|  |  |  |  |
| --- | --- | --- | --- |
| **修订历史** | | | |
| 日期 | 版本 | 说明 | 作者 |
| 7/11/2018 | 1.0 | 第一次迭代总结 | 金瑞洋、王见思、宋逸凡、李翌珺 |
| 7/26/2018 | 2.0 | 第二次迭代总结 |
|  |  |  |
|  |  |  |

**“慧眼识踪“——基于深度学习的人员即时搜寻系统**

**迭代评估报告**

版本2.0

# 简介

本项目针对寻找走失人员、跟踪犯罪嫌疑人和定位传染病人等众多现实场景下都必须要解决的搜寻指定人员的问题，设计并开发了一个“基于深度学习的人员即时搜寻系统”。

## 1.1目的

本迭代评估将对GETS项目各个迭代的执行进行各方面评价，并根据实际情况对各个迭代的内容加以调整。

## 1.2范围

该迭代评估适用于GETS项目的开发，主要针对项目开发人员与项目经理编写。

## 1.3定义、首字母缩写词和缩略语

搜索目标：需要搜寻系统根据监控画面锁定位置的人物

GETS：God Eye Tracking System， “慧眼识踪”人员搜寻系统

## 1.4概述

本迭代评估报告包含四次迭代的执行情况与整体评价，并详细列明了各个迭代的已完成目标，计划遵循程度，外部变更等信息。

# 已达到的迭代目标

第一次迭代：原计划中系统版本1（R1）必须实现的功能（查看摄像头参数、查看实时监控）均已实现。

第二次迭代：原计划中系统版本2（R2）必须实现的功能（查看历史视频、从视频中截取图片上传并单层搜索目标）均已实现。

# 计划遵循程度

第一次迭代：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 任务 | 预期时间 | 实际时间 | 成果 | 相关人员 | 完成情况 |
| 阶段会议 | 7/2/2018 | 7/2/2018 | 迭代计划 | 全体成员 | 完成 |
| 分析系统架构 | 7/2/2018 | 7/2/2018 | 系统架构文档 | 全体成员 | 完成 |
| 初步前端界面开发 | 7/2/2018 -  7/4/2018 | 7/2/2018 | 第一版前端界面 | 王见思、宋逸凡 | 完成 |
| 实现视频流接入与输出 | 7/2/2018 –  7/5/2018 | 7/2/2018-7/3/2018 | 实时监控在前端的输出 | 王见思、金瑞洋、李翌珺 | 完成 |
| 前端界面优化 | 7/5/2018 –  7/7/2018 | 7/3/2018 –  7/7/2018 | 第二版前端界面 | 全体成员 | 提前完成 |
| 收集识别技术相关信息 | 7/6/2018 –  7/10/2018 | 7/3/2018 –  7/11/2018 | 无 | 全体成员 | 完成 |
| 测试 | 7/10/2018 –  7/14/2018 | 7/11/2018 | 测试用例与测试报告第一部分 | 全体成员 | 完成 |
| 截取搜索目标 | 未计划 | 7/3/2018 – 7/6/2018 | 带有截取功能的界面 | 王见思 | 提前完成 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 任务 | 预期时间 | 实际时间 | 成果 | 相关人员 | 完成情况 |
| 阶段会议 | 7/16/2018 | 7/16/2018 | 第二次迭代计划 | 全体成员 | 完成 |
| 分析系统架构 | 7/16/2018 | 7/16/2018 | 新的系统架构文档 | 全体成员 | 完成 |
| 前端界面进一步开发 | 7/16/2018 -  7/20/2018 | 7/16/2018-  7/23/2018 | 前端界面中的管理界面和简单修正 | 王见思、宋逸凡 | 延迟完成 |
| TensorFlow-Serving应用 | 7162/2018 –  7/20/2018 | 7/16/2018-7/20/2018 | 将object - detection尝试打包并上线 | 王见思、李翌珺、宋逸凡 | 未完成 |
| Reid功能合并 | 7/16/2018 –  7/20/2018 | 7/16/2018 –  7/20/2018 | Reid整体功能代码 | 金瑞洋 | 完成 |
| 用flask将object-detection功能打包 | 7/21/2018-7/23/2018 | 7/21/2018-7/23/2018 | 成功部署在flask上的object-detection功能代码 | 李翌珺、宋逸凡 | 完成 |
| 测试 | 7/21/2018 –  7/25/2018 | 7/21/2018 –  7/25/2018 | 测试用例与测试报告第二部分、单元测试代码 | 王见思、宋逸凡 | 完成 |
| Rancher集群管理 | 未计划 | 7/16/2018 – 7/20/2018 | 带有截取功能的界面 | 王见思 | 完成 |
| object-detection与reid与前端合并 | 7/21/2018 –  7/25/2018 | 7/21/2018 –  7/25/2018 | 具有完整功能的前端页面 | 全体成员 | 完成 |

# 已实施的用例与场景

第一次迭代：

已实施用例：查看摄像头参数、查看实时监控

已实施场景：1. 允许用户查看给出的摄像头位置以及相关参数。2. 用户可以通过在地图中点击摄像头查看该摄像头的实时视频。3. 用户可以在实时视频中截取搜索目标。

第二次迭代：

已实施用例：查看历史视频、管理摄像头与地图、在直播视频中截屏并上传查找。

已新实施的场景：1. 允许用户查看历史视频。2. 允许用户管理摄像头以及地图信息。3. 用户可以在实时视频中截取搜索目标并选择在直播或者历史视频中识别，并能够得到返回信息。

