|  |  |
| --- | --- |
| **密 级** | **公开** |
| **学 号** | **2019310191** |



**毕 业 设 计（论 文）**

|  |
| --- |
| **基于OpenCV与AI视觉的** |
| **人脸识别算法研究** |

|  |  |
| --- | --- |
| **学 院：** | **信息工程学院** |
| **姓 名：** | **张一川** |
| **年 级：** | **2019级** |
| **专 业：** | **计算机科学与技术** |
| **指导教师：** | **余有明** |
| **教师职称：** | **副教授** |

2023 年 06 月 04 日·北京

**北 京 石 油 化 工 学 院**

**毕 业 设 计 （论 文） 任 务 书**

学院（系、部） 信息工程学院 专业计算机科学与技术班级计192

学生姓名 张一川 指导教师/职称 余有明/副教授

1.毕业设计（论文）题目

**基于OpenCV与AI视觉的人脸识别算法研究**

2.任务起止日期： 2022 年 12 月 26 日 至 2023 年6月 23日

3.毕业设计（论文）的主要内容与要求

（含课题简介、任务与要求、预期培养目标、原始数据及应提交的成果）

1. 课题简介

本课题来源于AI视觉的人脸识别算法研究的需求。可以采用Jetson Nano为主控，OpenCV为图像处理库，Jupyter Lab为开发工具，Python为编程语言。通过大量正类图像（人脸图像）和负类图像（不带人脸的图像）训练Haar级联分类器进行人脸检测，实现对生活实践中人脸检测与人脸识别的不同需要。

1. 任务与要求
2. 调查分析国内外人脸识别算法或技术的优劣，写出开题报告；
3. 开展相关数据库设计、算法设计和实现，形成设计文档、程序和数据库；
4. 完成算法测试与改进；
5. 撰写毕业论文。
6. 预期培养目标
7. 在数据库设计、软件开发、算法研究等方面培养综合实践能力；
8. 培养学生的调研分析能力；
9. 培养学生的学术论文写作水平。
10. 原始数据及应提交的成果
11. 以参考文献中的典型应用做参考；
12. 应提交开题报告、外文翻译、软件原型、源程序、数据库和毕业论文。

4.主要参考文献

[1]沈蕴梅.基于百度AI人脸识别技术的智能照片搜索系统[J].现代信息科技,2022, 6(21):17-20.DOI:10.19850/j.cnki.2096-4706.2022.21.004.

[2]马宇轩,潘纬航,廖腾烈,全爔宇,李咏楠.基于百度AI人脸识别的机场危险人员监控[J].现代信息科技,2022,6(01):25-27+31.DOI:10.19850/j.cnki.2096-4706. 2022.01.007.

[3]张静.AI人脸识别技术的改进及其在校园的应用[J].微型电脑应用,2021,37(04): 73-76.

[4]宋永生.基于OpenCV及AI的网络学习者情绪监测系统[J].信息技术与信息化, 2019(11):9-12.

[5]秦鸿,李泰峰,郭亨艺,许毅.人脸识别技术在图书馆的应用研究[J].大学图书馆学报,2018,36(06):49-54.DOI:10.16603/j.issn1002-1027.2018.06.008.

5.进度计划及指导安排

|  |  |  |  |
| --- | --- | --- | --- |
| 周次 | 起止日期 | 工作内容 | 具体要求 |
| 18 | 2022.12.26-30 | 熟悉题目；文献调研 | 按要求查阅中国知网和会议文献 |
| 19 | 2022.1.2-6 | 文献分析；调研企业产品和软件文档 | 查阅文献，调研同类商品软件 |
| 寒假  第1周 | 2023.1.9-13 | 启动岗位实习课程任务；  分析归纳各种文献资料，写出开题报告初稿；  查阅外文文献；  毕业设计开题修改和外文翻译；  在企业开展6周的岗位实习。 | 联系企业；整理文献，写开题报告初稿 |
| 寒假  第2周 | 2023.1.16-20 | 岗位实习；汇报开题报告 |
| 寒假  第3周 | 2023.1.23-27 | 岗位实习；修改开题报告 |
| 寒假  第4周 | 2023.1.30-2.3 | 岗位实习；外文文献翻译 |
| 1 | 2023.2.20-24 | 岗位实习；外文文献翻译 |
| 2 | 2023.2.27-3.3 | 岗位实习总结；  外文文献翻译 |
| 3 | 2023.3.6-10 | 系统分析； | 系统需求分析 |
| 4 | 2023.3.13-17 | 开题报告答辩。 | 开题报告答辩 |
| 5 | 2023.3.20-24 | 完成系统分析 | 完成系统分析 |
| 6 | 2023.3.27-31 | 系统设计 | 上交岗位实习装订材料；设计系统管理流程 |
| 7 | 2023.4.3-7 | 系统设计 | 设计数据库 |
| 8 | 2023.4.10-14 | 系统设计 | 设计管理流程 |
| 9 | 2023.4.17-21 | 系统实现； | 实现功能模块； |
| 10 | 2023.4.24-28 | 系统实现；毕业设计中期检查 | 实现功能模块；中期检查 |
| 11 | 2023.5.2-5 | 系统实现 | 实现功能模块 |
| 12 | 2023.5.8-12 | 系统实现 | 实现功能模块 |
| 13 | 2023.5.15-19 | 系统实现 | 实现功能模块 |
| 14 | 2023.5.22-26 | 系统测试和完善；  撰写毕业论文 | 系统初验；撰写毕业论文 |
| 15 | 2023.5.29-6.2 | 系统验收 | 系统验收和答辩；  提交毕业论文 |
| 16 | 2023.6.5-9 | 论文答辩 | 准备论文答辩材料 |
| 17 | 2023.6.12-16 | 论文答辩 | 答辩后完善论文；  整理毕业设计资料 |
| 18 | 2023.6.19-23 | 资料归档 | 按要求归档所有资料 |

任务书审定日期 2022 年 12 月 19 日 系（教研室）主任（签字） \_\_\_

任务书批准日期 2022 年 12 月20 日 教学院（系、部）院长（签字） \_\_\_\_\_\_任务书下达日期 2022 年 12 月 20 日 指导教师（签字）\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

计划完成任务日期 2023年 6 月 23 日 学生（签字）\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# 摘 要

目前人脸识别算法的应用越来越广，刷脸的场景在生活中比比皆是：刷脸支付、安检、解锁等都是人脸识别的应用案例。虽然人脸识别技术给人们生活带来了便利，但存在的问题也不少，技术还不够完善，如识别速度慢、识别准确率有待提高等。

本文针对人脸识别存在的问题，提出了一种基于OpenCV和AI视觉技术的人脸识别算法。首先，该算法利用OpenCV和Haar特征分类器，对输入图像进行人脸检测和定位。然后，通过积分图对检测到的人脸图像进行特征提取，生成对应的特征值。最后，再将这些特征值输入Haar级联分类器进行分类，实现了对人脸快速准确的识别。

本文以PC为硬件平台，Python为设计语言，通过OpenCV技术与摄像头的AI视觉特性捕捉画面，研发了稳定的人脸识别系统。从识别阈值及人脸样本数量测试了系统可行性及实效性。实验结果表明:当识别阈值设置为50，人脸样本数据为100张时，人脸识别精准度高达94%，人脸识别时间小于2s。本文设计的人脸识别系统在人脸不同角度和不同环境光亮度的情况下均能比较准确地识别人脸信息，克服了人脸识别对环境参数要求高的缺点。对于开展复杂环境下的人脸识别研究有一定的参考价值。

**关键词**：人脸识别；OpenCV；AI视觉；Haar特征

# Abstract

At present, face recognition algorithms are being used more and more widely, and scenes of face brushing abound in life, such as face brushing payment, security check, unlocking, etc. are all application cases of face recognition. Although face recognition technology has brought convenience to people's lives, there are many problems, and the technology is not perfect, such as slow recognition speed and recognition accuracy rate to be improved.

