摘要

目的：随着科学技术不断地进步，随着时代车轮的缓缓前行，以前的一些技术也越来越不能适应当下的需求，其中最为典型的就是打卡技术。在旧时代，人们往往需要雇佣一个人来记录人们上下班的工作信息，这明显是对人力的极大浪费。而后又迎来了指纹打卡时代，而指纹打卡也存在极大的弊端，比如：手上有水或者由于长期泡水导致指纹不清晰等原因就会对打卡造成困扰，要么不能识别，要么识别错误。于是，通过人脸识别进行打卡的系统应运而生。

方法：本系统在Windows系统下利用python语言，借助opencv的图像处理技术对人的面部进行图像处理，然后通过使用face-recognize库对处理后的图像匹配，最终返回显示打卡信息。注册功能则是对捕捉到的人脸保存。

结果：打卡成功

在这篇文章中，对人脸识别的发展历程，相关技术以及图形用户界面的操作做出详细解释，同时会讲解自己实现本系统的详细过程，同时会对编程中遇到的一些问题做出讲解以及解决办法。

关键词：人脸识别，图形化，opencv

**Abstract**

Purpose: With the continuous progress of science and technology and the slow progress of The Times, some previous technologies are increasingly unable to meet the current needs, among which the most typical one is the punching card technology. In the old days, it was often necessary to hire someone to keep track of people's work as they went to and from work, an obvious waste of manpower. Then ushered in the era of fingerprint punching in, and fingerprint punching in also has great disadvantages, such as water on hands or unclear fingerprints due to long-term soaking in water and other reasons will cause trouble to punching in, or can not identify, or identify the wrong. As a result, the system of punching in through face recognition came into being.

Methods: The system uses Python language under Windows system, and uses OpenCV image processing technology to process the image of human face, and then uses face-recognition library to match the processed image, and finally returns to display the punch card information. The registration function is to save the captured face.

Result: successful clocking.

In this article, article will explain in detail the development process of face recognition, related technologies and the operation of the graphical user interface. At the same time, I will explain the detailed process of realizing this system by myself. At the same time, I will explain and solve some problems encountered in programming.

Keywords: face recognition, graphics, OpenCV

目录

[１ 绪论 1](#_Toc72350974)

[1.1 研究背景 1](#_Toc72350975)

[1.2 国内外研究现状 1](#_Toc72350976)

[1.3 本文论文结构 2](#_Toc72350977)

[2 系统开发环境 3](#_Toc72350978)

[2.1 开发环境 3](#_Toc72350979)

[2.2 开发中所使用的核心技术 3](#_Toc72350980)

[2.2.1 opencv 3](#_Toc72350981)

[2.2.2 face-recognition 3](#_Toc72350982)

[2.2.3 GUI图形界面 4](#_Toc72350983)

[3 目前流行的人脸识别算法及其简介 5](#_Toc72350984)

[3.1 子空间算法 5](#_Toc72350985)

[3.1.1 PCA算法 5](#_Toc72350986)

[3.1.2 LDA算法 6](#_Toc72350987)

[3.2 人工特征+分类器 6](#_Toc72350988)

[3.3 基于深度学习的人脸识别 7](#_Toc72350989)

[3.3.1 卷积神经网络 7](#_Toc72350990)

[3.3.2 SphereFace 9](#_Toc72350991)

[4 需求分析与可行性分析 10](#_Toc72350992)

[4.1 需求分析 10](#_Toc72350993)

[4.2 可行性分析 11](#_Toc72350994)

[5 详细设计与具体实现过程 12](#_Toc72350995)

[5.1 打卡模块 12](#_Toc72350996)

[5.2 注册模块 14](#_Toc72350997)

[5.3 显示及备份模块 14](#_Toc72350998)

[5.4 GUI界面设计（tkinter） 15](#_Toc72350999)

[6 系统测试 19](#_Toc72351000)

[6.1 环境测试 19](#_Toc72351001)

[6.2 打卡功能测试 19](#_Toc72351002)

[6.3 注册功能的测试 20](#_Toc72351003)

[6.4 显示模块测试 21](#_Toc72351004)

[7 结论 22](#_Toc72351005)

[8 参考文献 23](#_Toc72351006)

# １ 绪论

## 1.1 研究背景

当人类进入21世纪以来，科技进入了蓬勃发展的黄金时期，信息也随着科技的发展以及互联网的普及而迅速贴近人们的生活。然而，随着人们对于信息使用的越来越频繁，特别是自世纪之初轰动世界的911案件以来，人们对于社会公共安全愈发重视。信息识别体现出其越来越重要的位置，其运用范围之广几乎覆盖了我们日常生活的各个领域。而如今所使用的身份识别方法，如：电话号码、身份证、特殊语句等等，但是这些都存在着易丢失、易遗忘等问题，而随着科技的发展，这些传统的验证方法都遭到了严重挑战，其可信度也越来越底。由此，人们渐渐把目光转向了生物特征识别方面，由于生物特征都是由人类的DNA所决定，是每个生物体独一无二的特征，这就大大增加了身份验证安全性。生物识别大致分为物体体征和行为体征两大方面。行为体征有笔迹，动作等；物体体征有指纹、虹膜、人脸等等。而相比于其他的体征，人脸识别则有着易采集，非接触，静态等其他方法所不具有的优势，因此也就成为人们比较容易接受的方法。

## 1.2 国内外研究现状

实际上，人脸识别领域有着非常广泛的运用领域和前景。在我们日常生活比如在公司门禁，银行取款，医疗护理等等都是非常适合使用人脸识别的场景。在历史上，人脸识别的运用可以追溯到19世纪的法国。当时有个名叫Galton的人，他工作中就使用了这个方法。而我国的相关研究则起源于上世纪八、九十年代。尽管我国对于人脸识别的研究起源很晚，但仍旧取得了许多重大成果。人脸识别技术的发展经历大致可以分成三个阶段：首先是一般性能下的面部特征研究；然后是人脸识别技术成果井喷期；最后就是实现真正的机器自动识别时期。

第一阶段：一般性模式下人脸面部特征研究所采用的技术主要是基于人的面部几何结构的方法。Bertillon使用一个简单的数据将人的面部特征和数据库中数据联系起来。这一阶段是人脸识别初级阶段，尽管成果不多，实用性不强并且对人工依赖很大，但仍然为以后的人脸识别的发展奠定了基础。

