

分类号： TP311

密 级： 公 开

UDC：

单位代码： 10424

学 位 论 文

Android 平台上基于人脸识别的身份认证系统的设计与实现

王志远

申请学位级别： 硕士学位 专业名称： 电子与通信工程

指导教师姓名： 刘 山 职 称： 教 授

山 东 科 技 大 学

二〇一八年四月

论文题目：

Android 平台上基于人脸识别的身份认证 系统的设计与实现

作者姓名：王 志 远 入学时间：2015 年 9 月

专业名称：电子与通信工程 研究方向：图像处理与模式识别

指导教师：刘 山 职 称：教 授

论文提交日期：2018 年 4 月

论文答辩日期：2018 年 6 月

授予学位日期：

**DESIGN AND IMPLEMENTATION OF FACE
RECOGNITION IDENTITY AUTHENTICATION SYSTEM
BASED ON ANDROID PLATFORM**

A Dissertation submitted in fulfillment of the requirement of the degree of

MASTER OF PHILOSOPHY

from

Shandong University of Science and Technology

by

Wang Zhiyuan

Supervisor: Professor Laxmisha Rai

College of Electronic Communication and Physics

April 2018

学位论文原创性声明

本人呈交给山东科技大学的这篇硕士学位论文，除所列参考文献和世所公认的文献外，全部是本人攻读学位期间在导师指导下的研究成果。除文中已经标明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，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。

若有不实之处，本人愿意承担相关法律责任。

硕士生签名：

日 期：

学位论文使用授权声明

本人完全了解山东科技大学有关保留、使用学位论文的规定，同意本人所撰写的学位论文的使用授权按照学校的管理规定处理。

作为申请学位的条件之一，学校有权保留学位论文并向国家有关部门或其指定机构送交论文的电子版和纸质版；有权将学位论文的全部或部分内容编入有关数据库发表，并可以以电子、网络及其他数字媒体形式公开出版；允许学校档案馆和图书馆保留学位论文的纸质版和电子版，可以使用影印、缩印或扫描等复制手段保存和汇编学位论文；为教学和科研目的，学校档案馆和图书馆可以将公开的学位论文作为资料在档案馆、图书馆等场所或在校园网上供校内师生阅读、浏览。

（保密的学位论文在解密后适用本授权）

硕士生签名：

日 期：

导师签名：

日 期：

学位论文审查认定书

研究生 在规定的学习年限内，按照培养方案及个人培养计划，完成了课程学习，成绩合格，修满规定学分；在我的指导下完成本学位论文，论文中的观点、数据、表述和结构为我所认同，论文撰写格式符合学校的相关规定，同意将本论文作为申请学位论文。

导师签名：

日 期：

摘要

随着移动互联网与人工智能技术的发展,基于个人特征的生物识别技术代替传统的身份验证方式已经是大势所趋。而人脸识别是生物识别技术的一个重要组成部分,拥有其他生物识别技术没有的独特优势。本文主要针对 Android 平台上基于人脸识别的身份认证系统进行了设计和实现。

本文的主要工作概括如下:

1.对原始图像预处理算法进行了介绍和研究,首先,对采集的图像进行灰度归一化的操作,增强图像对比度,让不同成像条件下拍摄的同一个人的图像保持一致。然后,对于灰度后的图像进行了直方图均衡化的操作,其作用对图像进行非线性拉伸,重新分配图像灰度值,扩展像素个数多的灰度级,压缩像素个数少的灰度级,提高了对比度和灰度色调的变化。最后一步预处理工作是对图像进行滤波操作,消除噪点。

2.对 Adaboost 人脸检测算法的概念进行了介绍,对特征提取算法进行了介绍,对人脸区域进行特征提取,可以有效的降低人脸图像的维数,提高下一步分类器分类识别的效率和准确率。这里重点介绍了基于 PCA 的特征提取算法,并详细描述了 PCA 特征提取的几个过程。

3.对人脸识别的概念和分类算法进行了介绍,重点介绍了常见的几种机器学习分类器算法的概念和原理。重点研究了最近邻分类器算法和支持向量机分类器算法。通过分析人脸身份认证系统的使用场景和尽可能提高人脸识别准确率,提出了基于最近邻算法(KNN)和支持向量机算法(SVM)相结合的分类器算法。经过在一些公共数据集上的测试,这种相结合分类器算法比单独使用最近邻算法或支持向量机算法在人脸识别上具有更高的准确率和性能优势。

4.整个身份认证系统在 Android 平台上的实现。在进行系统需求分析后,根据 Android 系统的特点进行了系统框架的搭建。主要应用了 Android SDK 实现界面和摄像头采集图片的工作,用 C++语言和 OpenCV 库实现了图像处理的算法,并用 JNI 和 Android NDK 将两者结合起来,构成一个完成的系统。

关键词: Android, 人脸识别, OpenCV, Adaboost, 身份验证, SVM

ABSTRACT

With the development of mobile Internet and artificial intelligence technology, the biometric technology based on personal characteristics has replaced the traditional authentication method. Face recognition is an important part of biometric technology and has unique advantages that other biometric technologies do not have. This paper focuses on the design and implementation of face recognition authentication system based on Android platform.

The main work of this article is summarized as follows:

1. The original image preprocessing algorithm is introduced and researched. Through the image preprocessing algorithm, the original image's contour is clearer and the contrast is stronger. First of all, the original image is grayscale normalized. In order to reduce the influence of illumination and other factors on the image, the grayscale image is further subjected to histogram equalization processing. After the histogram equalization processing, the image will be More clear and soft, and finally introduced several image filtering solutions to eliminate noise.

2. The concept of the Adaboost face detection algorithm is introduced. The feature extraction algorithm is introduced. The feature extraction of the face region can effectively reduce the dimensions of the face image, and the efficiency of the classification and recognition of the next classifier can be mentioned. Accuracy. This article focuses on the feature extraction algorithm based on PCA, and describes in detail the process of PCA feature extraction.

3. The concepts and classification algorithms of face recognition are introduced, and the concepts and principles of several common machine learning classifier algorithms are highlighted. Focused on the nearest neighbor classifier algorithm and support vector machine classifier algorithm. By analyzing the use scenario of face identity authentication system and improving the accuracy of face recognition as much as possible, a classifier algorithm based on the combination of nearest neighbor algorithm (KNN) and support vector machine algorithm (SVM) is

proposed. After testing on some common data sets, this combined classifier algorithm has higher accuracy and performance advantages in face recognition than using the nearest neighbor algorithm or support vector machine algorithm alone.

4. The entire identity authentication system is implemented on the Android platform. After system requirements analysis, the system framework was built according to the characteristics of the Android system. The main application of the Android SDK to achieve the interface and the camera to capture the picture work, using C++ language and OpenCV library to achieve the image processing algorithm, and use JNI and Android NDK to combine the two to form a complete system.

Key words: Android, Face recognition, OpenCV, Adaboost, Authentication, SVM

目录

摘要	I
目录	IV
1 绪论	1
1.1 课题研究背景及意义	1
1.2 国内外研究现状	2
1.3 本文主要研究内容与组织结构	4
2 人脸图像预处理	7
2.1 灰度归一化	7
2.2 直方图均衡化	9
2.3 图像滤波	11
2.4 本章小结	13
3 人脸检测与识别算法	14
3.1 基于 Adaboost 的人脸检测算法	14
3.2 基于 PCA 的人脸特征提取	19
3.3 基于 KNN 和 SVM 的人脸识别算法	20
3.4 本章小结	28
4 Android 环境搭建和开发技术	30
4.1 Android 系统概述与优势	30
4.2 Android 应用组件	31
4.3 Android 开发准备	33
4.4 本章小结	38
5 基于 Android 平台的身份认证系统的设计与实现	39
5.1 系统功能分析	39
5.2 实验与结果分析	41
5.4 本章小结	45

6 总结与展望	46
6.1 本章小结	46
6.2 未来的展望	46
参考文献	48
致谢	52
攻读硕士期间主要研究成果	53

Contents

Abstract	I
Contents.....	V
1 Introduction.....	1
1.1 Project research background and significance.....	1
1.2 Research status at home and abroad	3
1.3 This article mainly studies the content and organizational structure.....	6
2 Face image preprocessing.....	7
2.1 Grayscale normalization.....	7
2.2 Histogram equalization	9
2.3 Image filtering	11
2.4 Chapter summary.....	13
3 Face Detection and Recognition Algorithm	14
3.1 Face detection algorithm	14
3.2 Face feature extraction	19
3.3 Face recognition algorithm.....	20
3.4 Chapter summary.....	28
4 Android environment to build and develop technology	30
4.1 Android system overview and advantages	30
4.2 Android application components	31
4.3 Android development preparation	33
4.4 Chapter summary.....	38
5 Construction of Android Environment and Implementation of Face Recognition System	39
5.1 System function analysis.....	39
5.2 Experiment and result analysis	41
5.3 Chapter summary.....	45
6 Summary and outlook.....	46
6.1 Chapter summary.....	46
6.2 Future outlook	47
Main Reference Documents	48

