<慧眼识踪>

测试报告

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <27日/7月/2018年> | <1.0> | <第二次迭代测试报告> | <陈志扬、余心如、袁添镱、陶心旋> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

1. 简介 4

1.1 目的 4

1.2 范围 4

1.3 定义、首字母缩写词和缩略语 4

1.4 参考资料 4

1.5 概述 4

2. 测试概要 4

3. 测试环境 5

4. 测试结果及分析 5

4.1 需求覆盖率及缺陷分布 5

4.2 缺陷严重程度 7

5. 缺陷清单 7

5.1 功能性缺陷 7

5.2 非功能性缺陷 8

6. 测试结论与建议 8

测试报告

# 简介

## 目的

本测试报告为慧眼识踪软件项目的系统测试报告，目的在于对系统开发和实施后的的结果进行测试以及测试结果分析，发现系统中存在的问题，描述系统是否符合项目需求说明书中规定的功能和性能要求。

## 范围

适用范围包括用户、项目开发人员和需要阅读本报告的高层领导。

## 定义、首字母缩写词和缩略语

功能测试：测试软件各个功能模块是否正确，逻辑是否正确。

非功能测试：测试软件的性能、易用性、可靠性等方面。

Reid：Re-identification，行人再识别。

Object detection：检测图片中的物体（本系统中针对行人）。

## 参考资料

1. 测试用例.xls
2. 软件需求规约.docx

## 概述

本测试报告分为简介、测试概要、测试环境、测试结果及分析、缺陷清单、测试结论与建议六个部分。简介简要介绍了本测试报告，测试概要概述了测试的五项基本信息，测试环境描述了测试是在什么环境下进行的，测试结果及分析说明了测试后得到的需求覆盖率与缺陷分布结果以及项目开发人员对于该结果的分析，缺陷清单列出了测试发现的系统的各个缺陷的出现情况与严重程度，最后的测试结论与建议总结了通过本次测试项目开发人员得出的结论与针对该结论对系统进行改进的建议措施。

**注：前后端单元测试相关结果及代码见 *web测试报告.docx***

**识别系统模块的相关测试结果见 *识别系统测试报告.docx***

# 测试概要

## 单元测试

## 集成测试

测试时间：2018/07/12 - 2018/07/13

测试地点：软件学院3108教学实验室

测试人员：全组项目开发人员

测试方法：采用集成测试，将两个已经测试过的单元组合成一个组件，并且测试它们之间的接口。

测试内容：通过python脚本将一段样本视频每隔30帧截取1帧作为样本，前200帧中的样本帧作为query样本，之后的样本作为test样本，再使用query样本与test样本进行Reid，测试object detection与Reid是否组合成功，能否返回正确的结果。

## 系统测试

测试时间：2018/07/23 - 2018/07/24

测试地点：软件学院3108教学实验室

测试人员：全组项目开发人员

测试方法：在完全不考虑程序内部结构和内部特性的情况下，只需检查程序功能是否按照系统需求规格说明书的规定正常使用，是否能在输入适当的数锯下产生正确的输出信息，并且能保持外部信息的完整性。因此采用了着眼于程序外部结构、不考虑内部逻辑结构、针对软件界面和软件功能进行测试的测试方法：黑盒测试。