第二阶段是人脸识别技术发展的黄金时期。在这一阶段中诞生了许许多多具有代表性，创造性的识别算法，在这一时期，美国官方还组织一场十分出名的人脸识别的测试，同时在这一时期还诞生了一些商业化用途的人脸识别系统，比如：Visionics的FaceIt系统。这一时期，美国的特克和潘特兰提出了当时最出名的人脸识别算法--“特征脸”方法—它被证明是最佳的均方原始图像和重构图像时产生误差极小化的压缩方法。当然，取得人脸识别算法进展的还有实验室的布鲁内和波基奥，在1992年左右他们做的一个比较实验，他们比较了基于结构特征和基于模板匹配的方法的识别性能差距，并给出了一个相对肯定的结论：模板匹配是要比基于特征更好的手段。这一结论的出现加上特征脸的技术促使了纯粹基于结构特征进行的人脸识别这一研究方向走向没落。但同时又激起了基于线性子空间建模和基于统计模式识别的方向的兴起并使之成为之后研究的主流方向。

人脸识别发展的第三个阶段实现了完全的自动识别，在这一阶段中，研究人员攻克了部分对人脸识别精确度会造成影响的比如：光照，表情等因素的影响。在研究者对它进行的研究并获取结果中，人们对这一技术的了解和运用越来越纯熟，该项技术逐步向商业用途靠近。美国政府还为此举办了两场系统测评。基奥盖蒂斯等人提出了基于光照锥的多光照、多姿态条件下的人脸识别方法，他们证明同一人脸在同一角度的不同光照下所形成的所有图像在图像空间中形成一个突起的光锥—光照锥。同时以支持向量机为代表的统计学理论也在这一时期被人们用来进行人脸识别和确认，这部分学者认为人脸识别是一个多类问题，此时通常有三种方法来解决这一问题：类内差/类间差法、一对多法以及一对一法。与此同时以布兰兹为首的研究团队又提出了以3D变形模型为基础的复杂条件下人脸图像分析与识别方法，这项工作在此时是一项具有开创性的举动，由于他在本质上是基于合成的分析技术，所以他让人脸形状，纹理等的内部属性和光照，摄像机配置等外部因素完全分开，以达到人来能识别图像的分析和识别的精确度大大提高。在2018年8月，由英伟达和MIT联合推出一款极其逼真的高清视频生成AI—vid2vid技术，在这个技术下，你能够无需实拍，只是勾勒出心中所想，该算法就能够实现相应的图像。这是计算机视觉领域的一大突破，同时促进了计算机人脸识别的向前发展。

总体而言，目前基于视频的人脸识别，三维空间的人脸识别以及跨年龄，少样本，有遮挡，少模型等类似问题逐渐成为研究的热点。而利用深度学习进行的像卷积神经网络，双镜头人脸检测，使用少量计算即可得到精确结果等逐渐成为备受重视的技术发展趋势。

## 1.3 本文论文结构

本文一共分为6部分，其大致安排如下：

第一章为绪论。其大致陈述了人脸识别的研究背景及其目的，简述了人脸识别技术的发展历史以及当今的发展现状。

第二章则简述了本系统所使用的开发环境以及关键技术。分析了对于所用技术的理论基础以及使用方法以及其拥有的优势。

第三章讲述了当前所流行的一些人脸识别技术和相应原理以及这些人脸技术的理论基础、发展历史。

第四章需求分析和可行性设计，这一章主要介绍了本系统主要实现了哪些功能，这些功能的必要性；实现这些功能时可能遇到的问题以及相应的解决方法以及以目前的发展状况实现该功能的可能性。

第五章详细设计以及具体实现过程；这一部分则讲解了对于本系统各个部分的实现逻辑，所使用的方法，这一部分包括了关键代码部分以及实现的方式。

第六章是系统测试，此部分则是对系统相应的GUI界面加以展示，同时说明此系统的运行流程。

第七章主要是对本系统的优缺点做出总结。

# 2 系统开发环境

## 2.1 开发环境

本系统使用win10操作系统为基础，以python语言将opencv技术和face-recognition库相结合起来为工具，实现人脸信息捕捉，识别功能，最终达到人脸识别打卡的目的。

## 2.2 开发中所使用的核心技术

### 2.2.1 opencv

自从他被inter创造到现在由Garage公司供应维持以来，他一直是一款跨平台的轻量级的、高效的软件库。它提供了计算机视觉和机器学习的功能，并支持多个平台的软件开发，如：windows、linux等等。它拥有包括几百个函数的API，这使得它在图像处理方面拥有着强大的功能，并且它不依赖于其它的外部库，这让他与其他外部库保持着低耦合性；并且它虽然是基于C语言以及C++开发的软件库，但仍然支持python等多种语言的应用。它实现了很多关于图像的处理和机器学习的通用算法，由于其在人机互动，图像分割识别，图片处理等等很多方面都可以很好的应用，这使得它被很多开发者喜爱。

### 2.2.2 face-recognition

无论是在政府还是商业的系统应用中，面部识别技术有着无限的潜力，因此在以前的几十年中这项技术一直吸引着许多研究人员注意，这使得它一直都能保持发展。在1995年，切拉帕等人列举了一部分面部识别技术，同时讲述了其中的优点和缺点。然而，他们并没有对已经应用的系统进行分析。1997年，至少来自13家公司的25项面部识别系统已经被应用。自此以后，由于许多新兴应用领域的出现，推动着面部识别技术的进一步发展，客户也逐渐将目光转向这个领域，这使得这项技术在商业领域逐渐成熟。本系统所使用的，用来实现人脸识别考勤的核心就是face-recognition库，由于python语言将他封装成一个十分简单的人脸识别接口，这让开发难度大大降低。并且，因为它使用深度学习来训练数据模型，所以他训练出来的模型的准确率很高。他利用自身所有的函数接口对图像人脸进行捕捉然后使用对捕捉到的人脸特征分析，最终达到分辨人脸的目的。

### 2.2.3 GUI图形界面

计算机历史上最伟大的发明创造之一就是图形用户界面，它对计算机的普及与进一步发展具有深远的意义。GUI图形界面是采用图形的方式实现计算机和用户之间的交互。用户能够使用按键，窗口，菜单等等对象对计算机进行操作以实现自己的目的。由于传统的控制台界面操作复杂，普通人很难理解加上它的显示并不美观，于是就诞生了GUI图形界面。通过这种手段，用户可以不必具备专业知识就能对，软件进行相关操作。GUI的发展历经萌芽期，发展期。随着不断地发展，图形用户界面逐渐趋向善用隐喻、直接操作、能见能点、视觉艺术完美性的方向发展。

