

分 类 号: TP311
研究生学号: 201253H308

单位代码: 10183
密 级: 公 开



吉 林 大 学

硕士学位论文

(专业学位)

基于 OpenCV 的人脸识别系统的设计与实现

THE DESIGN AND IMPLEMENTATION OF FACE RECOGNITION
SYSTEM BASED ON OPENCV

作 者 姓 名: 卜秋月

类 别: 工程硕士

领域(方向): 软件工程

指 导 教 师: 包 铁 副教授

彭 君 讲 师

培 养 单 位: 计算机科学与技术学院

2015 年 6 月

未经本论文作者的书面授权，依法收存和保管本论文书面版本、电子版本的任何单位和个人，均不得对本论文的全部或部分内容进行任何形式的复制、修改、发行、出租、改编等有碍作者著作权的商业性使用（但纯学术性使用不在此限）。否则，应承担侵权的法律责任。

吉林大学硕士学位论文原创性声明

本人郑重声明：所呈交学位论文，是本人在指导教师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：



日期：2015年6月1日

基于 OpenCV 的人脸识别系统的设计与实现

THE DESIGN AND IMPLEMENTATION OF FACE
RECOGNITION SYSTEM BASED ON OPENCV

作者姓名：卜秋月

领域（方向）：软件工程

指导教师：包铁 副教授

彭君 讲师

类别：在职工程硕士

答辩日期：2015 年 05 月 16 日

摘要

基于 OpenCV 的人脸识别系统的设计与实现

随着信息技术的飞速发展及人们安全意识的不断提高,传统的身份识别方法容易被窃取和泄露,已经不足以满足社会的需求。生物特征识别已经逐渐成为当今炙手可热的研究领域。生物特征识别技术主要是运用计算机视觉、图形图像处理、模式识别等技术来提取和描述人类的生理特征或行为特征,从而对个人的身份进行验证。在多种多样的生物识别技术中,人脸识别方式更自然,更直观,而且具有非接触性、非强制性、并发性和简便性等特点,因此吸引了众多研究者对其进行研究,是当今社会研究和应用的热点方向。人脸识别技术的研究起始于 60 年代末期,经过了三个阶段的发展,在各国学者的共同努力下,形成了较为成熟的技术,并在国家安全、证件核实、信息安全、人事考勤等许多领域都得到了广泛应用。

OpenCV 是一个开源跨平台的计算机视觉库,实现了图像处理和计算机视觉方面的很多通用算法。它可以直接应用于很多领域,其中包括物体识别,图像分区,人脸识别,运动跟踪,运动分析等等,还可以作为一个二次开发的理想工具,来实现人脸识别的主要算法。MFC 是一个微软公司提供的类库,以 C++类的形式封装了 Windows API,并且包含了一个应用程序的框架,进而减少了开发人员在编程时的工作量。它可以通过面向对象的方法来调用 Windows API,还可以便捷的实现应用程序的开发,因此本文使用 MFC 来实现人脸识别系统的框架。

目前人脸识别的算法主要有三种:第一种是基于 PCA 的 Eigenface 方法,它的基本思想是通过 K-L 变换对图像进行投影降维,找到图像的主要成分来对人脸进行描述及比对;第二种是基于 LDA 的 FisherFace 方法,它的主要思想是将高维的样本投影到最佳判别向量空间,从而能够有效的提取分类信息、压缩特征空间的维数,在新的子空间形成投影后,类间距离达到了最大和类内距离达到了最小,从而达到最佳的可分离性;第三种是基于局部特征提取的 LBP 方

法, 它的主要思想是对图像中的像素点和该点周围像素点进行对比, 然后根据对比结果进行求和。

本文对三种人脸识别算法进行了研究, 运用 OpenCV 实现了三种算法中计算效率较高的 LBP 算法进行人脸识别。然后运用 Visual Studio 2010 搭建了基于 OpenCV 的人脸识别系统, 用 MFC 设计了用户界面, 实现了人脸模型训练功能、读入待识别图像功能、提取待识别图像功能、识别人脸功能四种功能。本文还运用 ORL 人脸数据库对系统进行了训练和测试, 识别效率高、结果准确。同时, 本文对系统的各个模块和整体流程也进行了测试, 测试结果良好, 实现了完整的基于 OpenCV 的人脸识别系统。

关键词:

人脸识别, OpenCV, LBP

Abstract

The Design and Implementation of Face Recognition System based on OpenCV

With the continuous improvement of information technology and the rapid development of safety awareness, the traditional identification methods which can easily be stolen and leaked have been insufficient to meet the needs of society. Biometric authentication has gradually become a hot area of research. Biometric authentication technology is mainly using computer vision, image processing and pattern recognition techniques to identify a person's identity by physical characteristics or behavioral characteristics of human beings. In a variety of biometric technology, face recognition is more natural and more intuitive. It has many characteristics, such as non-contact, non-mandatory, concurrency and simplicity and so on. So it has attracted many researchers to study it. Nowadays, It has been the hotspot of research and applications in the society. Face Recognition technology, whiche starting in the late 1960s, has formed a relatively mature technology with the joint efforts of scholars from various countries after the development of three phases. It has been widely used in many fields, such as national security, document verification, information security, personnel attendance and so on.

OpenCV is an open source cross-platform computer vision library that implements many common algorithms for image processing and computer vision. It can be directly applied in many fields, including object recognition, image partition, face recognition, motion tracking, motion analysis, etc., and it can also be used as a secondary development of the ideal tool to achieve the main algorithm of face recognition. MFC is a library provided by Microsoft, which encapsulates the

Windows API in the form of a C++ class. It includes a framework for the application, thereby reducing the workload of the developers at the time of programming. It can be invoked through an object-oriented approach to Windows API, and also can achieve a convenient application development. So this paper uses the MFC framework to implement face recognition system.

Currently there are three face recognition algorithms: The first one is Eigenface method based on PCA, which basic idea is to project and reduce the dimensions of the image by K-L transform, and then find the main component of the image to describe and compare the human faces. The second method is FisherFace which based on LDA, its main idea is to project the high-dimensional sample to the best determine vector space, which can effectively extract classified information, compression dimension feature space. In the new projection subspace, the distance between classes has been reached the maximum and the distance in classes has been reached the minimum, so as to achieve the best separability. The third method is LBP which based on local feature extraction method, which basic idea is to compare the image pixels and the points around the pixel, and then summing the results based on comparison.

This paper has studied on the three face recognition algorithm, and then used OpenCV to implement LBP which is the most efficient of the three algorithms of face recognition to recognize faces. Then it uses Visual Studio 2010 to build a face recognition system, and designs user interface by MFC. This paper has realize the functions of training model of face, reading the image that need to be recognized, extracting image and recognizing face. This paper also uses ORL face database system to train and test the system, which has achieved highly effective and accurate recognition. The paper has tested every module of the system and the process of the system. This paper implements a whole face recognition system based on OpenCV.

Keywords:

Face Recognition, OpenCV, LBP

目 录

第 1 章 绪论.....	1
1.1 研究背景和意义.....	1
1.2 国内外研究现状.....	3
1.3 论文研究内容和结构.....	6
第 2 章 系统相关技术介绍.....	7
2.1 OPENCV 简介.....	7
2.2 MFC 简介.....	9
2.3 本章小结.....	10
第 3 章 人脸识别算法.....	11
3.1 EIGENFACE.....	11
3.1.1 K-L 变换.....	11
3.1.2 PCA 方法.....	13
3.2 FISHERFACE.....	15
3.2.1 二类问题的线性判别分析.....	15
3.2.2 多类问题的线性判别分析.....	18
3.2.3 FisherFace 算法流程.....	20
3.3 LBP.....	20
3.3.1 LBP 算法的描述.....	21
3.3.2 基于 LBP 的人脸识别.....	23
3.4 本章小结.....	24
第 4 章 系统设计.....	25
4.1 功能设计.....	25

4.2 模块设计.....	26
4.3 数据库设计.....	28
4.4 本章小结.....	29
第5章 系统实现.....	30
5.1 模块实现.....	30
5.2 算法实现.....	36
5.3 本章小结.....	38
第6章 系统测试.....	39
6.1 算法测试.....	39
6.2 功能测试.....	40
6.3 整体测试.....	41
第7章 总结与展望.....	44
7.1 总结.....	44
7.2 展望.....	45
参考文献.....	46
作者简介.....	49
致谢.....	50

第1章 绪论

1.1 研究背景和意义

随着信息技术的飞速发展及人们安全意识的不断提高，人脸识别领域的研究越来越受到人们的关注。当今社会，信息技术已经渗入人们生活的各个方面，如网上银行、网上购物、在线支付、社交软件等。这些技术为人们带来方便的同时，涉及个人财产和隐私等的信息安全问题也已成为人们关注的热点。传统的身份识别方法，如证件、密码、钥匙等，容易被窃取和泄露，已经不足以满足人们对安全保护的要求。例如一旦个人的银行卡和密码落入不法之徒手中，其很容易在自动取款机上盗取钱财。对于传统的身份识别方法，即使是这样的双重保护也无法达到期望的效果。因此，一种更安全、更可靠、更高效的身份识别技术——生物特征识别技术，逐渐进入人们的视野并迅速得到发展和应用。

生物特征识别技术主要是运用计算机视觉、图形图像处理、模式识别等技术来提取和描述人类的生理特征或行为特征，从而对个人的身份进行验证。每一个人都有不同的生理特征和行为特征，这些特征也都是可以测量的，同样也是可以通过各种技术来自动识别和验证的。这些可以检测和验证的生理特征有：指纹、虹膜、掌纹、人脸特征、DNA、视网膜等，行为特征有声音、笔迹、步态等。对于人类的生物特征，目前已经研究出的生物识别技术包括人脸、指纹、声音等方面的识别。通常来说，识别人类的生物特征技术具有方便性和可靠性，这种识别方式不用人们随身携带各种证件、卡和钥匙，也不用记住繁琐的口令，只需认定这个人本身即可。人脸识别比通过虹膜、指纹等识别方式的识别准确率低，但是它对人体没有侵害性，而且它的识别方式也更自然，更直观，所以人脸识别也就成为了目前比较容易被接受的生物特征识别方式。^[1]

根据目前生物特征识别技术的研究情况来看，人脸识别技术的优势有：

非接触性：在获取人脸图像的过程中，图像采集设备不会接触到用户，这

种方式更易于被用户接受；

非强制性：人脸图像的采集不需要强制用户同意配合，完全可在用户没有注意的情况下获取其面部影像；

并发性：可以同时对个人脸进行采集和识别，更加快速和高效；

简便性：人脸识别对设备的要求不高，只需有相机或摄像头记录下人脸图像即可。

基于以上这些优势，人脸识别技术在许多领域都得到了广泛的应用。人脸识别的主要应用方面如下：

(1) 国家安全。国家稳定持续发展是建立在国家安全的基础上的。但是现在恐怖分子的行为日渐猖狂。人脸识别技术的发展和运用也为维护国家安全起到了一定的帮助作用。人脸识别为公安机关有效和快速地追踪、鉴别犯罪分子提供了技术支持，使得警方可以迅速抓捕犯罪分子，从而保障了国家和社会的和谐稳定。2008 年的北京奥运会就采用了人脸识别技术对参赛人员和观众进行身份验证，从而有效的避免了可能存在的冲突隐患。

(2) 电子商务。随着网络的发展和普及，现今电子商务逐渐出现在人们生活的各个方面。人们经常使用在线支付在各种电商网站进行购物，或者用网上银行进行转账等交易活动。人脸识别可以为用户的身份认证提供技术支持，防止他人盗取用户的钱财，保障了信息安全，让人们能够更放心、更方便的进行在线金融活动。

(3) 访问控制。对于一些要求高安全、高机密的单位，如政府机关、医院、公安机关等场所，可以运用人脸识别技术来建立安全防卫系统，单位人员需进行人脸识别来验证身份后才可进入。同时，在登录个人电脑时，也可用人脸识别来控制是否可以访问该电脑。

(4) 视频监控。在地铁、火车站、机场、超市等人流较大的公共场所，通过遍布的摄像头进行视频监控可以很容易的获取人脸图像，从而进行身份识别，找到要找的人，监测可疑人物，追踪嫌犯轨迹，维护和保障公共安全。对于这一类领域，其他的生物特征识别技术是很难做到较好的监控效果的。

