

---

教育部高等学校大学计算机课程教学指导委员会

# 中国大学生计算机设计大赛



软件开发类作品文档简要要求

作品编号： 2021033965

作品名称： 基于树莓派的考生身份辅助核查系统

作 者： 武新纪 张恒 徐慧

版本编号：

填写日期： 2021 年 4 月 8 日

## 填写说明：

- 1、本文档适用于所有涉及软件开发的作品，包括：软件应用与开发、大数据、人工智能、物联网应用；
- 2、正文一律用五号宋体，一级标题为二号黑体，其他级别标题如有需要，可根据需要设置；
- 3、本文档为简要文档，不宜长篇大论，简明扼要为上；
- 4、提交文档时，以PDF格式提交本文档；
- 5、本文档内容是正式参赛内容组成部分，务必真实填写。如不属实，将导致奖项等级降低甚至终止本作品参加比赛。

## 目 录

第一章 需求分析 .....	2
1.1 教育行业存在的考生身份识别的问题 .....	2
1.2 现有竞品的不足 .....	2
1.3 本系统的优点 .....	2
第二章 概要设计 .....	3
第三章 详细设计 .....	4
3.1 系统架构 .....	4
3.2 硬件连接 .....	5
3.3 软件设计 .....	6
3.3.1 监考端设计 .....	6
3.3.2 后台系统设计 .....	7
3.3.2 巡考端设计 .....	8
3.4 功能实现 .....	9
第四章 测试报告 .....	11
第五章 安装及使用 .....	13
5.1 硬件安装 .....	13
5.2 软件安装 .....	13
5.2.1 监考端 .....	13
5.2.2 管理后台 .....	14
5.2.3 巡考端 .....	14
第六章 项目总结 .....	15
参考文献 .....	16

# 第一章 需求分析

## 1.1 教育行业存在的考生身份识别的问题

在教育行业的大规模各类教育考试中，考生的身份识别至关重要，是保证公平的前提和基础，在传统的考试流程中采用的是人工鉴别考生身份的方法，虽然人工判断可以做到较为准确的判断与比较，但人力消耗大，智能化程度低，再者，传统的纸质证件(准考证)很难进行加密操作，因此很容易被伪造，降低了考试的公平公正性，此外，大规模考试中考生数量比较多，监考老师大多数也都没有受过专业的人脸识别训练，仅凭人工辨别的方式根本无法在短时间内完成大量考生的对比，也很容易出现错判、漏判或其他人为因素的干扰。另外，在进行监考时，监考教师和巡考教师，监考教师即使在识别到与证件不匹配的考生时，无法及时将情况报告给巡考教师，巡考教师也来不及及时处理，最后可能会耽误考生考试。

## 1.2 现有竞品的不足

针对上述情况，不少地区纷纷出台通过智能设备实现考生人脸人证系统，采用人脸识别与身份证核验相结合，虽然已经大大的提高了公平性，但还有很多的不足。比如目前采用的PC设备成本较高也不易携带，不能很好的满足教育行业目前的需求，再者，在使用硬盘存储采集到的考生信息，存在较大的安全隐患。目前，市场上也存在着和我们相类似的便携式人证核实设备，采用的基本是安卓系统，功能繁多却缺乏安全性，没有管理后台对识别记录进行管理，也无法及时将异常考生及时报告给巡考教师。此外，定价基本上都在千元以上。

## 1.3 本系统的优点

我们的系统利用低成本的树莓派和自动对焦摄像头基本实现了考生身份辅助核查，提高了核查考生身份的效率，由于本系统基于手持移动终端，具备更大的通用性和便携性，操作简单，其次树莓派成本较低，具有经济性和实用性。我们的系统具有完善的管理后台，不仅能够实时查看对比信息，而且能够对其高效管理。便携终端使用ssh和sftp协议传输数据至服务器，本地实时删除，后台管理系统使用https安全加密传输技术，可以保证数据访问的安全性。此外，我们还拥有巡考终端，可以及时地将监考端未通过地记录发送给巡考端，巡考教师在巡考端可以实时所有考场的异常考生，发现之后便可以通过巡考端上显示的信息前往相应的教师对异常考生进行进一步核对，从而判别考生身份。