它拥有以下特点：第一图形用户界面能够实现人与计算机等电子设备的人机交互，并将交互结果通过显示屏反馈给用户；第二点是随着现在时代的发展，图形用户界面发挥越来越重要的作用，一个美观的程序界面通常更能吸引用户。恰好，由于GUI是多元素组合，在这其中包含了很多艺术性设计，具有构成美、色彩美、光影美、人性美，界面美观就能吸引客户使用产品。第三个特点则是它方便了用户的使用，能从实验室走进千家万户，给用户带来良好的使用体验，实用性强。最后一点就是用户在操作时由于发送和接受都是图形对象因此不用记忆繁杂的命令，这使得用户不必具备专业知识和操作技能，虽然对于用户来说降低了其专业性，但对于开发者则是将加强了程序的专业性。在我们设计GUI时必须遵守以下原则：

1. 界面风格一致性；
2. 界面布局逻辑性原则；
3. 界面设计具有启示性；
4. 界面设计遵循习惯性用法。

# 3 目前流行的人脸识别算法及其简介

## 3.1 子空间算法

### 3.1.1 PCA算法

PCA算法又叫主成分分析算法，它是一种常用的降维手法，它通过某种线性投影将高维的图像映射到低维的位置，希望通过以此种方式用最少的维度保存到最多的特征值[[1]](#endnote-1)。此算法首先利用图3.1中的公式计算出最大化数据映射后的方差，

附图

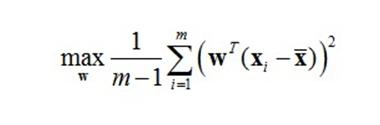


图3.1方差公式

然后经过线性变换得到如下图3.2优化后的公式，其中W’W=I是希望所有的向量都能够正交以达到不会有冗余信息的目的。

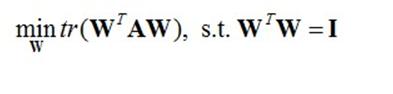


图3.2 优化后的公式

PCA的输出就是Y = W'X。经历了这一系列操作后，x的维度由原始维度降低到了n维。虽然他追求能够最大化保存图像内部信息，但是，这种处理后反而使许多数据杂糅在一起，导致无法区分，因此这成为了PCA一个极大的缺陷，也导致在很多时候此算法的效果不能达到理想状态。

### 3.1.2 LDA算法

LDA算法，是一种监督学习的降维技术。也就是说他的数据集的样本输出都是有分类的。但是，虽然它拥有分类的功能，但是我们常用的还是其降维的方面。他是经典的模式识别相关算法，于1996年被引入模式识别和人工智能领域。他的基本思想是对数据进行降维以尽量满足以下条件：

1、尽量保存数据的样本信息；

2、找出样本最好分的最佳投影方向；

3、投影后可以使得同一类的样本距离自己最近，不同类的则尽量远离自己。

所以这种算法在空间中特征分离的性能几乎达到当时水平下的最佳。它通过计算类内，类间的散度矩阵Sw，Sb，而后利用这些矩阵，得到最大的多个特征值和特征向量同时得到投影矩阵，最终输出样本集；因此，他在将为过程中可以使用类别先验知识经验，这也是其相较于PCA算法的一大优势。但是，他不适合非高斯分布样本的降维并且可能会产生过度的拟合数据。

## 3.2 人工特征+分类器

在机器学习中常用的分类器大致分为线性和非线性两种。其中线性的有LDA，逻辑蒂斯回归等等，其模型是线性的参数模型，分类平面是平面或者超平面；非线性分类器又包含朴素贝叶斯，朴素树等等，它的分界面可以是超平面和曲面的组合。

而这一步的关键则是人工特征的设计，因为他需要能够准确识别出人脸。在此过程中，很多用于描述图像的技术都运用到此处，像：HOG,SIFT,LBP等等，而在这中间最为典型的就是LBP（局部二值模式）特征，它简便有效并在一定程度上解决了光照的问题，但是仍然会被人面部的姿态，表情等的影响，随后的是联合贝叶斯，它是贝叶斯人脸的改进。它以局部二值模式特征和拉普拉斯映射特征提取为基础特征，将人脸的差异与因为姿态、表情等的差异分成了两个因素。同时用潜在变量组成的协方差将两张人脸进行关联。此方法能够将两张人脸进行关联，在建立关联的同时能够使用人脸先验知识经验，将两张照片当成一张人脸问题进行处理，使其能够更好的验证人脸的相关性，从而达到更高的精确度。

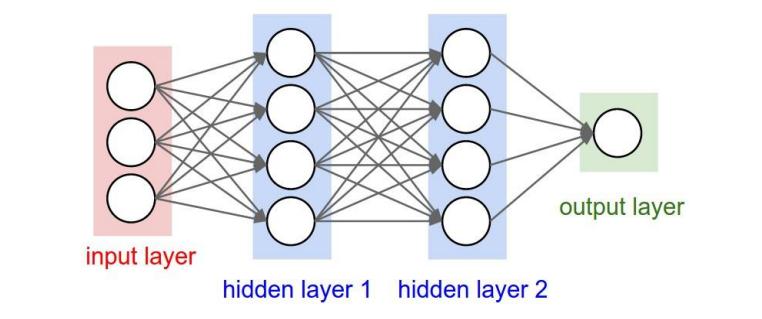
人工特征的巅峰是出自微软亚洲研究院的一篇翻译过来名为《二维许可：高维特征及其在人脸验证中的有效压缩》的文章，在文章中，作者举出案例证明了高维特征与验证的机能之间有着正相关的联系，进而证明了：人脸维度越高，其验证的准确性越高。在当时即使没有用到DeepLearning在LAWF上的精确度还是达到了95+%。因此，这篇文章在当时引起了许多学者的重视，大有成为人脸识别领域的新航标的势头。然而，随着深度学习的出现，这种方式逐渐没落，最终退出人脸识别的舞台。

## 3.3 基于深度学习的人脸识别

### 3.3.1 卷积神经网络

自从2012年深度学习开始展露锋芒后，出现了很多研究者试图将此技术应用到自己的领域，这一举动则大大推进了深度学习技术的发展。卷积神经网络从众多中脱颖而出，展示出其巨大的威力和潜力。

