

毕业设计说明书

基于视觉物联网技术的门禁系统设计 与实现

学生姓名： 陈怀颜 学号： 1607094143

学 院： 大数据学院

专 业： 物联网工程

指导教师： 秦品乐

2020 年 5 月

基于视觉物联网技术的门禁系统设计与实现

摘要

在人工智能不断发展的新科技时代中，人们的生活习惯开始不断偏向方便快捷的智慧生活，这也就导致了门禁系统不断地向高科技高水平的智能方向不断发展。然而由于每个人脸面部的不同特征导致把人脸应用在门禁系统成为了一个新的研究发展方向。我们知道人脸识别在很多地方已经有了非常成熟的应用包括手机的智能解锁，付款等。但是在门禁中人脸表情包括多人的通过和是否佩戴眼镜等的影响。都会导致识别人脸的准确度。

本文主要针对人脸识别的几个关键过程进行研究，其中包括人脸的图像的检测过程到面部特征的提取，再到与本地人脸数据库的比对从而达到身份的验证，利用数据库来管理人脸数据和实现人脸数据的可视化。

关键词：物联网，人脸识别，门禁系统

Design and Implementation of Access Control System Based on Visual Internet of Things Technology

Abstract

In the era of new technology in which artificial intelligence is constantly evolving, people's living habits are beginning to favor convenient and efficient smart life, which has led to the continuous development of access control systems towards high-tech and high-level intelligence. However, due to the different characteristics of each face and face, it has become a new research and development direction to apply the face to the access control system. We know that face recognition already has very mature applications in many places, including smart phone unlocking, payment, etc. However, facial expressions in access control include the influence of the passage of multiple people and whether to wear glasses. Will lead to the accuracy of face recognition.

This article focuses on the research of several key processes of face recognition, including the detection process of face images to the extraction of facial features, and then the comparison with the local face database to achieve identity verification, using the database to manage people Face data and visualization of face data.

Key words: face recognition, access control system, Internet of Things

目录

| | | |
|-------|------------------|----|
| 1 | 需求分析 | 1 |
| 1.1 | 背景分析 | 1 |
| 1.2 | 门禁系统的发展和国内外的研究现状 | 2 |
| 1.3 | 常用的人脸识别方法 | 2 |
| 1.4 | 技术可行性 | 4 |
| 1.5 | 项目概要 | 5 |
| 1.6 | 项目需求 | 5 |
| 1.7 | 用户角色和功能摘要 | 6 |
| 2 | 概要设计 | 7 |
| 2.1 | 开发环境 | 7 |
| 2.2 | 系统功能描述 | 7 |
| 2.3 | 系统功能设计 | 8 |
| 2.4 | 概念结构设计 | 9 |
| 2.5 | 系统功能结构 | 9 |
| 2.6 | 概念结构设计 | 11 |
| 2.7 | 逻辑结构设计 | 11 |
| 2.8 | 物理结构设计 | 11 |
| 3 | 具体功能分析 | 13 |
| 3.1 | 登录注册模块 | 13 |
| 3.2 | 人脸检测识别模块 | 14 |
| 3.2.1 | 人脸检测模块 | 14 |
| 3.2.2 | 人脸识别模块 | 15 |
| 3.3 | 后台数据管理模块 | 17 |
| 3.4 | 整个系统 | 17 |
| 4 | 详细设计 | 20 |
| 4.1 | 登录注册模块 | 20 |

中北大学 2020 届毕业设计说明书

| | | |
|---------|----------------|----|
| 4. 2 | 人脸检测模块 | 21 |
| 4. 3 | 人脸识别模块 | 22 |
| 4. 4 | 功能核心代码实现 | 23 |
| 4. 4. 1 | 登陆功能核心代码 | 23 |
| 4. 4. 2 | 人脸识别核心代码 | 24 |
| 5 | 测试 | 25 |
| 5. 1 | 登陆测试 | 25 |
| 5. 2 | 人脸检测测试 | 26 |
| 5. 3 | 人脸识别测试 | 27 |
| 6 | 总结 | 30 |
| | 参 考 文 献 | 31 |
| | 致 谢 | 33 |

1 需求分析

1.1 背景分析

在物联网技术不断地发展和普及过程中,传统的门禁系统逐渐展示出其不够方便快捷和灵敏度不够等的问题。智能化、网络化和高灵敏度的门禁将成为未来门禁系统的发展趋势^[1]。

我们在传统的门禁系统中加入快速人脸识别技术,将有助于在行人通过时实现快速准确的记录下通过的人员,方便管理和查询。传统的门禁系统需要钥匙,密码,IC 磁卡等来识别身份,这一定程度上加大了人力和物力的消耗,有时候还有忘记随身携带识别卡和忘记密码忘带钥匙的尴尬场面。显然,如果我们引用了人脸识别的门禁系统,可以有助于管理人员管理出入人员,方便出入人员无阻碍快速的通过门禁,极大地缩减了时间提高效率同时还有有效的改善了人们的生活方式^[2]。

现在比较常见的生物特征识别方法主要是人脸,指纹和虹膜三种。这三种方法在因为特征的不同在识别方面各有优势,虹膜的图像采集仪器比较昂贵所以应用的场所比较少。指纹识别虽然设备价格低廉,但是在安全和用户体验上不是很友好,在手上沾有水或是其他异物的时候会导致识别出错。相比于这两种识别方式,人脸识别的优点主要在于成本比较低,方便快捷,安全性高,人脸信息容易获取等方面。

人脸的识别还可以还能运用在许多不同的地方,如监控、视频通信等场合。人脸识别对于人们是很容易的事。实验显示,甚至出生几天的婴儿已能够区分已知的面孔。对电脑来说是人类的内在部件(眼睛,鼻子,嘴)和外部特征(头形,发际线)使之成功识别。脸识别过程中,需要考虑不同表情、光照变化、角度旋转等多种因素,故需要强健的算法来进行识别。本文就针对这些情况,深入学习两种识别人脸的算法特征脸方法和 Fisherface 方法,通过实验数据分析算法的优劣性并进行了对比。图像库包含了同一人脸的不同表情、不同角度等多种情况^[3]。

虽然人脸识别有许多优点,但是在很多物理条件上依旧限制着人脸识别的发展。比如环境的嘈杂,光照的强弱,和人脸与采集设备的距离等都影响着人脸识别得到精度。处于这次疫情,在必须戴口罩的情况下,人脸识别解锁手机的功能收到了很大的影响,极大地影响着人们的使用体验。因此为了提高安全性,使用的体验和更好的使用,对人脸识别技术的研究是非常有必要的。

1.2 门禁系统的发展和国内外的研究现状

随着社会的不断发展和人们生活品质的不断提升，对门禁的要求也越来越高了，出于出于安全和方便快捷的考虑。现在不断有各种新闻关于偷到，隐私泄露和各种各样的安全隐患，这些外在因素迫使着门禁系统的不断发展。于是只能的门禁系统开始摆在了项目研究人员的桌面上^[4]。

目前最典型的静态人脸图像识别研究方法是特征脸方法、神经网络方法以及弹性模板匹配方法在国外门禁系统的发展和使用都比较早，比如产业做的比较好的公司有美国的休斯和西屋公司，它们通过将 IC 卡应用到门禁系统中这一方式，给用户带来便利从而快速的占领当地市场，同时它们生产的产品性价比很高，于是拥有了大量的忠实客户进而有了更多的需求，不断的推动门禁系统向前发展进步^[5]。