(5) 人机交互。将人脸识别技术运用到机器人或游戏设备中,可以增加趣味性和游戏的真实感,使这些电子产品能够更好地为人们服务。

(6) 查验证件。在需要查验身份证、驾照、护照等身份证件时,肉眼往往很难快速准确的分辨身份是否相符,人脸识别技术则可以比较高效的对证件进行精准的查验,有效减少人力物力的耗费。例如,在我国出入境旅客最多的口岸——深圳罗湖口岸就采用了“深圳—香港生物护照旅客快速通关系统”,这个系统就是通过人脸识别来验证旅客身份,从而提高了通关效率,方便了旅客,保证了出入境的安全性。

(7) 人事考勤。许多工作单位都需要对员工进行考勤,通过人脸识别技术进行考勤可以避免传统考勤制度的仿冒、造假现象。

由此可见,人脸识别技术的研究对保护国家安全和信息安全、推动社会发展有着积极的作用。

1.2 国内外研究现状

目前,人脸识别技术在很多国家和地区都有较为积极的研究和发展。人脸识别技术的研究最早是由 Bledsoe 在一篇技术报告^[2]中提出的,该报告以人脸特征点的距离、比率等参数作为人脸特征,搭建了一个半自动的人脸识别系统。之后,在各国学者的共同努力下,人脸识别技术得到了不停的优化和改进,逐渐发展和成熟,从而在社会生活中得到广泛的应用。

人脸识别技术的研究起始于 60 年代末期,最早在文献^[3] 出现相关研究的描述。而使用计算机来实现的人脸识别技术是最近的二十年才慢慢发展研究起来的,一直到 90 年代时才成为当时研究的一个重点课题,从 1990 年到 1998 年,发表在 EI 检索上的文章便已经达到了数千篇。根据人脸识别技术的发展,可以划分为三个阶段:

第一阶段为从 1964 年到 1990 年这 27 年,人们研究的内容重点着重在人脸几何特征方法的研究^{[2][4][5]}。但是,这种方法对图像质量的要求很高,观察到的特征点比较少,因此这一阶段只是人脸识别的基础研究,不具有普遍的适应性,

检测很容易产生错误，无法推广到实际应用中。

第二阶段为从 1991 年到 1999 年这 9 年，这段时间人脸识别方法研究高速发展，人们开始研究基于代数特征和统计理论的识别方法。1991 年，Pentland 的研究小组提出了 EigenFace（特征脸）方法^[6]。其后所有的人脸识别算法都是以这个方法为基准算法衍生而来的。1997 年，Kriegman 的研究小组提出了 Fisherface 方法^[7]。这种方法的特别之处在于它先要对人脸进行降维，然后再采用线性判别方法对人脸进行分析处理。这一阶段，人脸识别方法产生了巨大进步，解决了特征提取的问题，开始进行了推广应用，但是在光照、姿态等条件下有所局限。

第三阶段为从 2000 年至今，这段时间研究人员继续对人脸识别方法进行探索，他们不再专注于发展新的算法，而是转向对提高人脸识别方法精度的研究。研究者开始研究消除光照、姿态、复杂背景等对人脸识别方法结果的影响。例如，2001 年，Shashua 提出了一种针对特定对象集合学习的绘制技术从而消除光照的影响^[8]，有效地提高了人脸识别的准确性。

国内对人脸识别技术的研究开始的较晚，在 90 年代后期才开始进行深入的研究，但是在众多研究组织的积极努力下，已经取得了卓越的成果^{[9][10][11]}，如我国的清华大学、北京大学、哈尔滨工业大学、上海交通大学、南京理工大学等高校以及中国科学院自动化研究所^[12]。在人脸识别的肤色检测方面，清华大学改进了识别方法，对人脸的颜色和特征进行优化，提出一种具有自动适应能力的检测方法^[13]。而哈尔滨工业大学则是采用特征子脸、模板匹配、彩色信息等人脸检测技术，实现了一个多级结构的人脸检测与跟踪，这个技术即便是在复杂变化的背景下，对多角度、多表情的人脸也能进行识别，并实现跟踪^[14]。在人脸识别技术的研究中，中国科学院自动化研究所的研究实验室研究出的识别模式对国内智能监控识别的发展起到了重要作用，尤其对运动目标的跟踪识别、人脸智能识别以及分析人体行为识别是重要的研究课题^[15]，研究成果也被广泛的应用。

目前国内外研究较为广泛的人脸识别方法有：（1）基于脸部几何特征的人

脸识别方法；(2) 基于特征脸的人脸识别方法；(3) 基于神经网络的人脸识别方法；(4) 基于局部特征的人脸识别方法；(5) 基于弹性匹配的人脸识别方法等。

基于几何特征的人脸识别方法是通过提取人的眼睛、眉毛、鼻子、嘴等脸部器官的几何形状作为识别特征的^{[16][17]}。世界上千千万万的人，每个人的脸也都是有各自的差异，这些差异都可以通过眼睛、眉毛、鼻子、嘴巴等器官的位置和形状来表现，因此这些器官的几何描述就可以作为人脸识别的一个判断标准。集合特征的人脸识别方法最初时，是通过人脸侧面的轮廓来描述和识别人脸，而后随着计算精度的提高，可以对眼睛、嘴巴等细节位置的特征点进行提取，从而达到更高的识别率。

基于特征脸的人脸识别方法是训练一组标准人脸图像，利用主成分分析方法构造主元子空间，最终这些主元便可以具有人脸的特征，这些主元也就被称为特征脸。它是一种比较简单、高效而且非常实用的算法，是基于变换系数特征实现的。但是这种算法对于训练数据库和待识别图像的灰度变换有较大的影响，这种方法在光照和表情变化较小的情况下性能比较好，但是受光照影响较明显，在光照等变化较大时识别性能会有所下降^{[18][19]}。

基于神经网络的人脸识别方法则是将图像空间投影到隐层子空间，根据不同的目的构造不同的神经网络^{[20][21]}。基于神经网络的人脸识别方法在人脸识别领域的应用中，相比其他几类算法是具有很大优越性，其他几类算法大部分都是对人脸识别思想的规律或规则进行显性的描述，而这种描述相对来说更难而且更不精确，神经网络的方法则是利用大量的学习，得到对人脸识别规律或规则的隐性表达，这种隐性表达的适应性相对较强，而且也比较容易实现。

基于局部特征分析的人脸识别方法是根据人脸的图像中表现出的局部特征传达出人脸信息^[22]。提取图像的特征，如果是使用整体的方法，数据是当作空间的高维向量，这种的识别效果会不太理想。如果仅仅是使用局部特征来描述人脸图像，则可以获得一个低维隐式的特征表达方式，这种方法的图像表示不仅仅会随着光照变化而变化，还对图像的尺度变化、形变、旋转等具有较强的

稳定性。

基于弹性匹配的人脸识别方法是对人脸图像进行建模，创建一个二维或三维的网格，将人脸模型匹配的问题转变成可变形曲面的弹性匹配问题。弹性匹配算法是利用了有限元分析进行曲面变形，从而将人脸进行匹配^{[23][24]}。

1.3 论文研究内容和结构

本文对基于 OpenCV 的人脸识别系统进行了设计与实现。首先利用 OpenCV 图像处理开发库实现了 Eigenface、FisherFace、LBP 三种人脸识别方法，使用 ORL 标准人脸数据库进行了训练，并对这三种算法进行了研究和实验测试，最终选用 LBP 算法来进行人脸识别。然后基于 Visual Studio 2010 进行系统搭建，并使用 MFC 来设计和实现系统的用户界面，最终实现完整的人脸识别系统。

第一章 绪论。介绍了人脸识别技术研究的背景和意义，以及国内外对人脸识别技术的研究和发展现状。同时还对论文研究的主要内容和总体结构进行了详细的介绍。

第二章 系统相关技术的介绍。对 OpenCV 进行了简单的介绍，并给出了它的优点和功能，还对 MFC 以及其主要成员类进行了简单的介绍。

第三章 人脸识别算法。介绍了 EigenFace、FisherFace 和 LBP 三种人脸识别算法的原理和各自的优缺点，并对它们进行了比较。

第四章 系统设计。介绍了系统的功能设计、模块设计以及数据库设计，对系统要实现的具体功能进行了相关介绍。

第五章 系统实现。对具体的模块实现和算法实现的方法进行了详细的介绍。

第六章 系统测试。对系统的算法、功能和整体方面进行了相关测试，证明了系统的可靠性。

第七章 结论与展望。对本文进行的工作进行了总结，并介绍了下一步的研究方向和工作重点。

第 2 章 系统相关技术介绍

2.1 OpenCV 简介

OpenCV^[25]的全称是 Open Source Computer Vision Library, 它是 Intel 公司提出并参与开发的开源跨平台, 是一个计算机视觉库, 在 Linux、Windows 和 Mac OS 操作系统上都能够运行。OpenCV 是用 C++ 语言进行编写的, 它对外提供的接口大部分也是 C++ 形式的, 但在 OpenCV 中保留了相当数量的 C 语言接口, 同时在库中还保留了相当数量的 Python, Java and MATLAB/OCTAVE (版本 2.5) 接口, 是一个轻量级且高效的开源库。

OpenCV 针对的主要是实际的应用, 并且它通过优化 C 语言代码的编写, 在很大程度上提高了程序的运行速度, 此外, 用户如果想进一步提高其处理速度, 可以购买 Intel 公司的 IPP 高性能多媒体函数库 (Integrated Performance Primitives) 来进一步提高其处理速度, 从而可以在很大程度上提高程序或系统的运行效率。OpenCV 将大部分图像处理、计算机视觉等方面的算法都进行了代码实现, 作为一个基本的计算机视觉、图像处理和模式识别的开源项目, OpenCV 在很多领域中得到了广泛应用, 例如物体识别, 图像分区, 人脸识别, 运动跟踪, 运动分析等等, 同时 OpenCV 还可以作为一个二次开发的理想工具, 来实现人脸识别系统的主要算法。

OpenCV 的特点主要包括:

- OpenCV 是一个开源的图像处理函数库, 它是基于 C/C++ 语言的;
- 其代码都是经过层层优化的, 可实现图像的实时处理;
- 具有良好的可移植性;
- 可以进行图像、视频等形式的读取、拷贝、保存等常规操作;
- 提供各种应用程序接口 (API);
- 提供面向英特尔 IPP 高效多媒体函数库的接口, 可以针对用户所使用的英特尔 CPU 对代码进行优化, 从而提高系统的性能。

OpenCV 的功能包括:

- 图像数据的基本操作,如内存申请和释放,图像的拷贝、缩放、旋转等;
- 图像、视频的输入输出,OpenCV 不仅具有对本地文件或摄像头捕捉到的图像进行输入的功能,还具有输出和显示图像和视频的功能;
- 矩阵、向量数据的操作以及在线性代数上的运算,OpenCV 完全能够实现计算矩阵乘积、矩阵方程特征值,并能够求解矩阵方程;
- 支持多种动态数据结构,OpenCV 能够支持的数据结构包括了链表、队列、数据集、树、图等形式;
- 基本图像处理,OpenCV 实现了图像去噪、图像边缘检测、图像像素点的采样和插值、图像色彩变换、腐蚀和膨胀等形态学处理、图像直方图等等;
- 结构分析,OpenCV 提供了图像连通域、霍夫变换、图像轮廓处理、连通分支的分析、多项式的逼近、图像的模板匹配、曲线拟合、狄劳尼三角化等多种结构分析形式;
- 摄像头定标,OpenCV 可以通过摄像头输入对目标进行寻找和跟踪、参数定标、基本矩阵估计、立体视觉匹配等;
- 运动分析,OpenCV 具有运动分析的功能,对光流、动作分割、目标跟踪等能够进行分析;
- 目标识别,OpenCV 提供了多种特征提取方法、HMM 模型;
- 基本的 GUI,通过 OpenCV 可以显示图像和视频、支持键盘、鼠标、滑动条的操作;
- 图像标注,OpenCV 可以实现直线、曲线、多边形、文本等各种形式的图像标注。

从 OpenCV2.4 开始,OpenCV 中添加了新的类——FaceRecognizer,目前 OpenCV 提供了三种人脸识别的方法:基于 PCA 的特征脸算法——EigenFace、基于 LDA 的 Fisher 脸算法——FisherFace,以及基于局部特征的人脸识别算法——LBPHFace。通过这些算法,我们可以较为便捷的实现人脸识别算法的实验和测试。