Thanks.....	52
Main Work Achievement of the Author during Working on Master Paper.....	53

1 绪论

1.1 课题研究背景及意义

随着移动互联网时代的发展,各种网络账号越来越多,账号的注册、登录、支付等操作越来越频繁。由于原始的账号密码的方式呈现出很大的弊端,越来越多的账号密码让用户产生了混淆。很多账号长时间不登录就忘记了密码甚至账号,而大部分网站找回账号密码等操作需要人工核对,费时费力,给用户带来了极大的不便。现在很多用户采用各种账号使用相同密码的方式,这种方式存在着很大的安全隐患,很多不法分子就是通过破解一个密码测试出其他账号的密码,例如之前的 12306 账号密码泄露事件,一旦账号密码泄露,会导致用户财产损失、隐私泄露。

寻找一种快速有效的身份验证的方式已经成为了当务之急,因此生物特征识别技术在近几年中得到了飞速的发展。生物特征具备高度的稳定性、安全性、唯一性,成为了身份验证的最理想依据,备受广大学者们的关注和研究^[1]。与其他识别方法相比,人脸识别由于其特有的优势。在人脸识别过程中,计算机通过观察人脸,学习到一些区分性的特征,通过比较后,确认个体的身份。这种识别方式和人类识别身份时使用的特征类似^{[2] [3]}。与指纹识别,视网膜识别,虹膜识别相比,人脸识别是不需要通过接触的,只需要通过照相机拍摄照片即可,可以较远距离进行实现身份确认,比较方便和友好^[4]。与步态识别、手势识别相比,人脸识别很难被模仿和伪造,所以更加安全。最后,人脸图像提取的特征信息更加容易存储^[5]。基于以上几点的考虑,关于人脸识别的一些技术更加受到了用户的喜爱与认可。

Android 是一种面向智能手机的基于 Linux 内核的操作系统^[6],它最初由 Andy Rubin 开发,Andy Rubin 开发这个系统的最初目的是为了创建一个智能的数码相机操作系统,但是当后来智能手机市场快速发展的时候,Android 被改造为主要面向智能手机的操作系统。后来 Android 系统被 Google 公司收购,成为 Google 旗下的一部分^[7]。Android 系统以 Linux 为内核,采用多级分层的架构,包括 Linux kernel、Applications、Framework 以及 Libraries 组成^[8]。由

于 Android 系统强大的可移植性和 Google 公司开源的性质, Android 系统理论上可以在任何电子产品上运行, 这对以后产生更智能化的电子产品有着巨大的影响和推动作用^[9]。Android 系统得到了众多手机硬件厂商的支持, 移动终端市场占有率年年升高。

和其他操作系统相比, Android 系统是开源的操作系统, 拥有众多的开发者, 更有着 Google 公司的支持^[10]。Android 系统拥有完整的开发生态, 从底层运行到应用程序的开发和发布都有着强大的监管措施, 极大地推动了移动产业的发展^[11]。因为 Android 系统的开放性和开源性允许任何移动设备制造厂商加入^[12], 对于众多的开发者来说, 有着大量的资源可以利用。对于用户来说, 不仅可以用到全世界的优秀开发者开发的应用程序, 还可以用众多开放的源代码进行二次开发^[13]。开放的平台会带来了良性的竞争, 能更好的服务用户, 让用户们能够用到更优质的产品^[14]。

从人脸识别技术发展潜力以及当前 Android 系统的市场占有率考虑, 基于 Android 系统的人脸识别身份认证系统的研究很有前景。特别是最近几年, 各种电商、O2O、O2P 公司的兴起, 例如美团、饿了么、支付宝、微信支付、滴滴打车、京东等等, 这些应用已经渗入到了人们生活的方方面面, 手机支付已经改变了人们消费的习惯, 在给人们带来极大便利的同时, 也带来了一些相应的安全隐患。基于人脸识别的身份验证取代基于传统账号密码的身份验证已经是未来发展的趋势, 同时这项工作具有很大的挑战性。首先, 使用手机摄像机进行拍照时光照与阴影的干扰一直是影响人脸识别效果的难题; 其次, 虽然手机性能在逐年提高, 但是相比于 PC 的性能还是有不小的差距, 在人脸识别效率上会慢一些, 甚至无法保证实时性, 所以人脸识别算法的优化也是工作过程中的一个重点也是难点; 最后, 人脸识别研究需要很高的理论知识, 涉猎范围较广, 需要工作者具备一定的理论基础。综上所述, 基于 Android 平台下的人脸识别身份认证系统具有较大的研究意义。

1.2 国内外研究现状

Galton^[15]^[16]在 19 世纪末 20 世纪初在《Nature》上首次提出人脸识别技术,

但当时计算机运算能力有限,人脸识别技术没有得到发展。20 世纪 60 年代左右,Chen 和 Bleasoe^[17]等人在 Panorimic Research Inc 上发表了一系列关于人脸识别的技术报告,标志着人脸识别技术研究的开始。人脸识别技术的发展主要分为 3 个阶段:基于几何特征、基于统计特征和基于大量数据的复杂模型。

早期的人脸识别技术研究一直是基于几何特征,受到当时计算机运算速度和存储空间的限制,识别精度难以提升。从 80 年代开始,计算机的性能按照摩尔定律开始飞速提升,关于人脸识别的研究也有了较大的进展,被逐渐应用到了实际场景中。1991 年美国麻省理工学院(MIT)的 M.Turk 和 A.Pentland 提出基于主成分分(PCA)的“特征脸”识别方法^[18]标志着基于统计特征的人脸识别研究的开始。1994 年, G.Yang 等人根据人脸位置的分布,用多个检测窗口来检测判断是否为人脸,提出了一种复杂背景的人脸检测算法^[19]。1997 年, Kotropoulos 等人提出了一种基于人脸先验知识和人脸五官的灰度积分投影进行比对的人脸检测算法^[20]。基于统计特征的人脸识别技术取得了很好的性能,也出现了多家人脸识别相关业务的商业公司。

进入 21 世纪以来,随着计算机性能的极大提升与互联网的快速发展,基于大数据的更复杂的模型计算成为了可能。2001 年, Viola 和 Jones 提出的基于自适应提升(AdaBoost)和类哈尔特征(Haar-like)的人脸检测算法就是以大量数据计算出的复杂模型^{[21] [22]}。该方法运用了几万张带 Haar-like 特征标注的人脸和非人脸图片,构建了 20 万以上数量的特征集,通过 AdaBoost 算法对这些特征进行选择 and 合并,从大量数据中自动学习分类。这一算法相比于当时的传统算法,数据量增大了十倍以上。2006 年左右, Geoffery Hinton 等在《Science》发表了关于深度神经网络(DeepBeliefNetworks, 简称 DBNs)的论文;这篇论文对深度学习的理论研究产生了巨大的影响,关于深度学习的研究开始越来越热^{[23] [24]}。2012 年,基于图形计算处理器(GPU)和海量数据,研究人员训练出具有千万量级的复杂深度神经网络模型,这标志着人脸识别基于大数据和复杂模型的时代到来^{[25] [26]}。

虽然人脸识别技术最早源于美国,我国研究该技术的时间较短,但是经过多年来国内的研究实验和实践,中国的人脸识别技术发展水平已经追赶甚至超过了当前的主流水平。早在 2001 年,公安系统就应用人脸识别技术来识别罪犯,

这一举措得到了国家的大力支持^[27]。现如今，人脸识别被应用在铁路安防、智能城市建设、公司考勤、门店支付^[28]等等，几乎覆盖了生活的方方面面。以后随着技术的快速发展，人脸识别将更加智能，更加快速。