在国内人脸识别这个方向虽然发展的比国外要晚一点，但是国家对新兴产物的重视以及推动使得人脸识别技术得到了快速的发展和壮大。这些都是由于国家提供了许多的科研资金，技术支持。政策支持，不断地鼓励着国内的科学家发展着我国的人脸识别技术。时至今日已经取得了不可忽视的辉煌成就。目前在这一方面比较领先的科研机构主要在一些比较知名的高校研究所包括：清华大学，北京大学，上海交通大学，南京大学，西安电子科技大学等。比较出名的科研公司有腾讯，百度在内的许多出名的互联网公司都取得了非常好的成就。在国内的应用上，已经有许多的 Alipay 的人脸识别自动销售机器了，未来的人脸识别进出站和普及消费也只是时间问题^[6]。

现在的门禁系统中添加了许多科技化元素比如语音，指纹，虹膜等技术使得门禁系统变得更加智能化。这些是生物体比较唯一的特征结合到门禁系统中可以有效的改变传统的门禁系统的格局。从而保护人们的隐私。因此门禁系统无论是在价值还是在用途上都是具有可以研发和发展的^[7]。

1.3 常用的人脸识别方法

目前最典型的静态人脸图像识别研究方法是特征脸方法、神经网络方法以及弹性模板匹配方法，常用的主要有以下几种：

1. 特征脸方法：

特征脸方法即 Eigenface，用的是主成分分析的方法即 PCA（Princial Component Analysis），主成分分析多用于人脸图像的表示与识别，主成分分析方法是人脸识别

最基本的方法之一,PCA 算法提供了一个从高维空间到低维空间的线性变换矩阵,它是在最小均方差意义上对原图像的估计,具有计算效率高,概念清晰,推广性强等优点,起初用于统计学,找出某集合的主要成分^[8]。

2. 人工神经网络 (ANN) 法:

常用的网络结构有线性自相联网、BP 网 (非线性) 以及隐马尔科夫模型等多种。采用人工神经网络方法进行人脸识别需要先借助训练把视觉刺激建成一种内部表示,而且分散地存储在某种神经结构中;采用此方法的搜索过程与人类大脑的联想方式有些类似,此方法即使在待测图像有小幅缺损时仍能有效识别;但这种方法训练过程较复杂,网络容量也对其有所限制^[9]。

3. 弹性匹配法:

此方法是一种动态模板匹配法,模板可以有基于图像表示法、基于特征表示法等几种表现形式,此方法使用点格阵表示,采用的是人脸特征点的信号向量信息。多是在人脸图像上应用栅格,取得特征信息^[10]。然而采用信号做分析也有其缺陷,因为此算法需要提取特征系数,尽管识别准确度较高,但其运算速率低下,无法实现实时性^[11]。

4. 几何特征分析方法:

几何特征分析方法是人脸识别的最早方法,具有直接、简单的优点,便于机器实现,首先提取人脸特征,将特征值组成向量形式,再比较待测人脸图像与人脸库中的图像的特征向量,利用距离公式计算并作出决策。该方法使用投影的方式来确定人脸的眼鼻嘴等区域特征,并计算各特征点间的距离、角度等参数作为该图像的特征向量用于比较^[12]。几何特征分析算法简单、识别速率高,但它要求图像均为正面人脸,不能有角度与形状的变化,因此精度不高,应用较少^[13]。

5. 局部特征分析法:

上面讲述的特征脸方法能够降低图像空间的维度,能更加简洁有效地表示人脸图像,但其无法了解到人脸图像的局部特征或拓扑结构,不适合后续的图像分割、特征提取等处理过程,这里的局部特征分析法也是用的低维人脸空间,并能够有效表示人脸图像的局部特征和拓扑结构,目前已在某商用软件中得以采用^[14]。

6. 柔性形状模型技术:

由于人脸的形状也会有些变化,因此固定的人脸模型不能有效体现那些变化。参数

化可变模板技术可用来有效表示人脸各部位特征，做出模板旋转、形变、便宜等操作后可以得到最佳匹配模板，能够有效提取人脸特征，然而其计算较复杂，运算量很大^[15]。柔性形状模型技术应运而生，它包含建模阶段和识别阶段，前者用于获得人脸灰度值分布模型和形状模型，以描述待测人脸图像，识别阶段利用这些信息作出识别^[16]。

7. Fisherface 方法：

Fisherface 方法目的是从高位特征空间里提取出最具有判别能力的低维特征，它采用了 PCA+LDA 的两步处理首先用 PCA 方法把人脸向量降到适当的低维空间，再用 LDA 方法获得最终的低维数据。不仅去掉原始图像的大量冗余信息，还把图像中的有用信息保留了下来，既提高了人脸识别率，又降低了识别过程的计算复杂度。Fisherface 方法用的是种类明确的变换矩阵，所以不必像 Eigenface 方法一样尽可能明确地捕捉光照^[17]。判别式分析并不是去找面部特征以区别不同人。有一点值得一提：Fisherface 方法的表现严重依赖输入数据。如果用 Fisherface 方法只学习光照好的图片而想去识别光照不好的图片，此方法很可能找到错误的成分，因为那些特征可能在光照不足的图片里没有支配力。这些是由逻辑的，因为此方法没机会处理光照度的问题^[18]。

8. 综合方法：

主要将特征脸法、神经网络法、几何特征法结合起来，取各家所长组成新的算法，识别效率得以显著提升，但也需要考虑各种环境因素。应用几何特征法能有效保持各特征点的相对距离，可以在非正常形变下保证识别效果；特征脸方法也不善于处理非正常形变图像；模板匹配法在光照度变化时处理能力相对要强一点；弹性匹配法使人脸图像可拉伸，有效解决了形变等问题，为上述几种算法做好基础；神经网络法也善于应对各种变化，处理能力较强^[19]。

1. 4 技术可行性

该系统是基于 Springboot, python 中的 Pytorch 库和 dlib 开源库等技术手段进行开发设计的。整个系统基于训练好的人脸检测模型加上人脸识别的 dlib 开源库结合 Pyqt5 的界面设计得到的。检测得到的人脸利用 Springboot 来管理^[20]。

PyTorch 的前身是 Torch，其底层和 Torch 框架一样，但是使用 Python 重新写了很多内容，不仅更加灵活，支持动态图，而且提供了 Python 接口。它是由 Torch7

团队开发,是一个以 Python 优先的深度学习框架,不仅能够实现强大的 GPU 加速,同时还支持动态神经网络,这是很多主流深度学习框架比如 Tensorflow 等都不支持的。Dlib 是一个现代化的 C++工具箱,其中包含用于在 C++中创建复杂软件以解决实际问题的机器学习算法和工具。它广泛应用于工业界和学术界,包括机器人,嵌入式设备,移动电话和大型高性能计算环境。Dlib 的开源许可证 允许您在任何应用程序中免费使用它。在管理选择 JSON 作为数据交互的方法,JSON 是一种轻量级的数据交互. 格式,安全、快速、通用、数据流量低。当移动端获取到 JSON 数据时,会根据得到的 JSON 数据把它解析为 JsonObject 对象,进而转化为需要的数据。

1.5 项目概要

本项目主要的用户就是后台的人脸管理员和使用人脸通过门禁的用户。其中管理员有注册登录功能,后台人脸数据的添加和人脸识别的通过程序。其中人脸检测部分是先检测得到人脸图片然后与本地数据库的正脸图片进行比对和匹配,得到相似度比例,这个数据值越是接近 1 说明这张脸与匹配的图片接接近本人。当数值比例不是很确定的时候根据与每一张图片比对得到数值时的大小,最大值就是最相似的图片。

1.6 项目需求

需求分析概要表管理员端如表 1.1 所示:

表 1.1 需求分析概要表管理员

| 需求编号 | 需求内容 |
|------|-------|
| X1 | 管理员注册 |
| X2 | 管理员登陆 |

需求分析概要表用户端如表 1.2 所示:

表 1.2 需求分析用户表

| 需求编号 | 需求内容 |
|------|------|
| X1 | 人脸录入 |
| X2 | 人脸检测 |
| X3 | 人脸识别 |

1.7 用户角色和功能摘要

用户在这个系统中主要是刷脸进入系统完成系统的功能。每一个用户都是固定的用户，有自己的正脸图片在数据库中。

用户角色说明如表 1.3 所示：

表 1.3 用户角色说明

| 用户角色 | 用户描述 |
|------|------------------------------|
| 管理员 | 对用户人脸数据进行管理，对管理员用户进行添加 |
| 用户 | 系统对用户进行人脸检测，然后与数据库进行比对进行人脸识别 |

2 概要设计

2.1 开发环境

服务器: Tomcat9.0

开发语言: Java、SQL、Python

开发模式: B/S

开发工具: IntelliJ IDEA

MySQL Workbench 8.0 CE

Pytorch 框架

Dlib 库

开发系统环境: Windows10 专业版 64bit、jdk 1.8

数据库: MySQL 5.7

2.2 系统功能描述

本系统主要用 Python 和 IntelliJ IDEA 通过引用外部 Dlib 库人脸识别的可视化界面主要用的是 PyQt5 来进行开发。人脸检测是使用了 Pytorch 库, pytorch 是一个由 facebook 开发的深度学习框架, 它包含了一些比较有趣的高级特性, 例如自动求导, 动态构图等。

在人脸检测中先将自己的人脸正脸图片放入待检测的位置, 然后运行程序进入检测部分, 检测出人脸后将检测到的图片放入待识别的位置, 程序继续运行进入识别的部分, 将正脸图片与检测图片进行比对, 从而判断出匹配度。根据匹配度的高低判断出人的姓名。检测得到的人脸与识别中的人脸用 PyQt5 做成可视化界面显示在桌面上, 方便可以调试使用。

在人脸正脸图片中加入管理用来管理人脸数据图片其中使用了 Springboot 来管理快速搭建环境, 数据放在 Mysql 数据库中进行管理。系统有注册登陆的简单功能, 方便管理。其中注册时需要重复输入密码方便记住密码。

人脸识别是对于本地存放的正脸图片的要求是尽量一张只有人脸的正脸图, 在检测人脸时, 人脸的摆放位置和角度也影响着这个检测效果不然出现了不一样的结果会导致失败。这个系统运行开始从管理界面开始运行, 然后进行到人脸的检测和识别模块, 对于整个系统的运行从点击开始运行开始进入 Springboot 的服务器,

然后运行 web 页面，然后加入可以运行的 CMD 命令，打开 Python 开始运行人脸检测。先打开摄像头拍照得到图片并保存，然后对保存的图片分析是不是正脸图片提取特征点。把提取到的特征点的图片保存，然后开始运行识别模块使用的是 Dlib 库，这是个开源的，免费的，持续更新维护人脸识别库。将图片与之匹配得到匹配的精确度，然后输出最高匹配度的名字。

2.3 系统功能设计

(1) 管理员注册登陆模块

用账号注册登陆的录入信息如表 2.1 所示：

表 2.1 登陆注册录入信息表

| 名称 | 字段 | 类型 | 备注（必填*） |
|-----|----------|--------|---------|
| 用户名 | username | String | * |
| 密码 | password | String | * |

注：注册信息判断：

用户是否已存在，密码是否正确

(2) 人脸检测功能模块

使用 Pytorch 框架，利用网上别人训练好的人脸检测模型，跑出人脸检测效果，存入本地，当人脸有两张时存入两张实时图片。

在人脸检测时对必要的系统关键软件的安装需要很大的耐心，在安装 Pytorch 时需要翻墙才能成功安装，不然在安装时就会发生安装失败的提示。

(3) 人脸识别功能模块

当获得了人脸检测到的人脸图片后把利用开源的 Dlib 库讲人脸与本地人脸正脸图片进行比对，得到匹配精度，这样得到的的最高的精度就是那张图片的人。输出图片人脸的名字在可视化操作界面中。

存入本地正脸图片最好是正脸高清的人脸图片，方便在提取脸部特征时获取到比较多的正确的点，可以用来确定是不是正确的结果。

(4) 人脸检测识别

在检测和识别放在一起运行的时，把检测得到图片存放的路径当做识别时提取图片的路径，做到实时检测和识别的进行。

检测识别放在一起时，可以做到检测和识别的同时进行，具有时效性。检测识

别同时运行时由于程序占用内存过大会导致整个系统非常卡顿，甚至会导致电脑死机的情况，在接入电源和保证其他程序没有运行的情况下一起运行整个程序。

2.4 概念结构设计

系统中管理端有管理员，管理员的实体 E-R 图 2.1 如图所示：

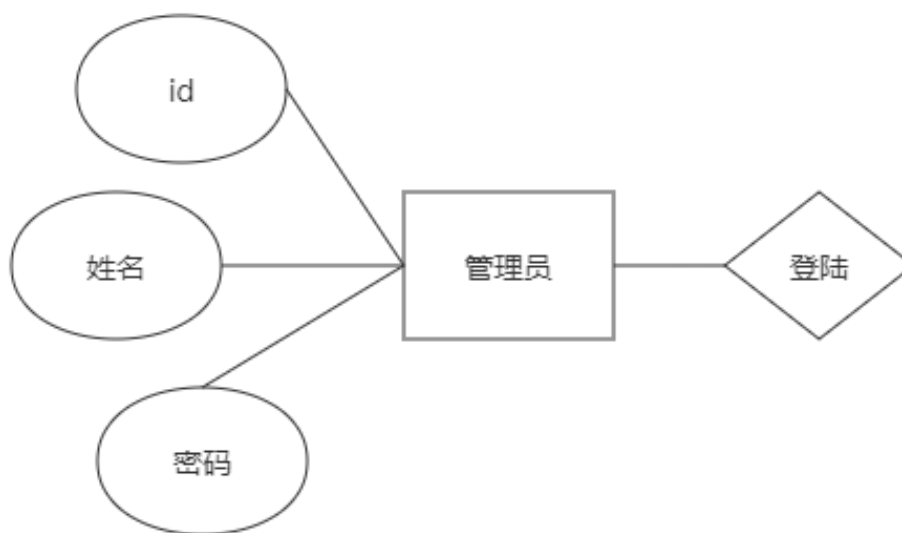


图 2.1 管理员 E-R 图

2.5 系统功能结构

本系统包含服务器 web 端和人脸检测识别端。整个系统运行开始从管理界面开始运行，可以注册登陆一个管理员信息，用来管理自己的本地图片及正脸图片功能，管理员有姓名和密码。进入系统必须登陆，登陆时向服务器请求数据是否存在，若正确返回登陆信息，若不匹配重新登陆。注册成功向服务器数据库添加信息系统的总体功能结构如图 2.2 所示：

