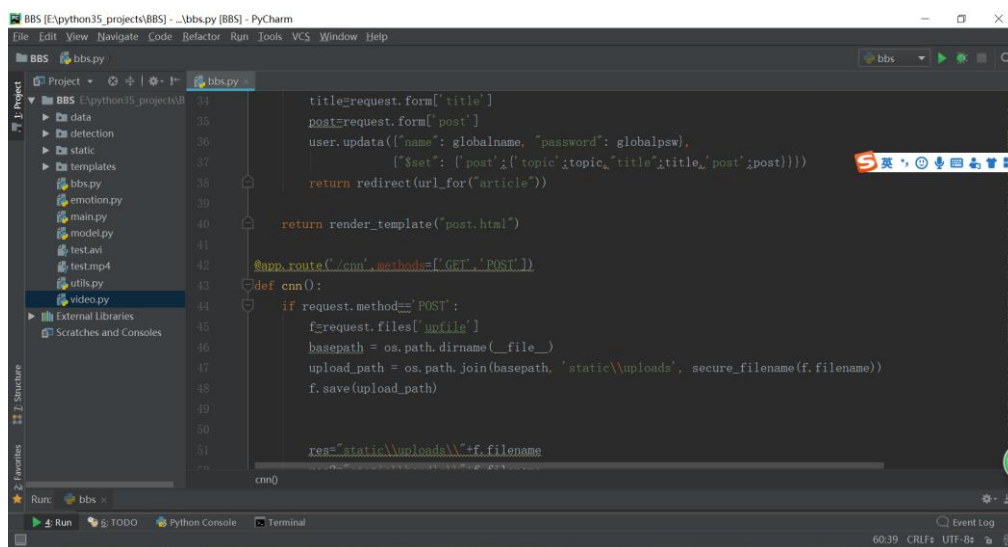
1.5 参考文献

- 《软件工程导论》 张海藩 等 清华大学出版社

2. 编码

2.1 程序设计语言

由于深度学习框架以及 web 框架 flask 都是基于 python 语言，所以本系统的程序设计语言也是 python.



2.2 编码风格

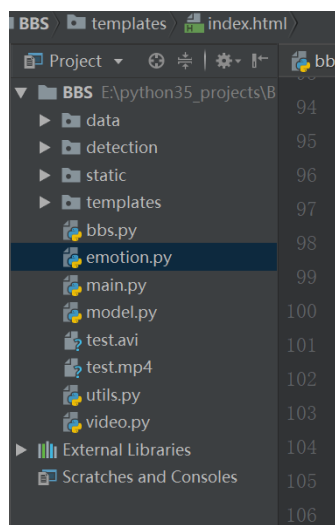
2.2.1 程序内部文档

程序内部文档是指恰当的标识符，适当的注释和程序的视觉组织等。在源码中可以读到。

2.2.2 输入输出

- 1) 对所有输入数据都会检验，包括视频数据，文字数据等
- 2) 输入格式简单，采用触屏按钮模式，十分简单
- 3) 本系统使用了数据结束标记，不用用户指定数据的个数
- 4) 本系统的输出报表是采用时序图模式，可读性很好，设计良好。

2.3 源程序结构



| | |
|------|-------|
| 文件 | 解释 |
| Data | 训练测试集 |

| | |
|-----------|---------------------------------|
| Detection | 权重参数模型 |
| Static | 静态文件（css/js/image）和上传文件以及处理后的文件 |
| Templates | Html 模板文件 |
| Bbs.py | Web 主模块，运行本模块即可 |
| Main.py | 算法模块 |
| Model.py | CNN 模型模块 |
| Utils.py | 功能函数模块 |
| Video.py | 后端处理视频模块 |

