**软件测试说明书**

**测试系统：公告地点人流量的云监管平台（0101组）**

**小组名称：Leet Code**

**参与人员：黄新昊 陈硕峰**

**2019 年 6 月 30日**

目录

[1引言 3](#_Toc9408)

[1.1系统概述 4](#_Toc26872)

[1.2文档概述 4](#_Toc5179)

[1.3测试范围 4](#_Toc15841)

[2引用文件 5](#_Toc17682)

[3测试结果概述 5](#_Toc20419)

[3.1总体评估 5](#_Toc2255)

[3.2测试环境与配置 10](#_Toc25729)

[4详细测试结果 12](#_Toc18015)

[4.1单元测试 12](#_Toc26006)

[4.1.1模型加载模块 12](#_Toc3694)

[4.1.2图片加载模块 13](#_Toc16795)

[4.1.3图片定位模块 16](#_Toc7015)

[4.1.4图片处理模块 21](#_Toc32442)

[4.1.5图片展示模块 25](#_Toc25226)

[4.1.6视频处理模块 29](#_Toc11756)

[4.1.7行人移动位置预测模块 33](#_Toc18116)

[4.1.8目标跟踪模块 34](#_Toc11509)

[4.1.9视频识别结果保存模块 38](#_Toc17190)

[4.2界面测试 42](#_Toc21884)

[4.3集成测试 43](#_Toc21035)

[4.3.1行人标识记录 44](#_Toc27064)

[4.3.2行人移动位置预测 46](#_Toc16440)

[4.3.3行人目标跟踪 47](#_Toc21366)

[4.3.4视频识别结果保存模块 49](#_Toc19183)

[4.4有效性测试 49](#_Toc12440)

[4.4.1测试功能点 49](#_Toc15441)

[4.4.2测试项目 51](#_Toc16439)

[4.4.3测试内容 52](#_Toc26566)

[4.4.4覆盖分析 61](#_Toc7682)

[4.4.5缺陷汇总及分析 64](#_Toc24703)

[4.5系统测试 65](#_Toc26021)

[4.5.1测试图表 65](#_Toc15312)

[4.5.2测试内容 67](#_Toc1907)

[5测试记录 6](#_Toc3522)9

[6评价 70](#_Toc6869)

[6.1能力 70](#_Toc14407)

[6.1.1功能 70](#_Toc25567)

[6.1.2性能 70](#_Toc8065)

[6.1.3易用性 71](#_Toc4815)

[6.1.4安全性 71](#_Toc25083)

[6.2缺陷和限制 71](#_Toc26146)

[6.3建议 75](#_Toc32216)

[6.4结论 77](#_Toc7848)

[7附录 78](#_Toc18196)

[测试工具介绍 78](#_Toc9511)

[测试程序 78](#_Toc23830)

# 1引言

编写该测试总结报告主要有以下几个作用：

1. 通过对测试结果的分析，得到对软件质量的评价。  
   2.分析测试的过程，产品，资源，信息，为以后制定测试计划提供参考。

3.评估测试测试执行和测试计划是否符合。  
4.分析系统存在的缺陷，为修复和预防bug提供建议 。

## 1.1系统概述

公共地点人流量实时计算的云监管平台根据校园大门、学校食堂、大型商超、政府大门、交通枢纽几个边缘地点拍摄的视频,通过机器视觉的技术,识别提取地点特征和人头数；边缘端程序对人头数做计算，如果人头数没有达到容量上限则数据不上报到云端，若人头数超出预置容量上限数，就将所在地点信息和人头数上报到云监管平台，云监管平台解析相关数据,界面呈现对应地点和人头数爆发的数据告警,边缘端人流数据实时变化,云端展示人流量数据也实时变化,如果人流量实时减少并小于阈值,报警自动解除;边缘端程序和云监管平台程序要求支持独立部署不同服务器,满足边缘计算云边协同架构的实际需求。

## 1.2文档概述

为了保证系统的各项功能可靠的实现，特编写此测试说明书。本文档作为测试说明书用于指导公共地点人流量计算的云监管平台项目开发，文档记录了测试过程中公共地点人流量计算的云监管平台项目的目标、运行环境、用户、数据精确度、查全率、时间特性、安全性、容错性、接口、数据、保密性、硬件需求、软件需求、通信需求、培训需求、后勤需求以及故障处理等各方面，并提出了修改建议。

## 1.3测试范围

依据用户需求说明书和软件需求规格说明书以及相应的设计文档进行系统测试，包括功能测试、性能测试、用户访问与安全控制测试、用户界面测试。主要功能包括:

1. 正确识别3个以上人数密集边缘区域的地点位置信息和人流量（即人头数）,并做相应排序；
2. 边缘端程序和云监管平台两个程序分开独立部署,满足边缘计算云边协同架构的实际需求;
3. 边缘端程序识别人头数和地点信息,并做相关计算,如果人头数没有达到容量上限则数据不上报到云端，若人头数超出预置容量上限数，就将所在人流异常视频数据、地点信息和人头数上报到云监管平台；
4. 云监管平台呈现对应地点和人头数爆发的数据告警,边缘端人流数据实时变化,云端展示人流量数据也实时变化,如果人流量实时降低并小于阈值,报警自动解除;
5. 异常视频中用动态标注的方式标注人头序号,并识别地点信息和人头总数和告警；
6. 异常视频数据、地点信息和人头数数据在云端进行数据存储,云监管平台可以查询; 对多个异常地点进行数据分析，并分类，排序重点突出经常容易出现人流异常爆发的地点；
7. 云监管平台实时展示地点信息和人头数数据，智能匹配给出疏导处理的相关保安组织人员推荐信息和联系方式;
8. 通知支持手机端和web端良好体验

# 2引用文件

1. 计算机软件文档编制规范GB/T 8567-2006
2. 贾铁军等.软件工程与实践[M].北京:清华大学出版社,2017
3. 王珊等.数据库系统概论[M].北京:高等教育出版社,2014
4. 朱少民等《软件项目管理》[M].北京:人民邮电出版社,2015
5. [康一梅](https://baike.baidu.com/item/%E5%BA%B7%E4%B8%80%E6%A2%85" \t "https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6%E9%A1%B9%E7%9B%AE%E7%AE%A1%E7%90%86/_blank)等《软件项目管理》[M].北京:清华大学出版社,2014
6. 《公共地点人流量计算的云监管平台软件需求说明书》
7. 《公共地点人流量计算的云监管平台软件开发计划书》
8. 《公共地点人流量计算的云监管平台软件设计说明书》

# 3测试结果概述

## 3.1总体评估

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 检测说明 | | | 检测结果 | | |
| 一、可靠性 | | | | | | |
| 1 | 软件应按照用户文档定义的可靠性来执行。 | | | F | | |
| 2 | 与差错处置相关的功能与产品说明和用户文档集中的陈述一致。 | | | F | | |
| 3 | 在用户文档集中陈述的限制范围内使用，软件不应丢失数据。 | | | F | | |
| 4 | 软件应识别无效的输入，并且不应作为许可的输入加以处理。 | | | F | | |
| 5 | 软件应该有异常处理机制。 | | | F | | |
| 6 | 软件应具有从致命错误中恢复的能力。 | | | F | | |
| 7 | 是否规定响应时间和精度 | | | F | | |
| 二、易用性 | | | | | | |
| 1 | 用户在看到产品说明或者第一次使用软件后，应确认产品或系统是否符合其需要 | | | P | | |
| 2 | 软件执行的结果、消息都应是易理解的 | | | F | | |
| 3 | 每个软件出错消息应指明如何改正差错或向谁报告差错 | | | F | | |
| 4 | 操作界面对非开发者用户友好 | | | F | | |
| 三、一致性 | | | | | | |
| 1 | 能否输出视频中的总人数和行人密度 | | | P | | |
| 2 | 能否对图片中的人标注人头序号 | | | P | | |
| 3 | 能否对视频中的人标注人头序号 | | | F | | |
| 4 | 能否对视频信息、人头数据进行存储 | | | F | | |
| 5 | 边缘端程序识别地点信息 | | | F | | |
| 6 | 是否具有云端程序 | | | F | | |
| 7 | 是否具有图形接口 | | | F | | |
| 8 | 是否符合通信接口需求 | | | F | | |
| 四、测试项目 | | | | | | |
| 序号 | 检测项目 | 检测说明 | 检测结果 | | | |
| 陈硕峰 | | 黄新昊 | |
| 1 | 加载模型，将训练好的模型加载到内存中 | | | | | |
| 1.1 | 能否将模型加载到内存中 | 启动测试函数，记录测试结果 | P | | | P |
| 2 | 图片加载，定义session加载待测试的图片文件。 | | | | | |
| 2.1 | 是否可以正确加载jpg格式的图片文件 | 选择jpg格式的图片，启动测试函数，记录测试结果 | P | | | P |
| 2.2 | 是否可以正确加载png格式的图片文件 | 选择png格式的图片，启动测试函数，记录测试结果 | P | | | P |
| 2.3 | 是否可以正确加载gif格式的图片文件 | 选择gif格式的图片，启动测试函数，记录测试结果 | F | | | F |
| 2.4 | 是否可以正确加载bmp格式的图片文件 | 选择bmp格式的图片，启动测试函数，记录测试结果 | P | | | F |
| 2.5 | 测试能否加载各种不同命名的图片文件 | 使用不同命名的.jpg图片 | P | | | P |
| 3 | 对各种常见格式图片进行目标检测定位。 | | | | | |
| 3.1 | 是否可以正确加载jpg格式的图片且可以实现目标检测定位 | 选择jpg格式的图片测试系统是否正确处理 | P | | | P |
| 3.2 | 是否可以正确加载png格式的图片且可以实现目标检测定位 | 选择png格式的图片测试系统是否正确处理 | P | | | P |
| 3.3 | 是否可以正确加载gif格式的图片且可以实现目标检测定位 | 选择gif格式的图片测试系统是否正确处理 | F | | | F |
| 3.4 | 是否可以正确加载bmp格式的图片且可以实现目标检测定位 | 选择bmp格式的图片测试系统是否正确处理 | P | | | F |
| 3.5 | 能否正确识别所有图片中的人 | 选择人数3至10人、10人至50人、50人以上的图片分别进行测试 | F | | | F |
| 3.6 | 能否对图片中的人标注人头序号 | 选择人数3至10人、10人至50人、50人以上的图片分别进行测试 | P | | | P |
| 4 | 对各种常见格式的图片进行处理。 | | | | | |
| 4.1 | 是否可以正确处理jpg格式的图片 | 选择jpg格式的图片测试其是否可以正确处理 | P | | | P |
| 4.2 | 是否可以正确处理jpeg格式的图片 | 选择jpeg格式的图片测试其是否可以正确处理 | P | | | P |
| 4.3 | 是否可以正确处理png格式的图片 | 选择png格式的图片测试其是否可以正确处理 | P | | | P |
| 4.4 | 是否可以正确处理gif格式的图片 | 选择gif格式的图片测试其是否可以正确处理 | F | | | F |
| 4.5 | 是否可以正确处理bmp格式的图片 | 选择bmp格式的图片测试其是否可以正确处理 | P | | | F |
| 5 | 图片展示，显示处理后的图片结果 | | | | | |
| 5.1 | 是否可以正确显示jpg格式的图片 | 选择jpg格式的图片测试其是否可以正常显示 | P | | | P |
| 5.2 | 是否可以正确显示jpeg格式的图片 | 选择jpeg格式的图片测试其是否可以正常显示 | P | | | P |
| 5.3 | 是否可以正确显示png格式的图片 | 选择png格式的图片测试其是否可以正常显示 | P | | | P |
| 5.4 | 是否可以正确显示gif格式的图片 | 选择gif格式的图片测试其是否可以正常显示 | F | | | F |
| 5.5 | 是否可以正确显示bmp格式的图片 | 选择bmp格式的图片测试其是否可以正常显示 | P | | | F |
| 6 | 视频处理 | | | | | |
| 6.1 | 系统是否可以正确处理mp4格式的视频 | 选择mp4格式的视频测试系统是否正确处理 | F | | | F |
| 6.2 | 系统是否可以正确处理avi格式的视频 | 选择avi格式的视频测试系统是否正确处理 | F | | | F |
| 6.3 | 系统是否可以正确处理mov格式的视频 | 选择mov格式的视频测试系统是否正确处理 | F | | | F |
| 6.4 | 系统是否可以正确处理gif格式的视频 | 选择gif格式的视频测试系统是否正确处理 | F | | | F |
| 6.5 | 系统是否可以正确加载各种命名方式的视频 | 选择各种不同命名方式的视频测试 | F | | | F |
| 7 | 行人移动位置预测 | | | | | |
| 7.1 | 算法是否提供正确中间结果 | 测试行人移动位置预测模块是否提供正确中间结果 | P | | | P |
| 8 | 视频行人目标标记与跟踪 | | | | | |
| 8.1 | 系统是否可以跟踪mp4格式视频中的目标 | 选择mp4格式的视频测试 | P | | | P |
| 8.2 | 系统是否可以跟踪avi格式视频中的目标 | 选择avi格式的视频测试 | P | | | P |
| 8.3 | 系统是否可以跟踪mov格式视频中的目标 | 选择mov格式的视频测试 | P | | | P |
| 8.4 | 系统是否可以跟踪gif格式视频中的目标 | 选择gif格式的视频测试 | F | | | F |
| 8.5 | 系统能否对视频中行人进行标记并跟踪 | 分别选择mp4、avi、mov格式视频进行测试 | P | | | P |
| 8.6 | 系统能否对视频中的人标注人头序号 | 分别选择mp4、avi、mov格式视频进行测试 | F | | | F |
| 9 | 视频识别结果输出与保存 | | | | | |
| 9.1 | 能否将视频信息、总人数和行人密度保存到数据库中 | 分别选择mp4、avi、mov格式视频进行测试 | F | | | F |
| 9.2 | 能否识别视频地点信息 | 分别选择mp4、avi、mov格式视频进行测试 | F | | | F |
| 9.3 | 若人数超过上限能否向云端上报相关信息 | 分别选择mp4、avi、mov格式视频进行测试 | F | | | F |
| 9.4 | 是否使用AES加密算法保护数据 | 分别选择mp4、avi、mov格式视频进行测试 | F | | | F |

## 3.2测试环境与配置

软件列表

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 版本号 | 用途 |
| 1 | Windows10 | 家庭中文版 | 检测用机1操作系统 |
| 2 | Chrome | 75.0.3770.1（32 位） | 浏览器应用软件 |
| 3 | MySQL | 5.4.0 | 数据库系统 |
| 4 | WPSOffice | 2019 | 办公软件 |
| 5 | JetBrains PyCharm | 2018.1.4 x64 | Python编程开发工具 |
| 6 | 360安全卫士 | 11.5 | 安全应用软件 |
| 7 | 网易有道词典 | 8.1.1.0 | 翻译应用软件 |
| 8 | Acrobat Reader DC | 7.2.13.3882 | 文档阅读应用软件 |
| 9 | CAJViewer | 7.2 | 文档阅读应用软件 |
| 10 | Xshell | 6 | 安全终端模拟软件 |
| 11 | 迅雷 |  | 文件下载应用软件 |
| 12 | TeamViewer | 13 | 安全终端模拟软件 |
| 13 | 微信web开发者工具 | 1.02.1904090 | 微信开发工具 |
| 14 | Eclipse | 4.3.2 | java编程开发工具 |
| 15 | Dev-C++ | 5.11 | C语言编程开发工具 |
| 16 | Firefox | 66.0.5（64位） | 浏览器应用软件 |
| 17 | QQ | 9.1.3.2.25532 | 社交应用软件 |
| 18 | 微信 | 7.0.4 | 社交应用软件 |

硬件列表

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 型号 | 用途 |
| 1 | 笔记本 | Shinelon notebook | 检测用机1 |
| 1 | 笔记本 | 惠普（HP）暗夜精灵 | 检测用机1 |
| 3 | 手机 | ONE PLUS 5 | 检测用机3 |
| 4 | 手机 | iPhone 6 | 检测用机4 |
| 5 | 服务器 | 腾讯云 | 服务器 |

第三方库

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 版本 | 用途 |
| 1 | Numpy | 1.16.2 | 用来存储和处理大型矩阵 |
| 2 | Moviepy | 1.0.0 | 用于视频编辑的python模块 |
| 3 | Opencv-python | 4.0.0.21 | 实现了图像处理和计算机视觉方面的很多通用算法 |
| 4 | Pillow | 5.2.0 | Python平台图像处理标准库 |
| 5 | Matplotlib | 2.2.3 | Python 的绘图库 |
| 6 | Tensorflow | 1.1.0 | 于各类机器学习算法的编程实现 |
| 7 | PyQt5 | 5.12.2 | 用来编写Python脚本的应用界面 |
| 8 | Imageio | 2.5.0 | [读取照片RGB内容，转换照片格式)](https://www.cnblogs.com/devilmaycry812839668/p/9344581.html) |
| 9 | Pykalman | 0.9.5 | 算法库 |
| 10 | opencv-contrib-python | 4.1.0.25 | Opencv附加库 |

# 4详细测试结果

## 4.1单元测试

### 4.1.1模型加载模块

功能说明：将训练好的模型加载到内存中，方便后续使用。

输入：无。

输出：深度学习模型索引。

测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-1 | 测试模块 | | 模型加载模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍小组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 能否将模型加载到内存，输出深度学习模型索引 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以将模型加载到内存，输出深度学习模型索引 | | 可以将模型加载到内存，输出深度学习模型索引 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-1 | 测试模块 | | 模型加载模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍小组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 能否将模型加载到内存，输出深度学习模型索引 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 启动测试函数 2. 检查测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以将模型加载到内存，输出深度学习模型索引 | | 可以将模型加载到内存，输出深度学习模型索引 | |