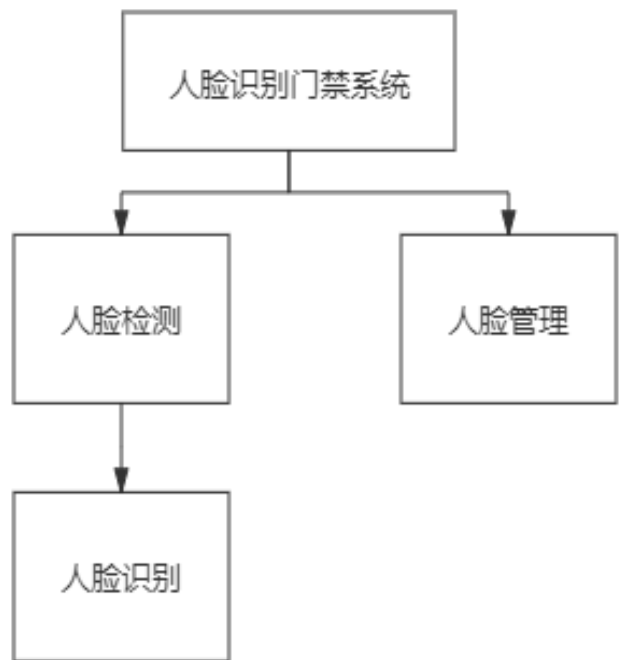


图 2.2 系统总体结构图

其中人脸检测部分的流程图如图 2.3 所示

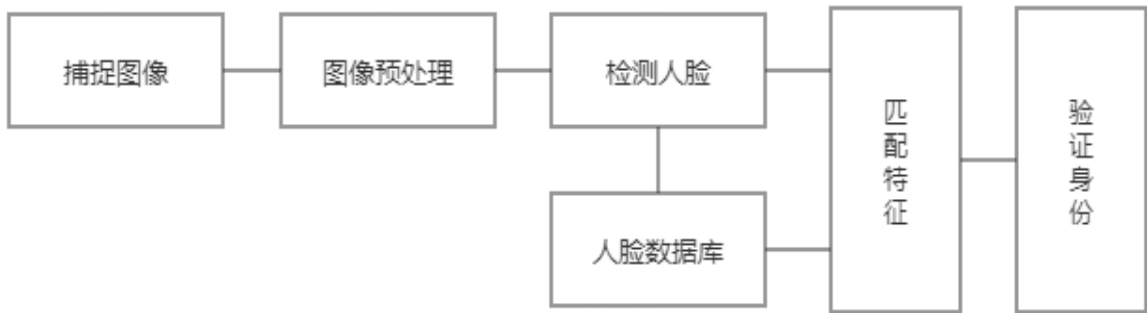
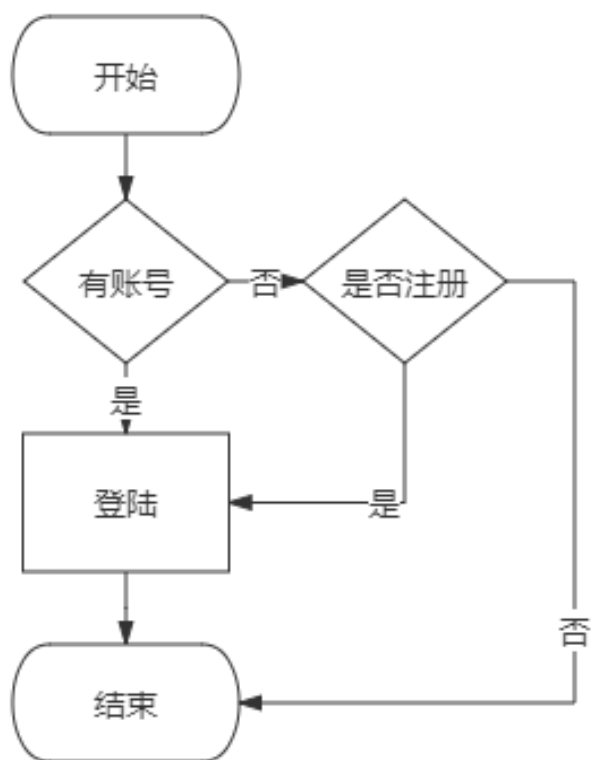


图 2.3 人脸检测流程图

对服务器端数据管理登陆主要是登陆流程的加入和账号的注册几点，账号的注册主要是确保 ID、姓名、密码的正确输入和不可重复性的要求。的流程图如图 2.4 所示：



图表 2.4 登录注册流程图

2.6 概念结构设计

本系统中存在人脸管理端和人脸识别端，在人脸数据的管理中，可以有注册多名管理者管理系统中多张图片。人脸的检测识别端，在检测和识别中 GPU 加速加快检测进程，识别中的人脸识别是调用的 Dilb 库完成。

具体的流程关系如图 2.5 所示：



图 2.5 检测识别流程图

2.7 逻辑结构设计

在该系统的关系模型中，关系的码用下划线标出，具体如下：

管理员（ID、姓名、密码）

用户（人脸）

2.8 物理结构设计

(1) 管理员信息表的设计

数据库中的管理员信息如表 2.2 所示：

表 2.2 管理员信息表

| 字段名 | 是否为空 | 类型 | 说明 |
|-----|----------|---------|----|
| ID | NOT NULL | int | * |
| 姓名 | NOT NULL | Varchar | * |
| 密码 | NOT NULL | int | * |

注：主键 ID

(2) 人脸数据存储表

人脸和数据暂时存储在本地文件夹中，项目启动时利用路径访问图片。在没有人脸数据管理的情况下，对文件夹中每个图片的管理就非常重要了，这会直接影响整个系统的结果，从而导致这个输出名字的正确性。名字和图片是一对一对应的，加上一定的字段保证图片的排序是储存人脸的名字。

3 具体功能分析

3.1 登录注册模块

(1) 功能描述

可以注册登陆一个管理员信息，用来管理自己的本地图片及正脸图片功能，管理员有姓名和密码。进入系统必须登陆，登陆时向服务器请求数据是否存在，若正确返回登陆信息，若不匹配重新登陆。

在系统中人脸数据是可以放在数据库中用来匹配的还可以放在本地文件夹中用来系统运行时调用。

(2) 用例图

登录注册模块用例图如图 3.1 所示：

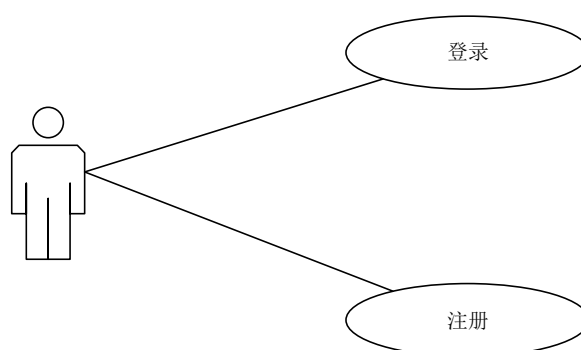


图 3.1 登陆注册用例图

整个系统的运行时的结构用例图如图 3.2 所示：

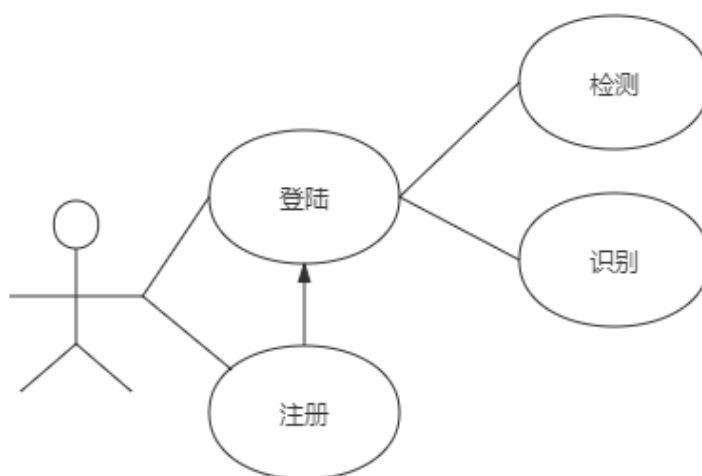


图 3.2 系统结构用例图

(3) 数据流图

登录注册模块数据流图如图 3.3 所示：

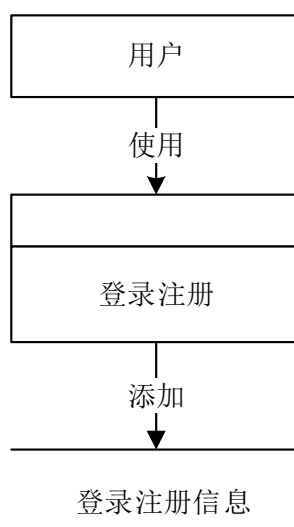


图 3.3 登录注册模块数据流图

3. 2 人脸检测识别模块

该模块主要是系统主要功能，是对人脸进行识别检测的部分。

3. 2. 1 人脸检测模块

该模块主要是对人脸的检测与存档，用 `pytorch` 框架下载训练好的人脸检测的模型，加上一定量的自己对模型框的调整和匹配，对精度的要求后可以实现一定程度上的比较精确的人脸的。