2.2 MFC 简介

MFC^[26]的全称是 Microsoft Foundation Classes (微软基础类库), 是一个微软公司提供的类库, 以 C++ 类的形式封装了 Windows API, 并且包含了一个应用程序的框架, 进而减少了开发人员在编程时的工作量。MFC 中包含了大量的 Windows 句柄封装类, 以及 Windows 的内建控件和组件的封装类。

MFC 可以通过面向对象的方法来调用 Windows API, 还可以便捷的实现应用程序的开发。MFC 将很多应用程序开发过程中所需的功能进行了自动化处理, 并提供了自定义的应用程序框架, 如文档框架视图结构和活动文档, 可以大幅度的缩减软件开发时间。

MFC 编程方式可以实现三种类型的应用程序: 单文档界面应用程序——SDI、多文档应用程序——MDI 以及基于对话框的应用程序——Dialog Based。MFC 形式的应用程序大部分都包含四个基本类: CMYAPP、CMAINFRAME、CMYDOC、CMYVIEW, 通过 MFC 类库中的各个类可以快速的生成面向对象形式的应用程序, 这些类之间也存在着相应的联系, 这些类的集合便构成了 MFC 应用程序的基本框架。

四个基本类成员的功能如下:

- 类 CMYAPP

类 CMYAPP 由 CWINAPP 类派生而来, 它是 MFC 应用程序运行的基础, 在 MFC 应用程序启动时, 会调用 CMYAPP 类当中的 InitInstance() 函数, InitInstance() 函数实现 MFC 应用程序的初始化, 创建文档模板, 创建应用程序的主窗口。

- 类 CMAINFRAME

类 CMAINFRAME 由 CFRAMEWND 类派生而来, 它是 MFC 应用程序的主框架。对于一个 Windows 应用程序来说, CMAINFRAME 所管理的范围包括程序主窗口中除了白色部分的视图外的所有其他部分, 在 MFC 应用程序中, 我们一般不会对 CMAINFRAME 类进行修改。

- 类 CMYDOC 和 CMYVIEW

类 CMYDOC 由 CDocument 类派生而来，类 CMYVIEW 由类 CWnd 派生而来。这两个类是有很强关联的，在一个 MFC 的应用程序中，文档对象 CMYDOC 一般表示一个已经打开的文件，用来管理数据；视图对象 CMYVIEW 则按特定的需求来显示当前文档中的数据，也就是对文档中的数据进行可视化。

2.3 本章小结

本章主要介绍了本文中所使用的相关技术，首先介绍了什么是 OpenCV，OpenCV 的主要特点和功能，以及本文系统中所使用的 OpenCV 的算法和功能；其次介绍了搭建本文中提出的人脸识别系统框架所使用的 MFC 技术，以及 MFC 中四个基本类的主要功能。基于 OpenCV 可以实现效率较高的人脸识别的基础算法，而基于 MFC 可以比较简单的实现人脸识别系统的框架，本文基于这两项技术来实现整体的人脸识别系统。

第3章 人脸识别算法

3.1 EigenFace

特征脸——Eigenface 就是一组特征向量，它应用于机器视觉领域来解决人脸识别的问题。1987 年，Sirovich 和 Kirby 首先提出使用特征脸进行人脸识别的方法，然后由 Pentland 研究小组将其用于人脸分类^[4]。特征脸是第一种能够有效地进行人脸识别的方法。图像问题是一个高维问题，二维的 $p \times q$ 的灰度图像，就是一个 $m = p \times q$ 维的向量空间。但是，并不是所有维度都是对我们有用的，因此我们需要找出主要的成分来表示大部分的图像信息。于是，Karl Pearson 和 Harold Hotelling 就提出了主成分分析（Principle Component Analysis, PCA）方法，将可能相关的变量转换成更小的不相关的子集，使得一个高维的数据集就能够用一些有意义的维度来表示出绝大部分的信息。

PCA 方法是一种以变量的协方差阵为基础，能够有效地对样本的信息数据进行处理、压缩和提取的方法。在方法的应用上，主要是通过寻找出数据主轴的方向，在找出的数据主轴上构成一个新的坐标系，这个新的坐标系维数于原坐标系的维数相比要低，然后把数据通过投影，从原坐标系投到新坐标系上，这个投影的过程即降维过程。在人脸识别领域，PCA 方法的基本流程就是先运用 K-L (Karhunen-Loeve) 变换从人脸的样本数据库中提取出人脸的主要特征，这些特征就构成了相应的特征脸子空间，然后将待识别的人脸图像也投影到这个特征脸子空间，将得到的这组投影系数与人脸数据库中的人脸的投影系数进行比较，从而识别出相应的人脸。PCA 方法能够保留数据的主分量（Principal Components），即原数据向量在其协方差矩阵的最大特征值所对应的特征向量方向上的投影，因此而得名为主成分分析。下面介绍 PCA 算法的基本原理和具体的实现过程。

3.1.1 K-L 变换

K-L 变换是一种以统计特性为基础的正交变换。它的优点是具有良好的去

相关性，是均方误差(Mean Square Error, MSE)意义下的最佳变换。

设 $X = (X_1, X_2, X_3, \dots, X_n)^T$ 为 n 维随机向量。它的均值向量为： $m_X = E\{X\}$ ，协方差矩阵为： $C_X = E\{(X - m_X)(X - m_X)^T\}$ 。

其中， $(X - m_X)(X - m_X)^T$ 为 $n \times n$ 阶的矩阵， C_X 的元素 C_{ii} 是 X 向量的第 i 个分量的方差， C_X 的元素 C_{ij} 是 X 向量的元素 X_i 和 X_j 的协方差。因为矩阵 C_X 是实对称的，所以如果向量中的元素 X_i 和 X_j 无关，则它们的协方差 $C_{ij} = C_{ji} = 0$ 。

我们可以从总体中随机抽取 k 个向量作为样本，则通过样本对总体的均值向量进行估计的公式如下：

$$m_X = \frac{1}{K} \sum_{k=1}^K X_k$$

通过样本对协方差矩阵进行估计的公式如下：

$$C_X = \frac{1}{K} \sum_{k=1}^K X_k X_k^T - m_X m_X^T$$

K-L 变换定义了一个将 X 向量映射到 n 维向量 Y 的正交变换 $A \in R^{n \times n}$ ，它能够使 Y 向量中的各个元素互不相关。因此 K-L 变换式为：

$$Y = A(X - m_X)$$

对于随机向量 Y ，它的均值向量为： $m_Y = E\{Y\}$ ，协方差矩阵为： $C_Y = E\{(Y - m_Y)(Y - m_Y)^T\}$ 。其中， C_Y 应为对角矩阵，即：

$$C_Y = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n \end{bmatrix}$$

因为 $C_Y = AC_X A$ ，所以计算 C_Y 的过程就相当于对 C_X 进行对角化，所以变换矩阵 A 的每一行应由 C_X 的特征向量组成，并且这些特征向量的顺序是按它们对应的特征值大小进行降序排序的，即：

$$A = \begin{bmatrix} \mathbf{e}_{11} & \mathbf{e}_{12} & \cdots & \mathbf{e}_{1n} \\ \mathbf{e}_{21} & \mathbf{e}_{22} & \cdots & \mathbf{e}_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{e}_{n1} & \mathbf{e}_{n2} & \cdots & \mathbf{e}_{nn} \end{bmatrix}$$

其中, \mathbf{e}_{ij} 为第 i 个特征向量的第 j 个分量。由此可知, $\lambda_1, \lambda_2, \dots, \lambda_n$ 就是 C_X 的特征值的降序排列。因此, C_X 与 C_Y 有相同特征值和特征向量。经过 A 的正交变换后, Y 的各个元素互不相关, 这就构成了对 X 的去相关操作。

K-L 变换还可以进行反变换, 由 Y 变换为 X 。因为 A 的各行是正交向量, 则有 $A^{-1} = A^T$ 。所以 K-L 反变换式为:

$$X = A^T Y + m_X$$

如果将 C_X 的最大的 K 个特征值所对应的 K 个特征向量组成维数为 $K \times n$ 的变换矩阵 A_K , 此时变换后的 Y 的维数会降为 K 维, 则 K-L 反变换式为:

$$\hat{X} = A_K^T Y + m_X$$

X 与 \hat{X} 间的均方误差的计算公式如下:

$$\text{MSE} = \sum_{j=1}^n \lambda_j - \sum_{j=1}^K \lambda_j = \sum_{j=K+1}^n \lambda_j$$

如果将所有的特征向量都用于正交变换中, 即 $k=n$, 则误差为零。因为特征值 λ_j 是递减的, 因此在计算时可以选择 K 个最大的特征值所对应的特征向量来组成变换矩阵从而降低误差。因为 K-L 变换可以将均方误差降至最小, 所以它是均方误差意义下的最佳变换。

3.1.2 PCA 方法

PCA 方法在人脸识别领域应用的具体步骤如下:

1. 将训练样本归一化。将大小为 $m \times n$ 的人脸图像进行重新排列, 使其成

为 mn 维列向量 X 。例如，图像矩阵为 $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ ，则将其排列为列向量 $\begin{bmatrix} 1 \\ 4 \\ 7 \\ 2 \\ 5 \\ 8 \\ 3 \\ 6 \\ 9 \end{bmatrix}$ 。计

算向量 X 的均值向量 μ ，计算公式如下：

$$\mu = \frac{1}{N} \sum_{i=1}^N X_i$$

然后运用均值向量将所有样本中心化。

2. 计算中心化后的样本向量的协方差矩阵。协方差矩阵计算公式如下：

$$S = \frac{1}{N} \sum_{i=1}^N (X_i - \mu)(X_i - \mu)^T$$

然后计算特征值 λ_i 和对应的特征向量 v_i ，如下式：

$$Sv_i = \lambda_i v_i, \quad i = 1, 2, 3, \dots, N$$

对特征值进行降序排列，并让特征向量和它的顺序一致。

3. 将第 2 步所得的 K 个最大的特征值对应的特征向量组成变换矩阵 A ，这 K 个特征向量就是主成分。然后运用 K-L 变换式 $Y = A(X - \mu)$ 对其进行变换投影，得到训练样本图像的降维向量 Y 。

4. 运用特征脸进行人脸识别的方法如下：

- (1) 把所有的训练数据投影到 PCA 子空间；
- (2) 把待识别的人脸图像投影到 PCA 子空间。

找到与待识别图像投影后的向量最接近的训练数据投影后的向量，从而识别出相应的人脸。

3.2 FisherFace

PCA 方法能够达到很好地降维和去相关效果，但是它不能利用已知的人脸图像样本的类别信息，属于无监督方法。统计学家 Sir R.A.Fisher^[27]在 1936 年最早提出了线性判别分析（Linear Discriminant Analysis, LDA）的思想，他成功对花进行了分类，找到了样本的最佳投影方向，使样本在投影后，类间的离散度能够达到最大，类内的离散度能够达到最小，但在当时这种算法仅适用于两类分类问题。因为经典的 LDA 采用的是 Fisher 判别准则函数，因此也称其为 Fisher 线性判别(Fisher Linear Discriminant, FLD)。在 Fisher 线性判别思想的基础上，Wilks^[28]和 Duda^[29]在判别向量集的概念上分别提出了自己的看法，也就是寻找判别向量构成子空间的一组数据，鉴别特征是使用原始的样本在子空间内投影向量，并用在识别上，使得 LDA 可以解决多类分类问题。1997 年，Belhumeur, Hespanha 和 Kriegman^[30]把 LDA 引入到人脸识别问题中。

LDA 的基本思想是将高维的样本投影到最佳判别向量空间，从而能够有效的提取分类信息、压缩特征空间的维数，在新的子空间形成投影后，类间距离达到了最大和类内距离达到了最小，从而达到最佳的可分离性。因此，LDA 是一种有效的特征提取方法，它能够使样本投影后的类间散布矩阵最大，类内散布矩阵最小。