### 4.1.2图片加载模块

功能说明：定义session加载待测试的图片文件。

输入：jpg、png、gif、bmp格式的图片。

输出：该图片在计算机中的存储矩阵。

测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-1 | 测试模块 | | 图片加载模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍小组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 测试该功能是否可以正确加载jpg格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择jpg格式的图片，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载jpg格式的图片文件 | | 可以正确加载jpg格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-1 | 测试模块 | | 图片加载模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍小组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 测试该功能是否可以正确加载jpg格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 选择jpg格式的图片作为参数 2. 启动测试函数 3. 查看测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载jpg格式的图片文件 | | 可以正确加载jpg格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-2 | 测试模块 | | 图片加载模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 测试该功能是否可以正确加载png格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择png格式的图片，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载png格式的图片文件 | | 可以正确加载png格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-2 | 测试模块 | | 图片加载模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 测试该功能是否可以正确加载png格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 选择png格式的图片作为参数 2. 启动测试函数 3. 查看测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载png格式的图片文件 | | 可以正确加载png格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-3 | 测试模块 | | 图片加载模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 测试该功能是否可以正确加载gif格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择gif格式的图片，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载gif格式的图片文件 | | 不可以正确加载gif格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-3 | 测试模块 | | 图片加载模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 测试该功能是否可以正确加载gif格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 选择gif格式的图片作为参数 2. 启动测试函数 3. 检查测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载gif格式的图片文件 | | 报错，无法加载gif格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-4 | 测试模块 | | 图片加载模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 测试该功能是否可以正确加载bmp格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择bmp格式的图片，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载bmp格式的图片文件 | | 不可以正确加载bmp格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-4 | 测试模块 | | 图片加载模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 测试该功能是否可以正确加载bmp格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 选择bmp格式的图片作为参数 2. 启动测试函数 3. 检查测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载bmp格式的图片文件 | | 可以正确加载bmp格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-5 | 测试模块 | | 图片加载模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 测试能否加载各种不同命名的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 选择bmp格式的不同命名的图片作为参数 2. 启动测试函数 3. 检查测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载各种不同命名的图片文件 | | 可以正确加载各种不同命名的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-5 | 测试模块 | | 图片加载模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 测试能否加载各种不同命名的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择bmp格式的不同命名的图片，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载各种不同命名的图片文件 | | 可以正确加载各种不同命名的图片文件 | |

### 4.1.3图片定位模块

功能说明：对原始图片进行目标检测定位的封装函数。

输入：jpg、png、gif、bmp格式的图片。

输出：该图片在计算机中的存储矩阵。

测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-1 | 测试模块 | | 图片定位模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍小组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 测试该功能是否可以正确加载jpg格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择jpg格式的图片，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载jpg格式的图片文件 | | 可以正确加载jpg格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-1 | 测试模块 | | 图片定位模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍小组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 测试该功能是否可以正确加载jpg格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 选择jpg格式的图片作为参数 2. 启动测试函数 3. 检查测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载jpg格式的图片文件 | | 可以正确加载jpg格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-2 | 测试模块 | | 图片定位模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 测试该功能是否可以正确加载png格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择png格式的图片，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载png格式的图片文件 | | 可以正确加载png格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-2 | 测试模块 | | 图片定位模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 测试该功能是否可以正确加载png格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 选择png格式的图片作为参数 2. 启动测试函数 3. 检查测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载png格式的图片文件 | | 可以正确加载png格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-3 | 测试模块 | | 图片定位模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 测试该功能是否可以正确加载gif格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择gif格式的图片，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载gif格式的图片文件 | | 不可以正确加载gif格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-3 | 测试模块 | | 图片定位模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 测试该功能是否可以正确加载gif格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 选择gif格式的图片作为参数 2. 启动测试函数 3. 检查测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载gif格式的图片文件 | | 不可以正确加载gif格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-4 | 测试模块 | | 图片定位模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 测试该功能是否可以正确加载bmp格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择bmp格式的图片，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载bmp格式的图片文件 | | 不可以正确加载bmp格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-4 | 测试模块 | | 图片定位模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 测试该功能是否可以正确加载bmp格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 选择bmp格式的图片作为参数 2. 启动测试函数 3. 查看测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载bmp格式的图片文件 | | 可以正确加载bmp格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-5 | 测试模块 | | 图片定位模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 测试能否正确识别所有图片中的人 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择人数不大于10人、10人至50人、50人以上的图片分别进行测试 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以检测并标记所有图片中的人 | | 只能检测并标记出部分图片中的人 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-5 | 测试模块 | | 图片定位模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 测试能否正确识别所有图片中的人 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择人数不大于10人、10人至50人、50人以上的图片分别进行测试 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以检测并标记所有图片中的人 | | 只能检测并标记出部分图片中的人 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-6 | 测试模块 | | 图片定位模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 测试能否对图片中的人标注人头序号 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择人数不大于10人、10人至50人、50人以上的图片分别进行测试 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以对图片中的人标注人头序号 | | 可以对图片中的人标注人头序号 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-6 | 测试模块 | | 图片定位模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 测试能否对图片中的人标注人头序号 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择人数不大于10人、10人至50人、50人以上的图片分别进行测试 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以对图片中的人标注人头序号 | | 可以对图片中的人标注人头序号 | |

### 4.1.4图片处理模块

功能说明：对原始图片进行处理的模块。

输入：jpg、jpeg、png、gif、bmp格式的图片。

输出：该图片在计算机中的存储矩阵。

测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-1 | 测试模块 | | 图片处理模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍小组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 测试该功能是否可以正确加载jpg格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择jpg格式的图片，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载jpg格式的图片文件 | | 可以正确加载jpg格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-1 | 测试模块 | | 图片处理模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍小组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 测试该功能是否可以正确加载jpg格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 选择jpg格式的图片作为参数 2. 启动测试函数 3. 检查测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载jpg格式的图片文件 | | 可以正确加载jpg格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-2 | 测试模块 | | 图片处理模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 测试该功能是否可以正确加载jpeg格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择jpeg格式的图片，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载jpeg格式的图片文件 | | 可以正确加载jpeg格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-2 | 测试模块 | | 图片处理模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 测试该功能是否可以正确加载jpeg格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 选择jpeg格式的图片作为参数 2. 启动测试函数 3. 检查测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载jpeg格式的图片文件 | | 可以正确加载jpeg格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-3 | 测试模块 | | 图片处理模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 测试该功能是否可以正确加载png格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择png格式的图片，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载png格式的图片文件 | | 可以正确加载png格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-3 | 测试模块 | | 图片处理模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 测试该功能是否可以正确加载png格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 选择png格式的图片作为参数 2. 启动测试函数 3. 检查测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载png格式的图片文件 | | 可以正确加载png格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-4 | 测试模块 | | 图片处理模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 测试该功能是否可以正确加载gif格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择gif格式的图片，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载gif格式的图片文件 | | 不可以正确加载gif格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-4 | 测试模块 | | 图片处理模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 测试该功能是否可以正确加载gif格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 选择gif格式的图片作为参数 2. 启动测试函数 3. 检查测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载gif格式的图片文件 | | 报错，不可以正确加载gif格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-5 | 测试模块 | | 图片处理模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 测试该功能是否可以正确加载bmp格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择bmp格式的图片，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载bmp格式的图片文件 | | 不可以正确加载bmp格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-5 | 测试模块 | | 图片处理模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 测试该功能是否可以正确加载bmp格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 选择bmp格式的图片作为参数 2. 启动测试函数 3. 检查测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载bmp格式的图片文件 | | 可以正确加载bmp格式的图片文件 | |

### 4.1.5图片展示模块

功能说明：显示处理后的图片结果。

输入：jpg、jpeg、png、gif、bmp格式的图片。

输出：将该图片显示在屏幕上。

测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-1 | 测试模块 | | 图片展示模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍小组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 测试该功能是否可以正确加载jpg格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择jpg格式的图片，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载jpg格式的图片文件 | | 可以正确加载jpg格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-1 | 测试模块 | | 图片展示模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍小组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 测试该功能是否可以正确加载jpg格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 选择jpg格式的图片作为全局参数 2. 启动测试函数 3. 检查测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载jpg格式的图片文件 | | 可以正确加载jpg格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-2 | 测试模块 | | 图片展示模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 测试该功能是否可以正确加载jpeg格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择jpeg格式的图片，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载jpeg格式的图片文件 | | 可以正确加载jpeg格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-2 | 测试模块 | | 图片展示模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 测试该功能是否可以正确加载jpeg格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 选择jpeg格式的图片作为全局参数 2. 启动测试函数 3. 检查测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载jpeg格式的图片文件 | | 可以正确加载jpeg格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-3 | 测试模块 | | 图片展示模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 测试该功能是否可以正确加载png格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择png格式的图片，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载png格式的图片文件 | | 可以正确加载png格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-3 | 测试模块 | | 图片展示模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 测试该功能是否可以正确加载png格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 选择png格式的图片作为全局参数 2. 启动测试函数 3. 检查测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载png格式的图片文件 | | 可以正确加载png格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-4 | 测试模块 | | 图片展示模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 测试该功能是否可以正确加载gif格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择gif格式的图片，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载gif格式的图片文件 | | 不可以正确加载gif格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-4 | 测试模块 | | 图片展示模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 测试该功能是否可以正确加载gif格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 选择gif格式的图片作为全局参数 2. 启动测试函数 3. 检查测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载gif格式的图片文件 | | 不可以正确加载gif格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-5 | 测试模块 | | 图片展示模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 测试该功能是否可以正确加载bmp格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择bmp格式的图片，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载bmp格式的图片文件 | | 不可以正确加载bmp格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-5 | 测试模块 | | 图片展示模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 测试该功能是否可以正确加载bmp格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 选择bmp格式的图片作为全局参数 2. 启动测试函数 3. 检查测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载bmp格式的图片文件 | | 可以正确加载bmp格式的图片文件 | |

### 4.1.6视频处理模块

功能说明：

1.使用VideoFileClip函数从视频中抓取图片；

2.用fl\_image函数将原图片替换为修改后的图片，用于传递物体识别的每张抓取图片；

3.将所有修改过的剪辑图像组合成一个新的视频。

输入：mp4、mov、avi、gif格式的视频

输出：无

测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-1 | 测试模块 | | 视频处理模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍小组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 对于mp4格式的视频能否正确处理 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择mp4格式的视频，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确处理mp4格式的视频 | | 未能正确处理mp4格式的视频 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-1 | 测试模块 | | 视频处理模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍小组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 对于mp4格式的视频能否正确处理 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 选择mp4格式的视频路径作为参数 2. 启动测试函数 3. 检查测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确处理mp4格式的视频 | | 未能正确处理mp4格式的视频 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-2 | 测试模块 | | 视频处理模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 对于avi格式的视频能否正确处理 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择avi格式的视频，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确处理avi格式的视频 | | 未能正确处理avi格式的视频 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-2 | 测试模块 | | 视频处理模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 对于avi格式的视频能否正确处理 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 选择avi格式的视频路径作为参数 2. 启动测试函数 3. 检查测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确处理avi格式的视频 | | 未能正确处理avi格式的视频 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-3 | 测试模块 | | 视频处理模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 对于mov格式的视频能否正确处理 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择mov格式的视频，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确处理mov格式的视频 | | 未能正确处理mov格式的视频 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-3 | 测试模块 | | 视频处理模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 对于mov格式的视频能否正确处理 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 选择mov格式的视频路径作为参数 2. 启动测试函数 3. 检查测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确处理mov格式的视频 | | 未能正确处理mov格式的视频 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-4 | 测试模块 | | 视频处理模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 对于gif格式的视频能否正确处理 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择gif格式的视频，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确处理gif格式的视频 | | 未能正确处理gif格式的视频 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-4 | 测试模块 | | 视频处理模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 对于gif格式的视频能否正确处理 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 选择gif格式的视频路径作为参数 2. 启动测试函数 3. 检查测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确处理gif格式的视频 | | 未能正确处理gif格式的视频 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-5 | 测试模块 | | 视频处理模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 能否处理各种不同命名方式的视频 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择gif格式的视频，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确处理各种不同命名方式的视频 | | 报错，不能正确处理各种不同命名方式的视频 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-5 | 测试模块 | | 视频处理模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 能否处理各种不同命名方式的视频 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 选择gif格式的视频路径作为参数 2. 启动测试函数 3. 检查测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确处理各种不同命名方式的视频 | | 报错，不能正确处理各种不同命名方式的视频 | |

### 4.1.7行人移动位置预测模块

功能说明：

Kalmon算法用于对象跟踪的卡尔曼滤波算法的帮助下载场景中滚动，借助卡尔曼滤波方式来对行人移动位置进行预测，提高目标跟踪的精度。

输入：numpy专用的二进制文件。

输出：kalmon算法输出值与测量值误差对比图。

测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-1 | 测试模块 | | 位置预测模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.19 |
| 开发人员 | 何艳萍小组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 算法能否正常工作，能否输出kalmon算法输出值与测量值误差对比图 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 算法可以正常工作，能输出kalmon算法输出值与测量值误差对比图 | | 算法可以正常工作，能输出kalmon算法输出值与测量值误差对比图 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-1 | 测试模块 | | 位置预测模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.19 |
| 开发人员 | 何艳萍小组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 算法能否正常工作，能否输出kalmon算法输出值与测量值误差对比图 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 启动测试函数 2. 检查测试结果，看是否输出对比图 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 算法可以正常工作，能输出kalmon算法输出值与测量值误差对比图 | | 算法可以正常工作，能输出kalmon算法输出值与测量值误差对比图 | |

### 4.1.8目标跟踪模块

功能说明：

利用mean-shift对已经标记的人进行目标跟踪，中间通过不断迭代更新行人目标位置并实时标记。

输入：mp4、mov、avi、gif格式的视频。

输出：对行人进行跟踪标记的视频。

测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-1 | 测试模块 | | 目标跟踪模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.20 |
| 开发人员 | 何艳萍小组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 模块能否正确加载mp4格式视频 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 模块可以正确加载mp4格式视频 | | 模块可以正确加载mp4格式视频 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-1 | 测试模块 | | 目标跟踪模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.20 |
| 开发人员 | 何艳萍小组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 模块能否正确加载mp4格式视频 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 选择mp4格式视频 2. 启动测试函数 3. 观察测试结果是否输出视频 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 模块可以正确加载mp4格式视频 | | 模块可以正确加载mp4格式视频 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-2 | 测试模块 | | 目标跟踪模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.20 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 模块能否正确加载avi格式视频 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择avi格式的视频，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 模块可以正确加载avi格式视频 | | 模块可以正确加载avi格式视频 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-2 | 测试模块 | | 目标跟踪模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.20 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 模块能否正确加载avi格式视频 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 选择avi格式的视频 2. 启动测试函数 3. 观察测试结果是否输出视频 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 模块可以正确加载avi格式视频 | | 模块可以正确加载avi格式视频 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-3 | 测试模块 | | 目标跟踪模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.20 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 模块能否正确加载mov格式视频 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择mov格式的视频，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 模块可以正确加载mov格式视频 | | 模块可以正确加载mov格式视频 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-3 | 测试模块 | | 目标跟踪模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.20 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 模块能否正确加载mov格式视频 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 选择mov格式的视频 2. 启动测试函数 3. 观察测试结果是否输出视频 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 模块可以正确加载mov格式视频 | | 模块可以正确加载mov格式视频 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-4 | 测试模块 | | 目标跟踪模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.20 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 模块能否正确加载gif格式视频 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择gif格式的视频，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以实现对视频中人物的实时标记 | | 未能实现对视频中人物的实时标记 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-4 | 测试模块 | | 目标跟踪模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.20 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 模块能否正确加载gif格式视频 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 选择gif格式的视频 2. 启动测试函数 3. 观察测试结果是否输出视频 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以实现对视频中人物的实时标记 | | 模块不可以加载gif格式视频 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-5 | 测试模块 | | 目标跟踪模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.20 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 能否对视频中所有行人进行标记并跟踪 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 分别选择mp4、avi、mov格式视频进行测试 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以对视频中所有行人进行标记并跟踪 | | 可以对视频中行人进行标记并跟踪 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-5 | 测试模块 | | 目标跟踪模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.20 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 能否对视频中所有行人进行标记并跟踪 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 分别选择mp4、avi、mov格式视频教学测试 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以对视频中所有行人进行标记并跟踪 | | 只能对视频中部分行人进行标记并跟踪 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-6 | 测试模块 | | 目标跟踪模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.20 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 能否对视频中的人标注人头序号 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 分别选择mp4、avi、mov格式视频进行测试 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以对视频中的人标注人头序号 | | 没有对视频中的人标注人头序号 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-6 | 测试模块 | | 目标跟踪模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.20 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 能否对视频中的人标注人头序号 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 分别选择mp4、avi、mov格式视频进行测试 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以对视频中的人标注人头序号 | | 没有对视频中的人标注人头序号 | |

### 4.1.9视频识别结果保存模块

功能说明：输出、保存识别结果

输入：mp4、mov、avi、gif格式的视频或

输出：将视频信息、总人数和行人密度保存到数据库中

测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-1 | 测试模块 | | 视频识别结果保存模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.20 |
| 开发人员 | 何艳萍小组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 能否将视频信息、总人数和行人密度保存到数据库中 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以将视频信息、总人数和行人密度保存到数据库中 | | 系统没有连接数据库，即未保存数据 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-1 | 测试模块 | | 视频识别结果保存模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.20 |
| 开发人员 | 何艳萍小组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 能否将视频信息、总人数和行人密度保存到数据库中 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以将视频信息、总人数和行人密度保存到数据库中 | | 系统没有连接数据库，即未保存数据 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-2 | 测试模块 | | 视频识别结果保存模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.20 |
| 开发人员 | 何艳萍小组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 能否识别视频地点信息 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以识别视频地点信息 | | 无视频地点识别机制 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-2 | 测试模块 | | 视频识别结果保存模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.20 |
| 开发人员 | 何艳萍小组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 能否识别视频地点信息 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以识别视频地点信息 | | 无视频地点识别机制 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-3 | 测试模块 | | 视频识别结果保存模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.20 |
| 开发人员 | 何艳萍小组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 若视频中人数超过上限能否向云端上报相关信息 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 若视频中人数超过上限可以向云端上报相关信息 | | 未设置人数上限，无上报机制 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-3 | 测试模块 | | 视频识别结果保存模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.20 |
| 开发人员 | 何艳萍小组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 若视频中人数超过上限能否向云端上报相关信息 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 若视频中人数超过上限可以向云端上报相关信息 | | 未设置人数上限，无上报机制 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | C-4 | 测试模块 | | 视频识别结果保存模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.20 |
| 开发人员 | 何艳萍小组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 是否使用AES加密算法进行数据保护 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 使用了AES加密算法进行数据保护 | | 未使用AES加密算法进行数据保护 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | H-4 | 测试模块 | | 视频识别结果保存模块 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.20 |
| 开发人员 | 何艳萍小组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 是否使用AES加密算法进行数据保护 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 使用了AES加密算法进行数据保护 | | 未使用AES加密算法进行数据保护 | |