检测得到的图片存放的位置直接决定着识别时的图片识别时的存放位置，这事需要放好记住，图片存放的位置，随之出来的图片有一开始打开摄像头获得的图片，因为获得的图片是动态的所示设置了动态的 `cnt` 来计数，用来确定图片的编号，可以准确得到检测的图片。

对于图片的处理可以使用符合实时人脸大小的框来获取人脸特征。在检测和识别放在一起运行的时，把检测得到图片存放的路径当做识别是提取图片的路径，做到实时检测和识别的进行。检测识别放在一起时，可以做到检测和识别的同时进行，具有时效性。

人脸检测的数据流程图如图 3.4 所示：



图 3.4 人脸检测的数据流程图

用户人脸检测的用例如图 3.5 所示：

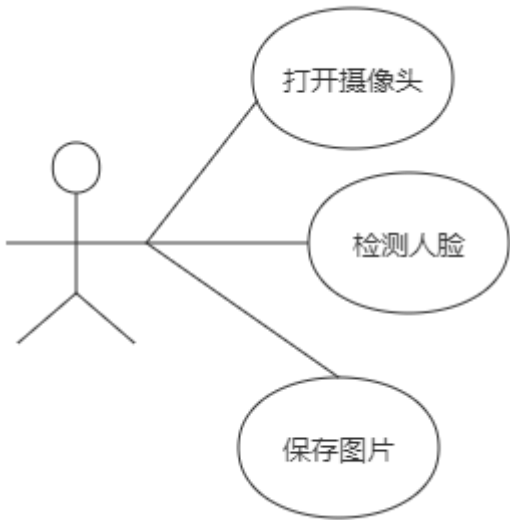


图 3.5 人脸检测的用例

3.2.2 人脸识别模块

调用人脸识别的 Dlib 库把检测得到的人脸图片与本地预存可进入人员的人脸数据进行匹配得到匹配的精度，其中比率最高的那张图的进度就是该张图片可以通过的人脸数据，及这张人脸可以通过门禁系统。人脸识别时的图片获取的路径也非常重要，这个是检测是的图片，因为获得的图片时动态的这时，获取图片也要加入 cnt 计数来动态获取，确定获得的图片时实时的。

调用 Dlib 库对于安装有一定的要求，在 python 的 pip install 中因为国内源的问题需要换源这时当你换上不是清华的 python 源时会因为版本问题导致安装时的失

败，这个问题虽然不大，但是也会耽误很大一部分自己的时间。当获得了人脸检测到的人脸图片后把利用开源的 Dlib 库讲人脸与本地人脸正脸图片进行比对，得到匹配精度，这样得到的的最高的精度就是那张图片的人。输出图片人脸的名字在可视化操作界面中。存入本地正脸图片最好是正脸高清的人脸图片，方便在提取脸部特征时获取到比较多的正确的点，可以用来确定是不是正确的结果。

人脸检测和识别的过程流程图如图 3.6 所示：

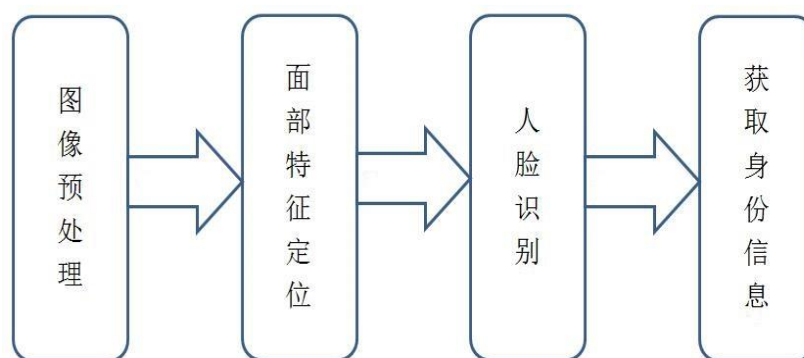


图 3.6 检测识别流程图

对于识别流程来说，在识别时的数据流程图如图 3.7 所示：

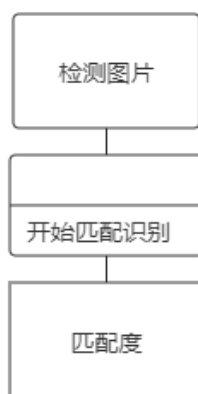


图 3.7 人脸识别的数据流程图

从流程图可以看出人脸识别与检测有着类似的步骤从提取图片，判断特征，到保存图片都有着大概相同的的操作步骤。

对于系统用户来说用户的功能如图 3.8 所示：

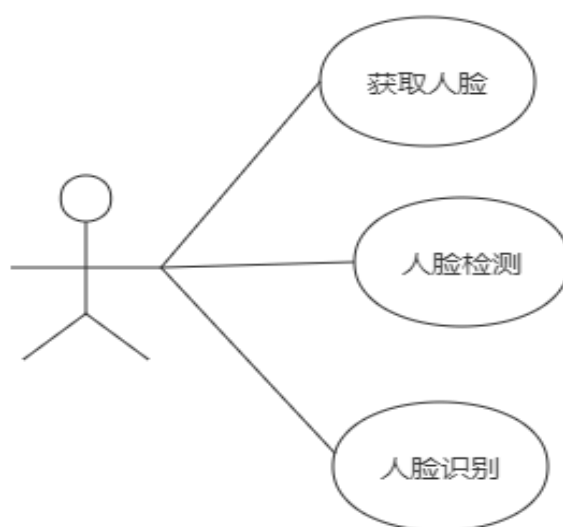


图 3.8 系统用户用例图

3.3 后台数据管理模块

本地库已存人脸按照顺序放在项目文件夹 candidate-faces 下，尽量存放清晰的正脸人脸图片，方便在检测时可以快速高效的获取图片的关键点特征。

系统的后台数据库数据流程如图所示 3.9 所示：



图 3.9 数据库数据流程图

3.4 整个系统

整个系统运行开始从管理界面开始运行，可以注册登陆一个管理员信息，用来管理自己的本地图片及正脸图片功能，管理员有姓名和密码。进入系统必须登陆，登陆时向服务器请求数据是否存在，若正确返回登陆信息，若不匹配重新登陆。

注册成功向服务器数据库添加信息。

然后进行到人脸的检测和识别模块，对于整个系统的运行从点击开始运行开始进入 Springboot 的服务器，然后运行 web 页面，然后加入可以运行的 CMD 命令，打开 Python 开始运行人脸检测。

先打开摄像头拍照得到图片并保存，然后对保存的图片分析是不是正脸图片提取特征点。把提取到的特征点的图片保存，然后开始运行识别模块使用的是 Dlib 库，这是个开源的，免费的，持续更新维护人脸识别库。将图片与之匹配得到匹配的精确度，然后输出最高匹配度的名字。

人脸正脸图片中加入管理用来管理人脸数据图片其中使用了 Springboot 来管理快速搭建环境，数据放在 Mysql 数据库中进行管理。系统有注册登陆的简单功能，方便管理。其中注册时需要重复输入密码方便记住密码。整个流程图如图 3.10 所示：