总的来说，我们的系统具备价格优势、便携而不失功能、便于管理，以及注重隐私和安全。监考端、巡考端和管理后台三部分相互协同，共同完成考生身份的核验。

## 第二章 概要设计

该系统分为三部分，一部分为采集与识别的监考端，部署在树莓派上，为便携式识别设备，该设备可以设置多个，分布于不同的考场，用于图片的采集和对比；一部分为巡考终端，可以部署在个人 PC 上，也可以部署在树莓派上等便携设备上，用于接收监考端的异常考生信息。还有一部分为后台管理系统，部署在云服务器上，用于数据的查看与管理，也用于和监考端、巡考端进行数据交互。监考端、巡考端通过 SSH 和 SFTP 协议与管理后台进行数据交互，当用户采集到人脸照片和证件照片后执行对比操作时，监考端会对采集到的照片进行对比，并将对比结果显示在屏幕上传递给用户，让监考教师第一时间知道对比结果。对比结束之后，系统会将对比结果以人脸照片和证件照片上传至后台管理系统，并立即从便携终端中删除采集到的图片。图片和识别结果被上传至服务器后，被存储在加密的 MySQL 数据库中。当我们登录管理后台的时候，就可以看见识别记录的时间、相似度以及人脸和证件照片，并且可以对其进行删除。巡考端与云端数据库进行交互，获取所有未通过的识别信息存储与本地的 sqlite3 数据库中，并实时将未通过的记录显示在巡考端，巡考教师可以根据巡考端显示的信息前往相应教室进行进一步核验。

系统设计多方面的知识，主要使用 python 作为开发语言：监考端和巡考端的系统使用 qt 作为基本的开发框架，qt 具有开源、跨平台、易于扩展等优点，非常适合我们的开发。后台管理系统我们使用 django 作为整体开发框架，并使用开源项目 SimpleUI 来美化了我们的后台。后台系统使用 Centos8 作为服务器系统，CentOS 具有开源免费、高效稳定、易于部署等优点，很多知名网站都是部署在 CentOS 系统上的。