我们都知道神经网络结构图正如图3.3所示，是一个层级网络；其实

 图3.3 神经网络结构

卷积神经网络同样也是层级网络，只是相较于神经网络，他的层的形式和功能都做出了很大的改动。他一般要分五层。

第一层：数据输入层。这一层是对要处理的进行一个预处理，即：去均值->归一化->PCA/白化。其中，去均值就是把图片数值各个维度全部去中心化为0，使得样本的中心能够落在原点上；归一化就是将幅度统一，减少各个维度因为幅度不同导致的误差，比如：A的幅度是0-10，B的幅度是0-1000，此时，如果我们不进行归一化处理，就会对图像识别的准确度造成极大影响，导致最终输出结果不准确。PCA/白化，就是先对各个维度的数据进行PCA降维。然后采用白化技术对数据中各个特征轴幅度进行归一化。

第二层：卷积计算层，正是由于这一层的存在，卷积神经网络才有了这个名称。这一层主要做两项工作：局部关联以及窗口滑动，滤波器对局部数据进行计算。其具体实现过程如图3.4所示；同时，在这有一个参数共享机制，在这个卷积层中每个窗口的神经元权重都是固定的每个神经元都只关注自己的权重，而每个神经元集合起来就好像是整张图像的特征提取器的集合。

附图

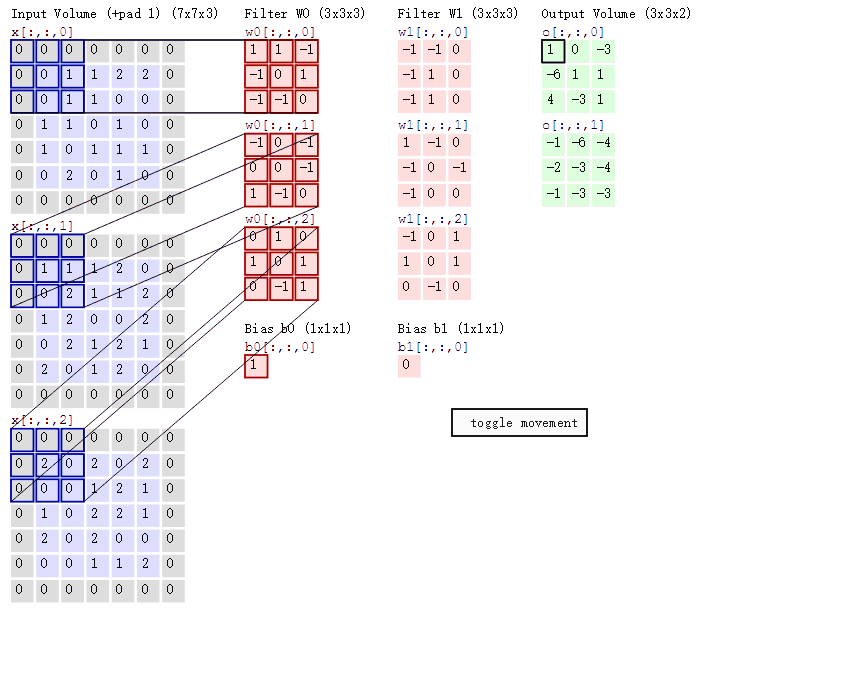


图3.4 滤波器工作原理

第三层是激励层，他将卷积层的输出结果做一个非线性映射，卷积神经网络一般采用修正线性单元（REUL），它操作简单同时收敛速度快，但会比较脆弱。在这一层要注意不要使用sigmoid，首先使用REUL，因为使用时速度很快但要小心，它很脆弱；如果不能有效使用推荐换用Leaky REUL。

第四层是池化层。这一层是用来压缩数据和参数的量，从而减少过拟合。pooling常用的技术手段一般有最大值池化，平均值池化，L2池化。但由于它是一种暴力降参的手法，因此在计算力充足的情况下这种操作的必要性就不太确定了，因此目前大一些的卷积神经网络只是偶尔使用这种方式。

第五层是全连接网络层。这一层的主要目的就是分类，之所以称他为网络，则是大多数CNNs采用多层全连接，而这样的结构即可被看作是网络，同时全连接层在卷积神经网络中处在神经网络的尾部，就和传统的神经元连接方式相同。

### 3.3.2 SphereFace

SphereFace，就是改进原来使用的softmax loss，形成的A-softmax loss；同时将此技术运用到人脸识别领域。因此，想要了解SphereFace，我们不得不先聊聊关于softmax的相关理论。Softmax功能就是将卷积神经网络计算后的神经元输出映射到0到1之间，并且这些值加在一起恰好等于1，这样就能得出每种分类的概率。相较于原来的softmax loss层中最后一个全连接层的输出，在A-softmax中，最后一个全连接层的输出对公式：wTx+bwTx+b优化,如果对w归一化，且b=0,则可以简化为||x||cos(θ)||x||cos(θ)，这就导致了学习到的特征xx发生了变化。通过该方法可以将特征xx转换到了球体空间，使得学习到的人脸特征具有更好的可区性。（softmax的损失函数，图3.5中的公式即为其定义）

附图

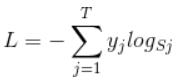


图3.5

# 4 需求分析与可行性分析

## 4.1 需求分析

如下图4.1所示。本系统以python语言为基础进行开发，其中包含了对opencv的使用附加了一些图形用户界面进行的开发。打卡部分：结合opencv调用摄像头对图像中人脸进行捕捉，然后利用opencv将捕捉到的图片进行灰度处理，二值化等等操作，之后利用face-recognition库对处理后的面部图像进行编码，利用得到的数组将其与库中人脸匹配，得到最佳匹配度人脸后输出其名称并返回给前端，同时将获取此时的系统时间，按照“人名+时间”的格式存储到Attendance文件中。

注册部分则是考虑都会有新员工加入所必需的一个部分，这一部分同样是先对图像中人脸进行捕捉处理，然后会在我们确定人员面貌时输入其人名，并保存在ImangesAttendance文件中，同时更新打卡系统中的信息。

显示记录部分则是对员工打卡情况的浏览，方便对员工的管理。这一部分涉及到GUI界面开发，首先需要创建表格，然后读取文件，拆分字符串，从中取出我们需要的子串，最后将分割出的字串返回到表格相对应的位置。