3. 测试

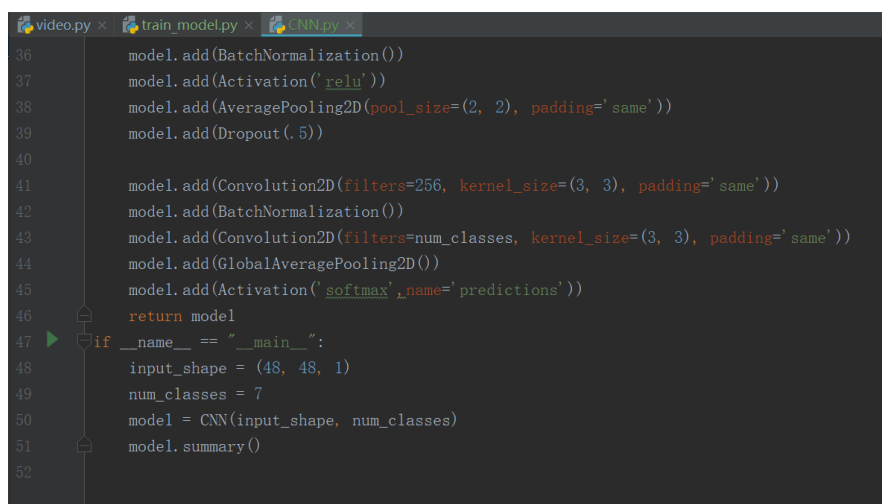
软件测试在软件生命周期中横跨两个阶段。通常在编写出每个模块之后就对它做必要的测试(称为单元测试)，模块的编写者和测试者是同一个人，编码和单元测试属于软件生命周期的同一个阶段。在这个阶段结束之后，对软件系统还应该进行各种综合测试，这是软件生命周期中的另一个独立的阶段，通常由专门的测试人员承担这项工作。

在本项目中，主要采用了单元测试，白盒测试，黑盒测试等技术。

3.1 单元测试

3.1.1 模块测试

对模块函数的接口数据流进行测试，检查数据是否正确进出，主要检查以下几个方面：形参，次序，属性，单位系统与变元等等，代码见源码的 test.py 文件，如下图，在 main 函数内测试了 CNN 模型



```

36     model.add(BatchNormalization())
37     model.add(Activation('relu'))
38     model.add(AveragePooling2D(pool_size=(2, 2), padding='same'))
39     model.add(Dropout(.5))
40
41     model.add(Convolution2D(filters=256, kernel_size=(3, 3), padding='same'))
42     model.add(BatchNormalization())
43     model.add(Convolution2D(filters=num_classes, kernel_size=(3, 3), padding='same'))
44     model.add(GlobalAveragePooling2D())
45     model.add(Activation('softmax', name='predictions'))
46     return model
47
48 if __name__ == '__main__':
49     input_shape = (48, 48, 1)
50     num_classes = 7
51     model = CNN(input_shape, num_classes)
52     model.summary()

```

3.1.2 局部数据测试

局部的数据测试同样也见源文件的 test.py.

```

6  def main():
7      f=input("please input you choose/n")
8      if f == "demo":
9          run_demo()
10     elif f == "model":
11         a1=input("train?valid?")
12         if a1=="train":
13             train_model()
14         else:
15             valid_model()
16
17     else:
18         print("usage: python3 main.py <function>")
19
20 if __name__ == '__main__':
21     main()