3.2.1 二类问题的线性判别分析

我们首先考虑比较简单的二类问题。给定一组属于两个类的 N 个特征为 d 维的样本 $x^{(i)} \{x_1^{(i)}, x_2^{(i)}, \dots, x_d^{(i)}\}$ ($i = 1, 2, \dots, N$)，其中有 N_1 个样本属于类 ω_1 ，剩余 N_2 个样本属于类 ω_2 。由于原始数据特征数太多，我们需要将 d 维特征降到只有一维，且这一维能够决定样本的类别，因此要找到一个最佳向量 $w \in R^d$ 。相应的投影函数为：

$$y = w^T x$$

我们通过投影后的 y 的值即可判别 x 的类别。

假设 x 的维数 $d = 2$ 时，我们就是要找一条方向为 w 的直线来做投影。 X 的两种投影方案如图 3.1，其中红色和蓝色分别代表两种不同的类别。

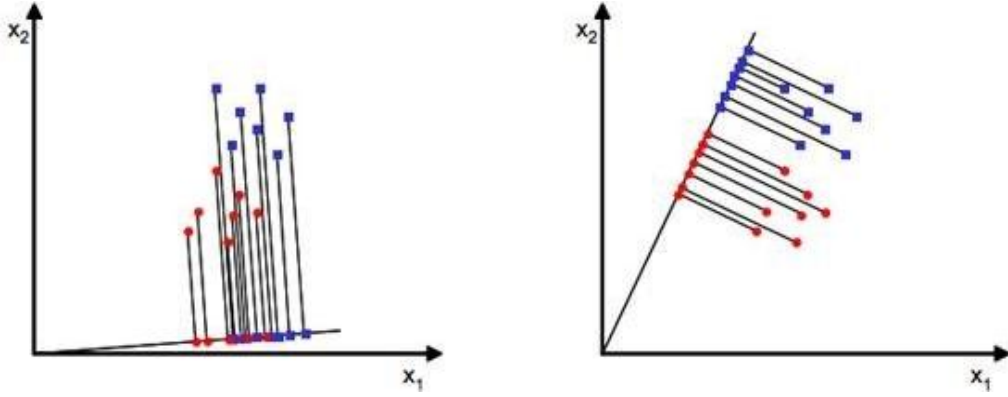


图 3.1 二维样本集的两两种投影方法比较

从图 3.1 中我们可以看出右边方案的投影点能够很好地区分出两种类别，优于左边方案。由此可见我们要通过计算来寻找出这个最佳向量 w 。

首先，计算每一类样本的均值（中心点），计算式为：

$$\mu_i = \frac{1}{N_i} \sum_{x \in \omega_i} x$$

其中 i 是样本的分类数。

样本 x 投影到 w 上的均值，即中心点为：

$$\tilde{\mu}_i = \frac{1}{N_i} \sum_{y \in \omega_i} y = \frac{1}{N_i} \sum_{x \in \omega_i} w^T x = w^T \mu_i$$

最佳向量 w 的判断标准之一就是要使投影后的两类样本点尽量分离。因此，可以定义一个度量值如下式：

$$J(w) = |\tilde{\mu}_1 - \tilde{\mu}_2| = |w^T(\mu_1 - \mu_2)|$$

$J(w)$ 表示不同分类的投影中心的距离，这个值越大分类效果越好。

最佳向量 w 的判断的另一个标准为同一个分类的投影点尽量聚合。因此，我们在考虑不同分类投影中心的同时还要考虑样本点之间的方差，方差越大，样本点越难分离。我们可以定义另一个度量值如下式：

$$\tilde{s}_i^2 = \sum_{y \in \omega_i} (y - \tilde{\mu}_i)^2$$

上式称为散列值 (scatter)，它表示同一个分类经过投影之后样本点的密集程度，密集程度越小表示投影点聚合度越高。因此，散列值越小越好。

我们可以结合这两个度量，第一个度量做分子，第二个度量做分母，形成最终的度量公式如下：

$$J(w) = \frac{|\widetilde{\mu}_1 - \widetilde{\mu}_2|^2}{\widetilde{S}_1^2 + \widetilde{S}_2^2}$$

由上式可知 $J(w)$ 的值越大，向量 w 的性能也就更好。因此，我们只需求解使 $J(w)$ 值最大的 w 即可。

散列值公式的展开式为：

$$\widetilde{S}_i^2 = \sum_{y \in \omega_i} (y - \widetilde{\mu}_i)^2 = \sum_{x \in \omega_i} (w^T x - w^T \mu_i)^2 = \sum_{x \in \omega_i} w^T (x - \mu_i)(x - \mu_i)^T w$$

上式的中间部分即可定义为散列矩阵，如下式：

$$S_i = \sum_{x \in \omega_i} (x - \mu_i)(x - \mu_i)^T$$

类内散列矩阵为：

$$S_w = S_1 + S_2$$

从而替换可得：

$$\widetilde{S}_1^2 + \widetilde{S}_2^2 = w^T S_w w$$

类间散列矩阵为：

$$S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$$

从而替换可得：

$$(\widetilde{\mu}_1 - \widetilde{\mu}_2)^2 = (w^T \mu_1 - w^T \mu_2)^2 = w^T (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T w = w^T S_B w$$

因此， $J(w)$ 最终可表示为：

$$J(w) = \frac{w^T S_B w}{w^T S_w w}$$

这个式子就是 Fisher 准则函数。

将上式分母归一，利用拉格朗日乘数法求导化简得：

$$w = S_w^{-1}(\mu_1 - \mu_2)$$

由此可知，我们只需求出样本的均值和方差就可计算出最佳向量 w 。

3.2.2 多类问题的线性判别分析

多类问题的线性判别分析是两类问题的扩展。对于有多种分类的问题，将样本向量降到一维已经不足以满足要求，投影向量的维数应该有所增加。

假设共有 C 个类别，投影向量的维数为 K ，则投影向量可表示为 $W = [w_1|w_2|\dots|w_K]$ ，其中 W 为 $d \times K$ 维的矩阵。

因此，样本 x 在向量 W 上的投影可表示为：

$$y = W^T x$$

与二类问题相同，我们依然用投影后的类间散列度和类内散列度来衡量是否为最优的投影向量 W 。

假设 x 的维数 $d = 2$ ，类别数 $C = 3$ 时，类间散列度和类内散列度的几何表示如图 3.2 所示。

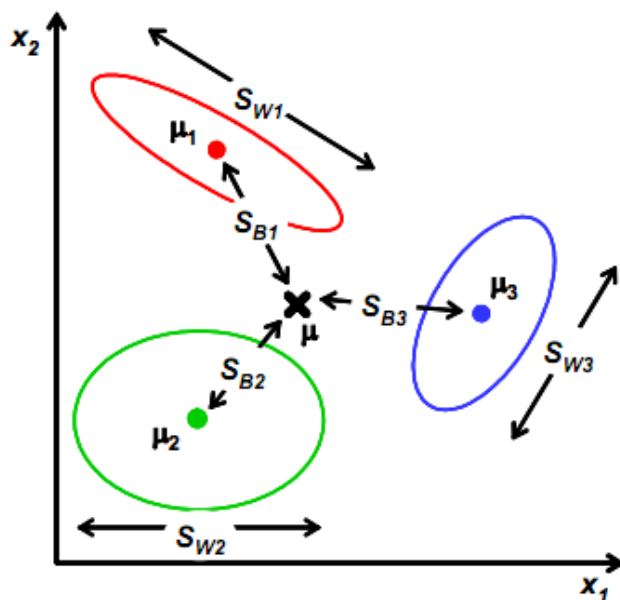


图 3.2 多类问题下类间散列度和类内散列度的几何表示

图 3.2 中的 μ_1 ， μ_2 ， μ_3 分别表示三个类别的中心点， μ 表示样本的总的中心点。类内散列矩阵 S_W 表示为：

$$S_W = \sum_{i=1}^C S_{w_i}$$

其中， S_{w_i} 相当于类 ω_1 中的样本点相对于该类中心 μ_i 的散列程度，计算公式如下：

$$S_{w_i} = \sum_{x \in \omega_i} (x - \mu_i)(x - \mu_i)^T$$

类间散列矩阵 S_B 度量的是每类中心点相对于样本的总的中心点的散列情况，因此， S_B 表示为：

$$S_B = \sum_{i=1}^C N_i (\mu_i - \mu)(\mu_i - \mu)^T$$

上面的式子均为投影之前的情况，而 $J(w)$ 度量的是投影之后的情况。投影之后每类的中心点、样本的总中心点、类内散列矩阵、类间散列矩阵的计算公式分别如下：

$$\tilde{\mu}_i = \frac{1}{N_i} \sum_{y \in \omega_i} y$$

$$\tilde{\mu} = \frac{1}{N} \sum_{\forall y} y$$

$$\widetilde{S}_W = \sum_{i=1}^C \sum_{y \in \omega_i} (y - \tilde{\mu}_i)(y - \tilde{\mu}_i)^T$$

$$\widetilde{S}_B = \sum_{i=1}^C N_i (\tilde{\mu}_i - \tilde{\mu})(\tilde{\mu}_i - \tilde{\mu})^T$$

因此，准则函数 $J(w)$ 可表示为：

$$J(w) = \frac{\widetilde{S}_W}{\widetilde{S}_B} = \frac{|w^T \widetilde{S}_B w|}{|w^T \widetilde{S}_W w|}$$

最后，可化简为下式：

$$S_W^{-1} S_B w_i = \lambda w_i$$

因为 S_W 为非奇异矩阵，所以最佳投影向量 W 就是矩阵 $S_W^{-1} S_B$ 所对应的特征

向量。只需先求出 $S_W^{-1}S_B$ 的特征值，然后按照特征值的降序排序，取前 K 个特征向量即可组成最佳投影向量，将样本由 d 维空间投影至 $C-1$ 维空间。

3.2.3 FisherFace 算法流程

利用 FisherFace 进行人脸识别的具体流程如下：

1. 将所有的大小为 $m \times n$ 的人脸图像进行重新排列，使其成为 mn 维列向量 x_i ，然后将所有的 N 幅图像组成一个大小为 $mn \times N$ 的矩阵。
 2. 计算每个类的样本的均值 $\mu_i = \frac{1}{N_i} \sum_{x \in \omega_i} x$ ，其中 $i = 1, 2, 3, \dots, C$ 。
 3. 计算所有样本的总的均值 $\mu = \frac{1}{N} \sum_{i=1}^C N_i \mu_i$ 。
 4. 计算类内散列矩阵 $S_W = \sum_{i=1}^C S_{W_i} = \sum_{i=1}^C \sum_{x \in \omega_i} (x - \mu_i)(x - \mu_i)^T$ 。
 5. 计算类间散列矩阵 $S_B = \sum_{i=1}^C N_i (\mu_i - \mu)(\mu_i - \mu)^T$ 。
 6. 求出矩阵 $S_W^{-1}S_B$ 的特征值和特征向量，对特征值进行降序排列，取前 $C-1$ 个特征向量组成最佳投影向量 W 。
 7. 利用投影函数 $y = W^T x$ ，将训练样本投影到 LDA 子空间，并将 y 的值存储到特征库中。
 8. 将训练样本投影到 LDA 子空间，计算出投影后的向量 y 。
- 利用分类器对样本进行判别分类。

3.3 LBP

Eigenface 算法和 Fisherface 算法都是通过整体特征的方法来实现人脸识别的，在这些算法中数据可以当作是图像空间的高维向量，但是高维向量的数据在计算效率等方面是比较差的，因此如果是可以使用一个低维子空间的数据来进行处理，其效率以及数据保存等方面都会有所改善。

在我们的日常生活中，我们不太可能在一样的光照下获取图像，或者说不太可能一个人有十张处于完全相同的环境中的照片。所以，如果一个人只有一张训练图像，而待识别图像又与训练图像所处的环境差异很大，那么通过子空

间的协方差估计方法所得到的结果可能就会出现错误，因此识别结果也就不正确。图 3.3 是 Eigenface 算法和 Fisherface 算法在 ORL 人脸数据库上的识别率：