In this paper, a face recognition algorithm based on OpenCV and AI vision technology is proposed to address the problems of face recognition. First, the algorithm uses OpenCV and Haar feature classifier to detect and localise faces on the input image. Then, feature extraction is performed on the detected face images through an integral map to generate the corresponding feature values. Finally, these feature values are then fed into the Haar cascade classifier for classification, achieving fast and accurate recognition of faces.

In this paper, a stable face recognition system is developed using PC as the hardware platform and Python as the design language, and capturing images through OpenCV technology and the AI visual characteristics of the camera. The feasibility and effectiveness of the system were tested in terms of recognition threshold and number of face samples. The experimental results show that when the recognition threshold is set to 50 and the number of face samples is 100, the accuracy of face recognition is as high as 94% and the time required for face recognition is within 1-2s. The face recognition system designed in this paper can recognise face information more accurately under different angles of the face and different environmental conditions, overcoming the shortcomings of face recognition which requires high environmental parameters.

**Key Words**：Face Recognition; OpenCV; AI Vision; Haar Features

目 录

[摘 要 I](#_Toc137972692)

[Abstract II](#_Toc137972693)

[第一章 绪 论 1](#_Toc137972694)

[1.1 研究背景 1](#_Toc137972695)

[1.2 人脸识别研究现状及意义 2](#_Toc137972696)

[1.2.1 人脸识别研究现状 2](#_Toc137972697)

[1.2.2 人脸识别技术的研究意义 3](#_Toc137972698)

[第二章 相关技术 4](#_Toc137972699)

[2.1 OpenCV 4](#_Toc137972700)

[2.2 PyCharm 4](#_Toc137972701)

[2.2 Haar-like特征 4](#_Toc137972702)

[2.2 Haar-like特征的计算——积分图 5](#_Toc137972703)

[2.3 计算Haar特征值 5](#_Toc137972704)

[2.4 Haar特征值含义 6](#_Toc137972705)

[2.5 Haar特征值归一化 7](#_Toc137972706)

[2.6 级联分类器 8](#_Toc137972707)

[2.6.1 级联分类器的原理 8](#_Toc137972708)

[2.6.2 级联分类器的原理 9](#_Toc137972709)

[2.6.3 级联分类器的检测 12](#_Toc137972710)

[第三章 人脸识别实验需求分析 14](#_Toc137972711)

[3.1 实验环境 14](#_Toc137972712)

[3.2 性能要求 14](#_Toc137972713)

[3.3 具体功能 14](#_Toc137972714)

[第四章 人脸识别功能设计 15](#_Toc137972715)

[4.1 总体设计 15](#_Toc137972716)

[4.2 详细设计 17](#_Toc137972717)

[第五章 人脸识别的实现及其测试 24](#_Toc137972718)

[5.1人脸识别的实现 24](#_Toc137972719)

[5.2 评价指标对人脸识别影响测试 28](#_Toc137972720)

[第六章 结论与展望 30](#_Toc137972721)

[参考文献 31](#_Toc137972722)

[致 谢 33](#_Toc137972723)

# 第一章 绪 论

## 1.1 研究背景

在日常生活中，我们与大自然的信息交互方式多种多样。如果把获取信息的方式进行比较，则视觉获取的信息比例占60%左右，听觉占20%，其它获取方式加和占20%左右[1]。由此可知，视觉是我们获取信息的主要来源和途径。通过眼睛捕捉到的他人面部图像里包含了人们的多数基础信息。

每个人都有不一样的身体特征，如面部、虹膜、指纹等，这些作为分辨个人身份的特征，应用于生活的方方面面。如存、取钱、车票无人销售、考勤认证等。其中虹膜识别技术更为精准，但会受到设备难以普及的限制；而指纹识别技术会因为手指受伤或职业原因导致无法获取指纹。现如今生活质量的提高、经济的腾飞、科技的进步，以往的生物特征识别技术现在无法满足人们对安全的需求。相比如今市场上其他的生物特征识别技术，人脸识别技术具有非直接接触性、隐私性、隐蔽性等多种优势，用于确保用户的安全性和唯一性。五官之间的间距、五官的大小以及形状、面部状态以及面部肌肉收放动作，这些都是面部特征的主要识别依据。

实现人脸识别，首先进行人脸图像信息的采集或录入，然后把人脸图像信息转为程序可以识别的数组，再将数组组合形成独一无二的人脸信息，最后将人脸信息存储到数据库当中。对数据库充分加密便可保障人们的信息安全，避免个人生物信息的泄露。随着电子证件已经逐渐代替了传统的实物证件，以及指纹痕迹存在被用来复制指纹的可能性；虹膜识别设备难以在民间普及；声波识别在采集声纹特征时容易受环境影响等问题，导致这些识别技术难以继续发展。因此在种种的识别技术当中，人脸识别技术是极其重要且实用性强的一个。

## 1.2 人脸识别研究现状及意义

### 1.2.1 人脸识别研究现状

人脸识别技术起源于十九世纪末Galton的关于人脸识别的文献。因为当时的专业技术以及设备的限制，人脸识别技术的发展特别缓慢。但是随着人们的生活需求提高，相关科研机构和学者意识到了人脸识别技术的前景和重要性，人脸识别技术自然而然地得到了进步。Bledose等学者以面部器官之间的间距和特征性比例等数值作为人脸特征在全世界最先实现了半自动人脸识别技术。模板匹配法和集合结构特征法在当时得到了大多数人的认可。模板匹配法第一步是要大量的人脸图像存储在一个数据库中，将这些人脸图像作为样本，把要识别的人脸和数据库中的人脸图像进行相似度比较。这个方法复杂度高，识别准确率低且不稳，而且还会因为人脸的动作以及被遮挡的程度导致识别成功率降低。

20世纪后半叶，伴随着计算机科学的进步，学术研究人员相继提出各种算法，如梯度特征、Haar-Like、Gabor、SIFT等基于深度学习以及数学几何的人脸特征。

最初的人脸识别技术多半依靠特征检测的传统技术，例如:特征脸、统计学理论、几何特征等方法。当深度学习技术越来越成熟，便出现了以卷积层为技术基础的人脸识别深度学习算法。计算机语言在当前以Python为主流，因为Python支持多数的计算机学习库。

OpenCV的图片算法是当下人脸识别技术的根本，其主要经过以下步骤:边缘平滑处理、图片光线补偿、彩色转灰度化操作。摄像头的采集过程中经常会出现光线不充足的情况，这会对面部特征点的识别造成负面影响。因此就要对待检测的人脸图片进行光线补偿操作。计算机处理人脸图像的过程中会受到各种因素的影响，使得人脸图片的照片质量下降，所以要进行人脸图像平滑处理。彩色的人脸图像相比灰度图容量大且不宜处理，因此人脸图像要经过灰度化处理。

人脸识别技术的飞速提升是依靠于计算机深度学习的出现。计算机学习方法中包括了深度学习算法，这是一种使用深度神经网络实现机器学习为主的一种算法。数以千计的人脸训练数据，经过计算机学习最终实现人脸识别技术。

张晓兵提出了多尺度分层特征融合的算法[11]，解决了LBP算子不能完全表达人脸信息的问题，识别效果明显提升。以人脸识别为出发点，引入了TensorFlow、Kubernetes等容器技术，实现了人脸识别程序在计算机上同时进行部署和任务编排。虽然人脸识别的技术研究在我国发展起步的时间相较于其他发达国家相对较晚，但是其的发展十分迅速，主要集中在中国的各个高校以及研究机构。