## 第三章 详细设计

### 3.1 系统架构

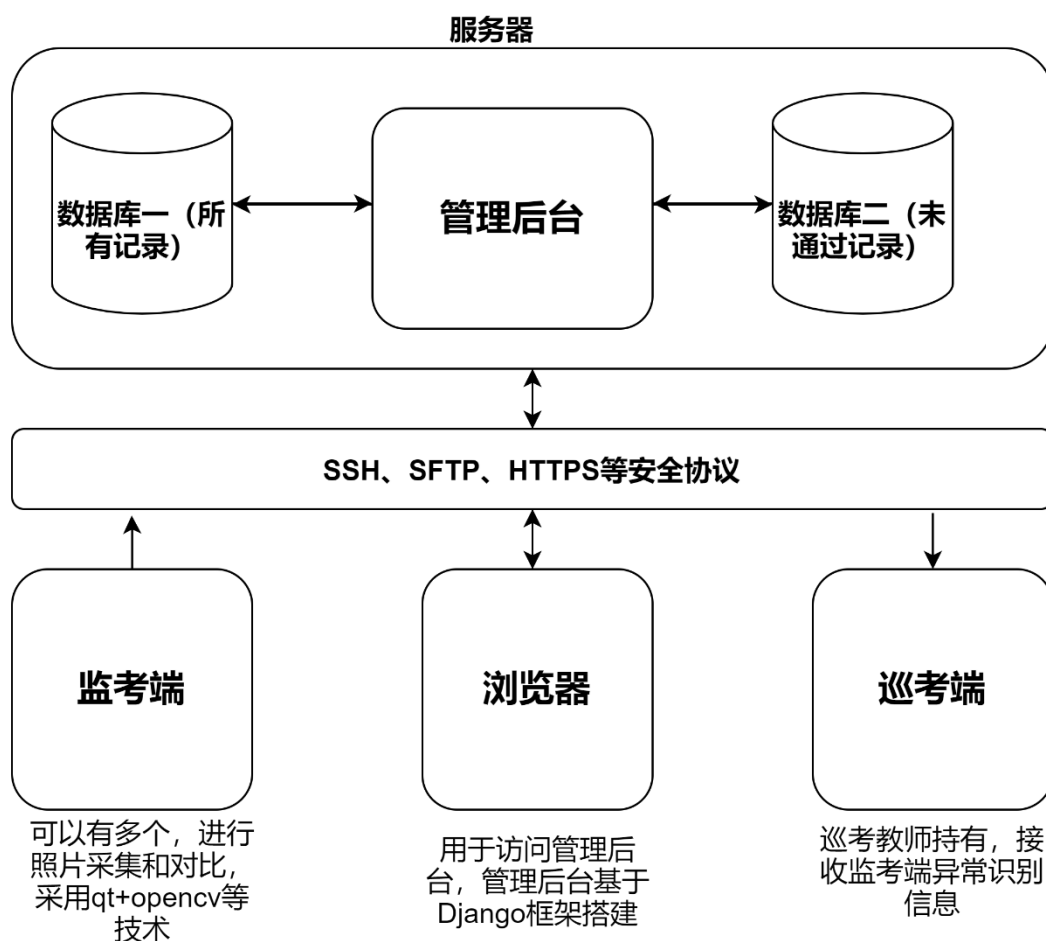


图 3.1 系统架构图

如上图所示，我们的系统由三部分组成，监考端、巡考端和服务器的管理后台。终端设备通过 SSH、SFTP、HTTPS 等安全协议与服务器相连，保证了数据传输的安全性，防止数据被劫持而造成数据和考生信息的泄露。管理后台和云服务器上的 MySQL 数据库交互，在浏览器登录管理后台，就可以对所有识别数据进行管理。

监考端采用 qt 作为整体开发框架，采用 python 作为主要开发语言，因此使用了 PySide2 进行 qt 开发，也可以使用 PyQt5 进行开发，但 PySide2 是 Qt 官方维护，更为稳定。使用 opencv 模块读取摄像头信息，并进行照片处理，使用 pymysql 模块将数据写入到云端数据库，使用 paramiko 模块通过 SSH 和 SFTP 协议将采集到的图片传输到云服务器。人证对比采用了两种解决方案，分别是在线识别和离线识别，在线识别方案采用百度 AI 开放平台的人脸对比 API，离线识别方案采用开源库人脸识别库 face\_recognition 作为离线识别方案。

管理后台部署在阿里云服务器上，采用的操作系统为 CentOS8 操作系统，目前部署在阿

里云服务器上。如果需要，管理后台也可以部署在学校的内网环境下，更进一步确保数据安全。管理后台使用 python 语言，基于 Django 框架编写，还使用了开源库 simpleUI 来进行界面的美化。管理后台连接着 MySQL 数据库，操作着两张数据表，两张表的结构类似，不同之处在于一个存储了全部的识别信息，另一张表仅存储未通过的识别信息，存储未通过的识别信息的表将会与巡考端产生数据交互，将未通过数据传递给巡考端。管理后台为 django 项目，采用 uwsgi+nginx 方式来进行部署，其中，uwsgi 用于处理动态文件，nginx 处理静态文件和进行反向代理，使管理后台不直接暴露地址与接口给外网，进一步保证管理后台信息的安全。

巡考端和监考端类似，同样采用 qt 作为开发框架，因为使用的 python 语言，所以使用 PySide2 来进行 qt 的开发，巡考端要操作两种数据库，一个是服务器端的 MySQL 数据库，通过 pymysql 模块实现，从云端数据库读取数据；另一个是 sqlite3 本地数据库，通过 sqlite3 模块将从服务器端 MySQL 数据库读取的文件写入到本地 sqlite3 数据库。Sqlite3 数据库是程序启动时被创建，当程序结束之后会被从本地删除。因为该系统还涉及到了人脸和证件的采集，为了让巡考端更直观的看到考生信息，故将监考端采集到的图片通过 SFTP 协议传输到本地，并显示在巡考端界面上，但一旦程序关闭，将会连通数据库一起删除巡考端存储的所有照片，确保学生的隐私信息不被泄露。

### 3.2 硬件连接

硬件部分，我们选用了拥有 1G 内存的树莓派 3B 来作为程序运行的硬件系统，一来可以控制成本，二来程序的相关优化在树莓派 3B 上能够流畅运行。我们将树莓派主板和摄像头、电池以及显示屏连接到了在一起，并做了精致的外壳，在实现主要功能的同时，又不失便携性和美观性。考虑到树莓派 3B 主板的供电问题和系统外壳和各硬件之间的协调性，系统只采用一个摄像头，但是并没有影响采集人脸照片和证件照片的功能上。