国内外也有不少 AI 公司提供了各个平台下的人脸识别服务。例如 Face++、百度、科大讯飞等公司官网上都提供了相关的 SDK，使开发者很方便的将人脸识别功能集成到自己的产品中。另外，也有一些开源的计算机视觉框架也涉及到人脸识别功能，例如 dlib、OpenCV 等^{[29] [30]}，这些开源框架封装了底层的实现细节，使学者将工作的中心更好的放到人脸识别的算法研究中，而不需要考虑其他无关的细节。而对于 Android 开发者本身的 SDK 中，Google 也提供了人脸检测的 API，小米手机、华为手机的部分机型均集成了人脸检测功能。

1.3 本文主要研究内容与组织结构

1.3.1 本文主要研究内容

本文主要研究了图像预处理、人脸检测、特征提取和人脸识别的算法，以及算法在 Android 平台下的具体实现。

对原始图像预处理算法进行了介绍和研究，首先，对采集的图像进行灰度归一化的操作，增强图像对比度，让不同成像条件下拍摄的同一个人的图像保持一致。然后，对于灰度后的图像进行了直方图均衡化的操作，其作用是对图像进行非线性拉伸，重新分配图像灰度值，扩展像素个数多的灰度级，压缩像素个数少的灰度级，提高了对比度和灰度色调的变化。最后一步预处理工作是对图像进行滤波操作，消除噪点。

对 Adaboost 人脸检测算法的概念进行了介绍，对特征提取算法进行了介绍，对人脸区域进行特征提取，可以有效的降低人脸图像的维数，提高下一步分类器分类识别的效率和准确率。这里重点介绍了基于 PCA 的特征提取算法，并详细描述了 PCA 特征提取的几个过程。

对人脸识别的概念和分类算法进行了介绍，重点介绍了常见的几种机器学习分类器算法的概念和原理。重点研究了最近邻分类器算法和支持向量机分类器算法。通过分析人脸身份认证系统的使用场景和尽可能提高人脸识别准确率，提出了基于最近邻算法（KNN）和支持向量机算法（SVM）相结合的分类器算

法。经过在一些公共数据集上的测试，这种相结合分类器算法比单独使用最近邻算法或支持向量机算法在人脸识别上具有更高的准确率和性能优势。

整个身份认证系统在 Android 平台上的实现。在进行系统需求分析后，根据 Android 系统的特点进行了系统框架的搭建。主要应用了 Android SDK 实现界面和摄像头采集图片的工作，用 C++ 语言和 OpenCV 库实现了图像处理的算法，并用 JNI 和 Android NDK 将两者结合起来，构成一个完成的系统。

1.3.2 本文组织结构

本文的组织结构如下：

第 1 章绪论。综述人脸检测和识别技术的研究背景与意义，人脸识别技术的发展历史与国内外研究现状，以及当前 Android 平台下的人脸识别的发展现状。

第 2 章图像预处理。介绍了几种图像灰度化归一化的方法，通过灰度归一化增强图像对比度，使图像的关键部分更清晰；对直方图的均衡化方法和原理进行了详细的介绍，其作用是对图像进行非线性拉伸，重新分配图像灰度值，扩展像素个数多的灰度级，压缩像素个数少的灰度级，提高了对比度和灰度色调的变化。对图像滤波的几种方式进行了介绍，并分析了优缺点。

第 3 章人脸检测与特征提取。本章介绍了人脸检测概念，并从 Haar 特征、弱分类器、强分类器、级联分类器几个方面详细阐述了基于 Adaboost 的人脸检测算法的具体过程，然后介绍了人脸特征提取的概念，介绍了 PCA 提取人脸特征的具体过程。对人脸识别的概念和分类算法进行了介绍，重点研究了最近邻分类器算法和支持向量机分类器算法。通过分析人脸身份认证系统的使用场景和尽可能提高人脸识别准确率，提出了基于最近邻算法（KNN）和支持向量机算法（SVM）相结合的分类器算法，并测试其效果。

第 4 章 Android 环境搭建和开发技术。概述了 Android 系统的发展历程以及基于 Android 平台开发的优势，其次系统的介绍了 Android 系统开发的应用组件，最后介绍了 Android 开发需要做的准备工作。

第 5 章 Android 下基于人脸识别的身份认证系统的设计与实现。本章首先分析了整个系统的需求，并对这些需求进行了功能划分，然后介绍了各模块的实现。最后分别在 HuaWei Mate9 和 XiaoMi5 手机上测试了实验结果。

第 6 章总结与展望。总结了全文的工作内容，对 Android 平台基于人脸识别

别的身份认证系统的下一步研究给予展望。

2 人脸图像预处理

人脸图像预处理环节是整个身份认证系统的第一个环节，也是比较重要的一个环节。由于在拍照的时候，经常会遇到各种光照、角度、阴影、距离和摄像机质量差异的影响，同样的图像拍摄出来可能差异较大，所以在对图像进行人脸检测和识别之前，必须要对图像进行前期的处理。本节中介绍了三种图像预处理操作，分别是灰度归一化，直方图均衡化，图像滤波。

2.1 灰度归一化

灰度归一化是一种经典的图像处理算法，其目的在于增强图像对比度，使图像的关键部分更清晰，让不同成像条件下拍摄的同一个人的图像保持一致^[31]。灰度归一化算法分为两步，首先是将图像灰度化，即去除彩色信息，只保留灰度信息。转换后的灰度图与之前原始图像描述一致，不会损失一些主要轮廓信息。第二步是归一化。归一化将灰度值的变化范围缩放到 0~1 之间，便于后续环节计算。

颜色空间描述了颜色相关信息的编码，它的种类有很多，本小节主要介绍两种，RGB 和 HSV。

2.1.1 RGB 颜色空间

RGB 色彩模式是工业界的一种颜色标准，RGB 即是代表红、绿、蓝三个通道的颜色，通过这三个通道的叠加可以得到各式各样的颜色，这个颜色几乎包括了人类视力所能感知的所有颜色，是目前运用最广的颜色系统之一^[32]。红、绿、蓝这三个通道每种色各分为 255 阶亮度，值取 0 的时候该通道亮度最低，取 255 的时候亮度最高，红绿蓝三种通道混合后可以取到 1600 多万种颜色。

将一个像素点的 R、G、B 三个分量值按照一定的规则计算即可得到该点的灰度值，不同的规则下得到的灰度值不同。其中较为常用的计算方法有平均值算法，即取三个分量的均值为该点的灰度值。该算法简单易懂，但是这种计算方法忽视了这三个分量互相独立的特点，导致转换过来的灰度图效果不好。另一种较常用的计算方法为加权平均值法，即将三个分量分别乘以一个权值，再求均值作为给点的灰度值。该算法可以通过修改权值来更好的适应分量的特

点。本文经过综合考虑，采用第二种算法。

$$D(x_i, y_i) = \sqrt{(r_i - r_j)^2 + (g_i - g_j)^2 + (b_i - b_j)^2} \quad \text{式 (2.1)}$$

$$D(x_i, y_i) = \sqrt{w_r(r_i - r_j)^2 + w_g(g_i - g_j)^2 + w_b(b_i - b_j)^2} \quad \text{式 (2.2)}$$

加权平均法根据应用场景不同取值也不同， (w_r, w_g, w_b) 通常的取值比例有(3,4,2)，(3,6,1)，(4,8,1)，本文取比例(3,6,1)。

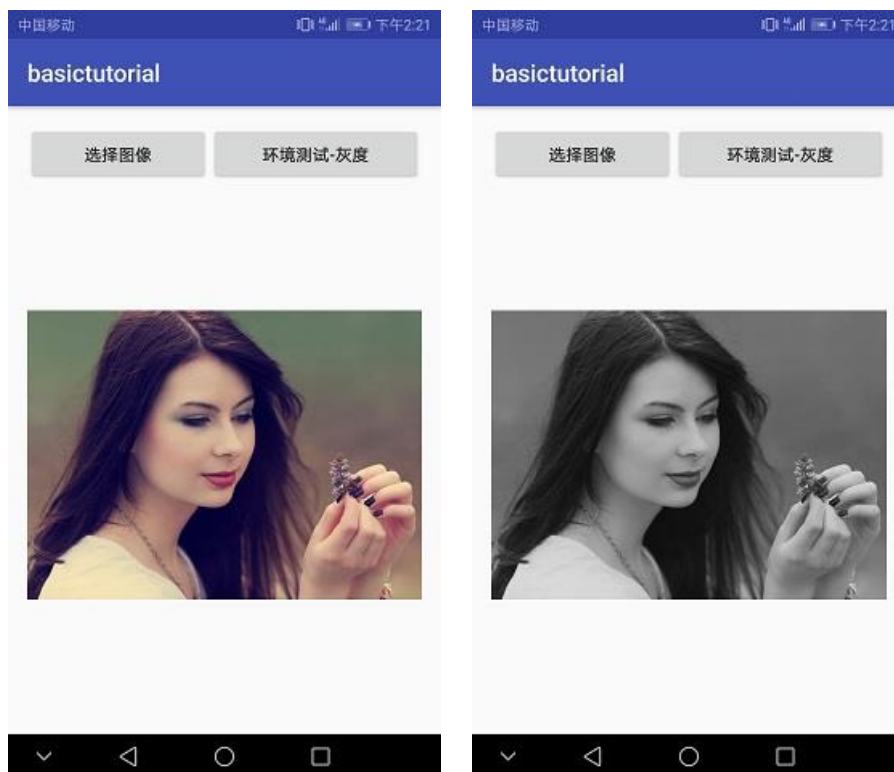


图 2.1 原图（左）与转换后的图（右）

Fig 2.1 Original (left) and converted (right)