### 1.2.2 人脸识别技术的研究意义

随着电子产品的飞速发展，电子证件和线上身份验证的使用频率也水涨船高。在随着AI技术的进步，人脸识别将会成为这些身份识别功能的根本，只有人脸识别更加准确，这些方面才能有更快地发展以及创新。发达国家的人脸识别技术一直在稳步前进，在全世界都在想攻破人脸识别技术的大环境下，我国对人脸识别领域的积极研究使得人脸识别系统进入快速发展的阶段。正是因为人脸识别具有其他生物特征识别不具备的优势如:隐蔽性、非侵犯性等多种长处，所以这对国家信息的保护以及对国家的安全起着很重要的作用。人脸识别的一些应用，如图1-1所示。



**图1-1 人脸识别的一些应用**

# 第二章 相关技术

## 2.1 OpenCV

[OpenCV](http://www.eepw.com.cn/news/listbylabel/label/OpenCV)是一个基于BSD开源发行的跨平台[计算机视觉](http://www.eepw.com.cn/news/listbylabel/label/%E8%AE%A1%E7%AE%97%E6%9C%BA%E8%A7%86%E8%A7%89)和机器学习软件库，可以用其对数字图像进行一系列处理工作，从而进一步开发实时的图像处理、计算机视觉以及模式识别程序，比如物品识别，人脸识别等机器视觉领域。[OpenCV](http://www.eepw.com.cn/news/listbylabel/label/OpenCV)可以运行在Linux、Windows、Android和Mac OS操作系统上。轻量级是[OpenCV](http://www.eepw.com.cn/news/listbylabel/label/OpenCV)的一个特点，而且也是开源免费的计算机视觉处理库，框架语言采用C语言和C++语言，拥有着Python、Java、MATLAB、C++等多种语言接口，现有的操作系统大多都支持其运行和调试，这些优势帮助[OpenCV](http://www.eepw.com.cn/news/listbylabel/label/OpenCV)实现了对于电脑图像和视觉处理等方面的通用算法。

## 2.2 PyCharm

PyCharm是一种PythonIDE，其工具自带多种可以帮助用户在使用Python语言开发时提高其编码工作效率的工具，比如，Tkinter(可视化工具)、自动完成、CV(视频输入模组)、Numpy(数学基础库)、Os(os模块)、Image(图像处理库)、智能提示等等。

## 2.2 Haar-like特征

Haar特征分类为线性特征、边缘特征、中心特征和对角线特征，这些特征模板由黑色和白色矩阵组成。模板的特征值被定义为白色矩形像素之和减去黑色矩形像素之和，目的是通过特征值反映灰度变化。这些灰度的变化可以体现出面部不同部位特定走向（对角走向、水平走向、垂直走向）的结构。

C:\Users\竖臂摘星\AppData\Local\Temp\WeChat Files\4abfb39076e61f329ec412e821b3e03.pngC:\Users\竖臂摘星\AppData\Local\Temp\WeChat Files\031041d2532818c6d20d139b19738ba.pngC:\Users\竖臂摘星\AppData\Local\Temp\WeChat Files\89a0ca970762f726f1a494b6c880795.pngC:\Users\竖臂摘星\AppData\Local\Temp\WeChat Files\b1a6d41170a9c7e34f0bbee0979ae63.png

**A B C D**

上图A，B，C的特征数值V=Σ白-Σ黑，图D的特征数值V=Σ白-2\*Σ黑。在计算图D特征数值时相比计算图A，B，C特征数值将黑色区域的像素和乘2，以达到在两种不同的矩形区域中的像素数目时相同的。区分人脸与非人脸，需要将矩形放在待测区域计算出特征数值，人脸区域与非人脸区域特称数值相差越大越有利于区分人脸与非人脸。

## 2.2 Haar-like特征的计算方法——积分图

主要方法是将图像中矩形区域的像素之和保存为一个数组元素，并将其存储在内存中。在计算矩形区域的像素和时，可以直接调用数组中的元素值，省去了再次计算像素和的步骤。积分公式:ii(i,j)=Σk≤i,l≤jf(k,l)ii(i,j)=Σk≤i,l≤jf(k,l)

## 2.3 计算Haar特征值

通过区域端点的积分图来计算像素值之和。如图2-1所示，矩形特征的特征值=A区域的像素值之和-B区域的像素值之和

*Y*

(1,1)

4

5

6

3

2

1

*X*

A

B

**图2-1 特征值计算样例图**

A区域的像素值:ii(5)+ii(1)-ii(2)-ii(4)

B区域的像素值:ii(6)+ii(2)-ii(5)-ii(3)

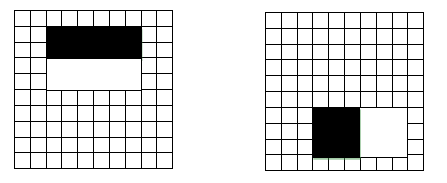
该区域的像素值:ii(5)+ii(1)-ii(2)-ii(4)-[ii(6)+ii(2)-ii(5)-ii(3)]=

[ii(5)-ii(4)]+[ii(3)-ii(2)]-[ii(2)-ii(1)]-[ii(6)-ii(5)]

由于特征矩形端点的积分图会影响矩形特征的特征值，而图像坐标不会影响矩形特征的特征值，因此计算出特征矩形端点的积分图后，只需进行加减运算即可得到矩形特征的特征值。这种方法显著提高了特征计算速度和检测目标速度。

## 2.4 Haar特征值含义

选取麻省理工人脸数据库中10×10大小的人脸样本图像2700张，分别记录如下图2-2与图2-3的Haar特征值。



**图2-2 人脸区域** **图2-3 无具体意义区域**

人脸区域特征

3000

2500

2000

1500

1000

500

0

-4000

-2000

0

2000

4000

6000

8000

无具体意义区域特征

特征值

样本

**图2-4 特征值分布图**

如上图2-4可知，2种不同Haar特征在相同一组样本中的特征值分布不同。人脸区域特征的特征值对样本的区分度高，绝大部分都大于0。无具体意义区域特征的特征值对样本的区分度低，均匀分布在0的两侧。由此可得，Haar特征分类的效果不同是由于Haar特征值的分布不均匀作造成的[18]。特征分类的效果越好正负样本区的分度越大，上图3-4中人脸区域特征的Haar特征分类明显优于无具体意义区域特征的Haar特征分类。

## 2.5 Haar特征值归一化

大小为128的Haar特征，其特征值变化范围大约在-2000到6000之间，其跨度较大。跨度很大不利于量化评定其特征值。为了减少Haar特征计算得出特征值的变化跨度，使其更易于量化评定，需要进行“归一化”。设:当前检测窗口中图像像素为i(x,y)，当前检测窗口为w×hw×h(如上图2-2和图2-3中大小为10×10的检测窗口)，OpenCV所采用的“归一化”如下:

平均值:mean=

平均值的平方和:=

灰度值:sum=∑i(x,y)

灰度值平方和:=∑i2(x,y)

计算归一化因子:varNormFactor=

归一化特征值:normValue=

计算出结果后将归一化的特征值(normValue)与阈值进行比对。

## 2.6 级联分类器

基于Adaboost算法，对于常规的提升算法所涉及到以下3个问题:

（1）如何具体描述Adaboost算法？

（2）对于同一训练数据集，如何改变其样本分布以达到重复训练的目的？

（3）强分类器阈值的设定与弱分类器数量如何有机结合寻找平衡点？

### 2.6.1 级联分类器的原理

级联分类器的原理是将数据分为多个层级，每个层级都有一个对应的分类器对数据进行正负分类，判定为正类的数据进入下一个层级进行再分类，否则判定为负类。级联分类器的树状结构如下图2-5所示:

······

拒绝

拒绝

第n层级

拒绝

第1层级

第2层级

通过

弱分类器1

弱分类器2

······

**图2-5 级联分类器模型图**

强分类器在对负样本进行判定时，具有非常高的准确性,只有检测窗口通过所有的强分类器时才能认定为正样本。一旦目标检测为负样本，就停止调取下层级的强分类器，大大减少了检测时间。级联分类器在检测图像的前几层级就筛选出了负样本，级联分类器的检测速度快。只有通过前一层级的强分类器检测出的正样本才会被传递到下一级进行检测，这种策略可以有效地避免伪正样本的出现，保证输出的正样本的准确性。(不排除如下图2-6检测对象佩戴针对人脸识别口罩等特殊情况)

**图2-6 骗过人脸识别系统的特殊口罩**

### 2.6.2 级联分类器的原理

首先训练出每个弱分类器，然后将弱分类器组成一个强分类器，接着把训练出的多个强分类器根据级联的方式组合，最终得到Haar级联分类器。

如下图2-7，是一个弱分类器的决策树。

非人脸

人脸

与特征

f3比较

非人脸

与特征

f1比较

与特征

f2比较

非人脸

待测图片的特征值

输入数据

决策树

符合

符合

符合

不符合

不符合

不符合

**图2-7 弱分类器决策树图**

在处理大量特征时，若使用AdaBoost直接训练，则工作量大效率低。因此挑选出若干出色的弱分类器(特征值)，再由AdaBoost对这些弱分类器进行训练。

在确认训练子窗口中矩形特征数量和特征值之后，需对每一个特征，训练一个弱分类器。设:h为弱分类器，f为特征，x为检训练子窗口，p为不等号方向，θ为阈值。

这样就可以确认特征的最优阈值，使得这个弱分类器对所有训练样本的分类误差最低。

将训练样本灰度化和归一化为规定尺寸后，训练弱分类器的具体步骤如下:

（1）对于每个特征f，计算所有训练样本的特征值。

（2）将所有训练样本的特征值进行排序。

（3）遍历排序后的特征值，计算每个元素的以下4个值:

1）该元素之前正例的权重和

2）该元素之前负例的权重和

3）正例的权重和

4）负例的权重和

（4）选取当前元素的特征值与其之前的一个特征值之间的数，将这个数作为阈值[21]。在当前选取的元素处，用所得到的弱分类器把训练样本分开。将得到当前元素处于阈值之前的部分记为人脸，当前元素处于阈值之后的部分记为非人脸。由于分类存在误差，导致记为人脸的元素部分存在非人脸元素，记为非人脸的元素部分存在人脸。阈值的分类误差为:e=min(+(-),+(-

遍历按特征所排序特征值的表,为弱分类器选择令分类误差最小的最优阈值，如下图2-8所示。

·

·

·

·

·

·

图像的特征值

(选定元素)

图像的特征值

图像的特征值

图像的特征值

图像的特征值

按

照

特

征

的

特

征

值

排

序

**图2-8 训练弱分类器的算法图**

训练强分类器的具体步骤(即Adaboost算法描述)如下:

（1）设:训练样本集(,)i=1,2,3,…N；当=1时为正样本，正样本的数量为；当=0时为负样本，负样本的数量为；训练的最大循环次数为T。

（2）初始化样本的权重(概率分布):

（3）训练循环:for t=1,2,3…T:

1）权重归一化，=

2）对每个特征j，训练对应的弱分类器，对每个分类器使用一种Haar特征进行训练，分类误差为:=|()-|

3）从训练完成的弱分类器中，找到一个分类误差最小的弱分类器。

4）更新每个样本所对应的权重:=

5）若样本的分类正确，那么=0,否则=1，而=

（4）最终强分类器组成为:其中:=

通过将所有弱分类器组合来构建强分类器，来检测输入图像。分类器的结果是通过将弱分类器加权和和平均数进行比较得到的[22]。

### 2.6.3 级联分类器的检测

单独的强分类器往往无法在精准的完成复杂的分类问题的同时，达到高检测速率和低误检率的要求。这时就需要由级联的强分类器来完成复杂的分类问题。在级联分类器时，将所级联的分类器由简单到复杂排列，降低每层级分类器的误检率要求，提高其检测速率。

对于第一层级的强分类器，其训练数据集为全部的原始训练样本。由于要求的误检率较低，下一层级的强分类器可以针对上一层级的强分类器难以区分且容易误检的样本进行训练。这种针对训练会产生更多的特征以及弱分类器，所以级联分类器时每层强分类器的复杂程度都是逐层递加的。

图像样本进行检测时，需要进行多区域和多尺度的检测。由于训练样本都已经灰度化和归一化为规定尺寸，所以在检测比规定尺寸更大的图像时需进行多尺度检测。第一种多尺度测量方法为固定取样的子窗口大小不变，通过缩放图片来实现检测。但图像的缩放和特征值的二次计算导致效率不佳；第二种多尺度测量方法为图像的大小不变，通过不断放大取样的子窗口来实现检测。

使用多区域和多尺度任意一种方法时，均会从子窗口中获得待测样本的取样图像，这些取样图像通过级联分类器逐级的检测。检测判定为负的区域会被排除，只有检测判定为非负的区域才能进入下一层级进行下一轮检测。级联分类器的检测过程如下图2-9所示:

子窗口样本图像

强分类器1

强分类器2

·

·

·

强分类器n

非目标区域

非目标区域

非目标区域

目标区域

**图2-9 级联分类器检测过程图**

增加弱分类器的数量可以降低强分类器的误识率，但引起计算时间的增加会导致检测速率降低。降低强分类器的阈值可以减少计算所需时间来提高检测速率，但会导致误识率的增高。在构造分类器时需在两者之间找到平衡点。

第三章 人脸识别实验需求分析

## 3.1 实验环境

CPU:500M以上；内存:8G，安装Windows 98、Windows Me、Windows 2000等操作系统中的一种。装有摄像头进行拍照，安装Python-3.6.1、PyCharm-2021.2.3、OpenCV-3.4.1、Designer5.14.3等软件，运行环境表，如表3-1所示。

**表3-1 运行环境表**

|  |  |
| --- | --- |
| 硬件 | 软件环境 |
| 处理器:inter酷睿i5 | 操作系统:Windows10 |
| 内存:8G | 运行环境:Python-3.6.1 |
| 显卡:GTX1050Ti 4GB | 运行环境:PyCharm-2021.2.3 |
| 运行环境:OpenCV-3.4.1 |

## 3.2 性能要求

人脸识别程序需要在短时间内通过算法对比数据库内容，进行人脸识别操作。超过2秒就会失去AI视觉特性捕捉画面的实效性，未来希望可以优化储存算法并在同一镜头内识别多个人脸信息。

## 3.3 具体功能

人脸识别程序应该具有的功能:对第一次使用者进行人脸的录入和储存，在视频流当中保存某一帧图片，将照片信息通过OpenCV算法转化为数组信息，并且储存在特定的数据库或者文件当中。在使用者第二次使用该程序时会进行人脸识别和检测，实时的比对摄像头AI视觉特性捕捉画面中的人物，是否是已录入信息的人员。如果是，将录入的信息实时显示出来；如果不是，则显示“unknow”并发出警报。

第四章 人脸识别功能设计

## 4.1 总体设计

实验程序是由Python作为框架，OpenCV作为调用技术，分别实现检测系统、录入系统、信息训练系统，在PyCharm中进行代码的书写、调试、实验以及设计。人脸识别系统的总体功能流程，如图4-1所示。