图 3.2 监考端设备安装图片

## 3.3 软件设计

### 3.3.1 监考端设计

在便携终端上，我们使用的是树莓派官方的系统，该系统针对树莓派硬件做了专门的优化，且安装各种软件和库都比较方便，我们的程序在该系统上可以完美运行。主程序除了用到了 Qt 的相关技术，还使用了 opencv 库读取摄像头画面，然后将其显示到 qt 框架的 label 标签上，实现了在主界面上实时显示摄像头画面的功能。我们的主程序，分为一个主界面和两个子界面，主界面主要包含了三个区域：最上方的标题区，其中标题区中间的标题还充当了状态栏的功能，可以将系统运行状态显示在上面；紧接着下方的是图像区，这个区域由两个框，用来显示拍摄到的图片，采集到的图片会实时显示到这个上面，其中左侧是人脸照片显示区，右边的是证件照片显示区；然后最后就是按钮区域，主界面上的按钮是三个，分别为“人脸采集”、“证件采集”和“对比”，当点击“人脸采集”按钮时，就会进入到人脸采集界面，同样，若点击“证件采集”按钮，就会进入证件采集界面。当人脸和证件照片都采集到了之后，中间的图像区就会显示采集到的图像，然后点击对比就会执行对比操作，然后会在标题区的中间区域显示对比结果，如果采集的图片不完整，那么就会提示采集的图片不完整，提示用户重新采集一个界面。主界面如图 3-3-1 所示。



图 3-3-1 主界面

两个子界面主界面的结构也很类似，都是由标题区、图像区和按钮区组成，不同的是主界面的图像区只显示拍摄到的画面，但是子界面的图像区会实时显示摄像头捕捉到的画面，当关闭窗口或点击拍照按钮后一秒，摄像头就会关闭，并且退出子窗口，回到主窗口。回到主窗口后，就可以根据主界面图像区显示的画面来观察是否拍摄了照片，然后执行对比功能。子界面如图 3-3-2 所示。



图 3-3-2 子界面

### 3.3.2 后台系统设计

在后台系统上，我们使用 Django 框架来搭建后台，并使用开源项目 SimpleUI 来美化我们的后台。Django 作为后台应用的主要框架，拥有轻量化、快速部署、可扩展性高的特点，适合快速的部署和开发。Django 项目的数据库我们使用的是 MySQL，相比于 sqlite，它具备更好的安全性，更快的性能，也更利于管理。在把 Django 项目部署到阿里云服务器上的时候，我们采用 uwsgi 和 nginx 结合的部署方式，uwsgi 是一个快速的、纯 C 语言开发的、自维护的 wsgi 服务器，也长期作为 django 项目部署的一种方式，但是这种 uwsgi 对静态文件的解析存在一些不足，所以我们使用 nginx 来弥补这个缺陷。Nginx 是一个小巧而搞笑的 Web 服务器，具有稳定、功能丰富、低占用和配置简单等优点，对于我们这个项目来说，nginx 拥有两部分的功能，一方面是对静态文件进行解析，相比于使用 uwsgi 解析，nginx 解析更为快速和稳定；另一方面，nginx 发挥反向代理的功能，django 项目开启的服务地址是在本地，但是我们却可以在公网上通过浏览器访问，这就是 nginx 的反向代理，反向代理有一个代理服务器，对外暴露的使代理服务器地址，隐藏了真是的服务器地址，从而可以减少被服务器被攻击，保证了网站的安全性。此外，我们考虑到使用服务器 IP 地址进行网站的访问比较麻烦，我们时常会记不住 IP，为此，我们将域名和服务器 IP 进行了绑定，并设置禁止通过 IP 进行访问，实现更方便访问的同时，又进一步提高了安全性。如图 3.3.2，为后台管理系统界面：