## 4.2界面测试

测试方法：等价类划分法

自动化测试工具：Selenium IDE（输出测试脚本为pytest脚本）

登陆界面测试：

|  |  |  |
| --- | --- | --- |
|  | 有效等价类 | 无效等价类 |
| 登陆 | 1. 用户名为任意长度任意字符组合 2. 密码为任意长度任意字符组合 3. 勾选记住密码 4. 不勾选记住密码 | 1. 不输入用户名 2. 不输入密码 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 覆盖等价类 | 测试用例 | | 测试结果 |
| 用户名 | 密码 |
| 1 | 1、2、3 | 我Ab4$ | Df#2 | 登陆成功 |
| 2 | 1、2、4 | 我Ab4$ | Df#2 | 登陆成功 |
| 3 | 2、4、5 |  | Df#2 | 登陆成功 |
| 4 | 1、4、6 | 我Ab4$ |  | 登陆成功 |

注册界面测试：

|  |  |  |
| --- | --- | --- |
|  | 有效等价类 | 无效等价类 |
| 注册 | 1. 用户名为任意长度任意字符组合 2. 密码为任意长度任意字符组合 3. 邮箱地址中要包含字符@ 4. 字符@后要有字符 5. 密码和确认密码输入一致 6. 勾选同意协议 | 1. 不输入用户名 2. 不输入密码 3. 不输入确认密码 4. 不输入邮箱 5. 邮箱地址没有@字符 6. 邮箱地址有@，但@后没有字符 7. 确认密码和密码不一致 8. 不勾选同意协议 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 覆盖等价类 | 测试用例 | | | | | 测试结果 |
| 用户名 | 邮箱 | 密码 | 确认密码 | 是否勾选同意协议 |
| 1 | 1、2、3、4、5、6 | 嗨#fsd | d15@qq.com | Df#1 | Df#1 | 是 | 注册成功 |
| 2 | 2、3、4、5、6、7 |  | d15@qq.com | Df#1 | Df#1 | 是 | 注册失败 |
| 3 | 1、3、4、5、6、8 | 嗨#fsd | d15@qq.com |  | Df#1 | 是 | 注册失败 |
| 4 | 1、2、3、4、6、9 | 嗨#fsd | d15@qq.com | Df#1 |  | 是 | 注册失败 |
| 5 | 1、2、5、6、10 | 嗨#fsd |  | Df#1 | Df#1 | 是 | 注册失败 |
| 6 | 1、2、5、6、11 | 嗨#fsd | cf5g | Df#1 | Df#1 | 是 | 注册失败 |
| 7 | 1、2、3、5、6、12 | 嗨#fsd | cf5g@ | Df#1 | Df#1 | 是 | 注册失败 |
| 8 | 1、2、3、4、6、13 | 嗨#fsd | d15@qq.com | Df#1 | ehf13d | 是 | 注册失败 |
| 9 | 1、2、3、4、5、14 | 嗨#fsd | d15@qq.com | Df#1 | Df#1 | 否 | 注册成功 |

登陆注册页面其他按钮测试：

|  |  |  |
| --- | --- | --- |
| 序号 | 测试功能 | 测试结果 |
| 1 | 在注册界面点击登陆，返回登陆界面 | 无法返回登陆界面 |
| 2 | 在登陆界面点击注册，跳转到注册界面 | 成功跳转到注册界面 |
| 3 | 在登陆界面点击忘记密码，跳转到找回密码界面 | 失败，无该界面 |

主页面测试：

|  |  |  |
| --- | --- | --- |
| 序号 | 测试功能 | 测试结果 |
| 1 | 导航栏浮动功能 | 正常 |
| 2 | 点击预警信息，显示相应内容 | 无反应 |
| 3 | 点击异常地点排序，显示相应内容 | 无反应 |
| 4 | 点击异常视频查看，显示相应内容 | 无反应 |
| 5 | 点击监控地点列表，显示相应内容 | 无反应 |
| 6 | 点击注销，返回登陆界面 | 无反应 |

## 4.3集成测试

**测试任务**

1. 各模块连接时，经过模块接口的数据丢失情况。  
   （2）某-模块的功能对另-个模块的功能的不利影响。  
   （3）各模块或子系统功能的组合,达到预期需求的集成子系统系统功能情况。
2. 检测软件系统所需的全局数据结构是否有问题。  
   （5）单个模块的误差累积后情况,是否会放大,是否可接受。

（6）单个模块的错误对数据库的影响。

**测试要求**

（1）确认模块之间无错误连接。

（2）验证被测系统满足设计要求的情况、根据设计要求的功能、性能和可靠性等要求，测试整个系统，验证达到设计要求情况 。

（3）以数据处理测试用例对被测系统的输人、输出、处理进行检测，达到设计要求。

（4）利用业务处理测试用例对被测系统业务处理过程进行测试.达到设计要求。

（5）测试软件正确处理的能力和容错能力所达到的标准。

（6）测试软件对数据、接口错误、数据错误、协议错误的识别及处理是否符合标准。

**集成测试通过的准则**

（1）各单元之间无错误连接。

（2）达到软件需求的各项功能、性能、可靠性等方面的指标要求。

（3）对偶发的错误输人有正确的处理能力。

（4）对测试中的异常问题有合理的提示反馈。

（5）人机界面及操作友好便捷。

**测试策略**

自顶向下增殖式集成测试。这种集成的方式是从主控模块开始，沿着程序的控制层次向下移动，逐渐把各个模块结合起来。在把附属于主控模块的那些模块组装到程序结构中。自顶向下增殖的步骤如下:

（1）以主模块为所测模块兼驱动模块，所有直属于主模块的下属模块全部用桩模块对主模块进行测试。

（2）采用深度优先或宽度优先的策略，用实际模块替换相应桩模块，再用桩代替它们的直接下属模块，与已测试的模块或子系统集成为新的子系统。

（3）进行回归测试( 即重新执行以前做过的全部测试或部分测试)，排除集成过程中引起错误的可能。

（4）判断是否所有的模块都已集成到系统中，是则结束测试，否则转到(2)去执行。

### 4.3.1行人标识记录

功能说明：通过已经训练好的Faster-Rcnn参数实现对行人的识别并标记（其中标记行人的阈值为0.7，即识别率必须达到70%）。

输入：图片

输出：将图片显示在屏幕上

测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 1 | 测试模块 | | 行人标识记录 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.24 |
| 开发人员 | 何艳萍小组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 测试该功能是否可以正确加载并处理jpg格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 图片展示模块调用模型加载模块以及图片处理模块  选择jpg格式的图片，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载并处理jpg格式的图片文件 | | 可以正确加载jpg格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 2 | 测试模块 | | 行人标识记录 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.24 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 图片展示模块调用模型加载模块以及图片处理模块  测试该功能是否可以正确加载并处理jpeg格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择jpeg格式的图片，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载并处理jpeg格式的图片文件 | | 可以正确加载jpeg格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 3 | 测试模块 | | 行人标识记录 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.24 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 测试该功能是否可以正确加载并处理png格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 图片展示模块调用模型加载模块以及图片处理模块  选择png格式的图片，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载并处理png格式的图片文件 | | 可以正确加载并处理png格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 4 | 测试模块 | | 行人标识记录 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.24 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 图片展示模块调用模型加载模块以及图片处理模块  测试该功能是否可以正确加载gif格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择gif格式的图片，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载gif格式的图片文件 | | 不可以正确加载gif格式的图片文件 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 5 | 测试模块 | | 行人标识记录 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.24 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 图片展示模块调用模型加载模块以及图片处理模块  测试该功能是否可以正确加载并处理bmp格式的图片文件 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择bmp格式的图片，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确加载bmp格式的图片文件 | | 不可以正确加载bmp格式的图片文件 | |

### 4.3.2行人移动位置预测

功能说明：借助卡尔曼滤波方式（卡尔曼算法用于对象跟踪的卡尔曼滤波算法的帮助下载场景中滚动）来对行人移动位置进行预测，提高目标跟踪的精度。

输入：.npy文件

输出：将图片显示在屏幕上

测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 1 | 测试模块 | | 行人移动位置 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍小组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 对于npy格式的视频能否正确处理 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择npy格式的文件，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确处理 | | 可以正确处理 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 2 | 测试模块 | | 行人移动位置 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 不输入文件是否有异常处理机制 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 不选择文件启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 程序有异常处理机制 | | 无异常处理机制 | |

### 4.3.3行人目标跟踪

功能说明：利用mean-shift对已经标记的人进行目标跟踪，中间通过不断迭代更新行人目标位置并实时标记。

输入：视频

输出：将视频中的人物标识出来并打印在屏幕上

测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 1 | 测试模块 | | 行人目标跟踪 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍小组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 对于mp4格式的视频能否正确处理 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择mp4格式的视频，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确处理mp4格式的视频 | | 可以正确处理mp4格式的视频 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 2 | 测试模块 | | 行人目标跟踪 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 对于avi格式的视频能否正确处理 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择avi格式的视频，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确处理avi格式的视频 | | 未能正确处理avi格式的视频 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 3 | 测试模块 | | 行人目标跟踪 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 对于mov格式的视频能否正确处理 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择mov格式的视频，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确处理mov格式的视频 | | 未能正确处理mov格式的视频 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 4 | 测试模块 | | 行人目标跟踪 |
| 编制人 | 黄新昊 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 黄新昊 | 测试负责人 | | 黄新昊 |
| 测试内容 | 对于gif格式的视频能否正确处理 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择gif格式的视频，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 可以正确处理gif格式的视频 | | 未能正确处理gif格式的视频 | |

### 4.3.4视频识别结果保存模块

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 1 | 测试模块 | | 视频识别结果保存模块 |
| 编制人 | 陈硕峰 | 编制时间 | | 2019.06.18 |
| 开发人员 | 何艳萍组 | 程序版本 | | v1.0 |
| 测试人员 | 陈硕峰 | 测试负责人 | | 陈硕峰 |
| 测试内容 | 视频识别效率是否符合响应时间和精度要求 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择gif格式的视频，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 符合 | | 没有给出响应时间和精度要求 | |

## 4.4有效性测试

### 4.4.1测试功能点

1. 正确识别3个以上人数密集边缘区域的地点位置信息和人流量（即人头数）,并做相应排序；
2. 边缘端程序和云监管平台两个程序分开独立部署,满足边缘计算云边协同架构的实际需求;
3. 从摄像头获取视频数据；
4. 边缘端程序识别人头数和地点信息,并做相关计算,如果人头数没有达到容量上限则数据不上报到云端，若人头数超出预置容量上限数，就将所在人流异常视频数据、地点信息和人头数上报到云监管平台；
5. 识别出的图片中的人数是否正确；
6. 在视频中人数改变时实时输出总人数；
7. 云监管平台呈现对应地点和人头数爆发的数据告警,边缘端人流数据实时变化,云端展示人流量数据也实时变化,如果人流量实时降低并小于阈值,报警自动解除;
8. 异常视频中用动态标注的方式标注人头序号,并识别地点信息和人头总数和告警；
9. 获取视频地点信息；
10. 异常视频数据、地点信息和人头数数据在云端进行数据存储,云监管平台可以查询;对多个异常地点进行数据分析，并分类，排序重点突出经常容易出现人流异常爆发的地点；
11. 云监管平台实时展示地点信息和人头数数据，智能匹配给出疏导处理的相关保安组织人员推荐信息和联系方式;
12. 同时支持web端和手机移动端良好体验；
13. 利用可靠的密码技术对用户数据进行加密；
14. 设备备份机制、容错机制，防止在系统出现单点失败时，系统的备份机制保证系统的正常运行；
15. 在技术可行的情况下，将安全系统网络与过程控制信息系统网络隔离开；
16. 登陆界面未输入用户名弹出警告；
17. 登陆界面未输入密码弹出警告；
18. 输入用户名、密码后勾选保存密码下此无需输入账号密码；
19. 在登陆界面点击忘记密码，跳转到密码找回界面；
20. 在登陆界面点击注册，跳转到注册界面；
21. 在注册界面未输入用户名时弹出警告；
22. 在注册界面未输入密码时弹出警告；
23. 在注册界面未输入邮箱时弹出警告；
24. 在注册界面输入邮箱格式错误时弹出警告；
25. 在注册界面两次输入密码不同时弹出警告；
26. 在注册界面不勾选同意协议就点击注册按钮弹出警告；
27. 在注册界面点击登陆，返回登陆界面；

### 4.4.2测试项目

|  |  |  |
| --- | --- | --- |
| 项目 | 内容 | 检测结果 |
| 功能点1 | 测试功能点1 | F |
| 功能点2 | 测试功能点2 | F |
| 功能点3 | 测试功能点3 | F |
| 功能点4 | 测试功能点4 | F |
| 功能点5 | 测试功能点5 | F |
| 功能点6 | 测试功能点6 | F |
| 功能点7 | 测试功能点7 | F |
| 功能点8 | 测试功能点8 | F |
| 功能点9 | 测试功能点9 | F |
| 功能点10 | 测试功能点10 | F |
| 功能点11 | 测试功能点11 | F |
| 功能点12 | 测试功能点12 | F |
| 功能点13 | 测试功能点13 | F |
| 功能点14 | 测试功能点14 | F |
| 功能点15 | 测试功能点15 | F |
| 功能点16 | 测试功能点16 | P |
| 功能点17 | 测试功能点17 | P |
| 功能点18 | 测试功能点18 | F |
| 功能点19 | 测试功能点19 | F |
| 功能点20 | 测试功能点20 | P |
| 功能点21 | 测试功能点21 | P |
| 功能点22 | 测试功能点22 | P |
| 功能点23 | 测试功能点23 | P |
| 功能点24 | 测试功能点24 | P |
| 功能点25 | 测试功能点25 | P |
| 功能点26 | 测试功能点26 | F |
| 功能点27 | 测试功能点27 | F |