人脸识别程序的总体设计关键点如下:

（1）下载并且安装配置Python、PyCharm、OpenCV、Designer软件，更改组件下载地址为清华地址，以防下载速度过慢；

（2）设计人脸信息录入系统的编程，实现人脸照片拍摄及人脸信息录入等功能。设计数据训练系统的编程，将收录的人脸照片通过OpenCV算法转化为数组，并且存入到对应的文件或者数据库中。设计解决人脸识别程序，通过人脸特征点和欧式距离计算等方式，输出对应的人员信息；

（3）设计人脸图片灰度转换的代码，将彩色或者拍摄的人物照转换为不同灰度的照片；

（4）照片的尺寸修改，将保存的不同大小尺寸的图片都修改为统一大小的；

（5）实现绘制矩形的代码，可以在人脸图像上绘制合适大小的矩形或者圆形，设计人脸检测的代码，用于检测图片中的人脸位置；

（6）在实现人脸检测的基础上进行设计人脸识别的代码，调用OpenCV实现人脸识别以及特征比对模块；

（7）实现调用摄像头程序，可以调用程序自带摄像头或者是外置摄像头；

（8）在PyCharm中用PYQT5组件创建程序主要窗体，及其按键以及样式美化，将每个功能编写到函数当中，最终实现整体功能；

（9）获取计算机时间，并且编写日志代码方便查看每一步的操作，并显示在程序合适的窗体中，编写使设备发声代码，当显示陌生人次数太多时触发。

**图4-1 应用程序整体流程图**

## 4.2 详细设计

人脸识别程序是由Python作为搭建语言编写程序，OpenCV技术作为图像处理以及人脸比对的核心技术[23]，PyCharm作为载体进行调用组件并执行程序，下文说明了程序的三个组成部分:人物身份录入、人物信息管理、人脸识别。

进行所需环境安装，访问Python官网点击下载Python-3.6.1安装包并且安装，进入PyCharm官网单击Downloads下载并安装，进行OpenCV的安装，采用的是PIP安装，进入命令窗口输入pip install opencv-python进行安装，最后安装OpenCV的源码进入官网单击libraty下载完成进行解压即可，最后进行软件Designer5.14.3的安装以及破解即可，在Python中调用OpenCV、Tensor、Flowtkinter、PyQt5等需要用到的服务组件，到此环境搭建完成。

编写调用摄像头拍照保存的代码，通过调用视频模块，调用系统摄像头，如需使用外置摄像头将参数0改为1或-1即可，设置视频在前端展示的时间设计帧率等参数，以及照片的命名方式该步骤也是为了人脸信息录入做准备，关闭摄像头并且释放内存空间，视频采集流程图具体流程,如图4-2所示。

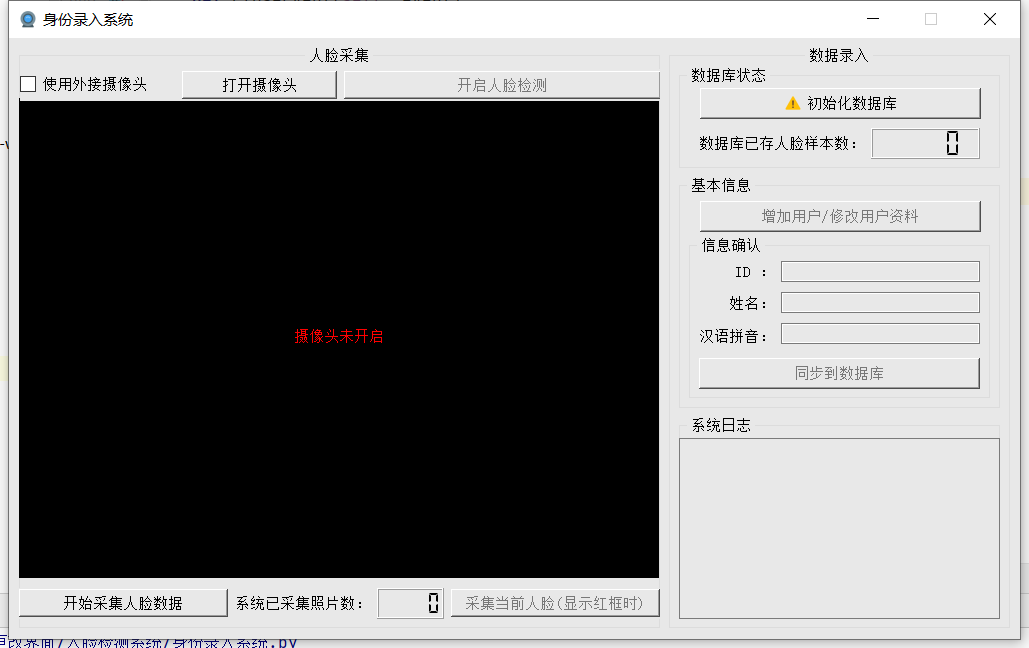


**图4-2 视频采集流程图**

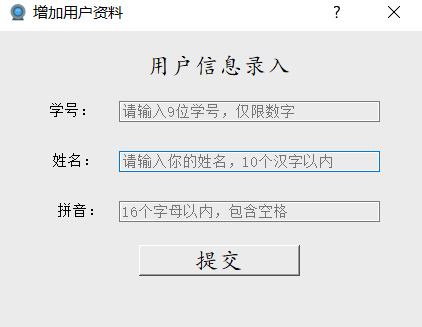
身份录入系统前端展示图，如图4-3所示，当进入身份录入系统后在会显示人脸数据库中已经拍照保存的样本数量。当打开系统摄像头开启人脸检测后，系统会使用Harr的算法找出人脸，画面会用方框框出人脸所在的位置，当点击添加用户/修改用户资料可以录入用户信息信息录入前端展示图，如图4-4所示，最后在右下角显示身份录入系统日志。身份录入的流程图，如图4-5所示，编写身份录入系统，主要通过Python实现程序逻辑，使用PyQt5组件编辑书写，其程序主要函数及其功能如信息录入系统函数表，如表4-1所示。

**表4-1 信息录入系统函数表**

|  |  |
| --- | --- |
| 主要函数 | 功能 |
| DataRecordUI(QWidget) | 该函数通过PyQt5组件编辑身份录入可视化界面，进行窗口布局 |
| UserInfoDialog(QDialog) | 该函数通过PyQt5组件编辑信息录入可视化界面，进行窗口布局 |
| self.isFaceDetectEnabled | 该函数通过OpenCV算法检测人脸的位置 |
| def enableFaceRecord(self) | 该函数通过指令进行拍照保存人脸图像 |
| self.cap.set(cv2.CAP\_PROP \_FRAME\_WIDTH，640) | 该函数通过指令进行调用开启或关闭摄像头 |
| def addOrUpdateUserInfo(self) | 该函数通过指令进行增加/修改用户信息 |
| def migrateToDb(self) | 该函数通过指令将人脸信息及照片上传到数据库 |



**图4-3 身份录入系统前端展示图**



**图4-4 信息录入前端展示图**

**图4-5 身份录入系统流程图**

使用者在采集人脸照片的时候会填写人员信息，这些人脸信息会同步到数据库，在人脸识别时将用到，信息管理系统会对人脸信息进行比对，比如学号要为9位数组，用户姓名必须为汉字，姓名拼音必须为英文字符，人脸信息数据结构表，如表4-2所示。

**表4-2 人脸信息数据结构表**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段标识含义 | 字段类型 | 字段大小 |
| stu\_id\_regx | 用户学号 | varchar | 9 |
| cn\_name\_regx | 用户姓名 | varchar | 10 |
| en\_name\_regx | 用户姓名拼音 | varchar | 50 |
| log | 系统时间 | Data | 默认长度 |