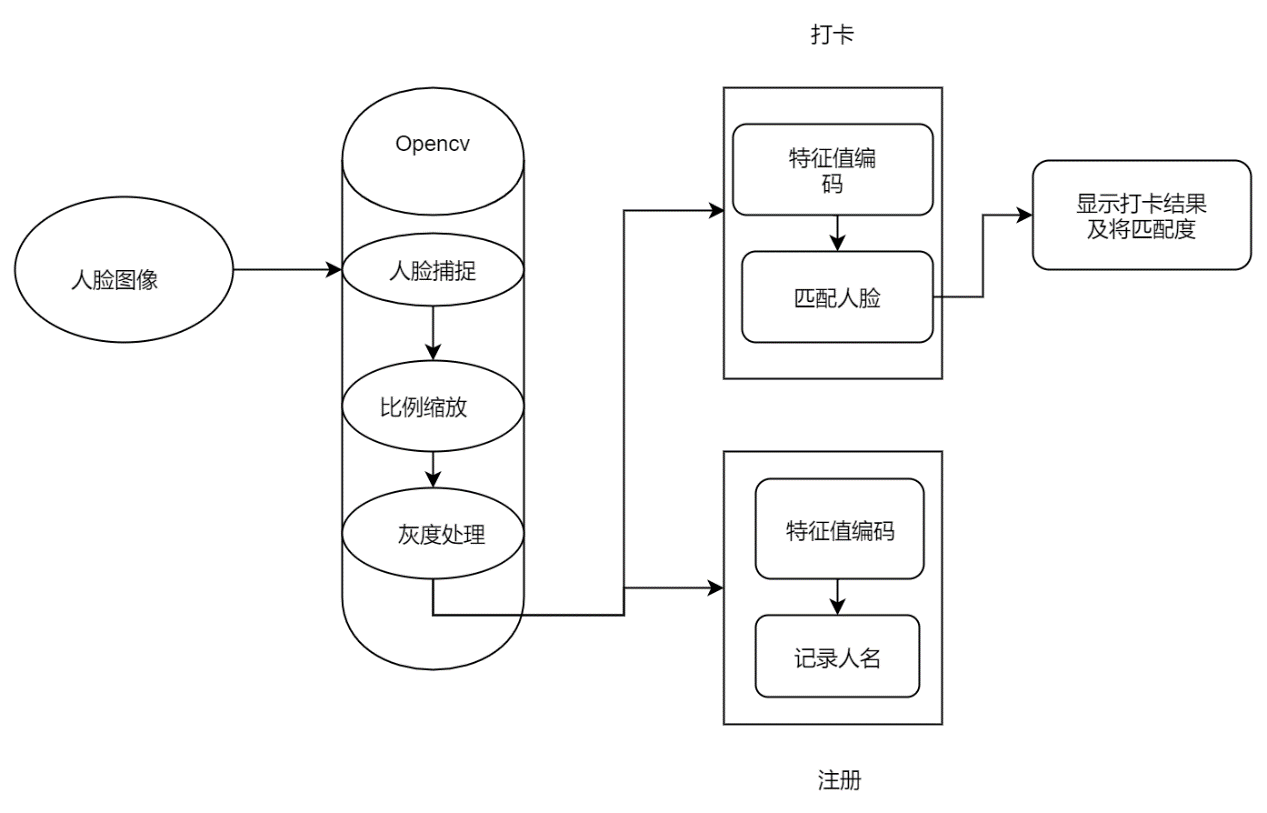


图4.1

## 4.2 可行性分析

首先软件支持：开发一个简单的运用opencv以及face-recognition来处理图像进行人脸识别主要就是帮助刚接触到图像处理的新人熟悉图像处理的流程以及相对应的功能方法。因此一台能够装载pycharm并顺利运行的电脑即可。在安装python解释器后在opencv官网下载安装包解压导入pycharm中所使用的工程即可。当然，我们也可以直接在命令行中输入pip install opencv-contrib-python命令即可自行安装。然后在命令提示符中安装facce-recogniztion。此时要先输入：pip.exe来查看自己的pip是否存在，这是python用来安装包的命令工具。然后下载CMake并将相关的环境变量加入到自己的电脑，之后利用install+dlib版本号安装dlib库，最后安装fece-recognition’库。

其次，目前对于人脸识别相关技术应用已经十分广泛且成熟。其识别率几乎能达到100%的程度，而且能够在各种状态，如：强光，弱光，表情等的影响下准确识别出人脸。我在课程设计验证中使用自己的设计结果进行过测试，其结果相当准确，所以说人脸识别这个关键技术是可以使用的。

另外，虽然在本次毕业设计之前，我并未接触到相关图像处理的算法以及实现过程，但在确定题目之前，老师曾让我们事先了解过对于相关技术的一些新成果，新论文，了解到关于图像处理的最前沿消息，对其大致的概念有了一个模糊的认知，这使得我在做课程设计时不至于完全陌生。

# 5 详细设计与具体实现过程

## 5.1 打卡模块

在这一模块中，我们的核心就是高精度识别不同的人脸。首先，我们要做的事情是将人脸库的数据读取出来，因此，我们先使用os.listdir()读取人脸库中每张照片的名称，而后遍历此列表去掉名称中的后缀，也就是“，jpg”。这样，我们就能得到所需的名字。第二部则是人脸识别。人脸识别部分先从摄像头中获取图像，利用cv2.resize()函数进行比例缩放得到我们所需要的合适比例，然后在对图片实施灰度处理，因为色彩对于人脸寻找没有什么帮助，之后利用face-recognition中的函数face\_locations（）查找到人脸的位置进行人脸分割，最终返回人脸位置list（ top，right, bottom, left）进行处理，由于在获取到人脸是不可能总是正的，因此我们要将人脸调成同一标准，之后利用face\_recognition函数库中的face\_encodings（）函数得到一个128位的特征向量列表。（图5.1 具体实现过程）

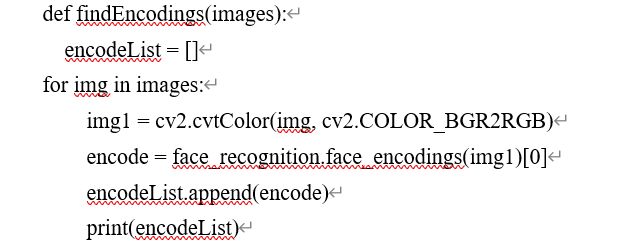


图5.1 代码实现

然后将文件进行遍历，使用.compare\_faces（）函数选择出匹配度最高的一个人脸特征值，同时使用face\_distance（）函数计算出两张人脸欧氏距离。之后返回图片的文件名以及欧式距离。当欧氏距离越大就表明两者越不想似。利用用着一点，此时我们也可以为欧式距离设置一个阈值：当最后得出的欧氏距离大于达到我们设置的阈值，返回“查找不到此人”，即当前库中并没有与之匹配的人脸信息。此时我们就需要下一个模块：注册功能。具体实现过程如图5.2所示：