### 4.4.3测试内容

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试人员 | 黄新昊 | 程序版本 | | v1.0 |
| 开发人员 | 何艳萍组 | 测试时间 | | 2019.06.26 |
| 测试内容 | 系统能否正确识别3个以上人数密集边缘区域的地点位置信息和人流量（即人头数）,并做相应排序 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择gif格式的视频，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 实现功能点1 | | 未能实现功能点1 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试人员 | 黄新昊 | 程序版本 | | v1.0 |
| 开发人员 | 何艳萍组 | 测试时间 | | 2019.06.26 |
| 测试内容 | 边缘端程序识别人头数和地点信息,并做相关计算,如果人头数没有达到容量上限则数据不上报到云端，若人头数超出预置容量上限数，就将所在人流异常视频数据、地点信息和人头数上报到云监管平台 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择gif格式的视频，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 实现功能点2 | | 未能实现功能点2 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试人员 | 陈硕峰 | 程序版本 | | v1.0 |
| 开发人员 | 何艳萍组 | 测试时间 | | 2019.06.26 |
| 测试内容 | 能否从摄像头获取视频数据 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 查看代码是否具有摄像头接口 2. 若有摄像头接口则使用摄像头进行测试 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 实现功能点3 | | 未能实现功能点3，未连接摄像头 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试人员 | 黄新昊 | 程序版本 | | v1.0 |
| 开发人员 | 何艳萍组 | 测试时间 | | 2019.06.26 |
| 测试内容 | 边缘端程序和云监管平台两个程序分开独立部署,满足边缘计算云边协同架构的实际需求 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择gif格式的视频，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 实现功能点4 | | 未能实现功能点4 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试人员 | 陈硕峰 | 程序版本 | | v1.0 |
| 开发人员 | 何艳萍组 | 测试时间 | | 2019.06.26 |
| 测试内容 | 识别出的图片中的人数是否正确 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 选择测试视频 2. 查看输出结果 3. 进行人工计数 4. 对比结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 实现功能点5 | | 未能实现功能点5，人群密度越高识别率越低 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试人员 | 陈硕峰 | 程序版本 | | v1.0 |
| 开发人员 | 何艳萍组 | 测试时间 | | 2019.06.26 |
| 测试内容 | 能否在视频中人数改变时实时输出总人数 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 选择测试视频 2. 运行代码 3. 检查是否实时输出总人数 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 实现功能点6 | | 未能实现功能点6，无法实时输出 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试人员 | 黄新昊 | 程序版本 | | v1.0 |
| 开发人员 | 何艳萍组 | 测试时间 | | 2019.06.26 |
| 测试内容 | 云监管平台呈现对应地点和人头数爆发的数据告警,边缘端人流数据实时变化,云端展示人流量数据也实时变化,如果人流量实时降低并小于阈值,报警自动解除 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择gif格式的视频，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 实现功能点7 | | 未能实现功能点7 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试人员 | 黄新昊 | 程序版本 | | v1.0 |
| 开发人员 | 何艳萍组 | 测试时间 | | 2019.06.26 |
| 测试内容 | 异常视频中用动态标注的方式标注人头序号,并识别地点信息和人头总数和告警 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择gif格式的视频，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 实现功能点8 | | 未能实现功能点8 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试人员 | 陈硕峰 | 程序版本 | | v1.0 |
| 开发人员 | 何艳萍组 | 测试时间 | | 2019.06.26 |
| 测试内容 | 能否获取视频地点信息 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 选择测试视频 2. 运行代码 3. 查看是否输出地点信息 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 实现功能点9 | | 未能实现功能点9 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试人员 | 黄新昊 | 程序版本 | | v1.0 |
| 开发人员 | 何艳萍组 | 测试时间 | | 2019.06.26 |
| 测试内容 | 异常视频数据、地点信息和人头数数据在云端进行数据存储,云监管平台可以查询; 对多个异常地点进行数据分析，并分类，排序重点突出经常容易出现人流异常爆发的地点 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择gif格式的视频，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 实现功能点10 | | 未能实现功能点10 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试人员 | 黄新昊 | 程序版本 | | v1.0 |
| 开发人员 | 何艳萍组 | 测试时间 | | 2019.06.26 |
| 测试内容 | 云监管平台实时展示地点信息和人头数数据，智能匹配给出疏导处理的相关保安组织人员推荐信息和联系方式; | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择gif格式的视频，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 实现功能点11 | | 未能实现功能点11 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试人员 | 黄新昊 | 程序版本 | | v1.0 |
| 开发人员 | 何艳萍组 | 测试时间 | | 2019.06.26 |
| 测试内容 | 同时支持web端和手机移动端良好体验 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 选择gif格式的视频，启动测试函数，记录测试结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 实现功能点12 | | 未能实现功能点12 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试人员 | 陈硕峰 | 程序版本 | | v1.0 |
| 开发人员 | 何艳萍组 | 测试时间 | | 2019.06.26 |
| 测试内容 | 是否利用可靠的密码技术对用户数据进行加密 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 该系统无用户接口，即无法获取用户数据，且未找到加密模块 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 实现功能点13 | | 未能实现功能点13 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试人员 | 陈硕峰 | 程序版本 | | v1.0 |
| 开发人员 | 何艳萍组 | 测试时间 | | 2019.06.26 |
| 测试内容 | 设备备份机制、容错机制，防止在系统出现单点失败时，系统的备份机制保证系统的正常运行 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 选择测试视频 2. 运行程序 3. 在当前节点上关闭该程序 4. 检查结果（该系统未搭建分布式，无备份机制，因此程序直接结束） | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 实现功能点14 | | 未能实现功能点14 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试人员 | 陈硕峰 | 程序版本 | | v1.0 |
| 开发人员 | 何艳萍组 | 测试时间 | | 2019.06.26 |
| 测试内容 | 在技术可行的情况下，将安全系统网络与过程控制信息系统网络隔离开 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 该系统无安全系统网络 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 实现功能点15 | | 未能实现功能点15 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试人员 | 陈硕峰 | 程序版本 | | v1.0 |
| 开发人员 | 何艳萍组 | 测试时间 | | 2019.06.26 |
| 测试内容 | 登陆界面未输入用户名弹出警告 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 使用Selenium IDE录制测试脚本 2. 运行脚本 3. 观察运行过程及结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 实现功能点16 | | 实现功能点16 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试人员 | 陈硕峰 | 程序版本 | | v1.0 |
| 开发人员 | 何艳萍组 | 测试时间 | | 2019.06.26 |
| 测试内容 | 登陆界面未输入密码弹出警告 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 使用Selenium IDE录制测试脚本 2. 运行脚本 3. 观察运行过程及结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 实现功能点17 | | 实现功能点17 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试人员 | 陈硕峰 | 程序版本 | | v1.0 |
| 开发人员 | 何艳萍组 | 测试时间 | | 2019.06.26 |
| 测试内容 | 输入用户名、密码后勾选保存密码下此无需输入账号密码 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 使用Selenium IDE录制测试脚本 2. 运行脚本 3. 观察运行过程及结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 实现功能点18 | | 未能实现功能点18 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试人员 | 陈硕峰 | 程序版本 | | v1.0 |
| 开发人员 | 何艳萍组 | 测试时间 | | 2019.06.26 |
| 测试内容 | 在登陆界面点击忘记密码，跳转到密码找回界面 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 使用Selenium IDE录制测试脚本 2. 运行脚本 3. 观察运行过程及结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 实现功能点19 | | 未能实现功能点19 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试人员 | 陈硕峰 | 程序版本 | | v1.0 |
| 开发人员 | 何艳萍组 | 测试时间 | | 2019.06.26 |
| 测试内容 | 在登陆界面点击注册，跳转到注册界面 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 使用Selenium IDE录制测试脚本 2. 运行脚本 3. 观察运行过程及结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 实现功能点20 | | 实现功能点20 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试人员 | 陈硕峰 | 程序版本 | | v1.0 |
| 开发人员 | 何艳萍组 | 测试时间 | | 2019.06.26 |
| 测试内容 | 在注册界面未输入用户名时弹出警告 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 使用Selenium IDE录制测试脚本 2. 运行脚本 3. 观察运行过程及结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 实现功能点21 | | 实现功能点21 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试人员 | 陈硕峰 | 程序版本 | | v1.0 |
| 开发人员 | 何艳萍组 | 测试时间 | | 2019.06.26 |
| 测试内容 | 在注册界面未输入密码时弹出警告 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 使用Selenium IDE录制测试脚本 2. 运行脚本 3. 观察运行过程及结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 实现功能点22 | | 实现功能点22 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试人员 | 陈硕峰 | 程序版本 | | v1.0 |
| 开发人员 | 何艳萍组 | 测试时间 | | 2019.06.26 |
| 测试内容 | 在注册界面未输入邮箱时弹出警告 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 使用Selenium IDE录制测试脚本 2. 运行脚本 3. 观察运行过程及结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 实现功能点23 | | 实现功能点23 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试人员 | 陈硕峰 | 程序版本 | | v1.0 |
| 开发人员 | 何艳萍组 | 测试时间 | | 2019.06.26 |
| 测试内容 | 在注册界面输入邮箱格式错误时弹出警告 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 使用Selenium IDE录制测试脚本 2. 运行脚本 3. 观察运行过程及结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 实现功能点24 | | 实现功能点24 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试人员 | 陈硕峰 | 程序版本 | | v1.0 |
| 开发人员 | 何艳萍组 | 测试时间 | | 2019.06.26 |
| 测试内容 | 在注册界面两次输入密码不同时弹出警告 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 使用Selenium IDE录制测试脚本 2. 运行脚本 3. 观察运行过程及结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 实现功能点25 | | 实现功能点25 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试人员 | 陈硕峰 | 程序版本 | | v1.0 |
| 开发人员 | 何艳萍组 | 测试时间 | | 2019.06.26 |
| 测试内容 | 在注册界面不勾选同意协议就点击注册按钮弹出警告 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 使用Selenium IDE录制测试脚本 2. 运行脚本 3. 观察运行过程及结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 实现功能点26 | | 未能实现功能点26 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试人员 | 陈硕峰 | 程序版本 | | v1.0 |
| 开发人员 | 何艳萍组 | 测试时间 | | 2019.06.26 |
| 测试内容 | 在注册界面点击登陆，返回登陆界面 | | | |
| 测试环境 | 见3.2章节 | | | |
| 测试步骤 | 1. 使用Selenium IDE录制测试脚本 2. 运行脚本 3. 观察运行过程及结果 | | | |
| 测试结果 | 预期结果 | | 实际结果 | |
| 实现功能点27 | | 未能实现功能点27 | |

### 4.4.4覆盖分析

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 检测点 | BUG | 检测结果 |
| 1 | 对包含人物的mp4格式视频的标记 | 0 | P |
| 2 | 对包含人物的avi格式视频的标记 | 1 | F |
| 3 | 对包含人物的mov格式视频的标记 | 1 | F |
| 4 | 对包含人物的gif格式视频的标记 | 1 | F |
| 5 | 对于mp4格式且不含人物的视频的标记 | 0 | P |
| 6 | 对于avi格式且不含人物的视频的标记 | 0 | P |
| 7 | 对于mov格式且不含人物的视频的标记 | 1 | F |
| 8 | 对于gif格式且不含人物的视频的标记 | 0 | P |
| 9 | 是否可以正确加载jpg格式的图片文件 | 0 | P |
| 10 | 是否可以正确加载jpeg格式的图片文件 | 0 | P |
| 11 | 是否可以正确加载png格式的图片文件 | 0 | P |
| 12 | 是否可以正确加载gif格式的图片文件 | 1 | F |
| 13 | 是否可以正确加载bmp格式的图片文件 | 1 | F |
| 14 | 是否可以正确加载jpg格式的图片且可以实现目标检测定位 | 0 | P |
| 15 | 是否可以正确加载jpeg格式的图片且可以实现目标检测定位 | 0 | P |
| 16 | 是否可以正确加载png格式的图片且可以实现目标检测定位 | 0 | P |
| 17 | 是否可以正确加载gif格式的图片且可以实现目标检测定位 | 1 | F |
| 18 | 是否可以正确加载bmp格式的图片且可以实现目标检测定位 | 1 | F |
| 19 | 是否可以正确处理jpg格式的图片 | 0 | P |
| 20 | 是否可以正确处理jpeg格式的图片 | 0 | P |
| 21 | 是否可以正确处理png格式的图片 | 0 | P |
| 22 | 是否可以正确处理gif格式的图片 | 1 | F |
| 23 | 是否可以正确处理bmp格式的图片 | 1 | F |
| 24 | 是否可以正确显示jpg格式的图片 | 0 | P |
| 25 | 是否可以正确显示jpeg格式的图片 | 0 | P |
| 26 | 是否可以正确显示png格式的图片 | 0 | P |
| 27 | 是否可以正确显示gif格式的图片 | 1 | F |
| 28 | 是否可以正确显示bmp格式的图片 | 1 | F |
| 29 | 系统是否可以正确处理mp4格式的视频 | 0 | P |
| 30 | 系统是否可以正确处理avi格式的视频 | 0 | P |
| 31 | 系统是否可以正确处理mov格式的视频 | 0 | P |
| 32 | 系统是否可以正确处理gif格式的视频 | 1 | F |
| 33 | 算法是否提供正确中间结果 | 0 | P |
| 34 | 系统是否可以正确跟踪mp4格式视频中的目标 | 0 | P |
| 35 | 系统是否可以正确跟踪avi格式视频中的目标 | 0 | P |
| 36 | 系统是否可以正确跟踪mov格式视频中的目标 | 0 | P |
| 37 | 系统是否可以正确跟踪gif格式视频中的目标 | 1 | F |
| 38 | 系统是否可以加载各种命名方式的图片文件 | 1 | F |
| 39 | 系统是否可以加载各种命名方式的视频文件 | 0 | P |
| 40 | 登陆界面未输入用户名弹出警告 | 0 | P |
| 41 | 登陆界面未输入密码弹出警告 | 0 | P |
| 42 | 输入用户名、密码后勾选保存密码下此无需输入账号密码 | 1 | F |
| 43 | 在登陆界面点击忘记密码，跳转到密码找回界面 | 1 | F |
| 44 | 在登陆界面点击注册，跳转到注册界面 | 0 | P |
| 45 | 在注册界面未输入用户名时弹出警告 | 0 | P |
| 46 | 在注册界面未输入密码时弹出警告 | 0 | P |
| 47 | 在注册界面未输入邮箱时弹出警告 | 0 | P |
| 48 | 在注册界面输入邮箱格式错误时弹出警告 | 0 | P |
| 49 | 在注册界面两次输入密码不同时弹出警告 | 0 | P |
| 50 | 在注册界面不勾选同意协议就点击注册按钮弹出警告 | 1 | F |
| 51 | 在注册界面点击登陆，返回登陆界面 | 1 | F |

### 4.4.5缺陷汇总及分析

|  |  |  |
| --- | --- | --- |
| 序号 | 缺陷 | 缺陷等级 |
| 1 | 所有模块均无无异常处理机制 | 致命 |
| 2 | 系统未能实现正确识别3个以上人数密集边缘区域的地点位置信息和人流量（即人头数） | 致命 |
| 3 | 系统未能实现相应排序 | 致命 |
| 4 | 系统未能实现边缘端程序和云监管平台两个程序分开独立部署 | 致命 |
| 5 | 系统未能满足边缘计算云边协同架构的实际需求 | 致命 |
| 6 | 系统未能实现识别地点信息功能 | 致命 |
| 7 | 系统未能实现人头数上报功能 | 致命 |
| 8 | 系统未能实现异常数据上报功能 | 致命 |
| 9 | 系统未能数据告警功能 | 致命 |
| 10 | 系统未能实现展示人流量数据实时变化功能 | 致命 |
| 11 | 系统未能实现报警自动解除功能 | 致命 |
| 12 | 系统未能实现数据存储功能 | 致命 |
| 13 | 系统未能实现数据分析和分类功能 | 致命 |
| 14 | 系统未能突出经常容易出现人流异常爆发的地点 | 致命 |
| 15 | 系统未能匹配给出疏导处理的相关保安组织人员推荐信息和联系方式 | 致命 |
| 16 | 系统未能支持web端和手机移动端良好体验 | 致命 |
| 17 | 系统不能正确处理avi格式的视频 | 严重 |
| 18 | 系统不能正确处理mov格式的视频 | 严重 |
| 19 | 系统不能正确处理gif格式的视频 | 严重 |
| 20 | 系统不能正确加载jpg格式的图片 | 严重 |
| 21 | 系统不能正确加载png格式的图片 | 严重 |
| 22 | 系统不能正确显示bmp格式的图片 | 严重 |
| 23 | 系统不能正确显示jpeg格式的图片 | 严重 |
| 24 | 系统不能正确跟踪gif格式视频中的目标 | 严重 |
| 25 | 系统不能加载其他命名方式的图片文件 | 轻微 |
| 26 | 系统没有统一化图片、视频命名的模块，需要手动更改 | 轻微 |
| 27 | 系统启动速度慢 | 轻微 |
| 28 | 系统处理延迟大，无法完成实时处理需求 | 严重 |
| 29 | 系统没有与摄像设备匹配的接口 | 严重 |
| 30 | 系统未进行权限设置 | 严重 |
| 31 | 系统未进行数据加密 | 严重 |
| 32 | 系统没有备份机制 | 致命 |
| 33 | 输入用户名、密码后勾选保存密码下此仍需输入账号密码 | 轻微 |
| 34 | 在登陆界面点击忘记密码，无法跳转到密码找回界面 | 严重 |
| 35 | 在注册界面不勾选同意协议就点击注册按钮不会弹出警告 | 轻微 |
| 36 | 在注册界面点击登陆，无法返回登陆界面 | 严重 |
| 37 | 对于用户名和密码未设置输入规范格式 | 轻微 |
| 38 | 无法检测到邮箱@字符后的内容是否正确 | 轻微 |
| 39 | 登陆界面不输入用户名和密码也可登陆 | 致命 |
| 40 | 系统未连接数据库 | 致命 |
| 41 | 主页面导航栏点击无反应 | 致命 |

缺陷总体概况分析：

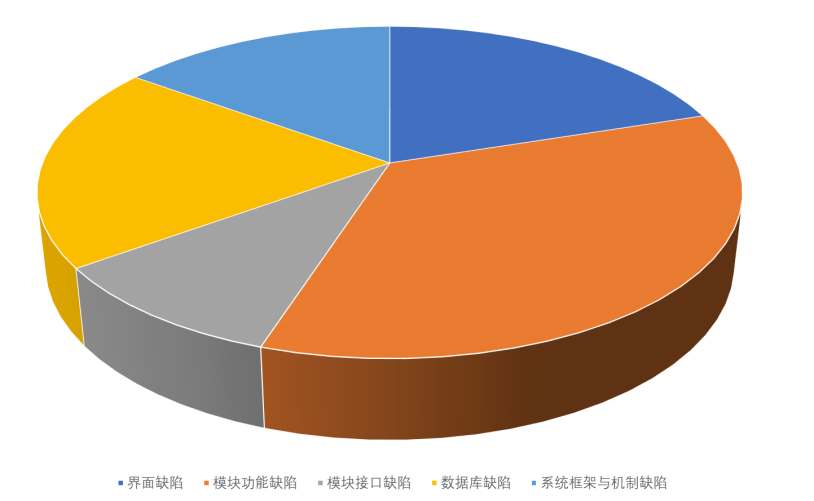


图 1 缺陷分布概况

由图1可知，近三分之一的缺陷分布在模块功能这一块，由此可见该系统的模块功能缺陷过多，经分析认为造成改现象的原因如下：

1. 需求说明书不细致、不完善。
2. 开发人员编码能力欠缺。
3. 开发人员之间沟通不足。

另外，界面的缺陷也占了总缺陷的20%左右，在测试过程中发现，该造成部分缺陷的主要原因为开发人员考虑的不周全、系统功能未完全开发完成这两方面原因导致的。而系统框架的缺陷则是由于开发人员基本没实现需求说明书中对于系统云端和边缘端的独立要求和云端未开发完成所导致的。