信息管理系统前端展示图，如图4-6所示，信息管理系统会展示人脸数据库中人脸的样本数以及每个样本的人员信息，可以根据9位学号数组查找到人员信息，能选择清除查询到的信息。点击训练后系统会通过OpenCV算法经过黑白化处理、转化为128个特征数组等操作训练没被训练的人脸模型，最后在右下角显示信息管理系统日志，人脸数据训练流程图见，如图4-7所示，信息管理系统界面的程序流程图，如图4-8所示。



**图4-6 信息管理系统前端展示图**



**图4-7 人脸数据训练流程图**



**图4-8 信息管理系统流程图**

人脸识别系统前端展示图，如图4-9所示，进入人脸识别系统后，可以选择开启系统摄像头或者外接摄像头，开启人脸检测后，画面会用方框框出人脸所在的位置，并显示摄像头的实时画面。打开人脸识别功能后，系统会通过计算欧式距离来判定是否与录入系统的人是否为一人，系统会检测实时画面中人脸，如果是人脸数据库中的则输出录入人员姓名拼音，最后在下面显示人脸识别系统日志。人脸识别系统主要通过将识别人员的128个特征值记录下来，通过欧式距离计算:

= **（4-1）**

比较与数据库的数值，即可判断是否为同一个人，其程序主要函数及其功能人脸识别系统函数表见表，如表4-3所示。

**表4-3 人脸识别系统函数表**

|  |  |
| --- | --- |
| 主要函数 | 功能 |
| CoreUI(QMainWindow) | 该函数通过PyQt5编辑人脸识别可视化界面，进行窗口布局 |
| self.faceTrackerCheckBox.stateChanged.connect | 该函数通过OpenCV算法进行人脸检测，检测人脸的位置 |
| self.faceRecognizerCheckBox.stateChanged.connect | 该函数通过OpenCV算法进行人脸预处理，处理摄像头采集的人脸信息 |
| def startWebcam(self) | 该函数通过指令调用开启或关闭摄像头 |
| def logOutput(self，log) | 该函数通过指令获取时间，实现日志效果 |
| def closeEvent(self，event) | 该函数通过指令将OpenCV线程、定时器、摄像头关闭 |



**图4-9 人脸识别系统前端展示图**

人脸识别实现的方法为:在人脸数据库中进行人脸检测，检测出人脸再将尺度归一减少误差，将检测的人脸提取特征进行全局特征比对，给出相似值并与预设的信任值对比，如果大于及输出人脸数据库录入的信息，在摄像头采集的画面情况下，先将人脸使用OpenCV算法检测出所在区域，在进行人脸特征点的标注,使用欧式距离公式计算在和上述人脸数据库操作全局比对，同理进行信任值和预设值的对比输出结果，当预设值大于信任值就输出unkonw且设备会发出警报声音，当信任值大于预设值就会输出对应的人脸信息，具体人脸识别流程图见图,如图4-10所示。

**图4-10 人脸识别流程图**

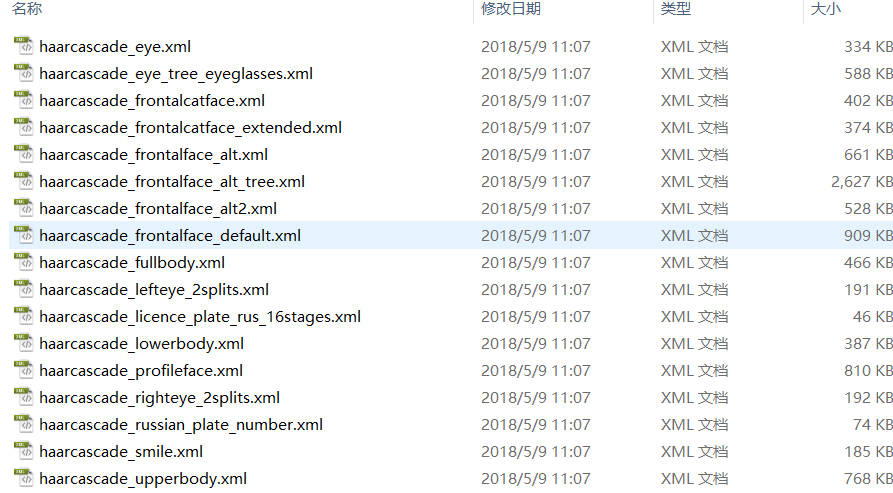
第五章 人脸识别的实现及其测试

## 5.1人脸识别的实现

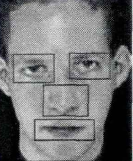
由OpenCV技术开源分类处理HAAR级联，对比训练好的数据进行人脸位置检测，再利用人脸识别的局部二进制算法做出一个相对准确的人脸识别程序，程序的ui界面是使用PyQt5组件设计。

HAAR目标检测算法:最先由Paul Viola等人提出，后经过Rainer Lienhart等扩展引入45°倾斜特征，是当今使用最多的检测框架之一[24]。HAAR在检测时利用conv9\_2、conv8\_2、conv7\_2、conv6\_2、onv4-3这些不同的feature maps，将这些面部进行分类和位置归一。该目标检测算法继承了YOLO中把Detection转变为regression的方法，进行一次便可以实现数据训练。

本文用到的两个人脸数据集是LFW和CeleA，前者数据集包括来自于6000人左右的人脸图片，并且这些人脸图片都是每个人从不同角度和不同背景拍摄的。后者包含两万多张人脸照片且提供多个点作为特征点坐标。该实验程序调用OpenCV中已经开源的级联分类器来依次检测嘴唇、眼睛和鼻子在人物中的确切坐标，所用到的Harr级联人脸特征分类器，如图5-1所示。用级联分类器检测人脸、鼻子、眼睛、嘴巴区域测试图，如图5-2所示。

****

**图5-1 Harr级联人脸特征分类器**



**图5-2 人脸鼻子眼睛嘴巴区域测试图**

Haar模型训练:将每个样本得到的重叠程度与最大的重叠程度进行比对，这样就能使得每个样本数据有比对的原型，当两个的交集或并集的绝对值大于设定的阈值，就可以认为图片是匹配的，输出样本就是原型，否则就是unknow。损失函数的计算和人脸图片的损失函数比较相似，即损失函数为损失位置与损失程度的对比，其中N是匹配到原型的重叠程度数量，参数是用来调整两个函数的数量关系。

人脸检测算法原理及设计方法:首先通过Haar人脸检测的方法先将人脸位置准确找到，使得Haar人脸检测通过此步骤在系统上可用。然后调用训练数据进行预处理，通过对比录入的人脸信息与刚才预处理结果，编写日志文件。最后，当识别已录入的人时，输出识别出的人脸信息；当识别出是陌生人时，输出unknow并通过播放设备发出警报声。人脸录入系统的主要模块包括调用摄像头进行拍照，将拍照得到的照片保存起来。进行信息录入，添加人脸信息并且同步到数据库，编写日志文件。人脸训练数据程序实现了显示全部以录入的人物数据，可以查看或者清除已录入的人物数据，将未训练的人脸信息用OpenCV的算法进行训练并保存，编写日志文件。