2.1.2 HSV 颜色空间

HSV 是 A. R. Smith 在 1978 年提出的，是根据颜色的直观特性的一种颜色空间，也称六角锥体模型^[33]。该模型中颜色参数分别是：色调(H)，饱和度(S)，明度(V)。色调 H 用角度度量，取值范围为 $0^\circ \sim 360^\circ$ ，从红色开始按逆时针方向计算，红色为 0° ，绿色为 120° ，蓝色为 240° 。它们的补色是：黄色为 60° ，青色为 180° ，品红为 300° ；饱和度 S 表示颜色接近光谱色的程度，取值范围为 $0\% \sim 100\%$ ，值越大，颜色越饱和。明度表示颜色明亮的程度，取值范围为 $0\% \sim 100\%$ ，值越大，颜色越亮。从 RGB 颜色空间转换成 HSV 颜色空间的公式如下：

其中 max 表示 R、G、B 三者值中的最大者，min 表示 R、G、B 三者值中的最小者。

$$H = \begin{cases} 0^\circ, & (max = min) \\ 60^\circ * \frac{g-b}{max-min} + 0^\circ, & (max = r, g \geq b) \\ 60^\circ * \frac{g-b}{max-min} + 360^\circ, & (max = r, g \leq b) \\ 60^\circ * \frac{b-r}{max-min} + 120^\circ, & (max = g) \\ 60^\circ * \frac{r-g}{max-min} + 240^\circ, & (max = b) \end{cases} \quad \text{式 (2.3)}$$

$$S = \begin{cases} 0, & (max = 0) \\ \frac{max-min}{max}, & (max \neq 0) \end{cases} \quad \text{式 (2.4)}$$

$$V = max \quad \text{式 (2.5)}$$

为了采用式 2.1、2.2 对 HSV 进行灰度化, 需要将 HSV 转化为 RGB:

$$\begin{aligned} h_i &= \left\lfloor \frac{h}{60} \right\rfloor \\ f &= \frac{h}{60} - h_i \\ p &= v * (1 - s) \\ q &= v * (1 - f * s) \\ t &= v * (1 - (1 - f) * s) \end{aligned} \quad \text{式 (2.6)}$$

2.2 直方图均衡化

直方图均衡化是一种常用的增强图像对比度的算法, 其基本思想是对图像进行非线性拉伸, 重新分配图像灰度值, 扩展像素个数多的灰度级, 压缩像素个数少的灰度级, 提高了对比度和灰度色调的变化^[34]。经过直方图均衡化处理, 可以使得拍摄到的人脸图像的灰度级分布均匀, 增强了图像中明暗区域的对比度, 进一步减少因光照、阴影等干扰给后续环节带来的影响。

直方图均衡化的变换函数如下:

$$s = T(r) = \int_0^r P(r)dr \quad (0 \leq r \leq 1) \quad \text{式 (2.7)}$$

其中, r 表示被增强图像的灰度, 经过灰度归一化的处理后, r 的最大值为 1, 最小值为 0. S 表示变换之后的灰度值。 $P(r)$ 表示概率密度函数, 必须需要满足以下条件:

- (1) 当 $0 \leq r \leq 1$ 时, $T(r)$ 的单调递增, 且 $0 \leq T(r) \leq 1$ 。
- (2) 反变换 $r = T^{-1}(s)$; $T^{-1}(s)$ 也单调递增, 并且 $0 \leq s \leq 1$ 。

对于灰度图分布离散的情况, 需要求出各灰度值的概率之和。计算公式如

下：

$$S_k = T(r_k) = \sum_{i=0}^k \frac{n_i}{N} = \sum_{i=0}^k p_r(r_i) \quad (0 \leq r_i \leq 1, k = 0, 1, 2, \dots, L-1) \quad \text{式 (2.8)}$$

N 为这幅图的像素之和， L 为级别的个数， K 为该级别出现的概率。

该步骤实验过程：

(1) 统计输入每个灰度级像素数目，其中 $i=0,1,2,\dots,L-1$ ， L 表示灰度级的总数目；

(2) 计算各个灰度级的概率密度，公式如下：

$$p_i(r_i) = \frac{n_i}{N} \quad \text{式 (2.9)}$$

(3) 计算累积分布函数：

$$s_k(r_k) = \sum_{i=0}^k P_i(r_i) \quad (k = 0, 1, 2, \dots, L-1) \quad \text{式 (2.10)}$$

(4) 最后的输出灰度级进行计算：

$$g_k = \text{INT} \left[(L-1)S_k(r_k) + \frac{0.5}{(L-1)} \right] \quad \text{式 (2.11)}$$

(5) 通过图像灰度级函数 f_k 和 g_k 的映射关系，修改灰度级的分布，最后得到直方图均衡化后的图像，效果如图 2.2 所示。

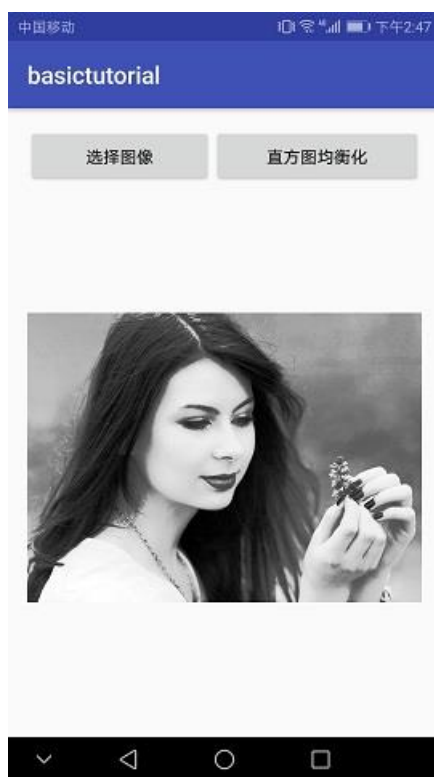


图 2.2 直方图均衡化后的图

Fig 2.2 Histogram equalized graph

2.3 图像滤波

对图像进行滤波可以进一步减少图像的噪声。本节介绍了空间域上的滤波和频域上的滤波。

2.3.1 空间域

(1) 均值滤波

均值滤波器是典型的线性滤波算法，其采用的主要方法为邻域平均法。线性滤波的基本原理是用图像中各像素点周围值的均值代替原图像中的各个像素值^[35]。具体计算公式如下：

$$g = \frac{1}{M} \sum_{(m,n) \in S} f(m,n) \quad \text{式 (2.12)}$$

其中， $x, y = 0, 1, 2, \dots, N-1$ ， S 为 (x,y) 点领域中不包括 (x,y) 点的坐标的集合， M 为集合中坐标点的总数。



图 2.3 4 点邻域图（左）与 8 点邻域图（右）

Fig 2.3 4-point neighborhood graph (left) and 8-point neighborhood graph (right)

图 2.3 表示领域选取最常用的两种方式：4 点和 8 点。领域的点数越大，图像的模糊度也会越大。

(2) 中值滤波器

相比均值滤波，中值滤波是一种非线性滤波算法，更适合处理图像中的孤立噪声。其基本原理是把图像中一点的值用该点的一个邻域中各点值的中值代替，让周围的像素值接近的真实值，从而消除孤立的噪声点^[36]。本文选用中值滤波的方式。