测试内容：针对慧眼识踪软件项目的软件需求规约书指出的系统应该具有的功能进行功能测试，对于系统的性能、易用性、可靠性等方面进行功能性测试。

# 测试环境

操作系统：Windows10

CPU：Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz

内存：16 GB

应用软件：Intellij IDEA、PyCharm、Red5

数据库软件：MySQL

# 测试结果及分析

## 需求覆盖率及缺陷分布

**表1 需求覆盖率及缺陷分布**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **测试内容** | | **测试**  **用例数** | **需求**  **覆盖率** | **缺陷数** | **缺陷率** | **备注** |
| **功能项** | **登录** | 4 | 100% | 0 | 0 | 包含正常及错误数据 |
| **注册** | 4 | 100% | 0 | 0 | 包含正常及错误数据 |
| **查看摄像头实时以及视频** | 4 | 100% | 0 | 0 | 摄像头包括已开启和未开启的摄像头 |
| **对视频进行截图并下载** | 4 | 100% | 0 | 0 | 无 |
| **对上传的图片进行裁剪和截图** | 4 | 100% | 0 | 0 | 包括上传不符合格式的文件和过大的文件 |
| **获取识别系统发来的信息并展示** | 6 | 100% | 0 | 0 | 无 |
| **分布式计算通信测试** | 10 | 100% | 0 | 0 | 测试两台计算机之间进行通信 |
| **物体检测及行人再识别测试** | 20 | 100% | 0 | 0 | 对一段视频或一张图片进行物体检测和行人再识别 |
| **对目标的预测搜索** | 14 | 100% | 0 | 0 | 无 |
| **功能项小计** | 70 | 100% | 0 | 0 | 无 |
| **非功**  **能项** | **性能** | 10 | 100% | 0 | 0 | 对系统各功能进行响应时间的测试（未包含行人再识别） |
| **可靠性** | 10 | 100% | 0 | 0 | 对系统各功能进行分模块的可靠性测试 |
| **易用性测试** | 2 | 100% | 0 | 0 | 邀请两位路人进行使用测试 |
| **安全性测试** | 5 | 100% | 0 | 0 | 检查数据库中用户信息的加密情况以及注册登陆时的信息检验 |
| **压力测试** | 10 | 100% | 1 | 10% | 网站访问的并发性以及对大量样本进行再识别 |
| **界面测试（兼容性测试）** | 8 | 100% | 0 | 0 |  |
|  | **非功能项小计** | *45* | *100%* | *1* | *2.2%* |  |
|  | **总计** | *115* | *100%* | *1* | *0.87%* |  |

## 缺陷严重程度

**表2 缺陷严重程度**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **严重程度** | **致命** | **严重** | **一般** | **细微** | **总数** |
| **缺陷个数** | 0 | 0 | 1 | 0 | 0 |
| **占缺陷百分比** | 0 | 0 | 100% | 0 | 0 |

# 缺陷清单

## 功能性缺陷

**表3 功能性缺陷列表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序**  **号** | **缺陷**  **编号** | **严重程度** | **功能模块** | **缺陷标题** | **缺陷描述** | **测试用例编号** |
| 1 | Bug\_ReID\_1 | 一般 | 物体检测及行人再识别 | ReID卡死（1000000数量级） | 在数据量规模过大时（大于100000张），系统将会内存溢出 | 慧眼识踪-testcase-022 |

表中相关项说明：

1. 缺陷编号：为每个缺陷赋予一个唯一的编号，可以通过此编号对缺陷进行跟踪。例如：Bug001。
2. 缺陷严重程度：缺陷可以根据严重程度分为以下几种情况。
3. 致命（fatal）：致命的错误，测试执行直接导致系统死机、蓝屏、挂起、或是程序非法退出；系统的主要功能或需求没有实现。
4. 严重（critical）：严重错误，系统的次要功能点或需求点没有实现；数据丢失或损坏。执行软件主要功能的测试用例导致系统出错，程序无法正常继续执行；程序执行过于缓慢或是占用过大的系统资源。
5. 一般（major）：不太严重的错误，这样的缺陷虽然不影响系统的基本使用，但没有很好地实现功能，没有达到预期的效果。如次要功能丧失，界面错误，打印内容、格式错误，提示信息不太正确，或用户界面太差，简单的输入限制未放在前台进行控制，删除操作未给出提示，操作时间长等。
6. 细微（minor）：一些小问题，对功能几乎没有影响，产品及属性仍可使用。如软件的实际执行过程与需求有较小的差异；程序的提示信息描述容易使用户产生混淆；有个别错别字、 文字排列不整齐；辅助说明描述不清楚，显示格式不规范，长时间操作未给用户进度提示，提示窗口文字未采用行业术语，可输入区域和只读区域没有明显的区分标志，系统处理未优化等。
7. 功能模块：所测试并出现该缺陷的功能模块名称。
8. 缺陷标题：描述缺陷的标题。
9. 缺陷描述：对缺陷的详细描述。描述缺陷的重现步骤和问题所在，描述需清晰明了，使研发人员可以按照描述将缺陷重现出来。必要时可以放上相应的截图，以便于研发人员理解问题所在。

## 非功能性缺陷

无

# 测试结论与建议

针对第一次迭代的遗留问题进行了解决和优化。限于硬件性能，部分性能测试无法很好地修复。