```

3.1.3 出错处理

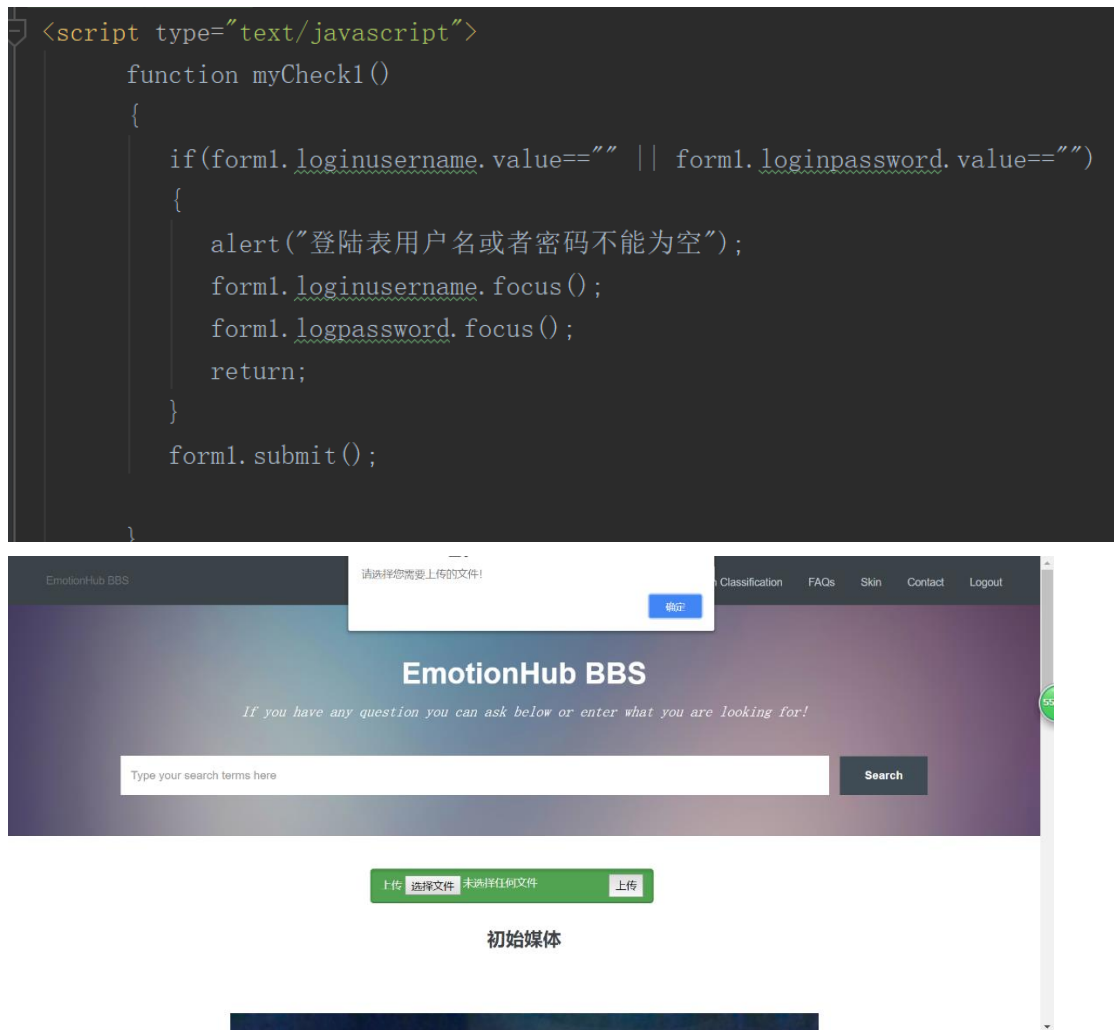
每种错误都有对应的错误通路页面显示，或者弹窗提示。

```

@app.route('/error/')
def error():
    return render_template("fornone.html")
@app.route('/results/',methods=['GET', 'POST'])
def results():
    global search_input
    global path
    if request.method=="POST":
        return redirect(url_for('index'))
    # print("哈哈哈哈哈")

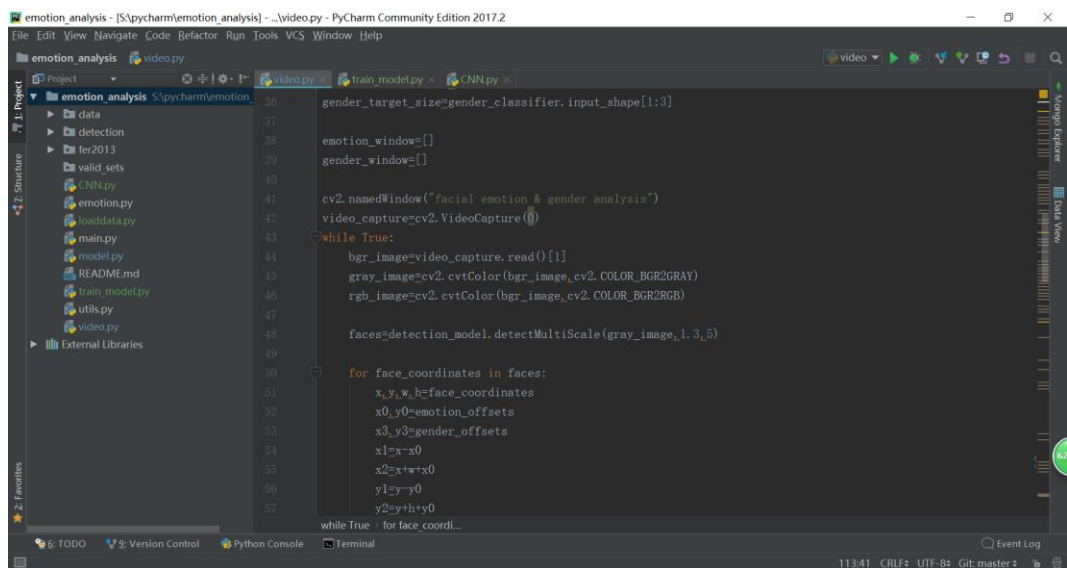
```