# 对照

# 评估标准评估结果

第一次迭代相关标准：

可支持性：

系统编码使用utf-8字符集，系统语言使用中文

变量命名使用驼峰式命名法

仅注册用户可以使用该系统的功能

设计约束：

编程语言：使用html、css、javascript、java、python等语言

开发工具：使用visual studio code等工具，react等框架进行前端开发，使用intellij idea等工具，spring boot等框架进行后端开发，使用pycharm等工具调用tensorflow框架

使用架构及设计约束：

系统必须采用B/S架构，系统中需要存储的数据类型必须被数据库支持，特征识别使用TensorFlow实现

用户界面：

- 首页

- 建筑平面图与摄像头信息

- 实时监控

- 基于首页信息

- 摄像头同步播放实时监控

硬件接口：

实现监控摄像头实时画面同步播放

软件接口：

与监控摄像头之间：实现输入视频流同步播放

与特征识别框架之间：接入TensorFlow框架接口

通信接口：

实现与监控摄像头间的通信

以上所有需求均已实现并在第一次迭代中得以体现。

第二次迭代新增：

用户界面：

- 首页

- 管理界面（摄像头与地图）

- 搜寻功能（由截图到识别到返回，有从直播中搜寻和在历史视频中搜寻两个功能）

软件接口：

与特征识别框架之间：将object-detection与reid结合并能由前端调用

以上所有需求均已实现并在第二次迭代中得以体现。

# 测试结果

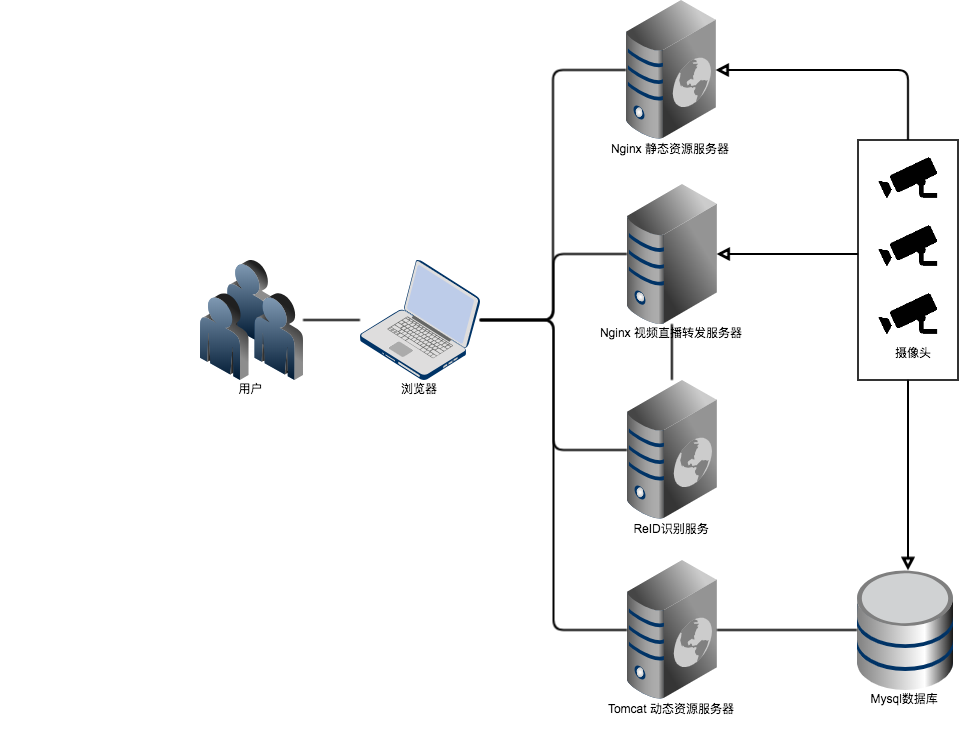
第一次迭代测试：使用人工测试，由测试员手动进入首页，点选摄像头并查看实时监控。

测试结果：系统可以正确显示首页，在测试员选定摄像头后能够正确给出参数并播放实时监控视频。

第二次迭代测试：集成测试使用人工测试，单元测试使用了Jest+Enzyme，压力测试使用了腾讯WeTest平台。

测试结果：未出现不可修复的缺陷，已实现功能均可正常运行，详见测试报告。

# 出现的外部变更

第一次迭代：1.迭代目标早于预期完成，经小组讨论决定于2018年7月11日提前结束第一次迭代，并在验收后开始第二次迭代。2.项目风险与预期不符，绝大多数风险为人物识别相关的技术风险，在之后应主要把时间和精力放在相关技术的学习与使用上。3.项目架构需要做出更改，更改后的架构图如下：

第二次迭代：1.在迭代前半段过程中关键技术部分探索碰壁而导致进度较预期稍为落后了一些，我们决定将验收时间放在最后一天即7月27日，以求保证项目功能等方面的完整性。2.在前半段过程中我们的分工出现了部分问题，一开始预期的是TensorFlow-Serving与flask两条路线并进，但在前期发现一条路线有进展后导致另一条路线搁置了两天，结果在前一条路线失败之后只得重新选择第二条路线，导致项目进度与预期出现了一些偏差。3.在直播中识别物体的逻辑过程发现与实际有一定距离，后期作了修改，但尚未完全完成。

# 需要进行的返工

第一次迭代：前端界面可进一步优化，摄像头参数有待具体化，实时监控延迟过长。

第二次迭代：前端一次识别完成后重新识别的界面流程可优化，识别时间过长，有待优化，新增历史视频时可马上对其做object-detection来进一步提高性能。