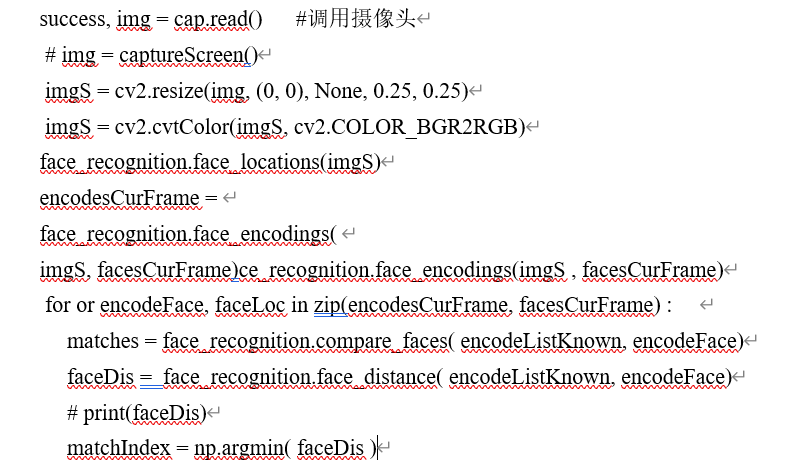


图5.2 具体实现代码

最后利用opencv中的函数进行人脸定位，框出打卡人的人脸位置，在框下显示打卡人名以及欧氏距离。具体实现如图5.3.

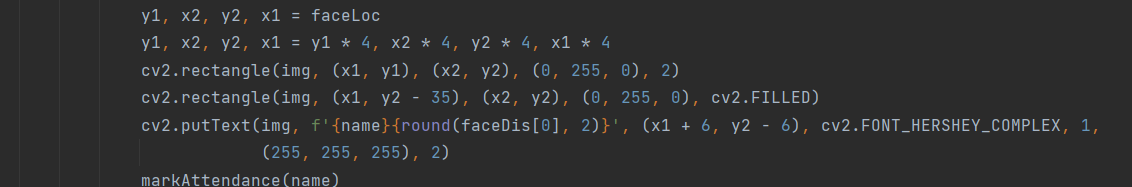


图5.3 定位人脸的实现

同时我也将系统识别出的人名用“人名+时间“的方式记录在Attendance文件中。这一步中，我将其封装成函数，当识别出人脸就将其传入markAttendance(name):中，然后在文件中查找此人名是否存在，如果存在就说明此人已经打卡，就不写入文件；如果文件中没有此人名，就将其写入，同时写入程序捕获到这张人脸的时间。这样做就防止了重复打卡的情况，节约了成本。代码实现如图5.4所示。

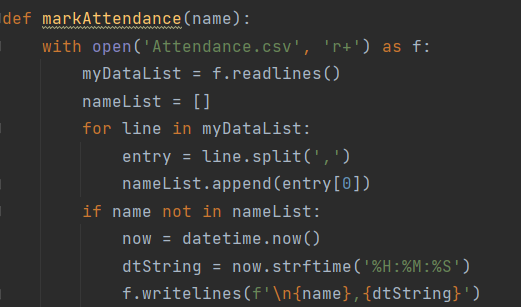


图5.4 记录考勤情况

## 5.2 注册模块

此模块实现起来比较简单，它先是使用cap = cv2.VideoCapture( 0, cv2.CAP\_DSHOW )调用摄像头，然后通过摄像头捕捉图像，当我们寻找到要注册的人脸图像时进行截图，此时键盘输入‘q’确认，然后调用askstring（）弹出一个输入框。在我们输入文件名—人名点击“确认”后，系统会将text框中字符获取传递给imwrite（）函数中name作为文件名保存在人脸库中而后关闭摄像头。为了降低代码耦合度同时方便图形界面的调用，我将其封装成函数，具体实现如下图5.5所示：



图5.5 注册功能的实现

## 5.3 显示及备份模块

显示打卡记录部分使用到了tkinter模块以及文件流的读取。首先，我们在主界面画布上创建一个窗口，然后，利用treeview（）函数创建一个表格，然后利用column（）函数设置每一列的列名以及大小，利用heading()函数设置表格的表头。这样就建好了一个表格。接下来就是从Attentance文件中使用readlines()函数读取到行数，使用循环利用reader（）函数的读取每一行的字符串。由于存储时的格式是“人名+时间”，因此我们可以利用“，”对字符串进行分割，先后获取到人名和时间。同时，利用insert（）函数分别将人名以及时间添加到表格中的相应位置进行显示。

tree = ttk.Treeview(win)

tree["columns"] = ("姓名", "打卡时间")

tree.column("姓名", width=100) # #设置列

tree.column("打卡时间", width=100)

tree.heading("名字", text="名字") # #设置显示的表头名

tree.heading("打卡时间", text=" time ")

count = len(open('Attendance.csv').readlines())

with open('Attendance.csv') as f:

f = csv.reader(f)

headers = list(f)

for i in range(1, count):

tree.insert("", 0, text="NO " + str(i), values=(headers[i][0], headers[i][1])) # #给第0行添加数据，索引值可重复

tree.pack()

## 5.4 GUI界面设计（tkinter）

它能够很好的适用于多个操作系统，并且它提供过了很多的组件用来支持图形界面开发。这些都是需要在界面先进行附着，然后才能创建。并且，这些组件是没有优先级之分的，都在同一高度。同时，它又是一款内嵌于python语言的插件。因此，对于用这种语言开发而又需要使用图形化开发的本系统来说，这个函数成了当下最合适的了。最够则是制作此程序的界面并不需要实现一些复杂的功能，因此这个插件刚好能够满足本系统的要求。正是其简单，方便的优点，让我最后决定选择它。当我们打开程序，

附图

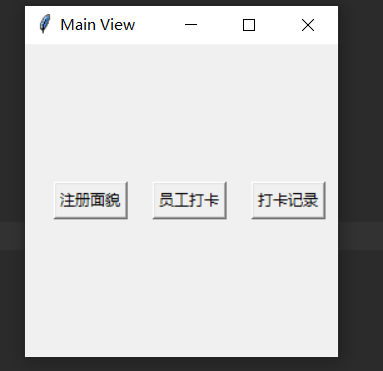


图5.1 程序主界面

如图5.1首先会弹出一个主页面的弹窗，画布上有三个按钮分别对应本程序的三大功能。这样布局能够清晰的展示出本程序用途，同时使各个部分模块化，能够独立操作，从而降低耦合，提高代码质量。之后当我们点击“注册面貌”按钮时，则会弹出一个窗体获取图像。如图5.2 所示，完成注册。