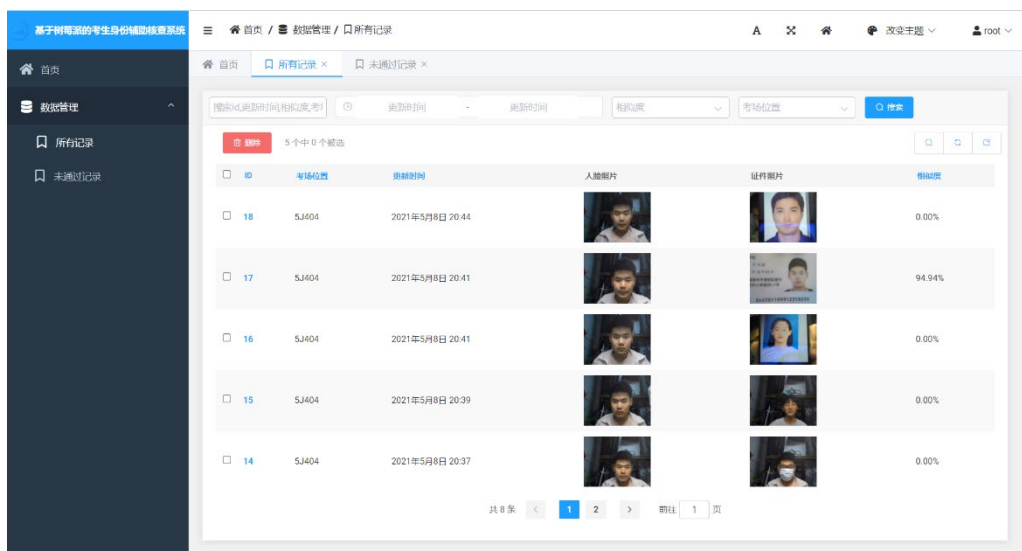
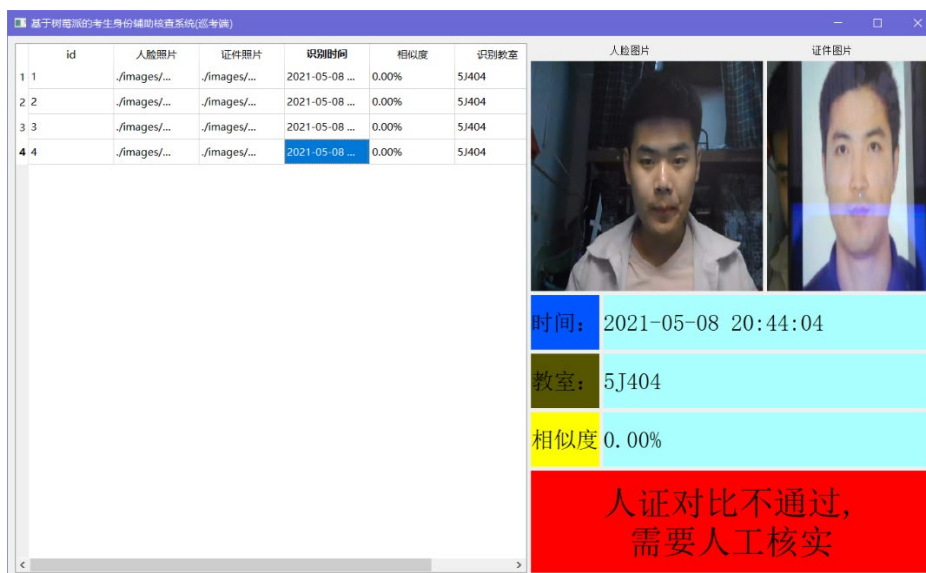


图 3.3.1 管理后台界面

### 3.3.2 巡考端设计

最后是巡考端，巡考端由巡考教师持有，监考端对比未通过的数据会通过管理后台反映给监考端，监考端教师在监考终端上可以看到未通过的考生信息和识别的教室，巡考教室可以前往相关教室进行进一步核实。巡考端依然采用的是 qt 作为整体的开发框架，依然具有跨平台特性，可以部署在任意设备上。巡考端也有相应的数据库，我们采用的是更轻量级的 sqlite3 数据库，监考终端会从服务器获取未通过的数据存储在本地的 sqlite3 数据库中，当监考端程序开启时会创建 sqlite3 数据库，当程序结束后，会删除创建的 sqlite3 数据库及所有的信息，进一步保证数据和隐私安全。以下为巡考端界面截图，左侧为所有的未通过数据，当点击时就会将详细信息显示在右侧，便于查看，左边的未通过数据每十秒将会更新一次，确保监考端教室能够实时获取对比不通过的考生信息。



图五：巡考端界面

### 3.4 功能实现

我们系统的核心功能——人脸对比：人脸对比我们设计了两种实现方式，一种是借助于百度 AI 开放平台的人脸对比 API，另一种是离线识别方案，基于开源图形库 face\_recognition。