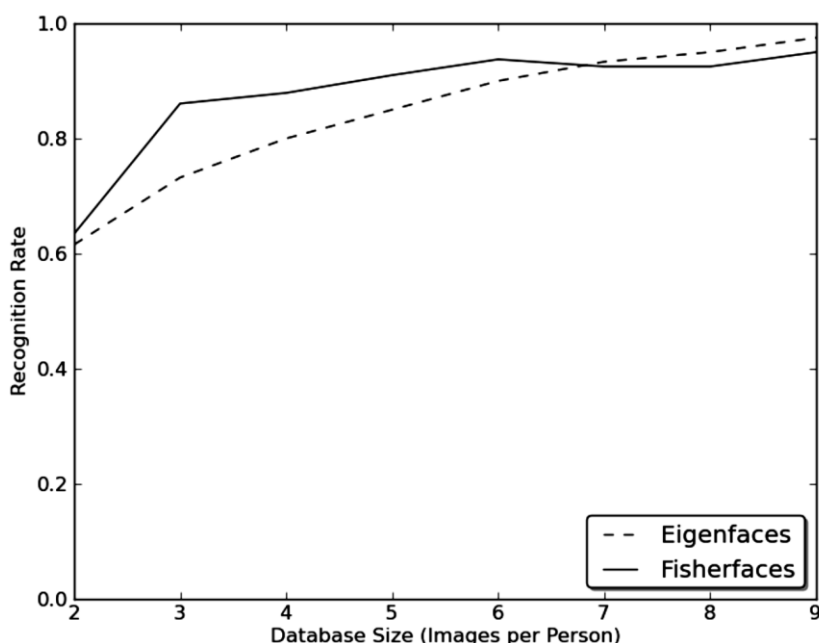


图 3.3 Eigenface 和 Fisherface 在 ORL 人脸数据库上的识别率

因此，想要得到较好的识别率，对于每一个人的训练图像最好是能有 7—9 张。

如果不再按照图像整体特征的思路，而是将人脸图像分块，分别获取其某一部分的局部特征，然后通过这些局部特征来描述一个人脸，这种人脸图像描述方法对光照变化会相对来说更为敏感，而且还对图像中的尺度变化、形变、旋转等变化也会更健壮，这也就是 LBP 算法。

3.3.1 LBP 算法的描述

局部二值模式（LBP, Local Binary Patterns）算法，是一种有效的纹理描述算法，具有旋转不变性和灰度不变性等特点。算法的基本思想是对图像中的像素点和该点周围像素点进行对比，然后根据对比结果进行求和。初始时，LBP 的定义在一个 3×3 的邻域内，以邻域的中心像素点为参考点，将相邻的 8 个像素点的灰度值与中心点进行比较，如果像素值大于中心点像素值，则该像素点

位置标记为1, 否则标记为0, 我们用一个二进制数来表示每个像素, 如 10101001, 对于这种周围有8个像素点的中心点, 最终能够得到 2^8 种不同的组合, 这个组合就是局部二值模式, 或被称为 LBP 码。这个值就用来反应该区域的纹理信息, 用公式表示则为:

$$LBP(x_c, y_c) = \sum_{p=0}^{p-1} 2^p s(i_p - i_c)$$

其中 (x_c, y_c) 是 3×3 邻域的中心点元素, 它的像素值为 i_c , i_p 是邻域内其他像素的值, $s(x)$ 是符号函数, 定义为:

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{else} \end{cases}$$

这种描述方法使得我们可以很好的捕捉到图像中的细节, 但是这种方法有一个缺点, 它只能对一个较小且半径固定的区域进行描述, 不能更好的适应不同大小, 不同频率纹理的需求。因此这种算法就会在图像邻域不变但变换邻域范围时失效。所以, 对这种方法进行改进, 使用一个变量的扩展方法^[31], 将 3×3 的邻域扩展到任意邻域, 用圆形邻域代替正方形邻域, 这样可以捕捉到如图 3.4 的近邻:

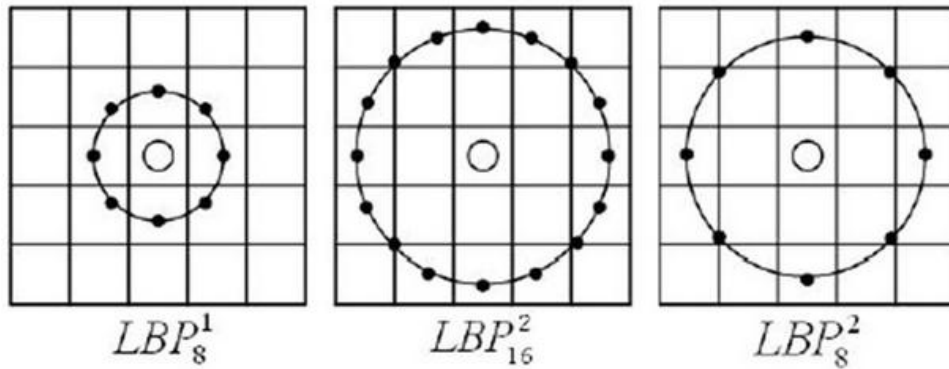


图 3.4 LBP 算子邻域示例

对一个给定的点 (x_c, y_c) , 它邻域内的某一点 (x_p, y_p) , $p \in P$ 的值则为:

$$x_p = x_c + R \cos\left(\frac{2\pi p}{P}\right)$$

$$y_p = y_c - R \sin\left(\frac{2\pi p}{P}\right)$$

其中， R 是圆的半径， P 是样本点的个数。

这个思路是对原始的 LBP 算法的优化，因此有的时候被称为扩展 LBP(或圆形 LBP)。通过上述公式可以计算任意采样点的值，但是计算得到的坐标未必是整数，某一个点在圆上，但是不在图像的任何坐标上，我们便需要取它的内插点来替代。OpenCV 中使用的是双线性插值：

$$f(x,y) \approx [1-x \quad x] \begin{bmatrix} f(0,0) & f(0,1) \\ f(1,0) & f(1,1) \end{bmatrix} \begin{bmatrix} 1-y \\ y \end{bmatrix}$$

通过上述算法，可以将一幅图像在每个像素点上得到 LBP 编码，从而获得一幅由 LBP 编码所组成的“LBP 特征图谱”。但是在 LBP 的应用中，一般不会直接将 LBP 特征图谱用于分类识别，因为这样的特征图谱跟位置信息关联性很大，直接将两幅图像通过 LBP 算子提取的特征图谱进行判别分析的话，会因为图像的对齐等位置问题造成很大的识别误差。而为了避免这种因为图像位置所造成的误差，可以将一幅图像划分的每个子区域所获得 LBP 特征分别建立统计直方图，然后再将统计直方图整合，获得整幅图像的统计直方图，这样就可以有效的减少位置误差，提高识别准确度。

3.3.2 基于 LBP 的人脸识别

基于 LBP 算法进行人脸图像特征提取，获得的局部纹理特征比较稳定，具有旋转不变性和灰度不变性等优点，而且适用于分类。基于 LBP 算法的人脸识别主要包括以下几个步骤：

1. 图像预处理：将输入的图像调整为标准的大小，且为了提高识别效率，一般将输入图像转换为灰度图像。
2. 图像分块：一般情况下，分的块数越多识别效果越好，但是相对应的计算复杂度也就越高，在 OpenCV 中，可以通过阈值来设定分块数。
3. 特征提取：LBP 的特征提取的步骤主要包括：
 - ① 根据分块后的人脸图像，得到若干局部区域；
 - ② 对每个区域中的一个像素点，与邻域内的像素点做比较，如果邻域内的像素点灰度值比中心像素点大，就要把邻域内的像素点位置标记为 1，

否则标记为 0，从而得出中心像素点的 LBP 值；

③ 对所有邻域的直方图进行计算，也就是计算每个数字出现的频率，然后对直方图进行归一化处理；

④ 将计算后得出的每个邻域的直方图连成一个特征向量，那么这个特征向量就是描述整个图像的 LBP 纹理特征的向量。

4. 描述人脸并匹配：特征提取所获得的 LBP 纹理特征向量即可以表示一幅人脸图像，可以利用 OpenCV 中的 SVM 分类器进行分类，从而达到人脸识别的目的。

3.4 本章小结

本章详细介绍了目前应用比较广泛，且在 OpenCV 中给出实现算法的三种人脸识别的算法：基于 PCA 的人脸识别算法，基于 LDA 的人脸识别算法，以及基于人脸局部特征提取的算法。详细介绍了 PCA 算法的基本思想，给出了算法描述；详细介绍了 Fisher 线性判别算法的基本思想，也给出了对应的算法描述；详细介绍了基于局部特征提取的人脸识别算法，给出了 LBP 的算法思想和描述。这三种算法都是经典的人脸识别算法，识别准确率也比较高，而且在 OpenCV 开源库中也实现了这三种算法的基本思想，可以在 OpenCV 基础上对这三种算法进行二次开发，从而实现本文系统中需要的对人脸识别的功能。

第 4 章 系统设计

4.1 功能设计

本系统的功能结构图如图 4.1 所示：

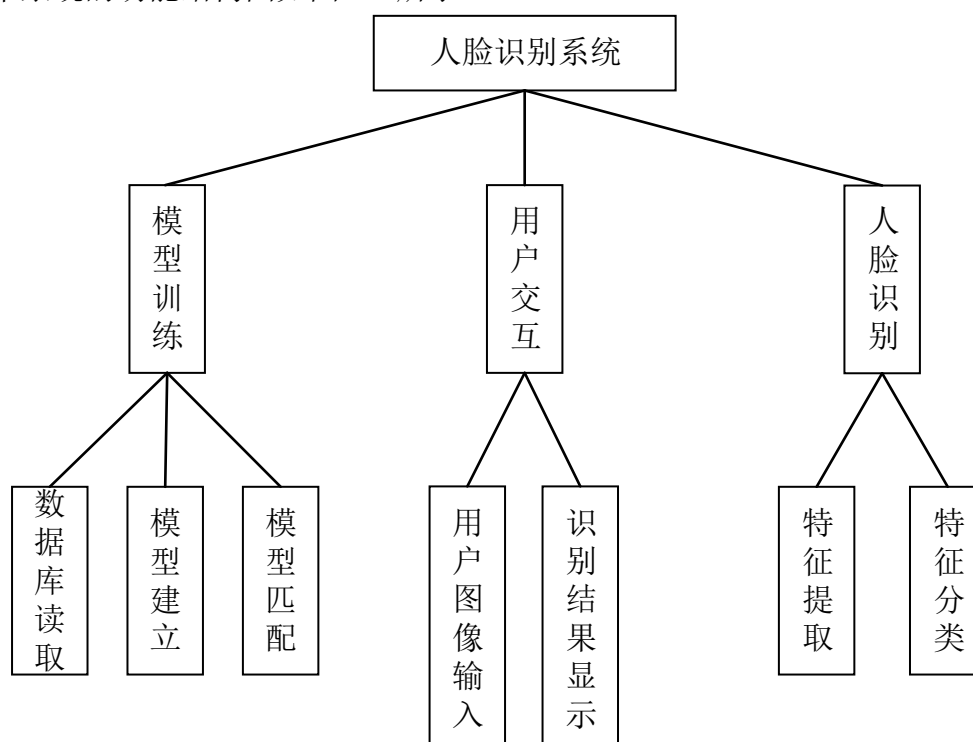


图 4.1 功能结构图

本系统的功能主要包括人脸模型训练功能、读入待识别图像功能、提取待识别图像功能、识别人脸功能四个方面。这些功能的具体设计如下：

1. 人脸模型训练功能

要实现人脸识别，首先就要对人脸库中的人脸进行训练，不同的人会得到不同的人脸模型，以用于识别时确定输入的图像与哪个人的特征匹配度更高。

人脸模型训练功能的实现是基于 OpenCV 技术的，在 OpenCV2.4.9 中提供了此算法，只需要将数据库中的人脸按照不同的人分类，然后分别读入，调用 OpenCV 内部函数即可得到训练好的模型。

2. 读入待识别图像功能

如果要识别一个人到底是不是在数据库中，或者到底是数据库中的哪个人，

就需要将待识别的人脸图像读入系统，以用于人脸图像特征的提取及识别。

待识别图像读入功能的实现是基于 OpenCV 中 `IplImage` 类中图像读取函数完成的，在 OpenCV 中直接提供了图像读取函数 `cvLoadImage`，目前此函数支持的文件格式有：Windows 位图文件-BMP、DIB，JPEG 文件-JPEG、JPG、JPE，便携式网络文件-PNG，便携式图像格式-PBM、PGM、PPM，Sun raster-SR、RAS，TIFF 文件-TIFF、TIF，OpenEXR HDR 文件-EXR，JPEG 2000 文件-jp2。

3. 提取待识别图像特征功能

人脸识别技术是基于提取的人脸图像特征来实现的，要对一幅人脸图像进行识别，首先就要对图像进行特征提取，然后根据特征来进行分类、识别。

待识别图像特征提取功能的实现是基于 OpenCV 技术的，在 OpenCV2.4.9 中提供了三种人脸特征提取的算法：EigenFace、FisherFace、LBP，这三种特征都是在人脸识别领域应用比较广泛且识别效果较好的特征。