图 3.10 系统流程图

系统的运行就是从登陆到获取获取人脸，然后检测人脸特征存储到本地，最后进行识别度的计算在系统中形象的用用例表示系统的整个运行结构如图 3.11 所示：

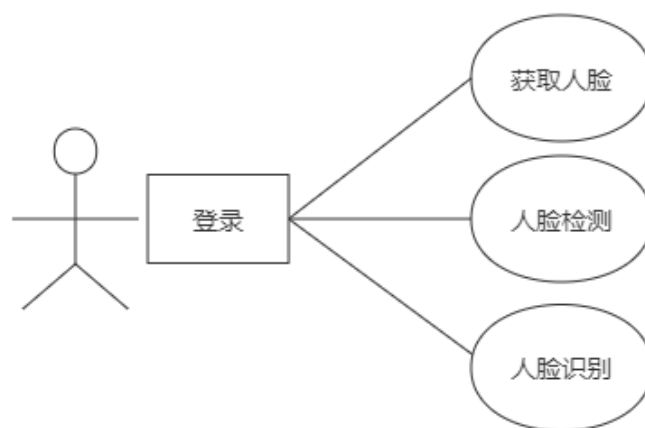


图 3.11 系统运行用例图

4 详细设计

4.1 登录注册模块

(1) 功能描述

可以注册登陆一个管理员信息，用来管理自己的本地图片及正脸图片功能，管理员有姓名和密码。进入系统必须登陆，登陆时向服务器请求数据是否存在。

若正确返回登陆信息，若不匹配重新登陆。注册成功向服务器数据库添加信息。

注册功能的 IPO 图如图 4.1 所示：

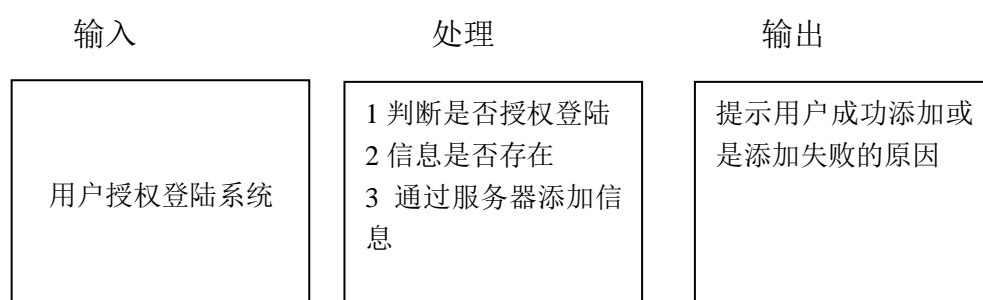


图 4.1 注册功能的 IPO 图

注册成功后，用户可以选择合适的账号登陆系统来管理图片然后启动服务，打开这个系统的构架。跳转到人脸检测与识别页面。若是失败则是返回注册注册界面重新注册，到注册成功为止。

其中登陆中把姓名与密码与服务器数据库进行比对，匹配用户信息通过时跳转到相应界面。

登陆成功的 IPO 图如图 4.2 所示：

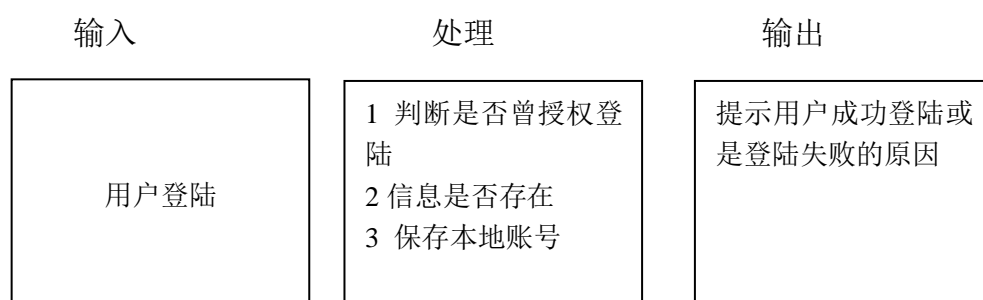


图 4.2 登陆功能的 IPO 图

(2) 流程图

系统登录注册模块测试计划如表 4.1 所示：

表 4.1 登陆注册测试计划表

| 序号 | 名称 | 目的 | 内容 |
|------|--------|--------|-----------------------------|
| 测试 1 | 注册功能测试 | 测试注册功能 | 测试用户账号唯一性、稳定性、非空判断、能否保存到数据库 |
| 测试 2 | 登录功能测试 | 测试登录功能 | 测试登录账号密码是否匹配，登录成功能否跳转到主页面 |