在线人脸对比使用百度 AI 开放平台的人脸对比 API，该 API 可以进行两张以及多张人脸照片的对比检测，从而判断两张照片中的人物是否为同一人，该 API 的人脸模型是基于大量人脸照片训练的，识别结果非常准确。受限于树莓派 3B 的性能，我们的人脸对比方式只设置了在线人证一种，这种人证方式准确率高、获得结果的速度快，是非常理想的一种人证对比检测方案。人脸识别、人脸对比这些技术是近些年来人工智能技术发展的优秀成功，但是很多人并不知道，我们的目的是以更低的成本、更好的方式，让人们去了解和应用，一项技术只有大规模应用才能够体现出它的价值，才更能够促进它的发展，永远躺在实验室里面的技术并不是一项好技术。在我们的程序中，我们将对比功能封装在了 Compare 类中，该类主要完成的功能是调用百度 API，以及读取本地拍摄的图片，并将其传入到百度的人脸对比 API 进行对比。对比之后，该类还会自动调用 Upload 类，Upload 类是用来将对比结果和拍摄到的图片上传到我们的私有阿里云服务器上，从而当我们访问网页后台时，就可以查看对比时间、参与对比的人脸照片和证件照片，以及最重要的相似度等信息。

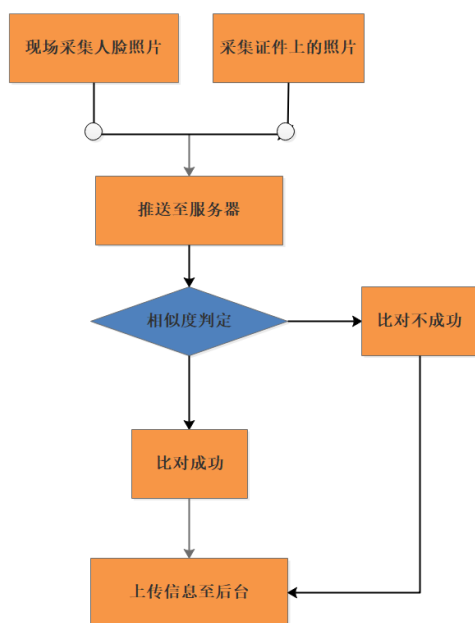


图 3-3-1 在线比对系统

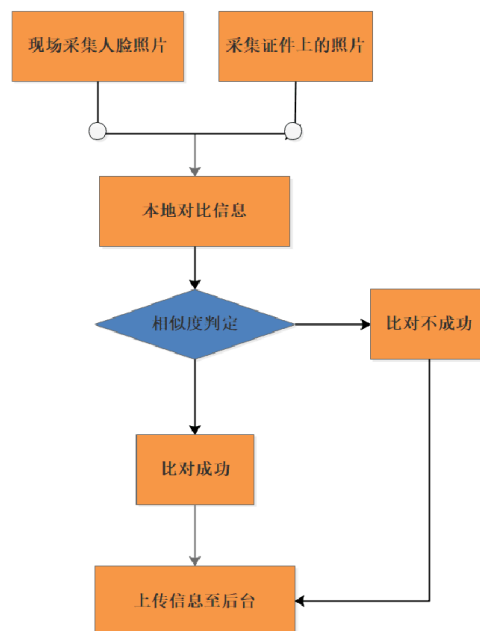


图 3-3-2 离线比对系统

离线人脸对比的方式是主要使用了开源库 face\_recognition，该库是一个基于 dlib 库的 C++ 图形库，通过 python 语言封装，只需要通过简单的 API 就能够实现人脸检测、人脸

对比等操作。这个开源库的原理是先把图片中的人脸数据提取出来，转化为 128D 向量。再进行欧式距离计算，通常使用的判别阈值是 0.6，即如果两个人脸的向量空间的欧式距离超过了 0.6，即认定不是同一个人；如果欧氏距离小于 0.6，则认为是同一个人。这种人脸对比方式可以不受网络的限制，在没有网络或者网络信号较差的时候可以替代在线识别，识别的准确度在可以接受的范围内，但相比于在线对比方案，还有所欠缺，所以我们主要使用在线对比的方案，这种方案作为备用方案。由以上的流程图可以看出，无论是在线对比，还是离线对比，对比过后，都会把图片上传到后台管理系统上去，这样无论在什么地方，都能够查看到识别的数据，管理非常的方便。