在 html 页面中嵌入 js 程序，进行检验与弹窗



3.1.4 边界条件

如下图所示，在读取视频的时候，对于第一帧图像和最后一帧图像会进行边界测试。



数据边界检测

```

if detected_face is not None:
    cv2.imwrite('a.jpg', detected_face)
    tensor = image_to_tensor(detected_face)
    result = sess.run(probs, feed_dict={x: tensor})
    # print(result)
if result is not None:
    for index, emotion in enumerate(EMOTIONS):
        cv2.putText(frame, emotion, (10, index * 20 + 20), cv2.FONT_HERSHEY_PLAIN, 0.5, (0, 255, 0), 1)
        cv2.rectangle(frame, (130, index * 20 + 10), (130 + int(result[0][index] * 100), (index + 1) * 20 + 4),
                      (255, 0, 0), -1)
    emoji_face = feelings_faces[np.argmax(result[0])]

```

3.2 白盒测试

本系统的白盒测试主要采用逻辑覆盖的方法：

3.2.1 语句覆盖

3.2.2 判定覆盖

3.2.3 条件覆盖

3.2.4 条件组合覆盖

等等测试都在源码中的注释部分。

3.3 黑盒测试

3.3.1 登陆验证

| 输入等价类 | 测试用例 | 测试数据 | 期望结果 | 实际结果 |
|-------|------------------|--------------------------|-------|----------------------|
| 无效等价类 | 用户名：空 密码：空 | 用户名： 密码： | 登陆不成功 | 弹出出错信息，提示用户名和密码输入错误。 |
| 无效等价类 | 用户名：空 密码：非空 | 用户名： 密码：1 | 登陆不成功 | 弹出出错信息，提示用户名输入错误。 |
| 无效等价类 | 用户名：非空 密码：空 | 用户名：rr 密码： | 登陆不成功 | 弹出出错信息，提示密码输入错误。 |
| 无效等价类 | 用户名：正确 密码：不正确 | 用户名：lishuai 密码：1 | 登陆不成功 | 弹出出错信息，提示密码输入错误。 |
| 无效等价类 | 用户名：不正确 密码：正确 | 用户名：li 密码：321321 | 登陆不成功 | 弹出出错信息，提示用户名输入错误。 |
| 有效等价类 | 用户名：正确 密码：正确 | 用户名：lishuai 密码：321321 | 登陆成功 | 进入导航页。 |

3.3.2 视频分析验证

| 输入等价类 | 测试用例 | 测试数据 | 期望结果 | 实际结果 |
|-------|----------|------------|---------|---------|
| 无效等价类 | 长度不合格的视频 | 500M 大小的视频 | 弹窗提示不合法 | 弹窗提示不合法 |

| | | | | |
|-------|---------|-------------|----------|-----------|
| 无效等价类 | 不含人脸的视频 | 70M 大小的风景视频 | 转到视频检测页面 | 转到视频检查页面 |
| 有效等价类 | 自拍视频 | 30M 大小的自拍视频 | 反馈页面 | 视频分析的反馈页面 |

4. 测试结论

经过大量测试，本软件系统具有一定的可靠性与可用性，足以适用各种浏览器以及访问并发情况