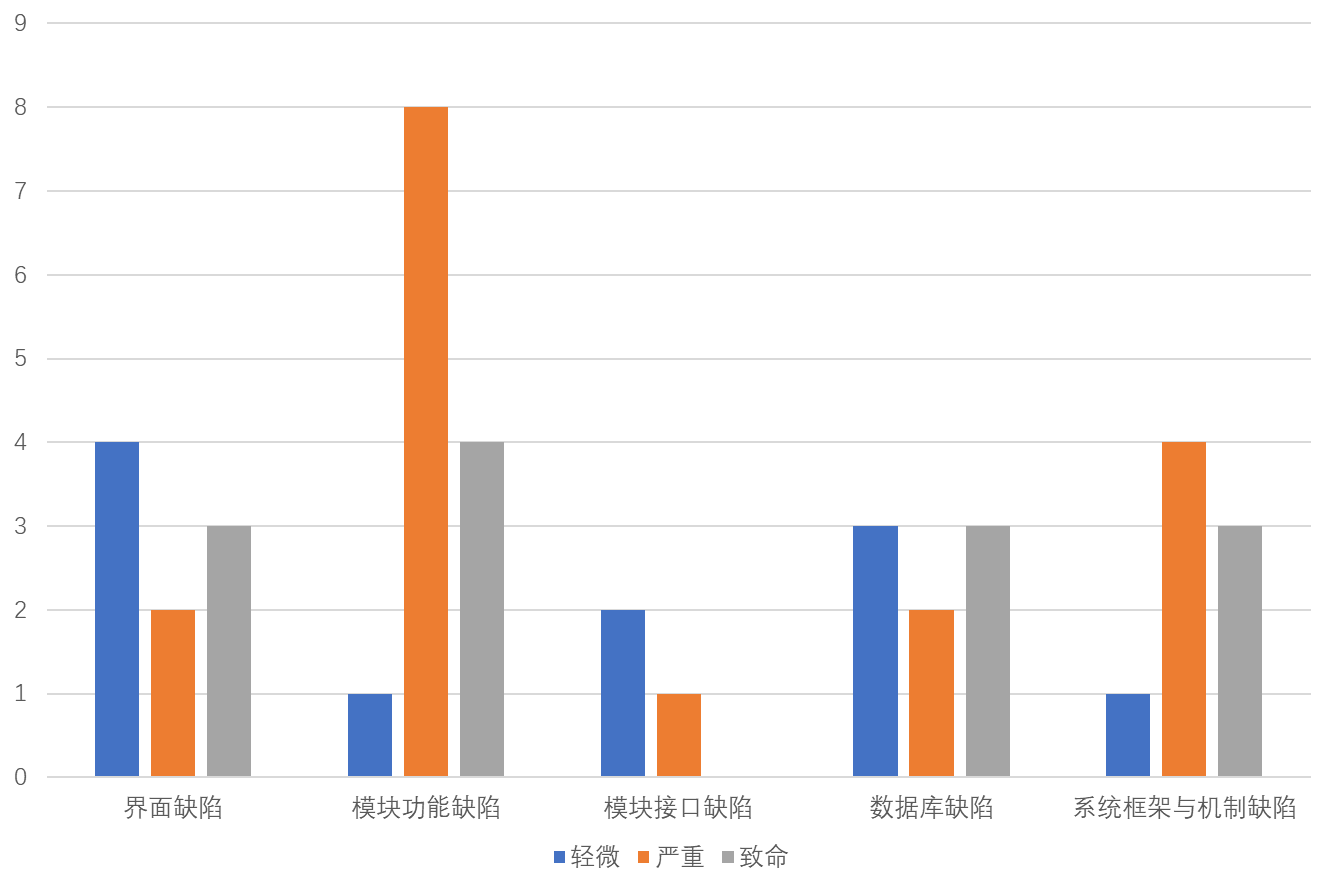


图 2 各部分缺陷级别分布情况

由图2 可知，模块功能部分的缺陷基本都为严重和致命缺陷，这表明开发人员在进行模块功能开发时不够细致、需求说明书对于模块功能的说明不够完备与全面。模块接口部分的缺陷相比其他部分的缺陷时程度最轻的。而界面、数据库、系统框架与机制这三个部分的缺陷的级别分布较平均

总体来说，通过图1和图2的分析可以看出该系统的缺陷还是比较多的，经过分析发现造成这种原因的最主要原因未：

1. 需求说明书不完备、不细致，导致模块功能缺陷多。
2. 规格需求不完善，导致出现部分模块接口缺陷。
3. 详细设计缺少界面设计，导致界面缺陷较多。
4. 概要设计没有数据库方面的内容，因此导致出现数据库缺陷。
5. 概要设计、详细设计、需求说明书等一系列文档的不完备，和开发人员能力不足等一系列原因综合导致了系统框架与机制的部分缺陷，并直接导致了另外四个部分大部分缺陷的出现。

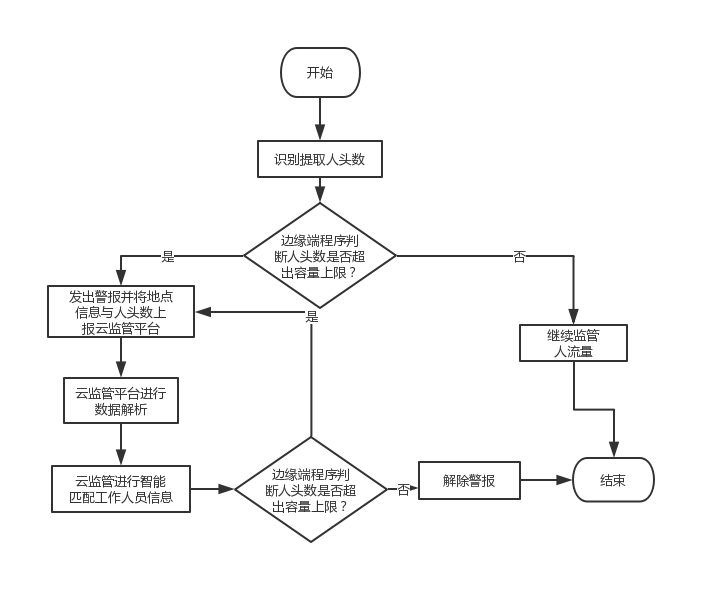
## 4.5系统测试

### 4.5.1测试图表

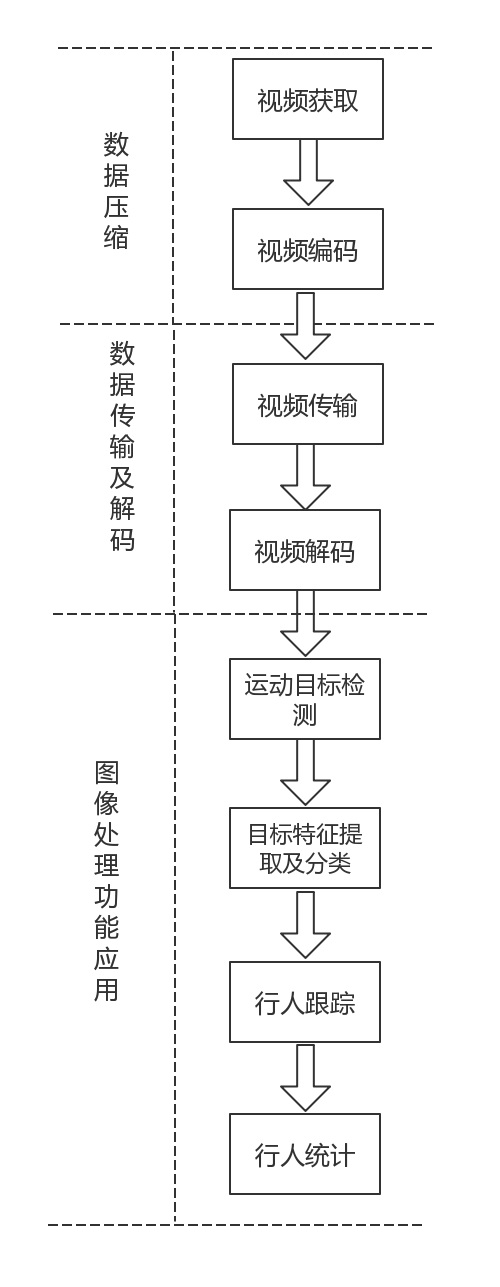
测试框图

### 

测试流程图



程序模块结构图



### 4.5.2测试内容

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 内容 | 输入 | 处理 | 输出 | 说明 |
| 识别提取地点特征和人头数 | 摄像机获取的视频图像 | ①检查输入数据的有效性；  ②如果合法，则对图像进行处理；否则不理会。  ③获取图像的地点特征及人头数 | 输出视频图像中的地点特征及人头数 | 获取某一视频图像中的地点特征及人头数；采用人脸识别、Mysql、图像处理等技术 |
| 边缘端计算人流数据 | 视频图像中的地点特征及人头数 | 人头数是否超过阙值，如果超过，则数据上报到云端；否则继续监测 | 上传数据到云监管平台 | Sobel边缘检测算法 |
| 云监管平台实时监控 | 边缘端人头数计算、地点特征信息 | 超过阙值则报警，排序重点突出经常容易出现人流异常爆发的地点，只能匹配给出疏导处理的相关安保组织人员推荐信息和联系方式 | 对应地点和人头数爆发的数据告警；匹配出给出的联系保安人员推荐信息和联系方式 |  |

|  |  |  |
| --- | --- | --- |
| 编号 | 内容 | 结果 |
| 1 | 在命令输入界面上输入，由系统在后台执行，并将结果反映到前台界面或者特定的文件内 | F |
| 2 | 采用图形化的操作界面 | P |
| 3 | 支持windows10、Android8.0及以上 | P |
| 4 | 客户端要求触摸发生动作 | F |
| 5 | 页面自适应，当页面在客户端及PC端切换时，要求宽度比自适应。 | F |
| 6 | 若操作人员发出意外动作时，数据刷新或重新选择动作 | F |
| 7 | 若发生无效操作时，尝试再次发生动作或等待系统刷新或重启系统。  容灾能力 | F |
| 8 | 设置权限问题 | F |
| 9 | AES加密算法保护数据 | F |
| 10 | 密码技术进行加密 | F |
| 11 | 记录或者历史数据集 | F |
| 12 | 授权的用户才能修改信息 | F |
| 13 | 漏洞检测、识别检测对象的系统资源 | F |
| 14 | 设备备份机制、容错机制 | F |
| 15 | 防病毒措施 | F |
| 16 | 安全系统网络与过程控制信息系统网络隔离开 | F |
| 17 | 正确识别3个以上人数密集边缘区域的地点位置信息和人流量 | F |
| 18 | 计算机视觉算法R-CNN | F |
| 19 | 云边协同架构 | F |
| 20 | 数据上传 | F |
| 21 | 页面长时间无响应，提出警告，要求用户刷新 | F |
| 22 | 页面载入时间过长，要有提示 | F |

# 5测试记录

1. 执行测试的日期、时间和地点
2. 对于每个测试的软硬件配置

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 测试人 | 日期 | 测试模块 |
| 1 | 黄新昊 | 2019.06.26 | 模型加载 |
| 2 | 陈硕峰 | 2019.06.18 | 图片加载 |
| 3 | 黄新昊 | 2019.06.18 | 图片定位 |
| 4 | 黄新昊 | 2019.06.18 | 图片处理 |
| 5 | 黄新昊 | 2019.06.18 | 图片展示 |
| 6 | 黄新昊 | 2019.06.18 | 视频处理 |
| 7 | 黄新昊 | 2019.06.19 | 行人移动位置预测 |
| 8 | 陈硕峰 | 2019.06.20 | 目标跟踪 |
| 9 | 陈硕峰 | 2019.06.20 | 视频识别结果保存 |
| 10 | 陈硕峰 | 2019.06.21 | 界面测试 |
| 11 | 黄新昊 | 2019.06.24 | 行人标识记录 |
| 12 | 黄新昊 | 2019.06.18 | 行人目标跟踪 |
| 13 | 陈硕峰 | 2019.06.18 | 视频识别结果保存 |
| 14 | 黄新昊 | 2019.06.26 | 有效性测试 |
| 15 | 陈硕峰 | 2019.06.28 | 系统测试 |

# 6评价

## 6.1能力

### 6.1.1功能

系统未能实现需求规格说明书中的以下功能需求：

1. 正确识别3个以上人数密集边缘区域的地点位置信息和人流量（即人头数）,并做相应排序；
2. 边缘端程序和云监管平台两个程序分开独立部署,满足边缘计算云边协同架构的实际需求;
3. 从摄像头获取视频数据；
4. 边缘端程序识别人头数和地点信息,并做相关计算,如果人头数没有达到容量上限则数据不上报到云端，若人头数超出预置容量上限数，就将所在人流异常视频数据、地点信息和人头数上报到云监管平台；
5. 识别出的图片中的人数是否正确；
6. 在视频中人数改变时实时输出总人数；
7. 云监管平台呈现对应地点和人头数爆发的数据告警,边缘端人流数据实时变化,云端展示人流量数据也实时变化,如果人流量实时降低并小于阈值,报警自动解除;
8. 异常视频中用动态标注的方式标注人头序号,并识别地点信息和人头总数和告警；
9. 获取视频地点信息；
10. 异常视频数据、地点信息和人头数数据在云端进行数据存储,云监管平台可以查询;对多个异常地点进行数据分析，并分类，排序重点突出经常容易出现人流异常爆发的地点；
11. 云监管平台实时展示地点信息和人头数数据，智能匹配给出疏导处理的相关保安组织人员推荐信息和联系方式;
12. 同时支持web端和手机移动端良好体验；

### 6.1.2性能

系统无法稳定运行。

### 6.1.3易用性

前端页面只有登录注册功能，无法向操作人员展示视频信息，无法实现展示边缘端程序人头数和地点信息,无法展示人流异常视频数据、地点信息和人头数。系统无法实时展示地点信息和人头数数据，智能匹配给出疏导处理的相关保安组织人员推荐信息和联系方式;

### 6.1.4安全性

系统没有采取有效措施保证用户信息安全、没有利用可靠的密码技术进行加密、没有历史数据集、没有漏洞检测、没有设置设备备份机制、容错机制，防止在系统出现单点失败时，系统的备份机制保证系统的正常运行、没有好防病毒措施，随时注意杀毒软件是否开启，及时在线升级杀毒软件、没有将安全系统网络与过程控制信息系统网络隔离开。

## 6.2缺陷和限制

|  |  |
| --- | --- |
| 序号 | 缺陷 |
| 1 | 所有模块均无无异常处理机制 |
| 2 | 系统未能实现正确识别3个以上人数密集边缘区域的地点位置信息和人流量（即人头数） |
| 3 | 系统未能实现相应排序 |
| 4 | 系统未能实现边缘端程序和云监管平台两个程序分开独立部署 |
| 5 | 系统未能满足边缘计算云边协同架构的实际需求 |
| 6 | 系统未能实现识别地点信息功能 |
| 7 | 系统未能实现人头数上报功能 |
| 8 | 系统未能实现异常数据上报功能 |
| 9 | 系统未能数据告警功能 |
| 10 | 系统未能实现展示人流量数据实时变化功能 |
| 11 | 系统未能实现报警自动解除功能 |
| 12 | 系统未能实现数据存储功能 |
| 13 | 系统未能实现数据分析和分类功能 |
| 14 | 系统未能突出经常容易出现人流异常爆发的地点 |
| 15 | 系统未能匹配给出疏导处理的相关保安组织人员推荐信息和联系方式 |
| 16 | 系统未能支持web端和手机移动端良好体验 |
| 17 | 系统不能正确处理avi格式的视频 |
| 18 | 系统不能正确处理mov格式的视频 |
| 19 | 系统不能正确处理gif格式的视频 |
| 20 | 系统不能正确加载gif格式的图片 |
| 21 | 系统不能正确加载bmp格式的图片 |
| 22 | 系统不能正确显示bmp格式的图片 |
| 23 | 系统不能正确显示gif格式的图片 |
| 24 | 系统不能正确跟踪gif格式视频中的目标 |
| 25 | 系统不能加载其他命名方式的图片文件 |
| 26 | 系统没有统一化图片、视频命名的模块，需要手动更改 |
| 27 | 系统启动速度慢 |
| 28 | 系统处理延迟大，无法完成实时处理需求 |
| 29 | 系统没有与摄像设备匹配的接口 |
| 30 | 系统未进行权限设置 |
| 31 | 系统未进行数据加密 |
| 32 | 系统没有备份机制 |

|  |  |
| --- | --- |
| 33 | 命令输入界面上输入较难操作 |
| 34 | 操作界面不友好 |
| 35 | 不支持手机端良好体验 |
| 36 | 没有实现客户端要求触摸发生动作 |
| 37 | 没有实现页面自适应，当页面在客户端及PC端切换时，要求宽度比自适应。 |
| 38 | 若操作人员发出意外动作时，没有实现数据刷新或重新选择动作 |
| 39 | 系统无容灾能力 |
| 40 | 系统无权限设置 |
| 41 | 系统未能运用AES加密算法保护数据 |
| 42 | 系统没有密码技术进行加密 |
| 43 | 系统没有记录或者历史数据集 |
| 44 | 系统没有做到授权的用户才能修改信息 |
| 45 | 系统没有漏洞检测、识别检测对象的系统资源 |
| 46 | 系统没有设备备份机制 |
| 47 | 系统没有防病毒措施 |
| 48 | 安全系统网络与过程控制信息系统网络隔离开 |
| 49 | 正确识别3个以上人数密集边缘区域的地点位置信息和人流量 |
| 50 | 系统没有容错机制 |
| 51 | 没有实现云边协同架构 |
| 52 | 没有实现数据上传 |
| 53 | 页面长时间无响应，没有提出警告，要求用户刷新 |
| 54 | 页面载入时间过长，无提示 |

## 6.3建议

|  |  |
| --- | --- |
| 序号 | 建议 |
| 1 | 系统无异常处理机制，应增加异常处理机制 |
| 2 | 系统未能实现正确识别3个以上人数密集边缘区域的地点位置信息和人流量（即人头数） |
| 3 | 系统应实现相应排序 |
| 4 | 边缘端程序和云监管平台两个程序分开独立部署 |
| 5 | 系统应满足边缘计算云边协同架构的实际需求 |
| 6 | 系统应实现识别地点信息功能 |
| 7 | 系统应实现人头数上报功能 |
| 8 | 系统应实现异常数据上报功能 |
| 9 | 系统应实现数据告警功能 |
| 10 | 系统应实现展示人流量数据实时变化功能 |
| 11 | 系统应实现报警自动解除功能 |
| 12 | 系统应实现数据存储功能 |
| 13 | 系统应实现数据分析和分类功能 |
| 14 | 系统应突出经常容易出现人流异常爆发的地点 |
| 15 | 系统应匹配给出疏导处理的相关保安组织人员推荐信息和联系方式 |
| 16 | 系统应支持web端和手机移动端良好体验 |
| 17 | 系统应能处理常用格式（如avi、mov、gif等）格式的视频 |
| 18 | 系统应能加载常用格式（如jpg、jpeg、png等）格式的图片 |
| 19 | 系统不能正确跟踪gif格式视频中的目标 |
| 20 | 系统只能加载特定命名方式的图片文件 |
| 21 | 系统应统一化图片、视频命名的模块，需要手动更改 |
| 22 | 系统启动速度应满足用户需求 |
| 23 | 系统处理延迟大，无法完成实时处理需求 |
| 24 | 系统没有与摄像设备匹配的接口 |
| 25 | 系统应进行权限设置 |
| 26 | 系统应进行数据加密 |
| 27 | 系统应有备份机制 |
| 28 | 在命令输入界面上输入不易操作 |
| 29 | 系统不支持手机端良好体验 |
| 30 | 系统应实现客户端要求触摸发生动作 |
| 31 | 页面应做到自适应 |
| 32 | 若操作人员发出意外动作时，应有数据刷新或重新选择动作 |
| 33 | 若发生无效操作时，尝试再次发生动作或等待系统刷新或重启系统。 |
| 34 | 系统应能记录或者历史数据集 |
| 35 | 系统应能实现授权的用户才能修改信息 |
| 36 | 系统应有漏洞检测机制 |
| 37 | 系统应有防病毒措施 |
| 38 | 页面长时间无响应时，提出警告 |
| 39 | 页面载入时间过长时，要有提示 |