4. 识别人脸功能

人脸识别系统的最终目的就是识别输入的人脸图像，这一功能也是本系统的最终功能。

识别人脸功能的实现也是基于 OpenCV 技术的，OpenCV2.4.9 中也集成了对特征的分类、识别的代码，它的主要算法思想就是将待识别图像的特征与数据库中人脸模型的特征进行比较，在满足一定阈值的基础上，相似度最好的那个模型就视为此人脸的最终识别结果。

4.2 模块设计

本系统中各模块之间的关系如图 4.2 所示：

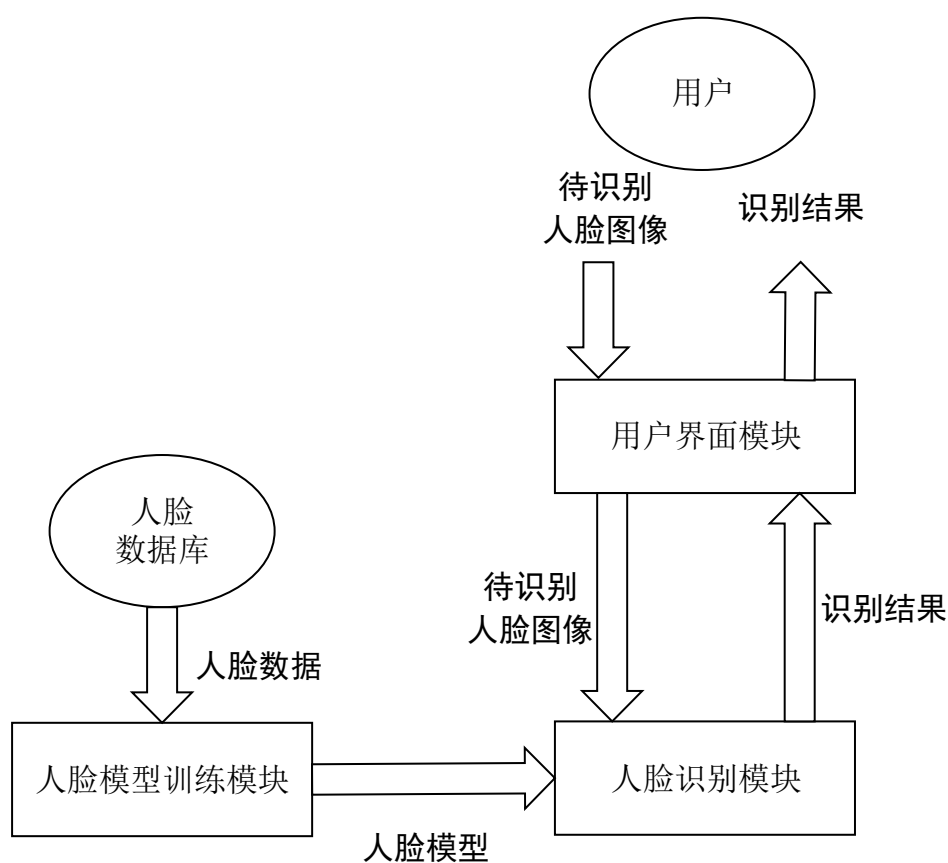


图 4.2 模块设计图

1. 人脸模型训练模块

人脸模型训练模块包括人脸模型训练功能，该模块实现对数据库中人脸数据的读取和训练，在该模块中，程序自动读取数据库中所有的人脸图像，并根据标签进行训练，生成对应的人脸模型，以用于人脸识别系统中人脸识别模块部分所需要的人脸匹配操作。

2. 用户界面模块

用户界面模块包括读入待识别图像功能，实现用户与系统之间的交互，用户上传一张人脸图像，然后系统返回一个识别结果。在该模块中，用户通过系统界面上的按钮来上传一张想要进行人脸识别的图像，然后系统将自动进入到人脸识别模块，待到计算出识别结果后，结果将从人脸识别模块返回到用户界面模块，用户界面模块则以图形和文字的形式将结果呈现给用户。

3. 人脸识别模块

人脸识别模块包括提取待识别图像特征和识别人脸功能，是本系统的核心算法所在的模块，实现人脸识别的目的。在该模块中，算法对用户界面模块传入的人脸图像进行一定的预处理，如大小调整、灰度图调整等，然后通过特征提取算法进行待识别图像的人脸特征提取，获得人脸特征后将对其与人脸模型训练模块生成的人脸模型进行匹配。在人脸识别模块中设定了一定的匹配度阈值，若待识别图像与数据库中某个人脸模型的匹配度最高，且其值高于设定的阈值，则识别成功，将识别结果返回给用户界面模块；若待识别图像与数据库中所有人脸模型的匹配度最高值小于设定的阈值，则识别失败，将失败结果返回给用户界面模块。

4.3 数据库设计

本文中所使用的数据库为 AT&T Face database，也就是应用比较广泛的 ORL 人脸数据库。ORL 人脸库由 40 个人组成，每个人包含 10 张不同的照片。这些照片分别在不同的时间，处于不同的光照条件下采集，待采集图像的人做出不一样的表情，如抬头、低头、侧脸、微笑、悲伤、严肃等等，也就呈现出不一样的人脸面部细节。但是这些图像的背景都是相同的，均为黑色背景，除了较小角度的侧脸和抬头低头外，全部都是正面人脸照片。

为了在程序中方便读取，每个人的人脸图像分别放在一个文件夹下，然后通过 CSV 文件给出人脸数据的存储位置和身份标签，CSV 文件如图 4.3 所示，例如“./at/s1/1.pgm;0”，其中“./at/s1/1.pgm”代表的就是一张人脸图像的位置，“0”则表示它所对应的身份标签。

```
1 ./at/s1/1.pgm;0
2 ./at/s1/2.pgm;0
3 ...
4 ./at/s2/1.pgm;1
5 ./at/s2/2.pgm;1
6 ...
7 ./at/s40/1.pgm;39
8 ./at/s40/2.pgm;39
```

图 4.3 数据库 CSV 文件示例

4.4 本章小结

本章详细介绍了基于 OpenCV 的人脸识别系统的架构设计，人脸识别系统主要包括人脸模型训练模块、读入待识别图像模块、提取待识别图像模块和识别人脸模块四个部分，这四部分分别对应本系统的人脸模型训练功能、读入待识别图像功能、提取待识别图像功能和识别人脸功能。其中人脸模型训练模块和识别人脸模块是本文的重点，基于 OpenCV 来实现。而数据库使用的是 AT&T Face database，也就是应用比较广泛的 ORL 人脸数据库，该数据库是人脸识别领域使用较为广泛的数据库之一。本文后续的系统实现就是基于本章的系统设计来完成的。

第 5 章 系统实现

5.1 模块实现

各个模块的实现如下：

1. 人脸模型训练模块

本系统中人脸图像数据库使用的是 ORL 开放人脸数据库，通过 CSV 文件的格式来存储人脸模型训练模块所需要的人脸图像的路径以及各个图像所对应的标签，CSV 文件是一种简单的数据交互格式，每一幅图像的信息（文件路径和对应的标签）占一行，中间用分号隔开。信息格式为：图像存储路径;标签，例如 E:\ORL\image.jpg;0。将数据库中的图像以每个人为一个文件夹，存储在本地，然后创建 CSV 文件来存储数据库中的所有图像路径和所对应的标签，接下来就可以对该数据库进行人脸模型的训练操作了。

人脸模型训练模块的第一步就是从 CSV 文件中获取图片数据库的信息，其中包括图片库中所有图像的存储位置的绝对路径以及每幅图像所包含的标签信息。获取 CSV 文件中的数据库信息之后，所要进行的操作是根据获取的图像路径去数据库中读取人脸图像数据，然后利用 OpenCV 的图像处理库对获取的图片数据和其相应的标签进行训练，得到人脸模型。

人脸模型训练模块代码实现如下：

```
//读取数据库训练数据

string fn_csv = "F:\\orl.txt";

//图像， 标签

vector<Mat> images;

vector<int> labels;

try {

    read_csv(fn_csv, images, labels);

} catch (cv::Exception& e) {
```



```

        cerr << "Error opening file \"" << fn_csv << "\". Reason: " << e.msg <<
endl;

        exit(1);
    }

    if(images.size() <= 1) {
        string error_message = "This demo needs at least 2 images to work. Please
add more images to your data set!";
        CV_Error(CV_StsError, error_message);
    }

    // 图像大小
    int height = images[0].rows;
    //利用 opencv 函数训练人脸模型
    Ptr<FaceRecognizer> model = createLBPHFaceRecognizer(1, 8);
    model->train(images, labels);

```

2. 用户界面模块

用户界面模块使用 MFC 来实现，用户界面运行结果如图 5.1 所示，用户可以通过“打开图片”按钮来上传待识别的人脸图像，上传的图像会在用户界面左侧显示，然后程序将在后台自动开始进行人脸识别计算，识别结果将显示在用户界面右侧，其中识别结果包括识别出的人员正面图像、人员姓名、性别。

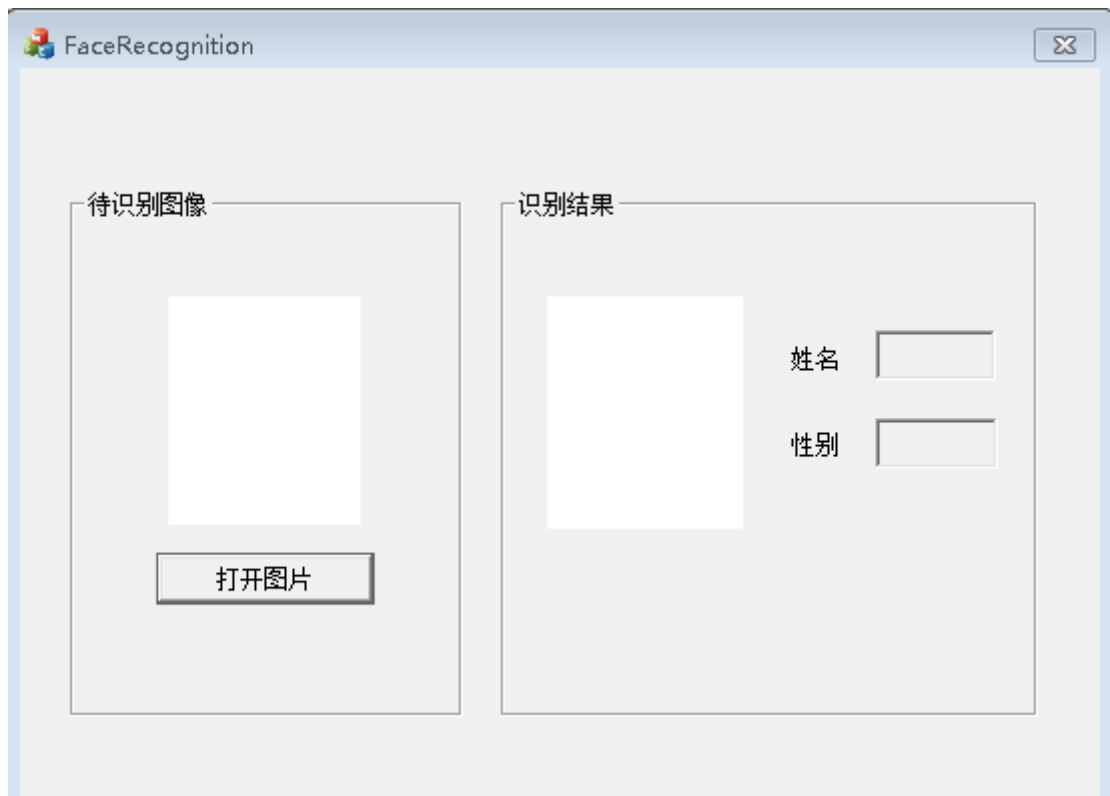


图 5.1 用户界面

用户界面模块代码实现如下：

```
void CFaceRecognitionDlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this); // 用于绘制的设备上下文

        SendMessage(WM_ICONERASEBKGND,
reinterpret_cast<WPARAM>(dc.GetSafeHdc()), 0);

        // 使图标在工作区矩形中居中
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
```

```
GetClientRect(&rect);

int x = (rect.Width() - cxIcon + 1) / 2;
int y = (rect.Height() - cyIcon + 1) / 2;

// 绘制图标
dc.DrawIcon(x, y, m_hIcon);
}
else
{
    CDialogEx::OnPaint();
}
//刷新人脸图像
DrawPicToHDC(image, IDC_PIC);
DrawPicToHDC(resultface, IDC_FACE);
}

//当用户拖动最小化窗口时系统调用此函数取得光标
//显示。
HCURSOR CFaceRecognitionDlg::OnQueryDragIcon()
{
    return static_cast<HCURSOR>(m_hIcon);
}

void CFaceRecognitionDlg::DrawPicToHDC(IplImage *img, UINT ID)
{
    CDC *pDC = GetDlgItem(ID)->GetDC();
```

```

HDC hDC = pDC->GetSafeHdc();

CRect rect;

GetDlgItem(ID)->GetClientRect(&rect);

CvImage cimg;

cimg.CopyOf(img);                                // 复制图片

cimg.DrawToHDC(hDC, &rect);                       // 将图片绘制到显示
控件的指定区域内

ReleaseDC(pDC);

}