中值滤波的计算公式如下：

$$g(x,y) = \text{Med}\{f(x-k,y-l), (k,l \in W)\} \quad \text{式 (2.13)}$$

其中， $f(x,y)$ ， $g(x,y)$ 分别为原始图像和处理后图像。 W 为二维模板。效果如下：

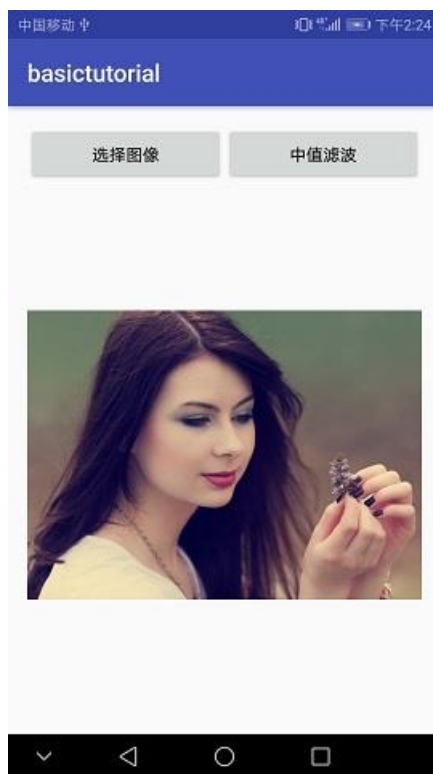


图 2.4 中值滤波后的图

Fig 2.4 Median filtered graph

2.3.2 频域

频域是描述信号在频率方面特性时用到的一种坐标系^[37]。对于图像来说，一般都会存在一些高频的分量，要消除这些高频分量，就需要通过低通滤波器。

(1) 理想低通滤波器

传递函数 $H(u,v)$ 的计算公式如下：

$$H(u,v) = \begin{cases} 1, & D(u,v) \leq D_0 \\ 0, & D(u,v) > D_0 \end{cases} \quad \text{式 (2.14)}$$

其中， D_0 是理想情况下的截止频率。

(2) 指数型低通滤波器

它的传递函数 $H(u,v)$ 为：

$$H(u,v) = \exp\{-[D(u,v)/D_0]^n\} \quad \text{式 (2.16)}$$

其中 D_0 是理想情况下的截止频率， n 为阶数。

(3) 梯形低通滤波器

它的传递函数 $H(u,v)$ 为：

$$H(u,v) = \begin{cases} 1, & D(u,v) < D_0 \\ \frac{D(u,v)-D_0}{D_0-D_1} & D_0 \leq D(u,v) \leq D_1 \\ 0 & D(u,v) > D_1 \end{cases} \quad \text{式 (2.17)}$$

其中 D_0 是理想情况下的截止频率， D_1 为预先设定值， $D_0 \leq D_1$ 。

2.4 本章小结

本章节从三个主要的方面介绍了图像预处理的过程。首先，对采集的图像进行灰度归一化的操作，增强图像对比度，使图像的关键部分更清晰，让不同成像条件下拍摄的同一个人的图像保持一致，归一化后的灰度值保持在 0 到 1 之间，便于后续计算。然后，对于灰度后的图像进行了直方图均衡化的操作，其作用是对图像进行非线性拉伸，重新分配图像灰度值，扩展像素个数多的灰度级，压缩像素个数少的灰度级，提高了对比度和灰度色调的变化。最后一步预处理工作是对图像进行滤波操作，主要分为空间域上的滤波和频域上的滤波。空间域分为均值滤波和中值滤波两种方式，其中中值滤波更利于消除鼓励噪点，本文选用了中值滤波的方式。频域上主要是通过制作低通滤波器的方式消除高频噪点。

3 人脸检测与识别算法

经过第二章的图像预处理操作后，下一步就是人脸检测环节，其目的是检测预处理后的图像中是否存在人脸信息，如果存在，就将其提取出来。在实际使用手机拍照的时候，由于拍摄距离，角度以及环境等因素，拍摄图像经常会携带一些非人脸的图像，直接进行人脸识别难度太大，必须先进行人脸检测来提取图像中人脸的主要区域，将非人脸的图像剔除。所以人脸检测环节是图像预处理与人脸识别之间的一个重要环节，该环节检测的准确率直接影响后续环节人脸识别的效率和准确率。

3.1 基于 Adaboost 的人脸检测算法

Adaboost 是一种迭代算法，其核心思想是针对同一个训练集训练不同的弱分类器，然后把这些弱分类器集合起来，构成一个分类效果更强的强分类器^[38]^[39]。其算法本身是通过改变数据分布来实现的，它根据每次训练集之中每个样本的分类是否正确，以及上次的总体分类的准确率，来确定每个样本的权值。将修改过权值的新数据集送给下层分类器进行训练，最后将每次训练得到的分类器最后融合起来，作为最后的决策分类器。使用 Adaboost 分类器可以排除一些不必要的训练数据特征，并放在关键的训练数据上面。目前 Adaboost 算法被广泛应用在人脸检测算法上。

基于 Adaboost 的人脸检测算法实现步骤如下：

- (1)使用 Harr-like 特征表示人脸，使用“积分图”实现特征数值的快速计算；
- (2)使用 Adaboost 算法挑选出一些最能代表人脸的矩形特征(弱分类器)，按照加权投票的方式将弱分类器构造为一个强分类器；
- (3) 将训练得到的若干强分类器串联组成一个级联结构的层叠分类器，级联结构能有效地提高分类器的检测速度。

3.1.1 Haar 特征

Haar 特征就是矩形特征^[40]。各矩形特征如图 3.1：矩形特征对一些简单的图形结构，比如边缘、线段，比较敏感，但是其只能描述特定走向（水平、垂

直、对角)的结构,因此比较粗略。如图 3.2,脸部一些特征能够由矩形特征简单地描绘,例如,通常,眼睛要比脸颊颜色更深;鼻梁两侧要比鼻梁颜色要深;嘴巴要比周围颜色更深。

对于一个 24*24 检测器来说,其内的矩形特征数量超过 160,000 个,必须通过特定算法甄选合适的矩形特征,并将其组合成强分类器才能检测人脸。

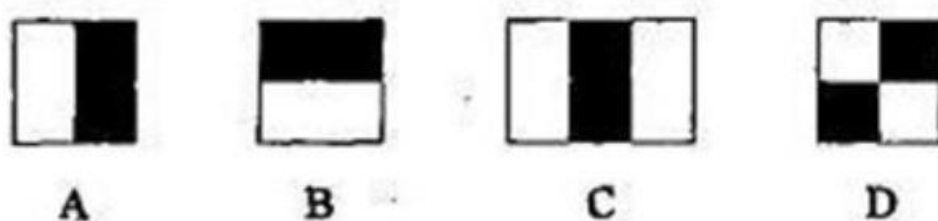


图 3.1 常见矩形特征

Fig 3.1 Common rectangular features

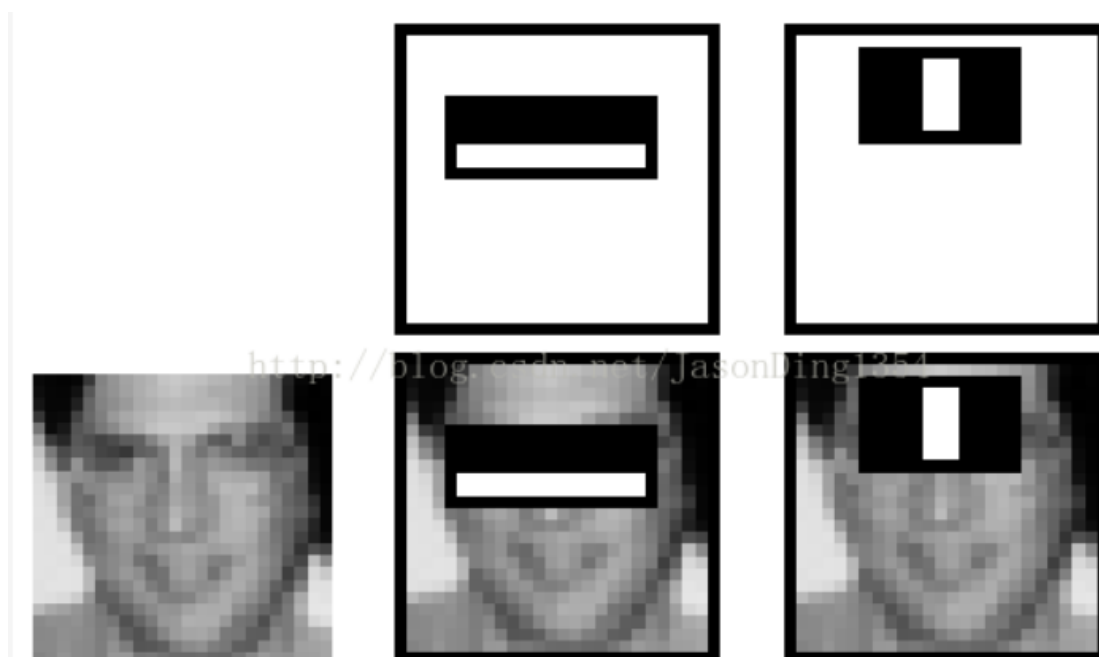


图 3.2 矩形特征在人脸上的特征匹配

Fig 3.2 Feature Matching of Rectangle Features on Human Faces

3.2.2 积分图

积分图是 Haar 特征值计算常用的一种方案,类似动态规划,主要的思想是将图像从起点开始到各个点所形成的矩形区域像素之存在数组中,当要计算某个区域的像素和时可以直接从数组中索引,不需要重新计算这个区域的像素和,从而加快了计算。