## 6.4结论

**一、系统未能实现需求规格说明书中的功能需求**

1. 正确识别3个以上人数密集边缘区域的地点位置信息和人流量（即人头数）,并做相应排序；
2. 边缘端程序和云监管平台两个程序分开独立部署,满足边缘计算云边协同架构的实际需求;
3. 从摄像头获取视频数据；
4. 边缘端程序识别人头数和地点信息,并做相关计算,如果人头数没有达到容量上限则数据不上报到云端，若人头数超出预置容量上限数，就将所在人流异常视频数据、地点信息和人头数上报到云监管平台；
5. 识别出的图片中的人数是否正确；
6. 在视频中人数改变时实时输出总人数；
7. 云监管平台呈现对应地点和人头数爆发的数据告警,边缘端人流数据实时变化,云端展示人流量数据也实时变化,如果人流量实时降低并小于阈值,报警自动解除;
8. 异常视频中用动态标注的方式标注人头序号,并识别地点信息和人头总数和告警；
9. 获取视频地点信息；
10. 异常视频数据、地点信息和人头数数据在云端进行数据存储,云监管平台可以查询;对多个异常地点进行数据分析，并分类，排序重点突出经常容易出现人流异常爆发的地点；
11. 云监管平台实时展示地点信息和人头数数据，智能匹配给出疏导处理的相关保安组织人员推荐信息和联系方式;
12. 同时支持web端和手机移动端良好体验；

**二、系统未能实现需求规格说明书中的接口需求**

用户接口、命令接口、程序接口、图形接口、硬件接口、软件接口、通信接口。将向用户提供的命令和它们的语法结构，以及软件的回答信息，提供用户使用软件产品实地的接口需求，软件产品和系统硬部件之间每一个接口的逻辑特点，包括信息的传递方式、响应时间和精度要求。

**三、系统未能实现需求规格说明书中的适应性需求、保密性需求。**

系统没有容灾能力、没有设置权限问题、没有采取有效措施保证用户信息安全、没有利用可靠的密码技术进行加密、没有历史数据集、没有漏洞检测、没有设置设备备份机制、容错机制，防止在系统出现单点失败时，系统的备份机制保证系统的正常运行、没有好防病毒措施，随时注意杀毒软件是否开启，及时在线升级杀毒软件、没有将安全系统网络与过程控制信息系统网络隔离开。

**四、系统操作不便，基本采取命令行方式，前端页面只有登录注册页面，没有支持电脑浏览器和手机端良好体验。**

# 7附录

## 附录1：测试工具介绍

PyCharm：PyCharm是一种Python IDE，带有一整套可以帮助用户在使用Python语言开发时提高其效率的工具，比如调试、语法高亮、Project管理、代码跳转、智能提示、自动完成、单元测试、版本控制。此外，该IDE提供了一些高级功能，以用于支持Django框架下的专业Web开发。

Selenium IDE：这个是selenium家族中的一个组件。Selenium IDE是用来做浏览器的自动化的，就是模拟人对浏览器的操作，例如输入，点击按钮等，还可以导出java、python等语言的测试脚本。

pytest是一个非常成熟的全功能的Python测试框架，是Python最流程的单测框架之一。主要特点有以下几点：

1、简单灵活，容易上手，文档丰富；

2、支持参数化，可以细粒度地控制要测试的测试用例；

3、能够支持简单的单元测试和复杂的功能测试，还可以用来做selenium/appnium等自动化测试、接口自动化测试（pytest+requests）;

4、pytest具有很多第三方插件，并且可以自定义扩展，比较好用的如pytest-selenium（集成selenium）、pytest-html（完美html测试报告生成）、pytest-rerunfailures（失败case重复执行）、pytest-xdist（多CPU分发）等；

5、测试用例的skip和xfail处理；

6、可以很好的和CI工具结合，例如jenkins

pytest支持以xUnit格式型的测试模型(setup/teardown)，如下：

模块形式----使用setup\_module/teardown\_module

函数形式----使用setup\_function/teardown\_function

类形式----使用setup\_class/teardown\_class

方法形式---使用setup\_method/teardown\_metho

## 附录2：测试程序

1. 模块测试：

# 测试文件：camshift2

# 对已经标记的人进行目标跟踪，并不断迭代更新行人目标位置并实时标记

def test\_camshift2\_demo1():

cap = cv2.VideoCapture('test\_videos/street.mp4')

print("mp4")

# 获取视频的第一个帧

ret, frame = cap.read()

# 设置窗口的初始位置

x, y, w, h = 192, 206, 18, 120

track\_window = (x, y, w, h)

# 设置跟踪的区域roi

roi = frame[y:y + h, x:x + w]

hsv\_roi = cv2.cvtColor(roi, cv2.COLOR\_BGR2HSV)

mask = cv2.inRange(hsv\_roi, np.array((0., 60., 32,)), np.array((180., 255., 255.)))

roi\_hist = cv2.calcHist([hsv\_roi], [0], mask, [180], [0, 180])

cv2.normalize(roi\_hist, roi\_hist, 0, 255, cv2.NORM\_MINMAX)

# 设置迭代次数，或者迭代10次或者至少移动1次

term\_crit = (cv2.TERM\_CRITERIA\_EPS | cv2.TERM\_CRITERIA\_COUNT, 10, 1)

while True:

ret, frame = cap.read()

# print(type(frame))

if ret == True:

hsv = cv2.cvtColor(frame, cv2.COLOR\_BGR2HSV)

dst = cv2.calcBackProject([hsv], [0], roi\_hist, [0, 180], 1)

# 对新的位置进行meanShift操作

ret, track\_window = cv2.CamShift(dst, track\_window, term\_crit)

# 在图上进行绘制

pts = cv2.boxPoints(ret)

pts = np.int0(pts)

img = cv2.polylines(frame, [pts], True, 255, 2)

cv2.imshow("img", img)

k = cv2.waitKey(60) & 0xff

if k == 27:

break

else:

break

cap.release()

cv2.destroyAllWindows()

def test\_camshift2\_demo2():

cap = cv2.VideoCapture('test\_videos/street.avi')

print("avi")

# 获取视频的第一个帧

ret, frame = cap.read()

# 设置窗口的初始位置

x, y, w, h = 192, 206, 18, 120

track\_window = (x, y, w, h)

# 设置跟踪的区域roi

roi = frame[y:y + h, x:x + w]

hsv\_roi = cv2.cvtColor(roi, cv2.COLOR\_BGR2HSV)

mask = cv2.inRange(hsv\_roi, np.array((0., 60., 32,)), np.array((180., 255., 255.)))

roi\_hist = cv2.calcHist([hsv\_roi], [0], mask, [180], [0, 180])

cv2.normalize(roi\_hist, roi\_hist, 0, 255, cv2.NORM\_MINMAX)

# 设置迭代次数，或者迭代10次或者至少移动1次

term\_crit = (cv2.TERM\_CRITERIA\_EPS | cv2.TERM\_CRITERIA\_COUNT, 10, 1)

while True:

ret, frame = cap.read()

# print(type(frame))

if ret == True:

hsv = cv2.cvtColor(frame, cv2.COLOR\_BGR2HSV)

dst = cv2.calcBackProject([hsv], [0], roi\_hist, [0, 180], 1)

# 对新的位置进行meanShift操作

ret, track\_window = cv2.CamShift(dst, track\_window, term\_crit)

# 在图上进行绘制

pts = cv2.boxPoints(ret)

pts = np.int0(pts)

img = cv2.polylines(frame, [pts], True, 255, 2)

cv2.imshow("img", img)

k = cv2.waitKey(60) & 0xff

if k == 27:

break

else:

break

cap.release()

cv2.destroyAllWindows()

def test\_camshift2\_demo3():

cap = cv2.VideoCapture('test\_videos/street.mov')

print("mov")

# 获取视频的第一个帧

ret, frame = cap.read()

# 设置窗口的初始位置

x, y, w, h = 192, 206, 18, 120

track\_window = (x, y, w, h)

# 设置跟踪的区域roi

roi = frame[y:y + h, x:x + w]

hsv\_roi = cv2.cvtColor(roi, cv2.COLOR\_BGR2HSV)

mask = cv2.inRange(hsv\_roi, np.array((0., 60., 32,)), np.array((180., 255., 255.)))

roi\_hist = cv2.calcHist([hsv\_roi], [0], mask, [180], [0, 180])

cv2.normalize(roi\_hist, roi\_hist, 0, 255, cv2.NORM\_MINMAX)

# 设置迭代次数，或者迭代10次或者至少移动1次

term\_crit = (cv2.TERM\_CRITERIA\_EPS | cv2.TERM\_CRITERIA\_COUNT, 10, 1)

while True:

ret, frame = cap.read()

# print(type(frame))

if ret == True:

hsv = cv2.cvtColor(frame, cv2.COLOR\_BGR2HSV)

dst = cv2.calcBackProject([hsv], [0], roi\_hist, [0, 180], 1)

# 对新的位置进行meanShift操作

ret, track\_window = cv2.CamShift(dst, track\_window, term\_crit)

# 在图上进行绘制

pts = cv2.boxPoints(ret)

pts = np.int0(pts)

img = cv2.polylines(frame, [pts], True, 255, 2)

cv2.imshow("img", img)

k = cv2.waitKey(60) & 0xff

if k == 27:

break

else:

break

cap.release()

cv2.destroyAllWindows()

def test\_camshift2\_demo4():

cap = cv2.VideoCapture('test\_videos/street.gif')

print("gif")

# 获取视频的第一个帧

ret, frame = cap.read()

# 设置窗口的初始位置

x, y, w, h = 192, 206, 18, 120

track\_window = (x, y, w, h)

# 设置跟踪的区域roi

roi = frame[y:y + h, x:x + w]

hsv\_roi = cv2.cvtColor(roi, cv2.COLOR\_BGR2HSV)

mask = cv2.inRange(hsv\_roi, np.array((0., 60., 32,)), np.array((180., 255., 255.)))

roi\_hist = cv2.calcHist([hsv\_roi], [0], mask, [180], [0, 180])

cv2.normalize(roi\_hist, roi\_hist, 0, 255, cv2.NORM\_MINMAX)

# 设置迭代次数，或者迭代10次或者至少移动1次

term\_crit = (cv2.TERM\_CRITERIA\_EPS | cv2.TERM\_CRITERIA\_COUNT, 10, 1)

while True:

ret, frame = cap.read()

# print(type(frame))

if ret == True:

hsv = cv2.cvtColor(frame, cv2.COLOR\_BGR2HSV)

dst = cv2.calcBackProject([hsv], [0], roi\_hist, [0, 180], 1)

# 对新的位置进行meanShift操作

ret, track\_window = cv2.CamShift(dst, track\_window, term\_crit)

# 在图上进行绘制

pts = cv2.boxPoints(ret)

pts = np.int0(pts)

img = cv2.polylines(frame, [pts], True, 255, 2)

cv2.imshow("img", img)

k = cv2.waitKey(60) & 0xff

if k == 27:

break

else:

break

cap.release()

cv2.destroyAllWindows()

#####################################################

#测试代码

# 测试用例名：test\_load

# 测试是否能够将训练好的模型及标签加载到内存

def test\_load\_demo1():

print("\n+++++++++++++++++++++++++++++++++")

print("load()函数返回结果为：")

print(load())

print("+++++++++++++++++++++++++++++++++")

# 测试实例名：test\_load\_image\_into\_numpy\_array\_demo1

# 测试是否定义session加载待测试的图片文件

def test\_load\_image\_into\_numpy\_array\_demo1():

image\_path = "test\_images/image3.jpg"

image = Image.open(image\_path)

image\_np = load\_image\_into\_numpy\_array(image)

print("\n+++++++++++++++++++++++++++++++++")

print("load\_image\_into\_numpy\_array()函数返回结果为：")

print(image\_np)

print("+++++++++++++++++++++++++++++++++")

def test\_load\_image\_into\_numpy\_array\_demo2():

image\_path = "test\_images/image1.png"

image = Image.open(image\_path)

image\_np = load\_image\_into\_numpy\_array(image)

print("\n+++++++++++++++++++++++++++++++++")

print("load\_image\_into\_numpy\_array()函数返回结果为：")

print(image\_np)

print("+++++++++++++++++++++++++++++++++")

def test\_load\_image\_into\_numpy\_array\_demo3():

image\_path = "test\_images/image1.gif"

image = Image.open(image\_path)

image\_np = load\_image\_into\_numpy\_array(image)

print("\n+++++++++++++++++++++++++++++++++")

print("load\_image\_into\_numpy\_array()函数返回结果为：")

print(image\_np)

print("+++++++++++++++++++++++++++++++++")

def test\_load\_image\_into\_numpy\_array\_demo4():

image\_path = "test\_images/image1.bmp"

image = Image.open(image\_path)

image\_np = load\_image\_into\_numpy\_array(image)

print("\n+++++++++++++++++++++++++++++++++")

print("load\_image\_into\_numpy\_array()函数返回结果为：")

print(image\_np)

print("+++++++++++++++++++++++++++++++++")

def test\_load\_image\_into\_numpy\_array\_demo5():

image\_path = "test\_images/dog.jpg"

image = Image.open(image\_path)

image\_np = load\_image\_into\_numpy\_array(image)

print("\n+++++++++++++++++++++++++++++++++")

print("load\_image\_into\_numpy\_array()函数返回结果为：")

print(image\_np)

print("+++++++++++++++++++++++++++++++++")

# 测试用例名：test\_detect\_objects\_demo1

# 测试能否对原始图片进行目标检测定位

def test\_detect\_objects\_demo1():

image = Image.open("test\_images/image3.jpg")

plt.figure(figsize=IMAGE\_SIZE)

plt.subplot(1, 2, 1)

plt.imshow(image)

image\_np = load\_image\_into\_numpy\_array(image)

category\_index = load()

detection\_graph = tf.get\_default\_graph()

with detection\_graph.as\_default():

with tf.Session(graph=detection\_graph) as sess:

image\_process = detect\_objects(image\_np, sess, detection\_graph, category\_index)

print("\n+++++++++++++++++++++++++++++++++")

print("load\_image\_into\_numpy\_array()函数返回结果为：")

print(image\_process)

print("+++++++++++++++++++++++++++++++++")

def test\_detect\_objects\_demo2():

image = Image.open("test\_images/image1.png")

plt.figure(figsize=IMAGE\_SIZE)

plt.subplot(1, 2, 1)

plt.imshow(image)

image\_np = load\_image\_into\_numpy\_array(image)

category\_index = load()

detection\_graph = tf.get\_default\_graph()

with detection\_graph.as\_default():

with tf.Session(graph=detection\_graph) as sess:

image\_process = detect\_objects(image\_np, sess, detection\_graph, category\_index)

print("\n+++++++++++++++++++++++++++++++++")

print("load\_image\_into\_numpy\_array()函数返回结果为：")

print(image\_process)

print("+++++++++++++++++++++++++++++++++")

def test\_detect\_objects\_demo3():

image = Image.open("test\_images/image1.gif")

plt.figure(figsize=IMAGE\_SIZE)

plt.subplot(1, 2, 1)

plt.imshow(image)

image\_np = load\_image\_into\_numpy\_array(image)

category\_index = load()

detection\_graph = tf.get\_default\_graph()

with detection\_graph.as\_default():

with tf.Session(graph=detection\_graph) as sess:

image\_process = detect\_objects(image\_np, sess, detection\_graph, category\_index)

print("\n+++++++++++++++++++++++++++++++++")

print("load\_image\_into\_numpy\_array()函数返回结果为：")

print(image\_process)

print("+++++++++++++++++++++++++++++++++")

def test\_detect\_objects\_demo4():

image = Image.open("test\_images/image1.bmp")

plt.figure(figsize=IMAGE\_SIZE)

plt.subplot(1, 2, 1)

plt.imshow(image)

image\_np = load\_image\_into\_numpy\_array(image)

category\_index = load()

detection\_graph = tf.get\_default\_graph()

with detection\_graph.as\_default():

with tf.Session(graph=detection\_graph) as sess:

image\_process = detect\_objects(image\_np, sess, detection\_graph, category\_index)

print("\n+++++++++++++++++++++++++++++++++")

print("load\_image\_into\_numpy\_array()函数返回结果为：")

print(image\_process)

print("+++++++++++++++++++++++++++++++++")

def test\_detect\_objects\_demo5():

image = Image.open("test\_images/image4.jpg")

plt.figure(figsize=IMAGE\_SIZE)

plt.subplot(1, 2, 1)

plt.imshow(image)

image\_np = load\_image\_into\_numpy\_array(image)

category\_index = load()

detection\_graph = tf.get\_default\_graph()

with detection\_graph.as\_default():

with tf.Session(graph=detection\_graph) as sess:

image\_process = detect\_objects(image\_np, sess, detection\_graph, category\_index)

print("\n+++++++++++++++++++++++++++++++++")

print("load\_image\_into\_numpy\_array()函数返回结果为：")

print(image\_process)

print("+++++++++++++++++++++++++++++++++")

def test\_detect\_objects\_demo6():

image = Image.open("test\_images/image5.jpg")

plt.figure(figsize=IMAGE\_SIZE)

plt.subplot(1, 2, 1)

plt.imshow(image)

image\_np = load\_image\_into\_numpy\_array(image)

category\_index = load()

detection\_graph = tf.get\_default\_graph()

with detection\_graph.as\_default():

with tf.Session(graph=detection\_graph) as sess:

image\_process = detect\_objects(image\_np, sess, detection\_graph, category\_index)

print("\n+++++++++++++++++++++++++++++++++")

print("load\_image\_into\_numpy\_array()函数返回结果为：")

print(image\_process)

print("+++++++++++++++++++++++++++++++++")

def test\_detect\_objects\_demo7():