(3) 界面设计

登陆注册界面如图 4.3 和 4.4 所示：

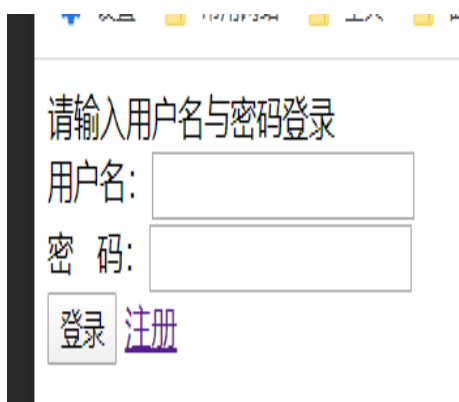


图 4.3 注册模块界面图



图 4.4 登陆模块界面图

4.2 人脸检测模块

人脸识别模块是先检测人脸，然后识别最后输出结果，当检测不到人脸时会一直在摄像头的检测界面，然后得到的图进行匹配。

检测的 IPO 图如图 4.5 所示：

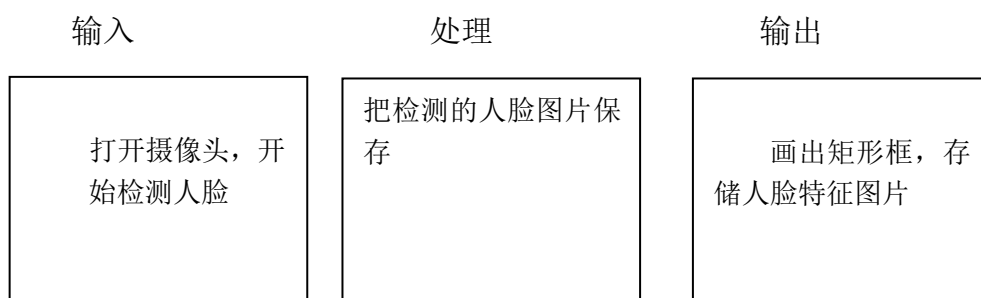


图 4.5 检测 IPO 图

人脸检测与识别模块的界面设计图在开始测试检测人脸时测试用例是从图片开始的，即没有调用摄像头的人脸检测模式，开始的本地图片检测测试图如图 4.6 所示：

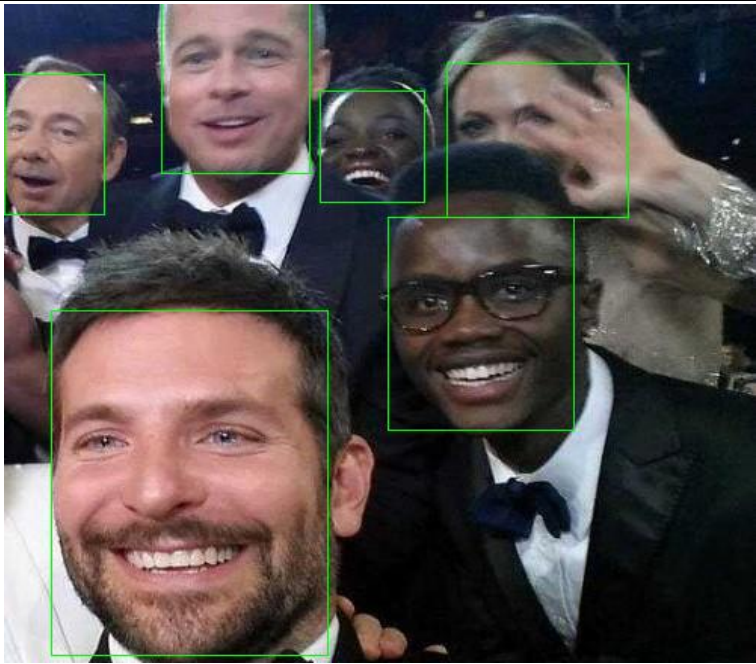


图 4.6 本地图片检测图

在检测到本地图片后加入摄像头的检测，摄像头的检测图片如图 4.7 所示：



图 4.7 人脸检测界面图

4.3 人脸识别模块

人脸识别是根据检测得到的人脸图片，然后与本地的人脸数据进行比较匹配，识别的 IPO 图如图 4.8 所示：

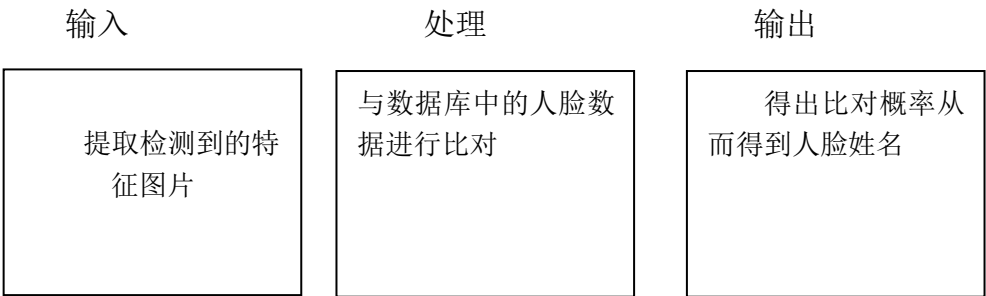


图 4.8 检测 IPO 图

人脸的识别模块的截面图如图 4.9 所示：



图 4.9 人脸识别模块截图

4. 4 功能核心代码实现

4. 4. 1 登陆功能核心代码

其中登陆注册的核心代码如下所示：

```
@ResponseBody
@RequestMapping(value = {"/userLogin"})
public String userLogin(@RequestParam("username") String username,
                        @RequestParam("password") String password,
                        HttpServletRequest request) {

    if(StringUtils.isEmpty(username)) {
        return "用户名不能为空";
    }

    if(StringUtils.isEmpty(password)) {
        return "密码不能为空";
    }

    User user = userService.userLogin(username, password);

    if(user != null) { //登录成功
        request.getSession().setAttribute("session_user", user);
        //将用户信息放入 session 用于后续的拦截器
    }
}
```

```
        return "登录成功";  
    }  
    return "登录失败，用户名或密码错误";  
}
```

4.4.2 人脸识别核心代码

人脸检测和识别的核心代码如下所示：

```
dets = dlib.rectangle(0,w,0,h)  
    dist = []  
    shape = self.sp(img_, dets)  
    face_descriptor_ = self.facerec.compute_face_descriptor(img_,  
shape)  
    d_test = numpy.array(face_descriptor_)  
    #print(d_test)  
    for i in self.descriptors:  
        dist_ = numpy.linalg.norm(i - d_test)  
        dist.append(dist_)  
    print(dist) //dist 匹配精度  
    index = np.argmax(np.array(dist))  
    return self.candidate[index]
```

5 测试

5.1 登陆测试

管理端通过测试是否可以登陆和注册，从用户角度出发，通过输入和观察各种输出的结果来发现系统中出现的各种问题。登录过程中，保证账号的唯一性是登陆的一个比较重要的因素，不能保证唯一就会导致难以确保安全性。其中登陆测试概要表如表 5.1 所示：

表 5.1 登录测试概要表

| 测试模块 | 测试内容 |
|------|--|
| 登陆模块 | 绑定本地账号的唯一性，验证码是否匹配；登录时验证账号密码是否正确，登录成功能否跳转至相应界面页面 |
| 注册模块 | 绑定本地账号的唯一性，验证码是否匹配；登录时验证账号密码是否正确，注册成功能否跳转至相应界面页面 |

登陆测试的测试效果图如图 5.1 所示：



图 5.1 登陆测试效果图

注册时的效果图如图 5.2 所示：

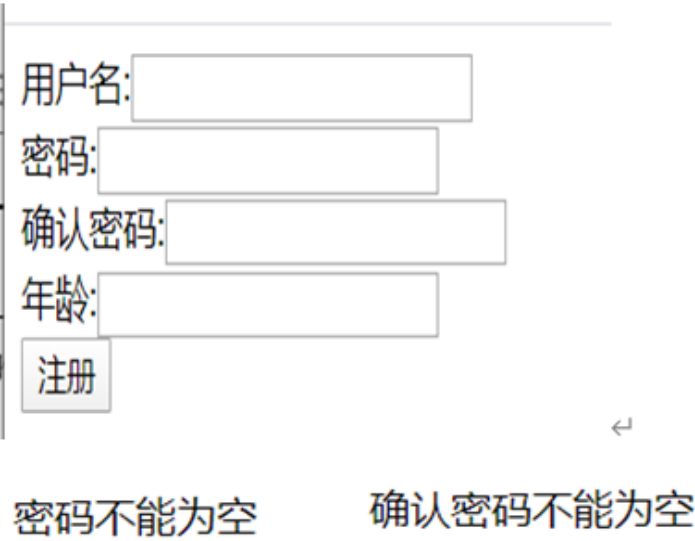


图 5.2 注册测试效果图

5. 2 人脸检测测试

人脸检测模块测试用例表如表 5.2 所示：

表 5.2 人脸检测用例表

| 测试模块 | 测试内容 |
|------|-----------------------------------|
| 人脸检测 | 能不能正常开启摄像头，能不能捕捉到人脸头像，检测人脸的范围是否正常 |
| 人脸识别 | 识别图片是否正常 |

其中人脸检测时的测试结果图可以分为有人脸是时的检测和没有人脸时的检测具体检测如下所示：

如图 5.3 为人脸检测时的测试图片，在有正脸时人脸是可以被检测到和输出并存入到本地文件夹然后供给人脸识别的，但是在没有人脸时，是不会检测到任何数据，当然也不会保存数据，这个就不会有后面的识别，节约内存空间和运行时的成本，保证运行的准确，和精确进行。



图 5.3 人脸检测侧视图

如图是在有遮挡物时运行这个检测系统并不会捕获到人脸图像。

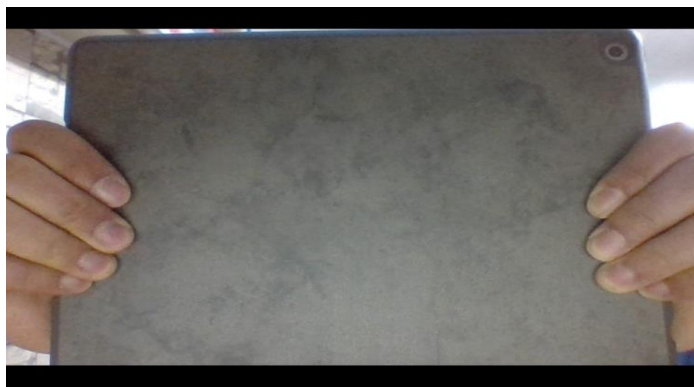


图 5.4 非人脸检测效果图

人脸识别模块测试用例表如表 5.3 所示：

5.3 人脸识别测试

表 5.3 人脸识别用例表

| 测试模块 | 测试内容 |
|------|--------------------------------|
| 人脸检测 | 检测图片是否正常 |
| 人脸识别 | 存放的正脸图片是否合格，匹配精度是否达标，人脸数据是不是完善 |