积分图每一点的值,等于图像的每一点(x,y)的像素值的左方和上方像素的

总和，即：

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad \text{式 (3.1)}$$

其中式子中的 $ii(x, y)$ 表示图像的积分图， $i(x, y)$ 表示图像的像素图。

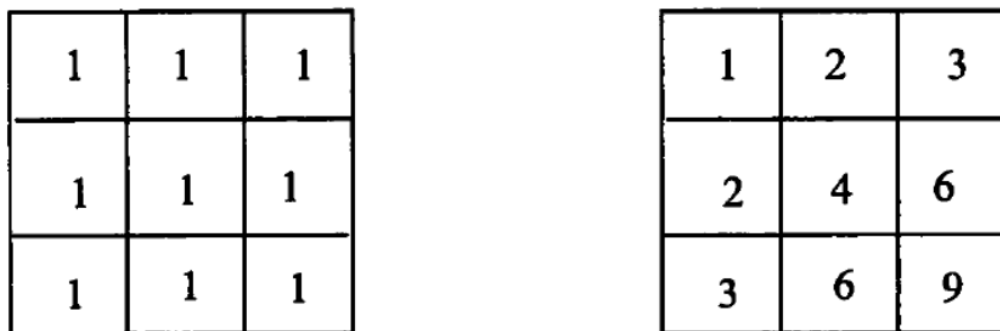


图 3.3 初始图及其积分图

Fig 3.3 Initial graph and its integral graph

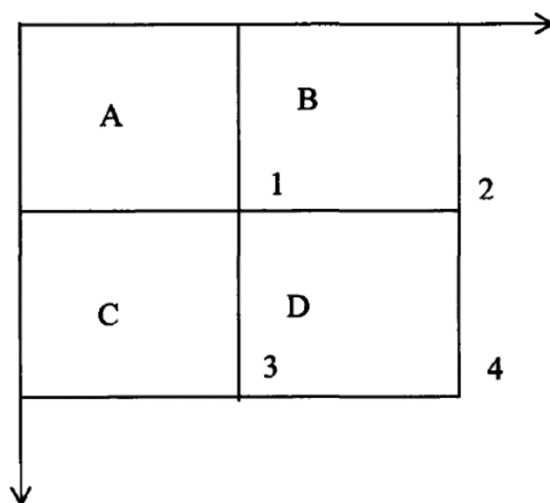


图 3.4 区域像素值计算

Fig 3.4 Area pixel value calculation

根据图 3.4 所示，积分图的计算方法为：

ii_1 = 区域 A 的像素值

ii_2 = 区域 A 的像素值 + 区域 B 的像素值

ii_3 = 区域 A 的像素值 + 区域 C 的像素值

ii_4 = 区域 A 的像素值 + 区域 B 的像素值 + 区域 C 的像素值 + 区域 D 的像素值

通过以上的推导可知，只需要通过 Haar 积分图以及区域端点的和差运算就

可求出 Haar 的值，这也是该算法检测效率高的原因。

3.1.3 分类器

AdaBoost 训练级联分类器的过程主要分为三步。一、计算训练样本的 Haar 特征，并将其训练成弱分类器；二、将这些弱分类器组合起来，可以得到一个强分类器，该强分类器的分类结果是由组合成该强分类器的所有弱分类器投票加权求和得来的；三、将多个强分类器组合起来就得到一个级联分类器，下一个强分类器将通过上一个强分类器的人脸图像再次检测，大大提高了检测的准确率。

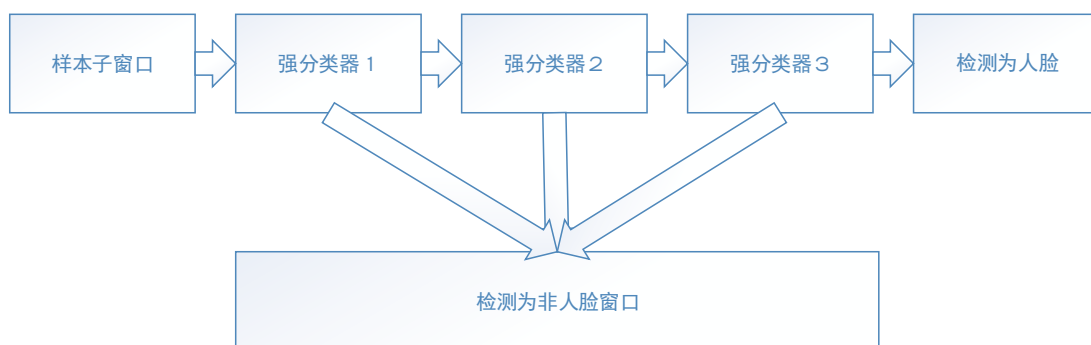


图 3.5 级联分类器的检测流程图

Fig 3.5 Cascade classifier detection flow chart

如图 3.5 所示，Adaboost 级联分类器的设计是由多个强分类器组成，这些分类器一级比一级严格。前一级可以将明显没有人脸特征的窗口排除，后一级则是将上一级分类错误的为人脸排除，所有控制好这些强分类器的检测权重是一个重点和难点。