image = Image.open("test\_images/image6.jpg")

plt.figure(figsize=IMAGE\_SIZE)

plt.subplot(1, 2, 1)

plt.imshow(image)

image\_np = load\_image\_into\_numpy\_array(image)

category\_index = load()

detection\_graph = tf.get\_default\_graph()

with detection\_graph.as\_default():

with tf.Session(graph=detection\_graph) as sess:

image\_process = detect\_objects(image\_np, sess, detection\_graph, category\_index)

print("\n+++++++++++++++++++++++++++++++++")

print("load\_image\_into\_numpy\_array()函数返回结果为：")

print(image\_process)

print("+++++++++++++++++++++++++++++++++")

def test\_detect\_objects\_demo8():

image = Image.open("test\_images/image4.jpg")

plt.figure(figsize=IMAGE\_SIZE)

plt.subplot(1, 2, 1)

plt.imshow(image)

image\_np = load\_image\_into\_numpy\_array(image)

category\_index = load()

detection\_graph = tf.get\_default\_graph()

with detection\_graph.as\_default():

with tf.Session(graph=detection\_graph) as sess:

image\_process = detect\_objects(image\_np, sess, detection\_graph, category\_index)

print("\n+++++++++++++++++++++++++++++++++")

print("load\_image\_into\_numpy\_array()函数返回结果为：")

print(image\_process)

print("+++++++++++++++++++++++++++++++++")

def test\_detect\_objects\_demo9():

image = Image.open("test\_images/image5.jpg")

plt.figure(figsize=IMAGE\_SIZE)

plt.subplot(1, 2, 1)

plt.imshow(image)

image\_np = load\_image\_into\_numpy\_array(image)

category\_index = load()

detection\_graph = tf.get\_default\_graph()

with detection\_graph.as\_default():

with tf.Session(graph=detection\_graph) as sess:

image\_process = detect\_objects(image\_np, sess, detection\_graph, category\_index)

print("\n+++++++++++++++++++++++++++++++++")

print("load\_image\_into\_numpy\_array()函数返回结果为：")

print(image\_process)

print("+++++++++++++++++++++++++++++++++")

def test\_detect\_objects\_demo10():

image = Image.open("test\_images/image6.jpg")

plt.figure(figsize=IMAGE\_SIZE)

plt.subplot(1, 2, 1)

plt.imshow(image)

image\_np = load\_image\_into\_numpy\_array(image)

category\_index = load()

detection\_graph = tf.get\_default\_graph()

with detection\_graph.as\_default():

with tf.Session(graph=detection\_graph) as sess:

image\_process = detect\_objects(image\_np, sess, detection\_graph, category\_index)

print("\n+++++++++++++++++++++++++++++++++")

print("load\_image\_into\_numpy\_array()函数返回结果为：")

print(image\_process)

print("+++++++++++++++++++++++++++++++++")

# 测试用例名：test\_process\_image\_demo1

# 测试能否对原始图片的处理

def test\_process\_image\_demo1():

image = Image.open("test\_images/image1.jpg")

plt.figure(figsize=IMAGE\_SIZE)

plt.subplot(1, 2, 1)

plt.imshow(image)

image\_np = load\_image\_into\_numpy\_array(image)

image\_process = process\_image(image\_np)

plt.subplot(1, 2, 2)

plt.imshow(image\_process)

plt.show()

print("\n+++++++++++++++++++++++++++++++++")

print("process\_image()函数返回结果为：")

print(image\_process)

print("+++++++++++++++++++++++++++++++++")

def test\_process\_image\_demo2():

image = Image.open("test\_images/image1.jpeg")

plt.figure(figsize=IMAGE\_SIZE)

plt.subplot(1, 2, 1)

plt.imshow(image)

image\_np = load\_image\_into\_numpy\_array(image)

image\_process = process\_image(image\_np)

plt.subplot(1, 2, 2)

plt.imshow(image\_process)

plt.show()

print("\n+++++++++++++++++++++++++++++++++")

print("process\_image()函数返回结果为：")

print(image\_process)

print("+++++++++++++++++++++++++++++++++")

def test\_process\_image\_demo3():

image = Image.open("test\_images/image1.png")

plt.figure(figsize=IMAGE\_SIZE)

plt.subplot(1, 2, 1)

plt.imshow(image)

image\_np = load\_image\_into\_numpy\_array(image)

image\_process = process\_image(image\_np)

plt.subplot(1, 2, 2)

plt.imshow(image\_process)

plt.show()

print("\n+++++++++++++++++++++++++++++++++")

print("process\_image()函数返回结果为：")

print(image\_process)

print("+++++++++++++++++++++++++++++++++")

def test\_process\_image\_demo4():

image = Image.open("test\_images/image1.gif")

plt.figure(figsize=IMAGE\_SIZE)

plt.subplot(1, 2, 1)

plt.imshow(image)

image\_np = load\_image\_into\_numpy\_array(image)

image\_process = process\_image(image\_np)

plt.subplot(1, 2, 2)

plt.imshow(image\_process)

plt.show()

print("\n+++++++++++++++++++++++++++++++++")

print("process\_image()函数返回结果为：")

print(image\_process)

print("+++++++++++++++++++++++++++++++++")

def test\_process\_image\_demo5():

image = Image.open("test\_images/image1.bmp")

plt.figure(figsize=IMAGE\_SIZE)

plt.subplot(1, 2, 1)

plt.imshow(image)

image\_np = load\_image\_into\_numpy\_array(image)

image\_process = process\_image(image\_np)

plt.subplot(1, 2, 2)

plt.imshow(image\_process)

plt.show()

print("\n+++++++++++++++++++++++++++++++++")

print("process\_image()函数返回结果为：")

print(image\_process)

print("+++++++++++++++++++++++++++++++++")

# 测试文件：detection

# 测试能否显示处理后的图片结果

#TEST\_IMAGE\_PATHS = [os.path.join(PATH\_TO\_TEST\_IMAGES\_DIR, 'image{}.jpg'.format(i))for i in range(1, 2)]

#TEST\_IMAGE\_PATHS = [os.path.join(PATH\_TO\_TEST\_IMAGES\_DIR, 'image{}.jpeg'.format(i))for i in range(1, 2)]

#TEST\_IMAGE\_PATHS = [os.path.join(PATH\_TO\_TEST\_IMAGES\_DIR, 'image{}.png'.format(i))for i in range(1, 2)]

#TEST\_IMAGE\_PATHS = [os.path.join(PATH\_TO\_TEST\_IMAGES\_DIR, 'image{}.gif'.format(i))for i in range(1, 2)]

#TEST\_IMAGE\_PATHS = [os.path.join(PATH\_TO\_TEST\_IMAGES\_DIR, 'image{}.bmp'.format(i))for i in range(1, 2)]

def test\_showImg\_demo1():

showImg()

# 测试用例名：test\_process\_video\_demo1

# 测试能否对视频进行识别标记

def test\_process\_video\_demo1():

video="test\_videos/street2.mp4"

process\_video(video)

def test\_process\_video\_demo2():

video="test\_videos/street2.mov"

process\_video(video)

def test\_process\_video\_demo3():

video="test\_videos/street2.avi"

process\_video(video)

def test\_process\_video\_demo4():

video="test\_videos/street2.gif"

process\_video(video)

def test\_process\_video\_demo5():

video="test\_videos/demo.mp4"

process\_video(video)

# detection模块测试

def test\_detection():

showImg()

# 测试文件：kalmon

# 测试该模块使用Kalmon算法是否能够提高跟踪准确度

def test\_kalmon():

# 加载保存好的小球移动文件

Measured = np.load("ballTrajectory.npy")

print(Measured.shape)

# print(Measured)

# 取出视频当中刚开始无小球的部分

while True:

if Measured[0, 0] == -1.:

Measured = np.delete(Measured, 0, 0)

else:

break

numMeas = Measured.shape[0]

print(Measured.shape)

# print(Measured)

# 使用卡尔曼滤波器来预测小球中间被阻挡住的位置

MarkedMeasure = np.ma.masked\_less(Measured, 0) # 屏蔽掉无坐标部分

# print(MarkedMeasure)

# 卡尔曼滤波器测量参数

Transition\_Matrix = [[1, 0, 1, 0], [0, 1, 0, 1], [0, 0, 1, 0], [0, 0, 0, 1]] # 转移矩阵

Observation\_Matrix = [[1, 0, 0, 0], [0, 1, 0, 0]] # 观察矩阵

# 其他参数

# 初始状态

xinit = MarkedMeasure[0, 0] # 当前位置的x坐标

yinit = MarkedMeasure[0, 1] # 当前位置的y坐标

vxinit = MarkedMeasure[1, 0] - MarkedMeasure[0, 0] # x方向的当前速度

vyinit = MarkedMeasure[1, 1] - MarkedMeasure[0, 1] # y方向的当前速度

initstate = [xinit, yinit, vxinit, vyinit]

initcovariance = 1.0e-3 \* np.eye(4) # 初始状态协方差，描述了初始状态的确定性

transistionCov = 1.0e-4 \* np.eye(4) # 过渡协方差，描述了过程模型的确定性

observationCov = 1.0e-1 \* np.eye(2) # 观测协方差，描述了观测模型的确定性

kf = KalmanFilter(transition\_matrices=Transition\_Matrix,

observation\_matrices=Observation\_Matrix,

initial\_state\_mean=initstate,

initial\_state\_covariance=initcovariance,

transition\_covariance=transistionCov,

observation\_covariance=observationCov)

# 通过调用KalmanFilter的filter()方法，计算轨道以及filtered\_state\_covariances的正确性

(filtered\_state\_means, filtered\_state\_covariance) = kf.filter(MarkedMeasure)

plt.plot(MarkedMeasure[:, 0], MarkedMeasure[:, 1], 'xr', label='measured')

plt.axis([0, 520, 360, 0])

# plt.hold(True)

plt.plot(filtered\_state\_means[:, 0], filtered\_state\_means[:, 1], 'ob', label='kalman output')

# plt.hold(True)

plt.legend(loc=3)

plt.title('Constant Velocity Kalman Filter')

plt.show()

1. 界面测试：

登陆功能

test\_demo1:

# Generated by Selenium IDE

import pytest

import time

import json

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.action\_chains import ActionChains

from selenium.webdriver.support import expected\_conditions

from selenium.webdriver.support.wait import WebDriverWait

from selenium.webdriver.common.keys import Keys

class TestDemo1():

def setup\_method(self, method):

self.driver = webdriver.Firefox()

self.vars = {}

def teardown\_method(self, method):

self.driver.quit()

def test\_demo1(self):

# Test name: demo1

# Step # | name | target | value | comment

# 1 | open | / | |

self.driver.get("http://127.0.0.1:8000/")

# 2 | setWindowSize | 1936x1096 | |

self.driver.set\_window\_size(1936, 1096)

# 3 | click | id=id\_username | |

self.driver.find\_element(By.ID, "id\_username").click()

# 4 | type | id=id\_username | 我Ab4$ |

self.driver.find\_element(By.ID, "id\_username").send\_keys("我Ab4$")

# 5 | click | id=id\_password | |

self.driver.find\_element(By.ID, "id\_password").click()

# 6 | type | id=id\_password | Df#2 |

self.driver.find\_element(By.ID, "id\_password").send\_keys("Df#2")

# 7 | click | css=.checkbox > label:nth-child(1) | |

self.driver.find\_element(By.CSS\_SELECTOR, ".checkbox > label:nth-child(1)").click()

# 8 | click | css=.btn | |

self.driver.find\_element(By.CSS\_SELECTOR, ".btn").click()

test\_demo2:

# Generated by Selenium IDE

import pytest

import time

import json

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.action\_chains import ActionChains

from selenium.webdriver.support import expected\_conditions

from selenium.webdriver.support.wait import WebDriverWait

from selenium.webdriver.common.keys import Keys

class TestDemo2():

def setup\_method(self, method):

self.driver = webdriver.Firefox()

self.vars = {}

def teardown\_method(self, method):

self.driver.quit()

def test\_demo2(self):

# Test name: demo2

# Step # | name | target | value | comment

# 1 | open | / | |

self.driver.get("http://127.0.0.1:8000/")

# 2 | setWindowSize | 1936x1096 | |

self.driver.set\_window\_size(1936, 1096)

# 3 | click | id=id\_username | |

self.driver.find\_element(By.ID, "id\_username").click()

# 4 | type | id=id\_username | 我Ab4$ |

self.driver.find\_element(By.ID, "id\_username").send\_keys("我Ab4$")

# 5 | click | id=id\_password | |

self.driver.find\_element(By.ID, "id\_password").click()

# 6 | type | id=id\_password | Df#2 |

self.driver.find\_element(By.ID, "id\_password").send\_keys("Df#2")

# 7 | click | css=.btn | |

self.driver.find\_element(By.CSS\_SELECTOR, ".btn").click()

test\_demo3:

# Generated by Selenium IDE

import pytest

import time

import json

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.action\_chains import ActionChains

from selenium.webdriver.support import expected\_conditions

from selenium.webdriver.support.wait import WebDriverWait

from selenium.webdriver.common.keys import Keys

class TestDemo3():

def setup\_method(self, method):

self.driver = webdriver.Firefox()

self.vars = {}

def teardown\_method(self, method):

self.driver.quit()

def test\_demo3(self):

# Test name: demo3

# Step # | name | target | value | comment

# 1 | open | / | |

self.driver.get("http://127.0.0.1:8000/")

# 2 | setWindowSize | 1936x1096 | |

self.driver.set\_window\_size(1936, 1096)

# 3 | click | id=id\_password | |

self.driver.find\_element(By.ID, "id\_password").click()

# 4 | type | id=id\_password | Df#2 |

self.driver.find\_element(By.ID, "id\_password").send\_keys("Df#2")

# 5 | click | css=.btn | |

self.driver.find\_element(By.CSS\_SELECTOR, ".btn").click()

test\_demo4:

# Generated by Selenium IDE

import pytest

import time

import json

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.action\_chains import ActionChains

from selenium.webdriver.support import expected\_conditions

from selenium.webdriver.support.wait import WebDriverWait

from selenium.webdriver.common.keys import Keys

class TestDemo4():

def setup\_method(self, method):

self.driver = webdriver.Firefox()

self.vars = {}

def teardown\_method(self, method):

self.driver.quit()

def test\_demo4(self):

# Test name: demo4

# Step # | name | target | value | comment

# 1 | open | / | |

self.driver.get("http://127.0.0.1:8000/")

# 2 | setWindowSize | 1936x1096 | |

self.driver.set\_window\_size(1936, 1096)

# 3 | click | id=id\_username | |

self.driver.find\_element(By.ID, "id\_username").click()

# 4 | type | id=id\_username | 我Ab4$ |

self.driver.find\_element(By.ID, "id\_username").send\_keys("我Ab4$")

# 5 | click | css=.btn | |

self.driver.find\_element(By.CSS\_SELECTOR, ".btn").click()

注册功能

test\_demo1:

# Generated by Selenium IDE

import pytest

import time

import json

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.action\_chains import ActionChains

from selenium.webdriver.support import expected\_conditions

from selenium.webdriver.support.wait import WebDriverWait

from selenium.webdriver.common.keys import Keys

class TestDemo1():

def setup\_method(self, method):

self.driver = webdriver.Firefox()

self.vars = {}

def teardown\_method(self, method):

self.driver.quit()

def test\_demo1(self):

# Test name: demo1

# Step # | name | target | value | comment

# 1 | open | /register/ | |

self.driver.get("http://127.0.0.1:8000/register/")

# 2 | setWindowSize | 1936x1096 | |

self.driver.set\_window\_size(1936, 1096)

# 3 | click | id=id\_username | |

self.driver.find\_element(By.ID, "id\_username").click()

# 4 | type | id=id\_username | 嗨#fsd |

self.driver.find\_element(By.ID, "id\_username").send\_keys("嗨#fsd")

# 5 | click | id=id\_email | |

self.driver.find\_element(By.ID, "id\_email").click()

# 6 | click | id=id\_email | |

self.driver.find\_element(By.ID, "id\_email").click()

# 7 | type | id=id\_email | d15@qq.com |

self.driver.find\_element(By.ID, "id\_email").send\_keys("d15@qq.com")

# 8 | click | id=id\_password1 | |

self.driver.find\_element(By.ID, "id\_password1").click()

# 9 | type | id=id\_password1 | Df#1 |

self.driver.find\_element(By.ID, "id\_password1").send\_keys("Df#1")

# 10 | click | id=id\_password2 | |

self.driver.find\_element(By.ID, "id\_password2").click()

# 11 | type | id=id\_password2 | Df#1 |

self.driver.find\_element(By.ID, "id\_password2").send\_keys("Df#1")

# 12 | click | css=.checkbox > label | |

self.driver.find\_element(By.CSS\_SELECTOR, ".checkbox > label").click()

# 13 | click | css=.btn | |

self.driver.find\_element(By.CSS\_SELECTOR, ".btn").click()

test\_demo2:

# Generated by Selenium IDE

import pytest

import time

import json

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.action\_chains import ActionChains

from selenium.webdriver.support import expected\_conditions

from selenium.webdriver.support.wait import WebDriverWait

from selenium.webdriver.common.keys import Keys

class TestDemo2():

def setup\_method(self, method):

self.driver = webdriver.Firefox()

self.vars = {}

def teardown\_method(self, method):

self.driver.quit()

def test\_demo2(self):

# Test name: demo2

# Step # | name | target | value | comment

# 1 | open | http://127.0.0.1:8000/register/ | |

self.driver.get("http://127.0.0.1:8000/register/")

# 2 | setWindowSize | 1936x1096 | |

self.driver.set\_window\_size(1936, 1096)

# 3 | click | id=id\_email | |

self.driver.find\_element(By.ID, "id\_email").click()

# 4 | type | id=id\_email | d15@qq.com |

self.driver.find\_element(By.ID, "id\_email").send\_keys("d15@qq.com")

# 5 | click | id=id\_password1 | |

self.driver.find\_element(By.ID, "id\_password1").click()

# 6 | type | id=id\_password1 | Df#1 |

self.driver.find\_element(By.ID, "id\_password1").send\_keys("Df#1")