```

3. 人脸识别模块

人脸识别模块的实现即为人脸识别算法的实现，而人脸识别算法是基于 OpenCV 的开源代码来实现的，此模块就是将待识别图像读取、特征提取、分类三个部分集成起来，实现人脸识别的功能。

人脸识别模块代码实现如下：

```

void CFaceRecognitionDlg::OnBnClickedOpen()
{
    // TODO: 在此添加控件通知处理程序代码

    CString FilePathName;

    CFileDialog dlg(TRUE,                                //TRUE 为 OPEN 对话框，
FALSE 为 SAVE AS 对话框
        NULL,
        NULL,
        OFN_HIDEREADONLY | OFN_OVERWRITEPROMPT,
        (LPCTSTR)_TEXT(" 图 片 文 件 |*.bmp;*.jpg;*.png;*.pgm|All  Files
(*.*)|*.*|"),
        NULL);

    if(dlg.DoModal() == IDOK)

```

```

{
    FilePathName = dlg.GetPathName();           // 文件名保存在了
FilePathName 里
}
else
{
    return;
}
if(image) cvReleaseImage(&image);
image = cvLoadImage(FilePathName, 0);          //显示图片
DrawPicToHDC(image, IDC_PIC);
int result = -1;
double confidence = 0.0;
facerec(image, result, confidence);
if (confidence > 50)
{
    result = 0;
}
CString temp;
temp.Format("%d", result);
CString resultpath = "result\\" + temp + ".bmp";
if(resultface) cvReleaseImage(&resultface);
resultface = cvLoadImage(resultpath, 1);
DrawPicToHDC(resultface, IDC_FACE);
//姓名
CEdit* pName;
pName = (CEdit*) GetDlgItem(IDC_NAME);

```

```
pName->SetWindowText(name[result]);  
//性别  
CEdit* pGender;  
pGender = (CEdit*) GetDlgItem(IDC_GENDER);  
pGender->SetWindowText(gender[result]);  
}
```

5.2 算法实现

系统中的主要算法是人脸识别算法,该算法是基于 OpenCV2.4.9 来实现的。从 OpenCV2.4 开始,OpenCV 中添加了新的类——FaceRecognizer,通过使用这个类可以较为简单的实现人脸识别的实验和测试。目前 OpenCV 提供的人脸识别方案有三种:特征脸 EigenFace、Fisher 脸 FisherFace 和 LBP 直方图 LBPHFace。这三种识别方案能够通过调用函数 createEigenFaceRecognizer、createFisherFaceRecognizer、createLBPHFaceRecognizer 建立模型。

人脸识别算法主要分为两部分:人脸图像的训练和预测。在 OpenCV 中分别有 train 函数和 predict 函数来实现这两部分的功能,此外 save 函数实现人脸数据的保存,load 函数实现人脸数据的读取。OpenCV 还设定了一些人脸识别的参数,如特征点数、识别阈值等,我们还可以通过调节这些参数值来对人脸识别进行调整以达到最优的识别效果。在训练阶段,train 函数中包含训练人脸图像组 vector<Mat>和对应标签组 vector<int>两组参数,其中对应标签组的参数是同一个人对应同样标签,在 OpenCV 中标签使用整形数值来表示,标签的数值及顺序没有要求,但是标签一定是唯一的,不可出现不同人重复标签。对在预测阶段,predict 函数包含的参数有待识别图像、识别结果的标签值、以及待识别图像和标签所对应的数据库模型的相似度。其中,若标签返回值为-1,则说明在数据库中不存在与待识别图像匹配的人脸模型。

根据 LBP 算法思想,算法流程图如图 5.2 所示:

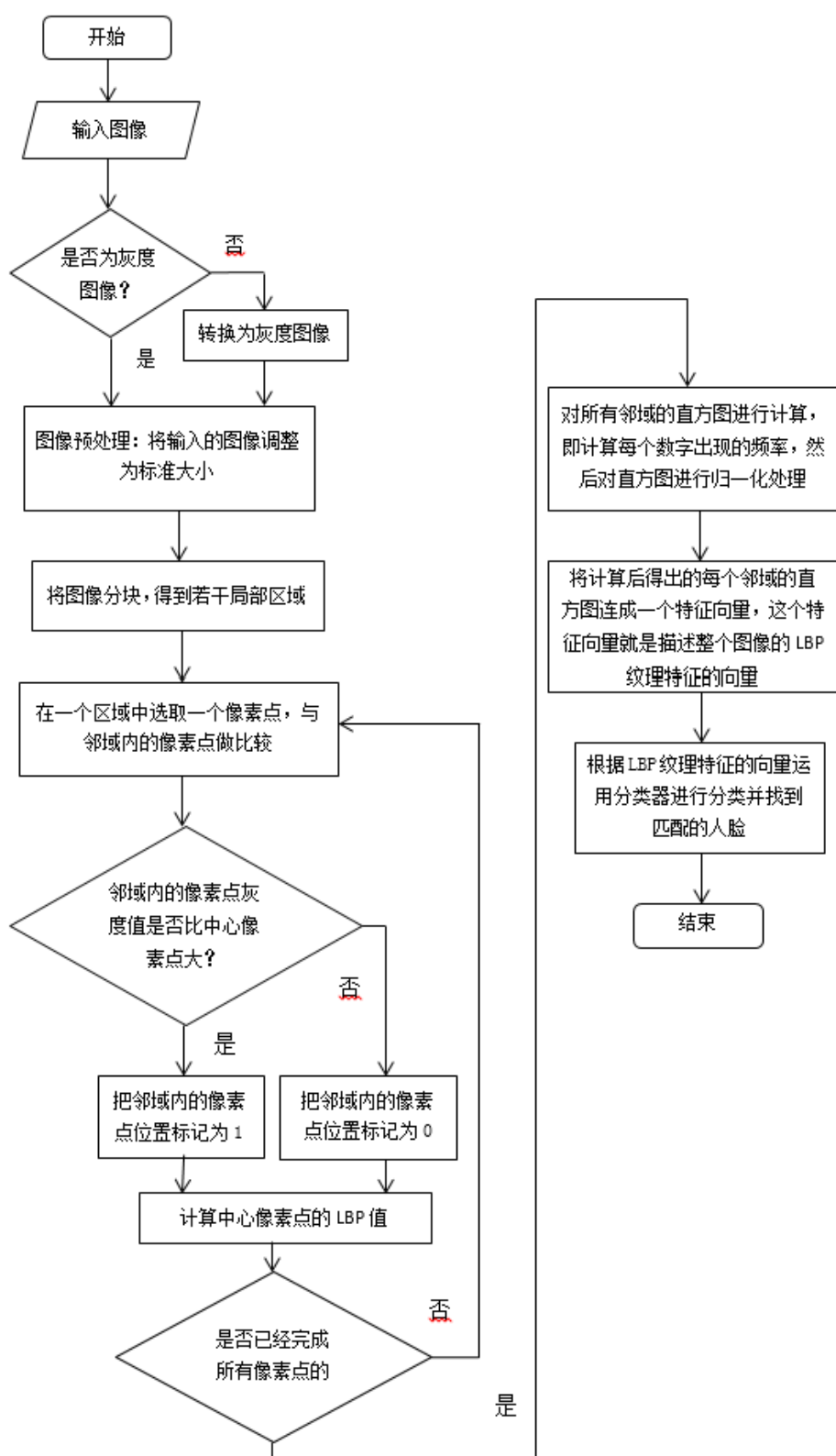


图 5.2 算法流程图

5.3 本章小结

本章详细介绍了基于 OpenCV 的人脸识别系统的实现，通过对第 4 章中系统设计的详细分析，以及对 OpenCV 中 FaceRecognizer 类的学习和研究，基于 OpenCV2.4.9，并结合 ORL 开放人脸数据库实现了人脸模型的训练；基于 OpenCV2.4.9 中的 LBP 人脸识别算法实现了识别人脸的功能。本系统的整体框架是基于 MFC 实现的，然后将 OpenCV 与 MFC 整合，便实现了整体的人脸识别系统。

第 6 章 系统测试

6.1 算法测试

算法测试基于 OpenCV 图像处理库中的人脸识别算法进行测试，测试数据在 ORL 人脸数据库中分别选取每人前 2 幅共 80 幅图像，每人前 5 幅共 200 幅图像，每人前 8 幅共 320 幅图像作训练样本。每人后 5 幅共 200 幅图像作测试样本。测试结果如表 6.1 和表 6.2 所示。

表 6.1 算法测试——识别率

方法 \ 训练集	80	200	320
EigenFace	80%	90%	98%
FisherFace	73.5%	81.5%	96.5%
LBPHFace	71.5%	88%	97%

表 6.2 算法测试——时间效率

方法 \ 训练集	80	200	320
EigenFace	4431ms	22807ms	56194ms
FisherFace	3448ms	15163ms	40826ms
LBPHFace	14706ms	24912ms	33051ms

由测试结果可看出，EigenFace 在较规则的图片中识别率较高。在训练样本较为适中的情况下，EigenFace 和 LBPHFace 的识别率都比较高，LBPHFace 的时间最短；当训练样本比较小时，EigenFace 的识别性能稍微有所下降，LBPHFace 的识别能力大大下降。所以得出结论为：用于人脸识别人脸训练集较小的情况下，EigenFace 可以优先考虑，而当人脸识别人脸训练集较大的情况下，优先考虑 LBPHFace 可获得更佳效果。

6.2 功能测试

1. 人脸模型训练功能

基于 OpenCV 实现的人脸模型训练功能可以实现对数据库人脸图像的读取和模型训练，图 6.1 和图 6.2 分别是 EigenFace 算法和 FisherFace 算法基于不同数量的特征训练得到的人脸模型。