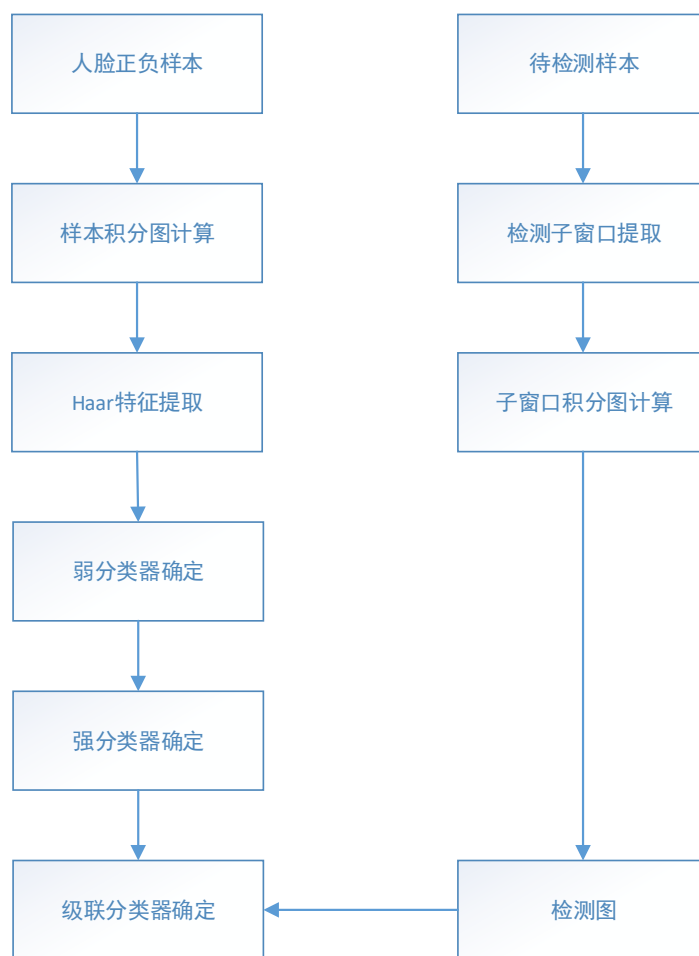


图 3.6 训练与检测图

Fig 3.6 Training and Detection Chart

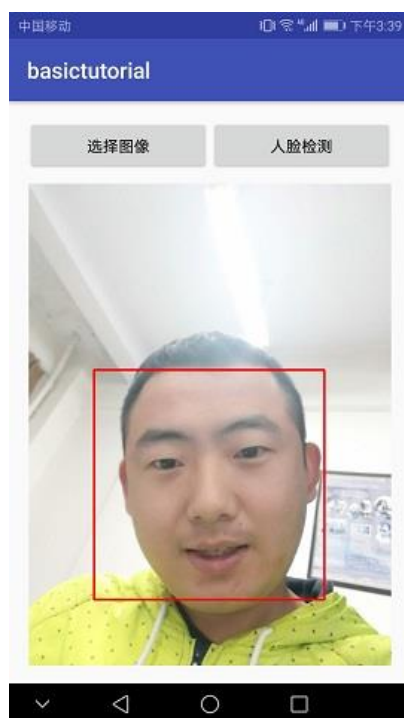


图 3.7 检测效果图

Fig 3.7 Detection effect chart

3.2 基于 PCA 的人脸特征提取算法

经过人脸检测获取到人脸主要区域后，下一步就是人脸特征提取。人脸特征提取主要是提取人脸主要特征，降低人脸特征复杂度，提高下一步人脸识别操作的运算量，提高识别效率^[41]。基于知识的特征提取和基于统计的特征提取是两种的主要方案。其中基于知识的特征提取比较直观，它提取的是一个人们可以理解的物体，例如眼睛、眉毛、嘴巴等。基于统计的特征提取提取的是图像中抽象的特征，常见方法有 PCA、LDA、ICA 等^[42]。本文中整个系统使用的是基于 PCA 的特征提取。在这里，我们首先要介绍下 K-L 变换。

3.2.1 K-L 变换

K-L 变换又被称为主成分变换，优点是去相关性好，是均方误差意义下的最佳变换，是一种比较常用的数据压缩方法。

假定 X 为 n 维矢量集合，其中矢量个数为 N ，那么 \bar{X} 为：

$$\bar{X} = \frac{1}{N} \sum_{i=1}^N \varphi_i \quad \text{式 (3.5)}$$

其中 \bar{X} 为均值， φ_i 表示 X 中的第 i 个矢量。第 i 个矢量与 \bar{X} 的差值 \bar{d} 可以表示为：

$$\bar{d} = \bar{x}_i - \bar{X} \quad \text{式 (3.6)}$$

计算协方差矩阵 P ：

$$P = \frac{1}{N} \sum_{i=1}^N \bar{d}_i \bar{d}_i^T \quad \text{式 (3.7)}$$

P 为 N 阶实对称矩阵，其对角元素为 x_i 的方差，其他元素则为 x_i 和 x_j 的协方差。求出协方差矩阵后，计算 P 正交化后的特征向量 S ，使得式(3.8)成立

$$Q = \frac{1}{N} \sum_{i=1}^N \bar{y}_i \bar{y}_i^T = \frac{1}{N} \sum_{i=1}^N S \bar{d}_i (S \bar{d}_i)^T = S \frac{1}{N} \sum_{i=1}^N \bar{d}_i \bar{d}_i^T S^T = S P S^T \quad \text{式 (3.8)}$$

则 K-L 变换就是将原本的矢量集 X 经过式(3.9)变换成一个新的矢量集 Y 。

$$\bar{y}_i = S \bar{d}_i \quad (i = 1, 2, 3 \dots, N) \quad \text{式 (3.9)}$$

其中，矢量集 Y 的每一个向量是不相关的。由此可以得出，K-L 变换，将原有数据从原有空间变换到了一个新的空间，消除了原有数据之间的相关性。

3.2.2 PCA 提取特征过程

本节介绍的 PCA 主要是通过 K-L 变换来将人脸图像从原有空间变换到一个新的空间，消除原有数据之间的相关性，得到一个线性无关的特征向量，每个特征向量对应一个特征值。这样，在降低了人脸图像特征维数的同时，还能表示人脸图像主要特征。

PCA 特征提取的一般过程如下：

- (1) 将训练样本数据用矩阵表示并进行均值归一化
- (2) 计算协方差矩阵并进行奇异值分解
- (3) 选取最大的前 K 个特征值对应的特征向量
- (4) 输出降维的投影特征矩阵并获得降维后的特征

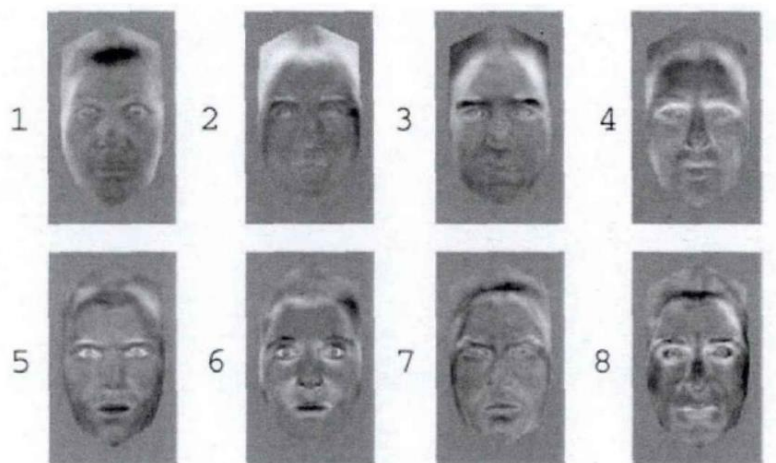


图 3.8 基于 PCA 表情识别的特征脸示意图

Fig 3.8 Face feature map based on PCA face recognition

3.3 基于 KNN 与 SVM 的人脸识别算法

获取到降维的人脸特征后，最后一步就是设计一个人脸识别分类器对降维后的人脸特征进行识别分类，最终达到人脸身份认证的效果。分类器的设计是整个系统中最后一个也是最重要的一个部分，直接决定着系统准确率的高低。

目前主流的人脸识别算法可以归结为基于几何特征、基于统计这两种方法。基于几何特征的方法是最早出现、比较传统的方法，是根据人眼睛、鼻子、嘴巴、下巴等这些部件的形状和结构关系的几何描述作为识别的特征，但是这种方法存在一个问题，一般几何特征只描述了部件的基本形状与结构关系，忽略了局部细微特征，在人脸识别的准确率上达不到要求，运算量也比较大。基于统计的人脸识别算法主要有最近邻算法、神经网络算法和支持向量机算法等，

这一类算法的准确度较高，但是需要一定数量样本的支持。最近邻算法的核心思想是如果一个样本在特征空间中的 k 个最相邻的样本中的大多数属于某一个类别，则该样本也属于这个类别，并具有这个类别上样本的特性^[43]，该算法原理简单易懂，计算速度较快，识别无表情变化的人脸图像时准确率较高。但是一般情况下，拍摄的人脸都会有一些表情的变化，面对这种情况，最近邻算法得不出很好的效果。神经网络算法的优势就是在于可以通过学习，获得人脸图像抽象特征，避免进行复杂的特征提取，有利于软件的实现^[44]。同时神经网络有个很大的缺点，该算法不易解释，表达抽象特征需要的网络层多，导致神经元数量多，运算时间长。支持向量机算法在解决小样本、非线性及高维模式识别中表现出许多特有的优势^[45]，本文中设计的分类器也利用到该分类器算法。

此外，本文设计的基于人脸识别的身份认证系统并不是类似于考勤机的人脸识别系统，而是针对手机持有者身份验证的人脸识别，是一个人脸确认的过程。因为手机持有者只有一个，所以所要做的工作只是确认摄像头拍的人脸是不是手机持有者，在本质上是一个二分类问题。而支持向量机算法优势在于对问题进行二分类，因此本文选用最近邻分类器和支持向量机分类器相结合的分类器作为最终分类器，并且通过测试验证其有效性。

3.3.1 最近邻算法

最近邻算法是一种常用的监督学习方法，其工作机制非常简单：给定测试样本，基于某种距离度量找出训练集中与其最靠近的 k 个训练样本，然后基于这 k 个样本的信息来进行预测^[46]。通常，在分类任务中可使用投票法，即选择这 k 个样本中出现最多的类别标记作为预测结果；在回归任务中可使用平均法，即将这 k 个样本的实值输出标记的平均值作为预测结果；还可基于距离远近进行加权平均或加权投票，距离越近的样本权重越大。

假如有 C 个类别，每类有 N 个样本，则第 i 类 ω_i 的判别函数为：

$$g_i(x) = \min_k \|X - X_i^k\|, k = 1, 2, \dots, N \quad \text{式 (3.10)}$$

其中， X_i^k 表示 ω_i 类的第 k 个样本。

假设 $X = (x_1, x_2, \dots, x_n)$ 是一个需要识别的特征向量， $Y = (y_1, y_2, \dots, y_n)$ 是一个标准样本的特征向量，则表示 X 和 Y 的距离有以下几种方式：