然后，就要介绍该系统的数据上传功能，上传功能封装在 Upload 类中，上传功能主要有两个子功能，一是对服务器数据库的操作，将对比的时间、图片在服务器中的路径以及对比结果写入到服务器的 MySQL 数据库，数据库与管理后台相连接，可以从网页后台上实时查看对比的信息，我们使用 python 的第三方库 pymysql 来对数据库进行管理；二是文件上传功能，就是将本地拍摄到的图片上传到服务器指定路径，服务器上的图片存储路径在对数据库操作时便写入到数据库中，网页端可以通过路径来寻找并将图片显示到网页后台，为了保证图片的安全性，我们使用 sftp 协议进行文件传输，sftp 协议相比于 ftp 协议更加的安全，可以在一定程度上保证图片在传输过程中不会被恶意劫持，从而造成用户隐私的泄露。在程序中，我们使用 python 第三方库 pymysql 来进行 ssh 连接和 sftp 文件传输。

# 第四章 测试报告

测试一：



图 4-1 测试结果一

测试二：



图 4-2 测试结果二

测试三:



图 4-3 测试结果三

测试四:

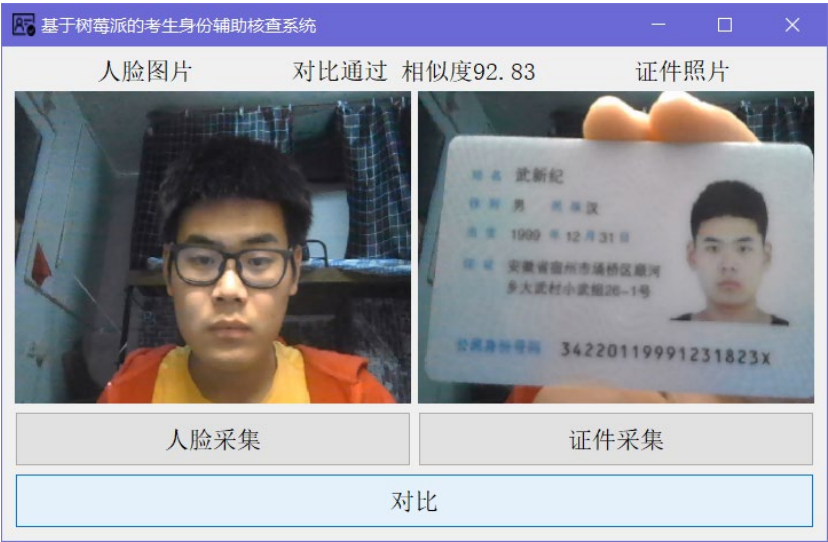


图 4-4 测试结果四

如上图所示，我们测试了三个不同的人，其中一个人在带眼睛和不带眼镜的情况下都进行了测试，识别结果较为准确，基本能够作为考生身份核验的设备使用。在测试中，在线识别的准确率和响应速度更高，所以最终我们选用在线识别的方式来作为主要的识别方式，离线识别方式作为备选方案，在没有网络的情况下使用。



## 第五章 安装及使用

### 5.1 硬件安装

树莓派安装好系统和所需软件后，将树莓派官方 CSI 摄像头连接到树莓派的 CSI 接口，进入树莓派设置，修改树莓派摄像头模式为开启，再将电池和升压模块和树莓派供电接口相连，确保给树莓派提足够的电压和电流，最后将树莓派、摄像头模块以及供电模块一起组装进预先设计好的外壳中，然后使用 Termius 将开发好的项目文件上传到指定目录并配置开机启动。将树莓派重启之后，就可以看到我们系统的主页，上面有一系列功能，可以进行人脸和证件照片的采集以及对比。相关连接如下图所示：



图 5-1 设备安装图一



图 5-2 设备安装图二

### 5.2 软件安装

详细的软件安装见软件安装部署文档，这里只列出使用的软件版本

#### 5.2.1 监考端

操作系统：树莓派官方系统 Raspbian, 对应的 Debian 版本为 Debian 10(buster)

Python 版本：python3.7

使用的 python 第三方库：

```
PySide2==5.15.2
opencv-python==4.5.1.48
opencv-contrib-python==4.4.0.46
pymysql==1.0.2
paramiko==2.7.2
dlib==19.21.1
face_recognition==1.3.0
requests==2.25.1
```

### 5.2.2 管理后台

操作系统: CentOS8

Python 版本: python3.8

MySQL 版本: 8.0.21

Nginx 版本: 1.14.1

Uwsgi 版本: 2.0.19.1

使用的 python 第三方库:

```
django==3.1.7
pymysql==1.0.2
django-simpleui==2021.4.11
```