附图：



图5.2打开摄像头获取人脸

当我们点击“员工打卡”则会弹出如图5.3所示的弹窗。当我们点击“打卡记录”按钮时则会弹出打卡人员的顺序，人名，时间。如图5.4所示

附图：

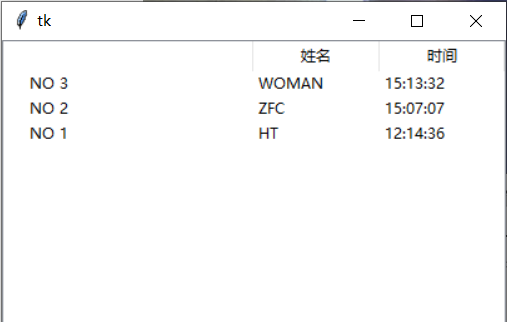


图5.4 考勤信息反馈

附图：

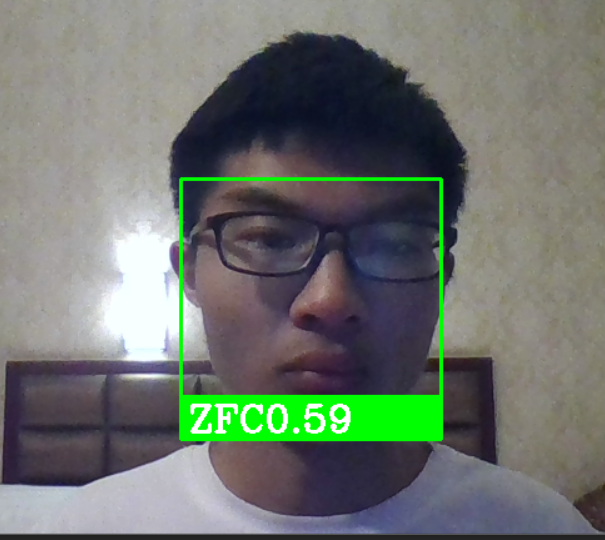


图5.3 检测出相应人名及欧氏距离

实现过程：将每一部分功能进行封装，写成函数方法。然后创建窗体，设置窗体大小及位置；创建对应按钮，使用pack（）方法定位出此button按钮需要放置的合适位置，然后在按钮上使用参数commond附上此按钮所需要实现的功能即可。

win1.title( 'Main View' ) # 添加窗体名称

win1.geometry( '250x250' ) # 设置窗体大小

Button(win1, text = "打卡记录",

command = MarkList).pack( side=tk.RIGHT, padx=10, pady=20 ) # 添加第一个按钮

Button(win1, text = "员工打卡" , command = remark).pack(side=tk.RIGHT, padx=10, pady=20) # 添加第二个按钮

Button(win1, text = "注册面貌", command = register).pack(side = tk.RIGHT, padx = 10, pady = 10) # 添加第三个按钮

win1.mainloop()

# 6 系统测试

## 6.1 环境测试

Python 3.7版本

Opencv 4.5.1.48版本

face-recognition 1.3.0版本

## 6.2 打卡功能测试

打卡功能测试主要时测试能否正确识别出人脸的名字，当我们在摄像头上出现人的面部图片就会进行判断，如果在图像下显示出正确的人名则说明识别正确，如图6.1，正确显示此人名字以及此人脸和库中已注册人脸的相似度。

否则识别失败，失败原因有两种：第一种情况是face库中没有此人的面部；如图6.2，我并没有设置活体检测，但系统仍然读不到这张照片的姓名，这是由于Kobe并没有在face库中注册，因此系统不会显示关于他的名字。

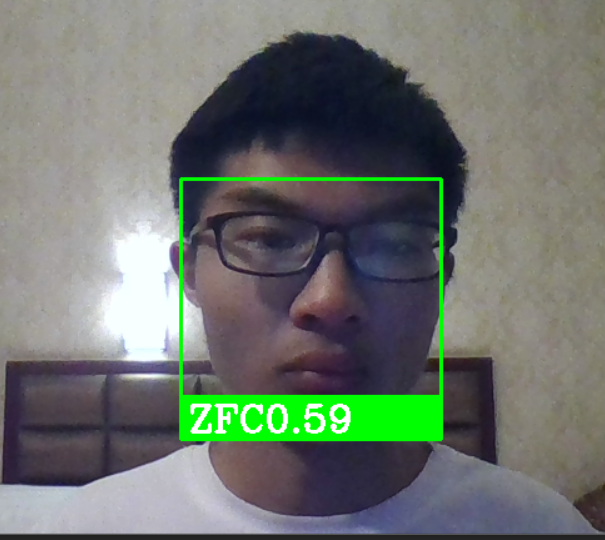


图6.1 打卡成功

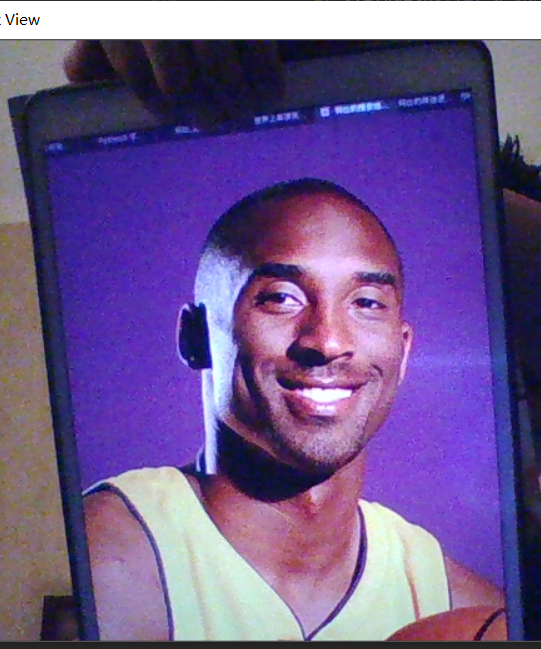


图6.2 未能识别出Kobe

第二种情况则是由于算法不完美，当人脸和库中保存到面部图片差距过大还是会出现失误，出现人脸与姓名不符合。

## 6.3 注册功能的测试

注册功能需要实现的功能是能够从图片中获取到人脸并将其保存在人脸库中。测试时，先查看待注册中的人脸是否已经入库，确认未入库后再进行人脸注册，所以我们先检查库，如图6.2所示