人脸识别程序的人脸拍照代码功能测试:运行对应代码成功拍照截取并保存人脸照片到指定文件夹/datasets中，人脸拍照代码功能测试图，如图7-3所示。



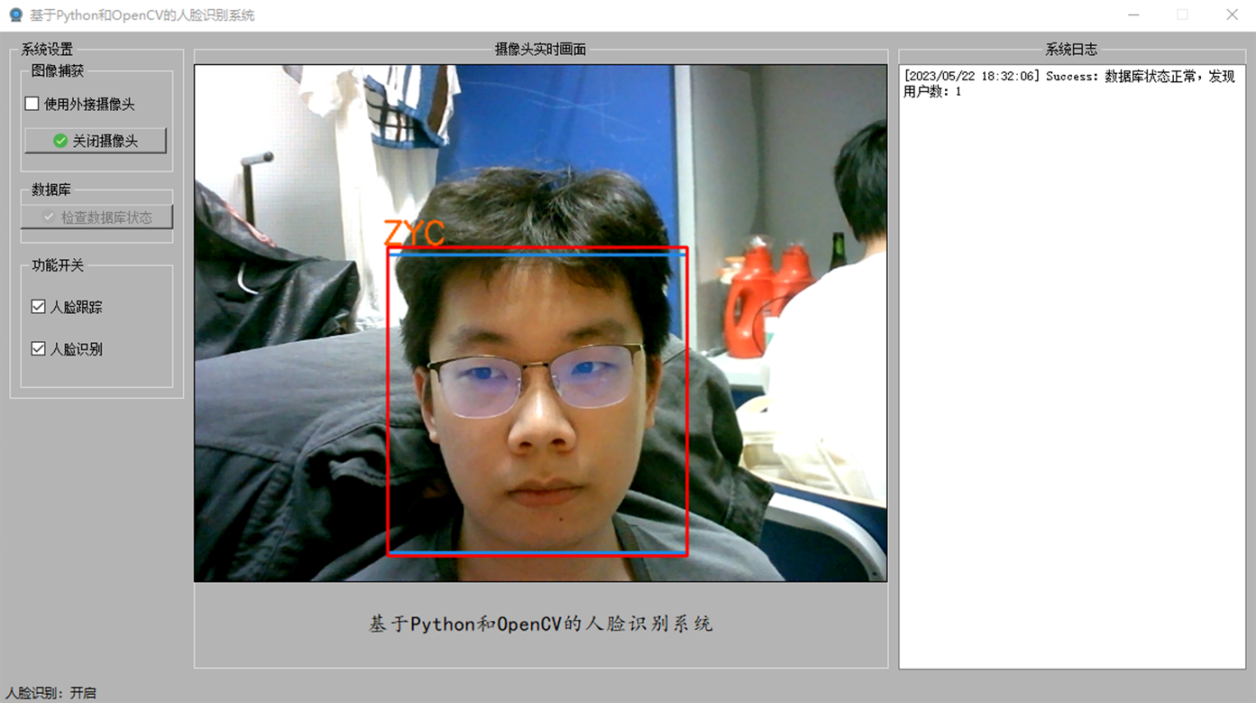
**图5-3 人脸拍照代码功能测试图**

人脸识别程序的人脸照片数组训练功能测试:运行对应代码成功将拍照保存的人脸照片通过OpenCV的算法转化为数组，保存到数据库文件中\人脸检测系统\recognizer\trainingData.yml，人脸照片转化为数组功能测试图，如图5-4所示。

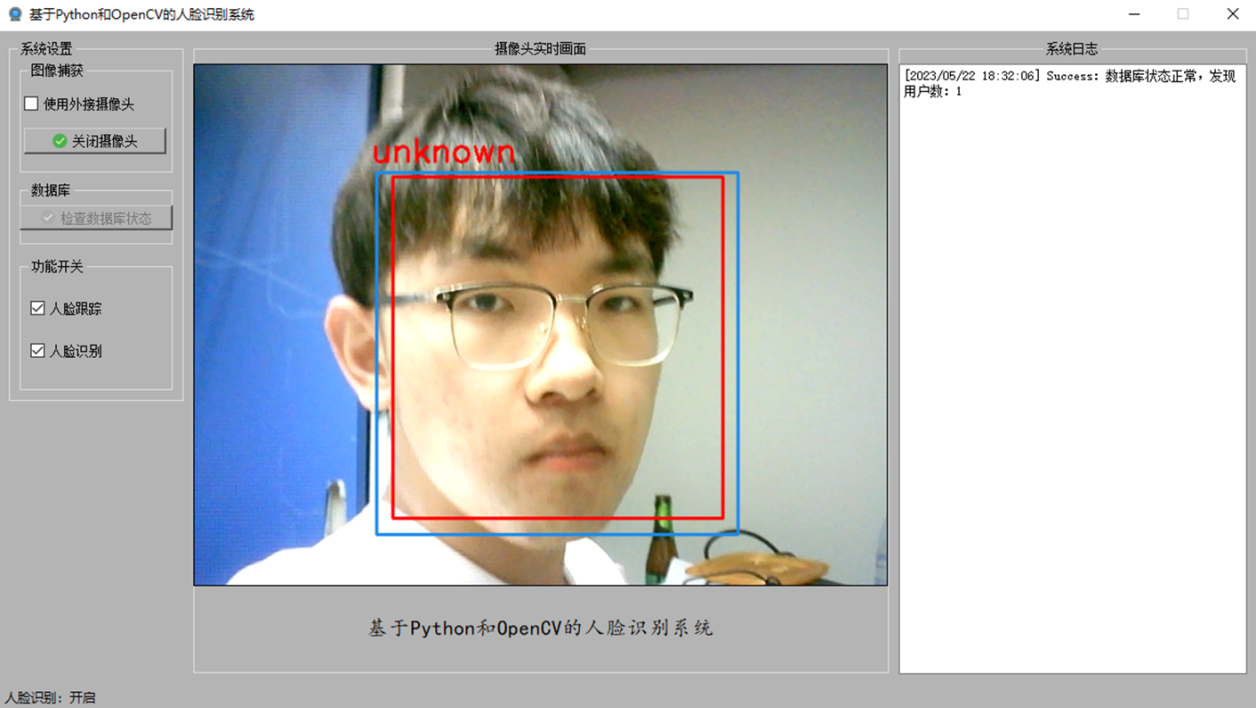


**图5-4 人脸照片转化为数组功能测试图**

人脸识别程序的人脸识别功能测试:运行对应代码成功用摄像头采集图像，将人脸信息于数据库中已录入的人脸信息进行比对，当数据库和摄像头预处理的人脸信息一致时输出录入的姓名，把姓名显示在运行红框之上，如图5-5所示。当数据库和摄像头预处理的人脸信息不一致时输出unknow并且发生报警，如图5-6所示。



**图5-5 人脸识别显示姓名测试图**



**图5-6 人脸识别不显示姓名测试图**

基于OpenCV与AI视觉的人脸识别实验程序通过三个核心系统设计并实现。采集的数据经过网络连接到识别程序所在的终端，再由计算机中的识别程序读取摄像头AI视觉特性捕捉到的画面，并对视频中的人进行人脸识别与检测，最后将识别的结果显示在窗口的显示屏上。其人脸识别流程图，如图5-7所示。

**图5-7 人脸识别流程图**

## 5.2 评价指标对人脸识别影响测试

影响程序识别时间的主要指标是confidence值的大小，若confidence值设置的大，程序识别时间会和其基本成正比，通过多次测试设置不同的confidence值得出以下的线性关系，confidence值与程序识别时间关系图，如图5-8所示。

**图5-8 confidence值与程序识别时间关系图**

影响程序识别精准度的主要指标是confidence值的大小，若confidence值设置的大，程序识别精准度会和其基本成正比，通过多次测试设置不同的confidence值得出以下的线性关系，confidence值与程序识别精准度关系图，如图5-9所示。**图5-9 confidence值与程序识别精准度关系图**