### 5.2.3 巡考端

系统: Windows10 64 位专业版

Python 版本: python3.8

使用的 python 第三方库:

```
PySide2==5.15.2
pymysql==1.0.2
paramiko==2.7.2
```

## 第六章 项目总结

目前市场有一些基于二代证的人证核实产品，但是成品较高，不能被广泛应用于大规模考试的考生身份验证，其功能也不能很好的满足教育行业的应用。本系统很好地解决了这些问题，利用树莓派和自动对焦摄像头实现人证核实功能，成本较低，且基于手持移动终端，能够随身携带，操作简单便捷，适用于辅助核查考生身份，提高身份核对的效率。

此外，我们的系统采用面向对象的编程方式，各个功能都封装在相应的类中，拥有很好的扩展性。支持深度定制，可以满足各个行业的需求，如学校的监考，为了不泄露隐私和保障信息安全，一般不会部署在公网上，而是部署在学校的内网，我们的系统，只需要更改相应的服务器地址配置，即可以满足这个需求，如果想添加其他功能，如人脸特征检测、眼睛检测等，只需要在主类中添加相应的函数即可，无需更改过多代码。

本系统在正式上线后可以高效地完成多而复杂的工作，最大程度上节省资金资源和人力资源，可广泛应用于一些机构系统中。如：

**考生身份识别系统：**在大规模考试中辅助核查考生的身份可以解决目前教育行业的一大难题，为该行业提供较大的便捷性。

**酒店人员入住信息核查：**随着计算机网络技术的发展，酒店行业也迎来了新的机遇和挑战，智能入住是酒店经营发展必不可少的一部分，利用人脸识别技术住客可以自行登记入住，减少了等待的时间。

**驾驶学员的身份信息认证：**随着时代的发展，考驾照的人数也在逐年增多，该系统可以辅助完成考生的到场验证、学员身份认证、上车下车签到等。

**宿舍楼门禁智能人脸识别：**大学生门禁也是大学教育的一部分，该系统可以智能识别进入的大学生，加强宿舍楼纪律的管理，避免非本校学生进入宿舍楼，减少不必要的损失。

**智能食堂管理系统：**系统设置在收餐台，在学生放置碗筷时进行人脸识别，对于学生单次浪费食物超 30% 的情况予以纪录，超过五次予以警告，减少食物浪费现象。

通过一番的开发和测试，我们发现，我们的系统有如下几个优点：一、成本低，核心部分只使用了一个树莓派 3B 以及一个树莓派 CSI 摄像头，总体成本不过二三百元，若考虑量产，成本将会更低，相比于市面上上千元的产品，我们拥有成本和价格优势；二、我们用于完善的后台管理系统，可以实时的将数据记录到后台数据库中，还可以使用后台管理系统进行查看和管理，无论是系统的安全性，还是人脸照片的隐私保护，我们都做的相对较好。三、我们的终端设备为手持式设备，小巧轻便，不需要携带其他设备，树莓派的低功耗和供电的大电池可以保证该系统具有较长的工作时间。

当然，没有完美的产品，就目前来看，我们的产品存在着几大缺陷：一、受限于树莓派性能，摄像头画面不够流畅；二、手持设备上没有配备物理按钮，电容触摸屏体验效果一般；三、功能相对单一，只有拍照和对比功能。不过我们会对我们的项目进行软件和硬件上的持续优化，以解决目前所面临的问题。



## 参考文献

- [1]. 余文利. 网络环境中数据加密技术实现与分析[J]. 网络与信息, 2005(10): 50-51.
- [2]. 张国亮. 云环境下密文全文检索技术研究[D]. 西安: 西安电子科技大学, 2015.
- [3]. 陆文周. 《Qt5 开发及其实例》电子工业出版社 2019. 4
- [4]. 陈河堆. 组件系统安全机制及其协议的设计与实现[D]. 南京: 南京航空航天大学, 2005.
- [5]. 何敏煌, 林亮昀 《Django2.0》 清华大学出版社 2019. 4
- [6]. 李立宗. 《OpenCV 轻松入门》. 电子工业出版社 2019. 10
- [7]. 小宋的博客. 面向监考业务的人证核实移动终端与系统 2019. 10
- [8]. 赵邢瑜. 一种人脸生物特征加密系统的研究与实现[J]. 哈尔滨理工大学 2013, 3