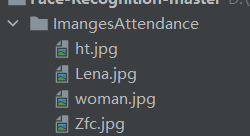


图6.2 未注册前人脸库

然后如图6.3所示进行人脸入库。



图6.3注册科比人脸

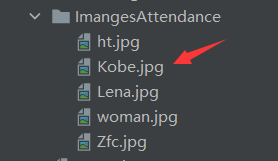
成功之后就会在人脸库中显示已注册人脸，如图6.4，程序运行成功，入库

图6.4 Kobe人脸入库

此时要注意：首先注册人脸时尽量保持只有一个人脸出现在图片中，从而保持之后的人脸识别的准确性。其次在注册时相同名字的人应对名字加以区分，否则库中照片不会重复保存同一名称的图片，而且已有图片不会被更新。

## 6.4 显示模块测试

这一模块需要实现的功能就是将已经打卡人员名单以及打卡时间显示出来，因此显示模块测试时只需要对比签到前和签到后表格中数据，查看是否已经在表格中显示即可。测试前，如图6.5

|  |  |  |
| --- | --- | --- |
| 名次 | 名字 | 时间 |
| NO 3 | WOMAN | 15：13：32 |
| NO 2 | ZFC | 15：07：07 |
| NO 1 | HT | 12：13：32 |

图6.5 目前已打卡人员记录

测试后，如图6.6，显示成功。

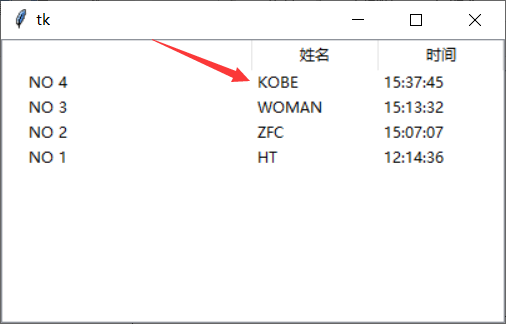


图6.6 kebe打卡结束后，记录更新

# 7 结论

本系统借助python语言以及opencv图像处理技术实现了实时人脸识别打卡系统。在此过程中，我完成了对python语言的了解，学会了运用python语言进行简单的程序编写，并学习了tkinter的相关部件，能够进行图形化编程,美化了界面；同时走到当前计算机视觉的前端，了解到关于人脸识别技术的发展现状以及未来的趋势，学习了一些人脸识别的相关算法。

随着人工智能的不断发展，计算机视觉领域相关的技术愈发成熟，但人们也对人脸识别的实现提出了更高的要求，希望能够完全实现不受外部因素的影响，对人脸做出最精确的判断。这次毕业设计中，系统仅仅实现了识别出人脸这一最基本的要求，离真正使用还有这不小的差距；同时这个考勤系统也需要改进，例如：当有员工离职，因该删除人脸库中该员工的人脸信息；当员工注册时人名写错，能够修改人名，还有对注册功能进行操作时应该加上权限，不能任何人都能对此操作，系统中应加入活体检测，防止员工使用照片打卡等等这些都是能够加以改进的地方。

# 8 参考文献

1. Matthew Turk,Alex Pentland. Eigenfaces for recognition. Journal of Cognitive Neuroscience，1999.2～5

   ［２］Peter N Belhumeur J P Hespanha David Kriegman.Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence，1997.３～４

   ［３］Dong Chen,Xudong Cao,Fang Wen,Jian Sun.Blessing of Dimensionality: High-Dimensional Feature and Its Efficient Compression for Face Verification.computer vision and pattern recognition，2013.

   ［４］Xu Runqin,Lv Pin,Xu Fanjiang,Shi Yishi. A survey of approach

   es for implementing optical neural networks[J]. Optics & Laser Tec hnology,2021,136.

   ［５］盖荣丽,蔡建荣,王诗宇,仓艳,陈娜.卷积神经网络在图像识别中的应用研究综述[J/OL].小型微型计算机系统:1-6[2021-05-16].

   ［６］李炳臻,刘克,顾佼佼,姜文志.卷积神经网络研究综述[J].计算机时代,2021(04):8-12+17.

   ［７］冉蓉,徐兴华,邱少华,崔小鹏,欧阳斌.基于深度卷积神经网络的裂纹检测方法综述[J].计算机工程与应用,2021,57(09):23-35.

   ［８］孟祥泽.基于深度卷积神经网络的图像目标检测算法现状研究综述[J].数字技术与应用,2021,39(01):112-116.

   ［９］Liang Yu,Li Siguang,Yan Chungang,Li Maozhen,Jiang Changjun. Explaining the black-box model: A survey of local interpretation methods for deep neural networks[J]. Neurocomputing,2021,419.

   ［１０］孙德刚.PCA算法的人脸识别技术研究[J].电子世界,2021(06):33-34.

   ［１１］袁旭,迟耀丹,吴博琦,刘安琪,王佳祺.基于PCA算法的人脸识别技术研究[J].信息技术与信息化,2021(03):34-37+41.

   ［１２］余江.改进的LDA行人再识别算法[J].科学技术创新,2020(19):69-70.

   ［１３］Liang Yu,Li Siguang,Yan Chungang,Li Maozhen,Jiang Changjun. Explaining the black-box model: A survey of local interpretation methods for deep neural networks[J]. Neurocomputing,2021,419.

   ［１４］张娜,刘坤,韩美林,陈晨.一种基于PCA和LDA融合的人脸识别算法研究[J].电子测量技术,2020,43(13):72-75.

   ［１５］杨彦荣,李亚雨.基于LDA主题模型的矩阵分解推荐算法[J].数字通信世界,2020(11):27-29.

   # 9 致谢

   本次毕业设计能够成功，首先我要感谢自己的毕业设计指导老师黄革新老师。在我选题之前，黄老师给我提出建议，让我多看看相关书籍；选题时黄老师还给我提出一些建议，帮我确定了方向；代码编写过程中，当我遇到不会的问题询问老师时，他耐心解释，帮助我成功完成此次程序设计。黄老师对待学生细心，耐心的态度让我敬佩。

   其次还要感谢同学的互相帮助，每次在遇到困难时，我询问同学时他们都会耐心解释，帮忙查资料，然后一起动手。让我在错误中积累经验，学习新知识。

   最后，感谢大学期间的任课老师帮助我积累专业知识，才让我能够顺利完成本次毕业设计。 [↑](#endnote-ref-1)