# 7 | click | id=id\_password2 | |

self.driver.find\_element(By.ID, "id\_password2").click()

# 8 | type | id=id\_password2 | Df#1 |

self.driver.find\_element(By.ID, "id\_password2").send\_keys("Df#1")

# 9 | click | css=label > input | |

self.driver.find\_element(By.CSS\_SELECTOR, "label > input").click()

# 10 | click | css=.btn | |

self.driver.find\_element(By.CSS\_SELECTOR, ".btn").click()

test\_demo3:

# Generated by Selenium IDE

import pytest

import time

import json

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.action\_chains import ActionChains

from selenium.webdriver.support import expected\_conditions

from selenium.webdriver.support.wait import WebDriverWait

from selenium.webdriver.common.keys import Keys

class TestDemo3():

def setup\_method(self, method):

self.driver = webdriver.Firefox()

self.vars = {}

def teardown\_method(self, method):

self.driver.quit()

def test\_demo3(self):

# Test name: demo3

# Step # | name | target | value | comment

# 1 | open | http://127.0.0.1:8000/register/ | |

self.driver.get("http://127.0.0.1:8000/register/")

# 2 | setWindowSize | 1936x1096 | |

self.driver.set\_window\_size(1936, 1096)

# 3 | click | id=id\_username | |

self.driver.find\_element(By.ID, "id\_username").click()

# 4 | type | id=id\_username | 嗨#fsd |

self.driver.find\_element(By.ID, "id\_username").send\_keys("嗨#fsd")

# 5 | click | id=id\_email | |

self.driver.find\_element(By.ID, "id\_email").click()

# 6 | type | id=id\_email | d15@qq.com |

self.driver.find\_element(By.ID, "id\_email").send\_keys("d15@qq.com")

# 7 | click | id=id\_password2 | |

self.driver.find\_element(By.ID, "id\_password2").click()

# 8 | type | id=id\_password2 | Df#1 |

self.driver.find\_element(By.ID, "id\_password2").send\_keys("Df#1")

# 9 | click | css=.checkbox > label | |

self.driver.find\_element(By.CSS\_SELECTOR, ".checkbox > label").click()

# 10 | click | css=.btn | |

self.driver.find\_element(By.CSS\_SELECTOR, ".btn").click()

test\_demo4:

# Generated by Selenium IDE

import pytest

import time

import json

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.action\_chains import ActionChains

from selenium.webdriver.support import expected\_conditions

from selenium.webdriver.support.wait import WebDriverWait

from selenium.webdriver.common.keys import Keys

class TestDemo4():

def setup\_method(self, method):

self.driver = webdriver.Firefox()

self.vars = {}

def teardown\_method(self, method):

self.driver.quit()

def test\_demo4(self):

# Test name: demo4

# Step # | name | target | value | comment

# 1 | open | http://127.0.0.1:8000/register/ | |

self.driver.get("http://127.0.0.1:8000/register/")

# 2 | setWindowSize | 1936x1096 | |

self.driver.set\_window\_size(1936, 1096)

# 3 | click | id=id\_username | |

self.driver.find\_element(By.ID, "id\_username").click()

# 4 | type | id=id\_username | 嗨#fsd |

self.driver.find\_element(By.ID, "id\_username").send\_keys("嗨#fsd")

# 5 | click | id=id\_email | |

self.driver.find\_element(By.ID, "id\_email").click()

# 6 | type | id=id\_email | d15@qq.com |

self.driver.find\_element(By.ID, "id\_email").send\_keys("d15@qq.com")

# 7 | click | id=id\_password1 | |

self.driver.find\_element(By.ID, "id\_password1").click()

# 8 | type | id=id\_password1 | Df#1 |

self.driver.find\_element(By.ID, "id\_password1").send\_keys("Df#1")

# 9 | click | css=.checkbox > label | |

self.driver.find\_element(By.CSS\_SELECTOR, ".checkbox > label").click()

# 10 | click | css=.btn | |

self.driver.find\_element(By.CSS\_SELECTOR, ".btn").click()

test\_demo5:

# Generated by Selenium IDE

import pytest

import time

import json

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.action\_chains import ActionChains

from selenium.webdriver.support import expected\_conditions

from selenium.webdriver.support.wait import WebDriverWait

from selenium.webdriver.common.keys import Keys

class TestDemo5():

def setup\_method(self, method):

self.driver = webdriver.Firefox()

self.vars = {}

def teardown\_method(self, method):

self.driver.quit()

def test\_demo5(self):

# Test name: demo5

# Step # | name | target | value | comment

# 1 | open | http://127.0.0.1:8000/register/ | |

self.driver.get("http://127.0.0.1:8000/register/")

# 2 | setWindowSize | 1936x1096 | |

self.driver.set\_window\_size(1936, 1096)

# 3 | click | id=id\_username | |

self.driver.find\_element(By.ID, "id\_username").click()

# 4 | type | id=id\_username | 嗨#fsd |

self.driver.find\_element(By.ID, "id\_username").send\_keys("嗨#fsd")

# 5 | click | id=id\_password1 | |

self.driver.find\_element(By.ID, "id\_password1").click()

# 6 | type | id=id\_password1 | Df#1 |

self.driver.find\_element(By.ID, "id\_password1").send\_keys("Df#1")

# 7 | click | id=id\_password2 | |

self.driver.find\_element(By.ID, "id\_password2").click()

# 8 | type | id=id\_password2 | Df#1 |

self.driver.find\_element(By.ID, "id\_password2").send\_keys("Df#1")

# 9 | click | css=.checkbox > label | |

self.driver.find\_element(By.CSS\_SELECTOR, ".checkbox > label").click()

# 10 | click | css=.btn | |

self.driver.find\_element(By.CSS\_SELECTOR, ".btn").click()

test\_demo6:

# Generated by Selenium IDE

import pytest

import time

import json

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.action\_chains import ActionChains

from selenium.webdriver.support import expected\_conditions

from selenium.webdriver.support.wait import WebDriverWait

from selenium.webdriver.common.keys import Keys

class TestDemo6():

def setup\_method(self, method):

self.driver = webdriver.Firefox()

self.vars = {}

def teardown\_method(self, method):

self.driver.quit()

def test\_demo6(self):

# Test name: demo6

# Step # | name | target | value | comment

# 1 | open | http://127.0.0.1:8000/register/ | |

self.driver.get("http://127.0.0.1:8000/register/")

# 2 | setWindowSize | 1936x1096 | |

self.driver.set\_window\_size(1936, 1096)

# 3 | click | id=id\_username | |

self.driver.find\_element(By.ID, "id\_username").click()

# 4 | type | id=id\_username | 嗨#fsd |

self.driver.find\_element(By.ID, "id\_username").send\_keys("嗨#fsd")

# 5 | click | id=id\_email | |

self.driver.find\_element(By.ID, "id\_email").click()

# 6 | type | id=id\_email | cf5g |

self.driver.find\_element(By.ID, "id\_email").send\_keys("cf5g")

# 7 | click | id=id\_password1 | |

self.driver.find\_element(By.ID, "id\_password1").click()

# 8 | type | id=id\_password1 | Df#1 |

self.driver.find\_element(By.ID, "id\_password1").send\_keys("Df#1")

# 9 | click | id=id\_password2 | |

self.driver.find\_element(By.ID, "id\_password2").click()

# 10 | type | id=id\_password2 | Df#1 |

self.driver.find\_element(By.ID, "id\_password2").send\_keys("Df#1")

# 11 | click | css=.checkbox > label | |

self.driver.find\_element(By.CSS\_SELECTOR, ".checkbox > label").click()

# 12 | click | css=.btn | |

self.driver.find\_element(By.CSS\_SELECTOR, ".btn").click()

test\_demo7:

# Generated by Selenium IDE

import pytest

import time

import json

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.action\_chains import ActionChains

from selenium.webdriver.support import expected\_conditions

from selenium.webdriver.support.wait import WebDriverWait

from selenium.webdriver.common.keys import Keys

class TestDemo7():

def setup\_method(self, method):

self.driver = webdriver.Firefox()

self.vars = {}

def teardown\_method(self, method):

self.driver.quit()

def test\_demo7(self):

# Test name: demo7

# Step # | name | target | value | comment

# 1 | open | http://127.0.0.1:8000/register/ | |

self.driver.get("http://127.0.0.1:8000/register/")

# 2 | setWindowSize | 550x694 | |

self.driver.set\_window\_size(550, 694)

# 3 | click | id=id\_username | |

self.driver.find\_element(By.ID, "id\_username").click()

# 4 | type | id=id\_username | 嗨#fsd |

self.driver.find\_element(By.ID, "id\_username").send\_keys("嗨#fsd")

# 5 | click | id=id\_email | |

self.driver.find\_element(By.ID, "id\_email").click()

# 6 | type | id=id\_email | cf5g@ |

self.driver.find\_element(By.ID, "id\_email").send\_keys("cf5g@")

# 7 | click | id=id\_password1 | |

self.driver.find\_element(By.ID, "id\_password1").click()

# 8 | type | id=id\_password1 | Df#1 |

self.driver.find\_element(By.ID, "id\_password1").send\_keys("Df#1")

# 9 | click | id=id\_password2 | |

self.driver.find\_element(By.ID, "id\_password2").click()

# 10 | type | id=id\_password2 | Df#1 |

self.driver.find\_element(By.ID, "id\_password2").send\_keys("Df#1")

# 11 | click | css=.checkbox > label | |

self.driver.find\_element(By.CSS\_SELECTOR, ".checkbox > label").click()

# 12 | click | css=.btn | |

self.driver.find\_element(By.CSS\_SELECTOR, ".btn").click()

test\_demo8:

# Generated by Selenium IDE

import pytest

import time

import json

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.action\_chains import ActionChains

from selenium.webdriver.support import expected\_conditions

from selenium.webdriver.support.wait import WebDriverWait

from selenium.webdriver.common.keys import Keys

class TestDemo8():

def setup\_method(self, method):

self.driver = webdriver.Firefox()

self.vars = {}

def teardown\_method(self, method):

self.driver.quit()

def test\_demo8(self):

# Test name: demo8

# Step # | name | target | value | comment

# 1 | open | http://127.0.0.1:8000/register/ | |

self.driver.get("http://127.0.0.1:8000/register/")

# 2 | setWindowSize | 550x694 | |

self.driver.set\_window\_size(550, 694)

# 3 | click | id=id\_username | |

self.driver.find\_element(By.ID, "id\_username").click()

# 4 | type | id=id\_username | 嗨#fsd |

self.driver.find\_element(By.ID, "id\_username").send\_keys("嗨#fsd")

# 5 | click | id=id\_email | |

self.driver.find\_element(By.ID, "id\_email").click()

# 6 | type | id=id\_email | d15@qq.com |

self.driver.find\_element(By.ID, "id\_email").send\_keys("d15@qq.com")

# 7 | click | id=id\_password1 | |

self.driver.find\_element(By.ID, "id\_password1").click()

# 8 | type | id=id\_password1 | Df#1 |

self.driver.find\_element(By.ID, "id\_password1").send\_keys("Df#1")

# 9 | click | id=id\_password2 | |

self.driver.find\_element(By.ID, "id\_password2").click()

# 10 | type | id=id\_password2 | ehf13d |

self.driver.find\_element(By.ID, "id\_password2").send\_keys("ehf13d")

# 11 | click | css=.checkbox > label | |

self.driver.find\_element(By.CSS\_SELECTOR, ".checkbox > label").click()

# 12 | click | css=.btn | |

self.driver.find\_element(By.CSS\_SELECTOR, ".btn").click()

test\_demo9:

# Generated by Selenium IDE

import pytest

import time

import json

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.action\_chains import ActionChains

from selenium.webdriver.support import expected\_conditions

from selenium.webdriver.support.wait import WebDriverWait

from selenium.webdriver.common.keys import Keys

class TestDemo9():

def setup\_method(self, method):

self.driver = webdriver.Firefox()

self.vars = {}

def teardown\_method(self, method):

self.driver.quit()

def test\_demo9(self):

# Test name: demo9

# Step # | name | target | value | comment

# 1 | open | http://127.0.0.1:8000/register/ | |

self.driver.get("http://127.0.0.1:8000/register/")

# 2 | setWindowSize | 550x694 | |

self.driver.set\_window\_size(550, 694)

# 3 | click | id=id\_username | |

self.driver.find\_element(By.ID, "id\_username").click()

# 4 | type | id=id\_username | 嗨#fsd |

self.driver.find\_element(By.ID, "id\_username").send\_keys("嗨#fsd")

# 5 | click | id=id\_email | |

self.driver.find\_element(By.ID, "id\_email").click()

# 6 | type | id=id\_email | d15@qq.com |

self.driver.find\_element(By.ID, "id\_email").send\_keys("d15@qq.com")

# 7 | click | id=id\_password1 | |

self.driver.find\_element(By.ID, "id\_password1").click()

# 8 | type | id=id\_password1 | Df#1 |

self.driver.find\_element(By.ID, "id\_password1").send\_keys("Df#1")

# 9 | click | id=id\_password2 | |

self.driver.find\_element(By.ID, "id\_password2").click()

# 10 | type | id=id\_password2 | Df#1 |

self.driver.find\_element(By.ID, "id\_password2").send\_keys("Df#1")

# 11 | click | css=.btn | |

self.driver.find\_element(By.CSS\_SELECTOR, ".btn").click()

登陆注册界面其他按钮

test\_demo1:

# Generated by Selenium IDE

import pytest

import time

import json

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.action\_chains import ActionChains

from selenium.webdriver.support import expected\_conditions

from selenium.webdriver.support.wait import WebDriverWait

from selenium.webdriver.common.keys import Keys

class TestDemo1():

def setup\_method(self, method):

self.driver = webdriver.Firefox()

self.vars = {}

def teardown\_method(self, method):

self.driver.quit()

def test\_demo1(self):

# Test name: demo1

# Step # | name | target | value | comment

# 1 | open | http://127.0.0.1:8000/register/index.html | |

self.driver.get("http://127.0.0.1:8000/register/index.html")

# 2 | setWindowSize | 550x695 | |

self.driver.set\_window\_size(550, 695)

# 3 | click | linkText=登录 | |

self.driver.find\_element(By.LINK\_TEXT, "登录").click()

test\_demo1:

# Generated by Selenium IDE

import pytest

import time

import json

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.action\_chains import ActionChains

from selenium.webdriver.support import expected\_conditions

from selenium.webdriver.support.wait import WebDriverWait

from selenium.webdriver.common.keys import Keys

class TestDemo2():

def setup\_method(self, method):

self.driver = webdriver.Firefox()

self.vars = {}

def teardown\_method(self, method):

self.driver.quit()

def test\_demo2(self):

# Test name: demo2

# Step # | name | target | value | comment

# 1 | open | http://127.0.0.1:8000/ | |

self.driver.get("http://127.0.0.1:8000/")

# 2 | setWindowSize | 550x695 | |

self.driver.set\_window\_size(550, 695)

# 3 | click | linkText=注册 | |

self.driver.find\_element(By.LINK\_TEXT, "注册").click()

test\_demo1:

# Generated by Selenium IDE

import pytest

import time

import json

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.action\_chains import ActionChains

from selenium.webdriver.support import expected\_conditions

from selenium.webdriver.support.wait import WebDriverWait

from selenium.webdriver.common.keys import Keys

class TestDemo3():

def setup\_method(self, method):

self.driver = webdriver.Firefox()

self.vars = {}

def teardown\_method(self, method):

self.driver.quit()

def test\_demo3(self):

# Test name: demo3

# Step # | name | target | value | comment

# 1 | open | http://127.0.0.1:8000/ | |

self.driver.get("http://127.0.0.1:8000/")

# 2 | setWindowSize | 550x695 | |

self.driver.set\_window\_size(550, 695)

# 3 | click | linkText=忘记密码？ | |

self.driver.find\_element(By.LINK\_TEXT, "忘记密码？").click()

主界面

test\_demo1:

# Generated by Selenium IDE

import pytest

import time

import json

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.action\_chains import ActionChains

from selenium.webdriver.support import expected\_conditions

from selenium.webdriver.support.wait import WebDriverWait

from selenium.webdriver.common.keys import Keys

class TestDemo1():

def setup\_method(self, method):

self.driver = webdriver.Firefox()

self.vars = {}

def teardown\_method(self, method):

self.driver.quit()

def test\_demo1(self):

# Test name: demo1

# Step # | name | target | value | comment

# 1 | open | http://127.0.0.1:8000/index/ | |

self.driver.get("http://127.0.0.1:8000/index/")

# 2 | setWindowSize | 1936x1096 | |

self.driver.set\_window\_size(1936, 1096)

# 3 | click | css=td:nth-child(7) > .badge | |

self.driver.find\_element(By.CSS\_SELECTOR, "td:nth-child(7) > .badge").click()

# 4 | click | css=td:nth-child(8) > .badge | |

self.driver.find\_element(By.CSS\_SELECTOR, "td:nth-child(8) > .badge").click()

# 5 | click | css=.ti-settings | |

self.driver.find\_element(By.CSS\_SELECTOR, ".ti-settings").click()

# 6 | click | css=.card-close > .ti-close | |

self.driver.find\_element(By.CSS\_SELECTOR, ".card-close > .ti-close").click()

# 7 | click | css=.ti-layout-grid4-alt | |

self.driver.find\_element(By.CSS\_SELECTOR, ".ti-layout-grid4-alt").click()

# 8 | click | css=.ti-view-list-alt | |

self.driver.find\_element(By.CSS\_SELECTOR, ".ti-view-list-alt").click()

# 9 | click | linkText=异常视频查看 | |

self.driver.find\_element(By.LINK\_TEXT, "异常视频查看").click()

# 10 | click | linkText=监控地点列表 | |

self.driver.find\_element(By.LINK\_TEXT, "监控地点列表").click()

# 11 | click | linkText=注销 | |

self.driver.find\_element(By.LINK\_TEXT, "注销").click()

# 12 | click | css=.ti-menu | |

self.driver.find\_element(By.CSS\_SELECTOR, ".ti-menu").click()

# 13 | click | css=.hamburger | |

self.driver.find\_element(By.CSS\_SELECTOR, ".hamburger").click()