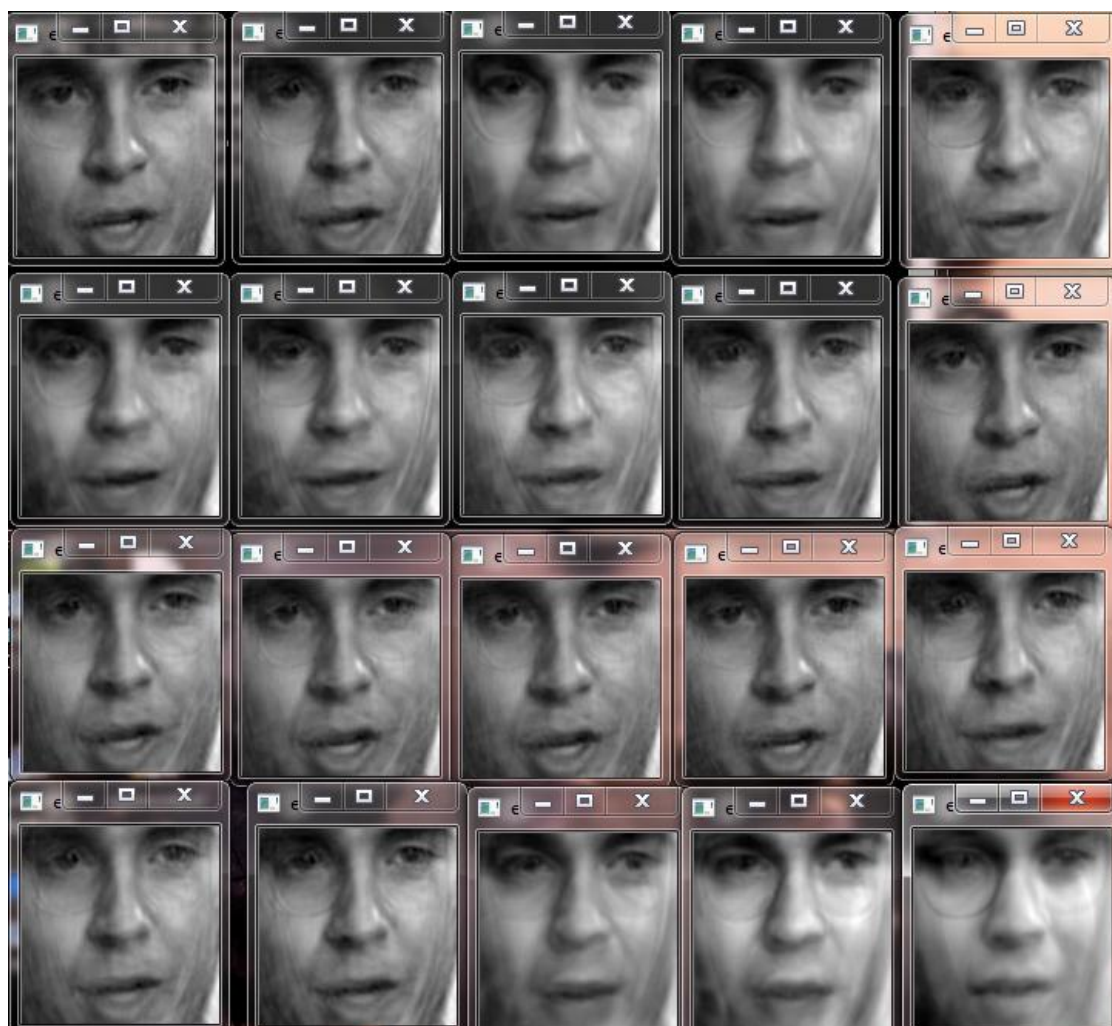


图 6.1 EigenFace 训练得到的人脸模型



图 6.2 FisherFace 训练得到的人脸模型

2. 读入待识别图像功能

通过用户界面可以简单便捷的读入待识别的图像，大部分常见的图像文件格式（如 jpg，png，bmp，pgm 等等）均可以成功读入到系统中。

3. 提取待识别图像特征功能

图像特征的提取是基于 OpenCV 中已有的算法来实现的，三种特征提取算法 EigenFace、FisherFace 和 LBP 均可以实现输入图像人脸特征的提取，其正确性也在识别过程中得到了验证。

4. 识别人脸功能

识别人脸功能是人脸识别系统中人脸识别模块的最后一步，它所需要完成的工作就是将提取的待识别图像特征进行分类，并给出相似度，从系统的输出结果来看，此功能也得到了正确的实现。

6.3 整体测试

系统整体测试是在各模块都已经成功测试的基础上进行的，测试的主要方

向是各个模块之间的接口是否正确，系统界面是否能正常加载，测试中运行均正常。系统整体运行结果如图 6.3 和图 6.4 所示，其中图 6.3 为上传照片为系统数据库中包含的人员的测试结果，该人员姓名为“一”，性别为“男”；图 6.4 为上传照片不是系统数据库中包含的人员的测试结果。



图 6.3 系统运行图——人员识别成功



图 6.4 系统运行图——人员识别失败

第7章 总结与展望

7.1 总结

人脸识别作为一个前沿技术,已成为当今炙手可热的研究领域,并在国家安全、证件核实、信息安全、人事考勤等许多领域都得到了广泛应用。本文总结了人脸识别的研究现状,对 EigenFace, FisherFace 和 LBP 三种人脸识别算法的基础理论进行了陈述,并用 OpenCV 开源视觉库实现了这三种算法,测试了它们的性能。根据测试的结果我们决定在系统中采用 LBP 人脸识别算法,运用 Visual Studio 2010 搭建了基于 OpenCV 的人脸识别系统,用 MFC 设计了用户界面,并对系统进行了测试,实现了完整的基于 OpenCV 的人脸识别系统,人脸识别效果良好。本文完成的主要工作有:

- (1) 详细介绍了 OpenCV, 以及 OpenCV 的优点和功能; 详细介绍了 MFC, 以及 MFC 中各个主要的类。
- (2) 详细介绍了 EigenFace, FisherFace 和 LBP 三种人脸识别算法的理论基础和具体实施过程。
- (3) 运用 OpenCV 开源视觉库实现了 EigenFace, FisherFace 和 LBP 三种人脸识别算法, 使用 ORL 标准人脸数据库进行数据库人脸模型的训练, 然后分别对这三种算法进行了性能测试, 比较了它们的各自优缺点, 最终选择了 LBP 人脸识别算法。
- (4) 将 LBP 人脸识别算法应用到本文系统中, 用 MFC 设计实现了界面较为美观、交互性良好的用户界面, 用户输入待识别的人脸图像, 系统则可以自动进行人脸识别计算, 并给出最终的识别结果, 实现了完整的基于 OpenCV 的人脸识别系统。
- (5) 对人脸识别系统进行了一系列的测试, 如功能测试、模块测试、整体测试, 并给出了详细的测试结果, 测试结果良好, 人脸识别准确、高效。

7.2 展望

人脸识别研究从二十世纪六十年代发展至今已经有近五十年的历史，取得了许多卓越的成绩，广泛应用到了各个领域。本文对已有的人脸识别算法进行研究并设计实现了人脸识别系统，但是该系统仍有许多不足之处，主要包括：

(1) 光照变化对人脸识别的准确度影响性还是比较大，应继续对算法进行优化，消除不利影响。

(2) 当要识别的人脸图像佩戴眼镜等配饰时会产生遮挡，增加人脸特征特区的难度，应在这方面进行改进。

(3) 进一步提高算法的效率，实现实时识别的功能。

在今后的研究和工作中，我将会进一步的探索上述问题，并争取找出解决问题的方法。

参考文献

- [1] 孙冬梅, 裘正定. 生物特征识别技术综述[J]. 电子学报, 2009 (z1): 1744-1748.
- [2] H.Chan and W.W.Bledsoe.A man-machine facial recognition system:some preliminary results[J].Panoramic Research Inc..Cal.1965.
- [3] Bledsoe W W. Man-machine facial recognition[J]. Rep. PRi, 1966, 22.
- [4] A.J.Goldstein.L.D.Harmon and A.B.Lesk.Identification of human faces[C].Proceedings of the IEEE.1971,59(5):748-760.
- [5] T.Kanade.Picture processing system by computer and recognition of human faces[D].Kyoto University.1973.
- [6] M.Turk,A.Pentland.Eigen-faces for recognition[J].Jounal of eognitive neurosciencem,3(1),1991:71-86.
- [7] Bellhumer P.N.,Hespanha J.,Kriegman D.,Eigenfaces vs. fisherfaces:Recognition using class specific linear projection[J].IEEE Transactions on Patten Analysis and Machine Intelligence,Special Issue on FaceRecognition,1997,17(7):711-720.
- [8] A.Shashua, T.Riklim. The quotient image: Class-Based Re-Rendering and recoginition with varying illiminations[J].IEEE Trans.on PAMI,23(2),2001:129-139.
- [9] 周激流, 张晔.人脸识别理论研究进展[J]. 计算机辅助设计与图形学学报, 1999,11(2):180-184.
- [10] 王芳.基于五官特征定位的人脸识别技术在防盗门中的应用[D].青岛: 中国海洋大学, 2005.
- [11] 张生亮.单样本多姿态人脸识别研究[J].计算机应用,2006,26(12):2851-2853.
- [12] 雷静.基于 OpenCV 的人脸跟踪识别系统研究[D].西安: 西安电子科技大学, 2010.
- [13] 卢春雨, 张长水, 闻方, 阎平凡.基于区域特征的快速人脸检测法[J].清华大学学报(自然科学版), 1999,39(1):101-105.

- [14] 刘明宝, 高文. 复杂背景下的人脸检测与跟踪系统[J]. 计算机研究与发展, 1997, 34 (增刊): 54-58.
- [15] 王蕴红等. 融合全局与局部特征的子空间人脸识别算法[J]. 计算机学报, 2005, 28(10): 1657-1663.
- [16] Chellappa R, Wilson C L, Sirohey S. Human and machine recognition of faces: A survey[J]. Proceedings of the IEEE, 1995, 83(5): 705-741.
- [17] Brunelli R, Poggio T. Face recognition: Features versus templates[J]. IEEE transactions on pattern analysis and machine intelligence, 1993, 15(10): 1042-1052.
- [18] Swets D L, Weng J J. Using discriminant eigenfeatures for image retrieval[J]. IEEE Transactions on pattern analysis and machine intelligence, 1996, 18(8): 831-836.
- [19] Moghaddam B, Pentland A. Probabilistic visual learning for object representation[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1997, 19(7): 696-710.
- [20] Lee S Y, Ham Y K, Park R H. Recognition of human front faces using knowledge-based feature extraction and neurofuzzy algorithm[J]. Pattern recognition, 1996, 29(11): 1863-1876.
- [21] Lawrence S, Giles C L, Tsoi A C, et al. Face recognition: A convolutional neural-network approach[J]. Neural Networks, IEEE Transactions on, 1997, 8(1): 98-113.
- [22] 周杰, 卢春雨. 人脸自动识别方法综述[J]. 电子学报, 2000, 28(4): 102-106.
- [23] Zhang J, Yan Y, Lades M. Face recognition: eigenface, elastic matching, and neural nets[J]. Proceedings of the IEEE, 1997, 85(9): 1423-1435.
- [24] Wiskott L, Fellous J M, Kuiger N, et al. Face recognition by elastic bunch graph matching[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1997, 19(7): 775-779.
- [25] OpenCV[EB/OL]. <http://zh.wikipedia.org/wiki/OpenCV>, 2014-10-08.

- [26] MFC[EB/OL]. <http://zh.wikipedia.org/wiki/MFC>, 2013-11-13.
- [27] R.A.Fisher.The Use of Multiple Measures inTaxonomic Problems.Ann.Engenics,1936,17: 179-188.
- [28] S.S.Wilks,MathematicalStatistics.Wiley,New York,1962:577-578.
- [29] R.Duda.P.Hart,D.Stork.PatternClassification.NewYork:John Wiley & Sons.2001.
- [30] Belhumeur, Hespanha, Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. IEEE Trans. Pattern Anal.Mach. Intell 1997, 19: 711–720.
- [31] Ahonen T, Hadid A, Pietikäinen M. Face recognition with local binary patterns[M]//Computer vision-eccv 2004. Springer Berlin Heidelberg, 2004: 469-481.

作者简介

作者姓名：卜秋月

作者性别：女

出生年月：1990 年 11 月 29 日

民族：汉

出生地：吉林省松原市

学位：学士学位

职称：科员

学历：本科

工作经历（职务）：松原市地方税务局综合大厅

致谢

在我读研究生的这三年中，我经历了很多，也成长了很多，在我硕士论文即将完成之际，我想借此机会感谢所有曾经给予我帮助和支持的人。

首先，我要感谢我的正导师包铁老师和指导老师彭君老师，在我读研期间，老师们一丝不苟的工作作风，严肃的科学态度，严谨的治学精神，深深地感染和激励着我，让我体会到了作为一名科研工作者应有的品质和素养，使我终生受用。特别是我完成毕业设计之中，老师们在百忙之中仍要抽出时间来对我进行指导，令我深受感动。老师们也很关心我的生活、工作情况，他们无微不至的关怀温暖着我，给了我力量。

同时，我还要感谢软件学院的同学们，他们在我的学习上给予了帮助和支持，在我的生活上给予了关心和照顾。在我遇到困难的时候，是他们鼓励着我继续前行。在此，我想向他们表达我衷心的感谢，并祝他们今后的工作顺利，生活美满。

最后，我要感谢父母对我的养育之恩，是他们给了我这么好的学习和成长环境，是他们的支持让我能够在我喜欢的领域研究、发展，感谢他们的理解和包容，感谢他们对我的默默付出。

对在学习、研究过程中给予我提供帮助的老师和同学们再次表示衷心的感谢！