在人脸测试时 由于如环境的嘈杂，光照的强弱，和人脸与采集设备的距离等都影响着人脸识别得到精度导致了有时的测试会导致失败，即识别成库中的其他的人脸的姓名，这主要是程序原因，在人脸特征的匹配度中由于人脸的匹配数值的影响，输出了在错误情况下的错误姓名。

人脸识别时的测试图片如图 5.5、5.6 所示

其中图 5.5 是在脸部情况不一样时的检测出错的情况。

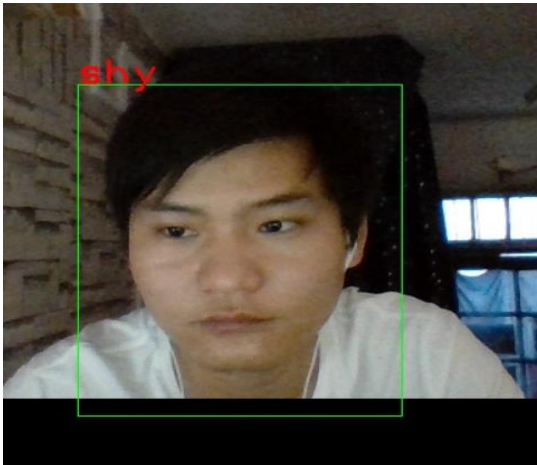


图 5.5 识别错误图



图 5.6 识别正确图

最终的结果成果展示图如图 5.7 所示：



图 5.7 成果展示图

放入手机图片依然可以检测 and 识别人脸的。加入了可以同时检测识别两张人脸的功能，测试结果图 5.8 所示：

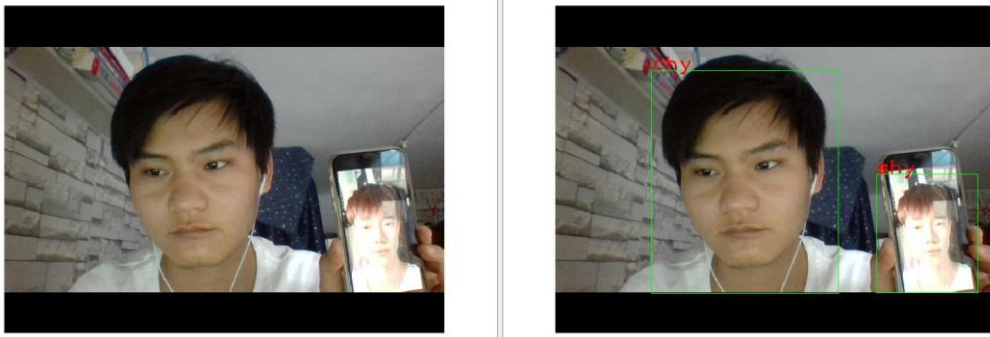


图 5.8 两张人脸展示图

6 总结

从这次的毕设中我学到了很多的东西，首先我从一个从来没有接触过深度学习的到了解并且会使用一点深度学习知识，然后巩固了自己对于编程的理解，以前的我对于编程没有很深的认识，总以为网上的代码拿来就可以用。但是在导师的教导下我知道，只有自己学着敲的代码才会属于自己，照搬网上的对自己的编程是没有什么帮助的。然后学会了怎么坚持的去做一件事，虽然这个课题对我来说比较难，但是只要不断地查和问，从零开始也不是什么难事。

毕设是对大学四年学习知识的总结，作为大学最后一份作业。在没有做毕设之前我以为知识对自己知识的总结，但是拿到课题后返现并不是，需要自己查阅学习很多的新知识。虽然完成的不是很完善，但是我也是用了很大的心思在上面，是对自己大学学习交上的一份答卷。

参 考 文 献

- [1]李佳南. 基于 ARM 的 IC 卡机房管理终端设计[D]. 四川:成都理工大学, 2009.
- [2] 马国浩, 石睿. 应用深度学习的智能门禁系统设计[J]. 福建电脑, 2019, 35(11):1-4.
- [3]石晨阳. 3D 人脸识别技术专利分析[J]. 工业技术创新, 2018, 5(5):4-7.
- [4]江白华. 基于深度学习的人脸识别研究[D]. 安徽:安徽理工大学, 2019.
- [5]田拯. 基于人脸识别的门禁系统设计与实现[D]. 上海交通大学, 2016.
- [6]黄鑫. 我国两项突破性技术获世界认可[J];广西质量监督导报;2018 年 11 期
- [7]玉士蒙. 基于云平台的智能门锁开发[D]. 北京建筑大学, 2018.
- [8]赵英男, 何祥健, 陈北京, 等. 增强 KMSE 及人脸识别应用[J]. 东南大学学报(英文版), 2016, 32(1):35-38. DOI:10.3969/j.issn.1003-7985.2016.01.007.
- [9]蒋罗. 三维人脸重建与人脸识别[D]. 安徽:中国科学技术大学, 2019.
- [10]王敏峻. 基于深度学习的人脸表情识别[D]. 内蒙古:内蒙古大学, 2019.
- [11]蒋博伟. 基于 ARM 的人脸识别系统的研究[D]. 安徽:安徽理工大学, 2019.
- [12]唐风高. 基于深度学习的人脸识别方法研究[D]. 江苏:江苏科技大学, 2019.
- [13]单聪. 非受限环境的视频人脸识别算法研究[D]. 河北:河北农业大学, 2019.
- [14]郭佳鹏. 基于卷积神经网络的人脸识别研究[D]. 辽宁:大连理工大学, 2019.
- [15]郑君. 基于人脸识别门禁控制系统的研究与实现[D]. 江苏:扬州大学, 2019.
- [16]Selen Işık Ulusoy, Şeref Abdurrahman Gülseren, Nermin Özkan, Cüneyt Bilen. Facial emotion recognition deficits in patients with bipolar disorder and their healthy parents[J]. General Hospital Psychiatry, 2020, 65.
- [17]Gargi Mishra, Virendra P. Vishwakarma, Apoorva Aggarwal. Constrained L₁-optimal sparse representation technique for face recognition[J]. Optics and Laser Technology, 2020, 129.
- [18]Muhammad Sajjad, Mansoor Nasir, Khan Muhammad, Siraj Khan, Zahoor Jan, Arun Kumar Sangaiah, Mohamed Elhoseny, Sung Wook Baik. Raspberry Pi assisted face recognition framework for enhanced law-enforcement services in smart cities[J]. Future Generation Computer

Systems, 2020, 108.

- [19]Nesrine Grati,Achraf Ben-Hamadou,Mohamed Hammami. Learning local representations for scalable RGB-D face recognition[J]. Expert Systems With Applications, 2020, 150.
- [20]Jian-Xun Mi,Yueru Sun,Jia Lu,Heng Kong. Robust supervised sparse representation for face recognition[J]. Cognitive Systems Research, 2020, 62.

致 谢

三个月时间匆忙而过，回首这三个月，我收获颇丰，期间得到了很多人的帮助、鼓励和支持，在此向他们表达我最诚挚的谢意。

首先，要感谢的是我的指导老师秦品乐，秦老师每周召开小组例会，督促我们努力完成毕业设计，同时在遇到问题时秦老师都能及时的给我们分析原因，寻求解决之道。正是由于秦老师的缜密的安排，才使我得以顺利完成毕设。在毕设期间，秦老师不厌其烦的指导我们，他教会我们的不仅是知识，更是做人的道理，勤奋刻苦，致知于行。

当然还要感谢毕业设计答辩组的各位老师，正是因为他们提出的建议和意见，才使我毕设项目能够更加完善的完成，非常感谢评审组各位指导老师对我的悉心教导。最后，还要感谢在寝室一起努力完成毕设的室友们，正是因为他们的鼓励和帮助，在遇到难题时帮我一起查阅资料解决，才使我在毕设期间坚持下来，非常感谢你们。