通过反复测试，综合考虑程序识别时间及程序识别精准度，得出confidence值设置为50且样本数量为100时，既可以满足实效性在一秒内输出人脸结果又可以满足可靠性使人脸识别准确率在94%左右。超过100张样本时，识别效率不会明显增加;少于100张样本时，识别效率明显降低。综上所述，在程序中confidence值和样本数的设定是尤为关键。如图5-10所示。

**图5-10 样本数和识别率关系图**

第六章 结论与展望

本文人脸识别实验程序实现的过程是在PyCharm软件中编写运行的，克服了Python环境不能可视化操作的难题。使用PyQt5组件编写主界面提供了轻量化的框架及运行环境。程序的核心识别技术是基于OpenCV的图像处理技术，分别在人脸录入检测，人脸与数据训练和将人脸特征信息转化为数组信息时使用。该程序用Python在PyCharm环境中编写代码，调用OpenCV技术去实现功能的设计思路去设计。在PyCharm环境当中编写Python的优势在于程序的可读性很高，对于提高Python编写习惯有很好的提升。

在传统编程中前后端都是分离的，通过接口去进行交互和显示。在此人脸识别程序当中通过PyQt5组件以及GUI图形用户界面，使得该程序是前后端为一体的轻量级程序，相比传统的程序前后端响应更加迅速，执行程序速度更为快捷，用户设计可视化界面更为直观。

调用OpenCV图像处理技术作为核心技术。使用OpenCV中自带的分类器快速准确的识别出面部区域，在程序的操作下进行面部特征转化为数组的操作，通过黑白化，灰度处理等一系类操作，将图像转变为数组ID存放在数据库中。在人脸检测环节中通过OpenCV中自带的联级分类器定位到人脸所在位置，并与数据库中人脸对应的数组进行比对，最终输出人物信息实现人脸识别功能。

关于人脸识别的展望。因为受到技术和专业知识的限制，在人脸识别实验程序中暂时不能添加随机开启多个摄像头之一的功能，希望在未来可以实现，这样可以用于监考程序的建设。本实验程序的安全性还未得到解决，区分不了3D和平面的图像，希望在今后的学习中可以将实验程序的算法进行完善确保人脸识别的安全性。同时也希望可以将人脸识别应用到监控程序中，实现监控并且显示每个人人脸信息的功能。

人脸识别应用前沿的技术，引进先进的思维去实现高效准确的身份识别，希望之后基于OpenCV与AI视觉的人脸识别算法得以改进，做到轻量化、可视化程度高、前后端不分离的程度。

参考文献

1. 郭明瑞.土地承包经营权流转的根据、障碍与对策[J].山东大学学报(哲学社会科学版),2014,5(05):1-9.
2. 肖阳.基于OpenCV的人脸识别的算法研究与实现[D].武汉:湖北工业大学,2018.
3. 李天昊.基于OpenCV的实时人脸识别系统的设计与实现[D].大庆:东北石油大学,2020.
4. 杜小甫,黄兴晗,刘沂杰.基于Python的人脸识别综述[J].电子测试,2021(16):55-87.
5. 叶继华,兰清平,刘长红等.结合互信息量和Log-Gabor特征的嵌入式人脸识别[J].系统仿真学报,2016,28(09):2214-2219.
6. 张晓兵.基于多特征融合的人脸识别研究[D].西安:西安科技大学,2018.
7. 黄大胜,林绅洋,李娟等.基于OpenCV的视频人脸识别研究[J].电脑编程技巧与维护,2020(06):115-117.
8. 黄子豪,孙沐阳,时田野.基于Python与OpenCV的人像检测和识别的研究[J].电子测试，2021(13):189-198.
9. 李莎.基于TensorFlow深度学习框架分析研究[J].电气传动自动化,2021，43(01):4-6.
10. 王羿.基于TensorFlow的人脸识别容器云平台框架设计[J].广播与电视技术,2021,48(07):139-144.
11. 王龙军,夏嘉杰,许靖唯.基于Android与Kotlin的人脸检测在智能图书馆系统中的应用[J].内蒙古科技与经济,2021(16):73-74.
12. 张伯辰,施鑫杰,霍梅梅.基于OpenCV的树莓派人脸识别疲劳驾驶检测系统[J].现代计算机,2021,27(23):129-132.
13. 缪来瑞，顾烨波.基于OpenCV的人脸识别考勤系统的研究和实现[J].科技视界，2021(13):2095-2457.
14. 李悦.基于Python+Tkinter的Linux GUI辅助管理工具设计 [D].长春:吉林大学,2009.
15. 朱宪飞. 交通系统监控环境下车辆异常行为识别算法研究[D].山东大学,2018.
16. 刘羽. 服装展示录像中人脸检测方法研究及应用[D].重庆理工大学,2010.
17. 赵团. 基于计算机视觉的驾驶员疲劳检测技术的研究与实现[D].东北大学,2010.
18. Bengio,Y.,& Hinton,G.(2015).Deep learning[J].自然,521(7553),436-444.
19. 余蓓蓓，王典洪 .基于灰度信息的人脸检测算法[J]. 苏州科技学院学报:工程技术版，2005，18（2）：81-83.
20. 邓耀. 基于单目视觉的汽车前碰撞预警系统研究[D].中国科学技术大学,2016.
21. 吴敏. 基于AdaBoost和独立分量分析的人脸检测与识别算法的研究[D].郑州大学,2010.
22. 张玉颖. 基于梯形模型及支撑向量机的非结构化道路检测[D].复旦大学,2010.
23. Viola, P.,& Jones,M.(2001).Rapid object detection using a boosted cascade of simple features[J].Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), December.
24. 王栩文. 基于图像的输电线路语义分割技术研究[D].浙江大学,2019.
25. Artificial Intelligence in Technology in Face Recognition Attendance Matecaachine[J].M2 Presswfire,2022:2095-2457.

致 谢

毕业论文的结束意味着我的大学生活即将圆满结束。回首往昔，在这四年的学习生涯中，有很多人给予了我宝贵的支持和帮助，使我能够顺利地完成这一重要的里程碑。

首先，衷心感谢余有明导师。您是我在学习道路上的引路人和榜样。感谢您对我毕业论文的精心指导和无私付出。您的严谨思维、专业知识和丰富经验启迪了我的思考，点燃了我对计算机科学领域的热爱。您耐心解答我的问题，指导我选择适当的毕业设计研究方向，并对我的论文提出宝贵的意见和建议，使我得以在论文的写作和研究过程中不断进步。

其次，我要感谢我的同学和朋友们。四年来，我们一起度过了艰辛却美好的求学时光。感谢你们在学业上的互助和支持，我们相互鼓励、相互学习，共同成长。在写论文的要点讨论中，你们的启发和建议给我带来了新的思路和灵感，使我的论文更加细致深入。

同时，我要感谢我的室友。你们是我大学生活中最重要的存在之一。赵庚辰，我们在同一个寝室度过了许多难忘的时光，每天的讨论和交流帮助我理解复杂概念；陈宇鹏，你的理解、鼓励和幽默化解了繁忙学习生活带来的压力。我们一起度过了各种高低起伏，你的陪伴使我感受到了温暖和家的亲切；毕经纬，感谢你对生活环境的共护和整理，使寝室成为一个适合学习的地方。你的积极态度和有序安排帮助我提高了学习效率。

此外，我要感谢北京石油化工学院。感谢学校为我提供了良好的学习环境与学术资源。特别感谢学院的教职员工们，你们的辛勤付出和悉心教导使我能够全面发展和成长。我深知自己取得的一切都离不开学校的培养和栽培。

最后，我要感谢我的父母和家人们。感谢你们多年来无私的关爱、支持和理解。是你们始终鼓励着我，给予我坚定的前行动力。特别是在我面临困难和挑战时，你们的支持让我加倍勇敢，克服了重重困难，为我铺就了追求梦想的道路。