(1) 均方误差距离

$$D(X, Y) = \sum_{i=1}^n |x_i - y_i|^2 \quad \text{式 (3.11)}$$

(2) 欧氏距离

$$D(X, Y) = \sqrt{\sum_{i=1}^n |x_i - y_i|^2} \quad \text{式 (3.12)}$$

(3) 角度距离

$$D(X, Y) = -\cos(X, Y) = -\frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2 \sum_{i=1}^n y_i^2}} \quad \text{式 (3.13)}$$

(4) 曼哈顿距离

$$D(X, Y) = \sum_{i=1}^n |x_i - y_i| \quad \text{式 (3.14)}$$

3.3.2 支持向量机算法

由于手机性能的局限，无法对大量样本进行训练，并且人脸图像的复杂度较高。针对这种小样本、维数高的分类情况，支持向量机算法就是一种很好的解决方案。首先，支持向量机算法在解决样本小、维数高的分类上有这很大的优势；其次，人脸是非刚性的，特征拥有非线性的特点，支持向量机算法能够很好的处理非线性的分类^[48]。本文也采用了该算法。

(1) 线性可分情况下的最优分类面

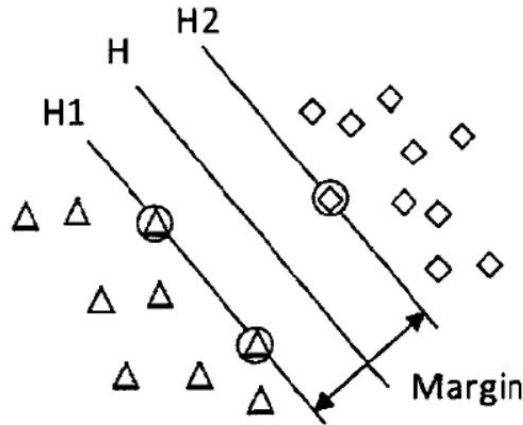


图 3.9 线性可分情况示意图

Fig 3.9 Linear separability

如图 3.9 所示，分别用三角行和菱形表示不同的训练样本，H 表示将不同类别的训练样本正确分开的分类线。最优分类线为正确分类前提下使分类间隔线之间距离最大的分类线^[49]。同样，再多维空间中，最优分类面为正确分类前提下使分类间隔面之间距离最大的分类面。

假设一个线性可分的样本集为 $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ 。其中， x_i 为样本特征向量， X 是 d 维实数空间中的向量，类标签 $y \in \{-1, +1\}$ 。

则分类面方程为：

$$\omega \cdot x + b = 0 \quad \text{式 (3.15)}$$

其中， ω 为权值向量， b 是分类平面的常数项。当 $g(x) \geq 0$ 时， y_1 取+1，反之， y_1 取-1。对判断函数 $g(x)$ 归一化，则：

$$\begin{cases} \omega \cdot x_i + b = -1, y_i = -1 \\ \omega \cdot x_i + b = 1, y_i = +1 \end{cases} \quad \text{式 (3.16)}$$

如果数据点 x_i 满足式(3.16)，则称 y_i 为支持向量。通过计算可知，最优分类面的单位法向量为：

$$\omega_0 = \frac{\omega}{\|\omega\|} \quad \text{式 (3.17)}$$

设 x_1, x_2 分别为平面 $H1$ 和 $H2$ 上的点，最大间隔 Dist 即是向量 $(x_1 - x_2)$ 在单位向量 ω_0 上的投影。即：

$$\text{Dist} = (x_1 - x_2) \cdot \frac{\omega}{\|\omega\|} \quad \text{式 (3.18)}$$

又因为 x_1, x_2 分别为平面 $H1$ 和 $H2$ 上的点，所以有：

$$\begin{cases} \omega \cdot x_i + b = +1 \\ \omega \cdot x_i + b = -1 \end{cases} \quad \text{式 (3.19)}$$

整理式得：

$$\omega \cdot (x_1 - x_2) = 2 \quad \text{式 (3.20)}$$

由以上公式可知：

$$\text{Dist} = (x_1 - x_2) \cdot \frac{\omega}{\|\omega\|} = \frac{2}{\|\omega\|} \quad \text{式 (3.21)}$$

由此可知,当 $\|\omega\|$ 最小时，间隔 Dist 最大。其约束条件为：

$$y_i(\omega \cdot x_i + b) \geq 1, i \in \{1, 2, \dots, N\} \quad \text{式 (3.22)}$$

因此，求解最优分类面问题就被转换为求解二次函数最优规划问题，拉格朗

日函数定义为:

$$L(\omega, b, \alpha) = \frac{1}{2} \omega^T \omega - \sum_{i=1}^n \alpha [y_i (\omega \cdot x_i + b) - 1], \quad \alpha_i \geq 0 \quad \text{式 (3.23)}$$

其中 α_i 表示拉格朗日乘子, 函数两边求偏导得:

$$\frac{\partial L(\omega, b, \alpha)}{\partial \omega} = 0 \Rightarrow \omega = \sum_{i=1}^N \alpha_i y_i x_i \quad \text{式 (3.24)}$$

$$\frac{\partial L(\omega, b, \alpha)}{\partial b} = 0 \Rightarrow \sum_{i=1}^N \alpha_i y_i = 0 \quad \text{式 (3.25)}$$

由以上两个公式, 整理得:

$$L(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j x_i^T x_j \quad \text{式 (3.26)}$$

约束条件为: $\sum_{i=1}^N \alpha_i y_i = 0, \alpha_i \geq 0$

问题变成了拉格朗日对偶问题, 可根据经典算法求解出 α^*, ω^*, b^* , 最后得到分类函数:

$$f(x) = \text{sgn}[(\omega^* \cdot x) + b^*] = \text{sgn}\left\{\sum_{j=1}^n \alpha_j y_j (x_j \cdot x) + b^*\right\} \quad \text{式 (3.27)}$$

其中, sgn 为符号函数, x_j 为支持向量, b^* 为阈值。

(2) 线性不可分情况下的最优分类面

因为人脸可以做出许多表情, 所以一般提取的特征往往是非线性的。在这种情况下, 上述介绍的求解方法会失效。此时可以通过在约束条件中引入松弛变量 $\zeta_i \geq 0$ 的方式求解^[50]。首先, 调整约束条件为:

$$y_i (\omega \cdot x_i + b) \geq 1 - \zeta_i, \zeta_i \geq 0, (i = 1, 2, \dots, N) \quad \text{式 (3.28)}$$

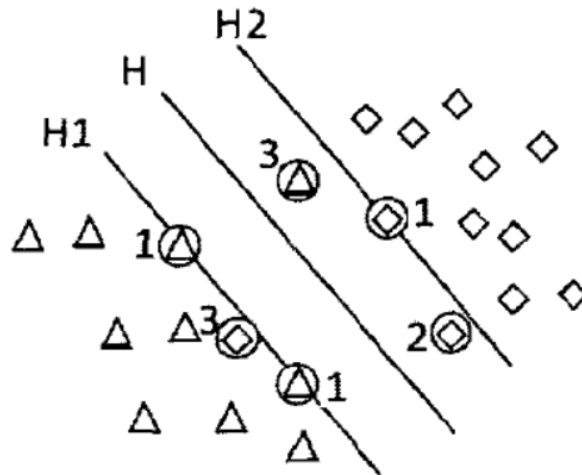


图 3.10 线性不可分情况示意图

Fig 3.10 Linearly inseparable case

- (1) 当 $\zeta_i = 0$ ，对应着能够正确线性分类的样本。如图 3.10 所示在分类间隔线外侧的样本。
- (2) 当 $0 < \zeta_i < 1$ ，对应着没有被错误分类的样本，如图 3.10 中在分类间隔线之间的标号为 1 的样本。
- (3) 当 $\zeta_i > 1$ ，对应允许可以被错误分类的样本，如图 3.10 中在分类线周围标号为 3 的样本。

引入代价系数 C 后，目标函数转变为：

$$\Psi(\omega, x, b) = \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^N \zeta_i \quad \text{式 (3.29)}$$

其中， Ψ 为目标函数， C 为错误代价系数，所以求得拉格朗日函数为：

$$L(\omega, x, \alpha) = \frac{1}{2} \|\omega\|^2 + C \sum_{i=0}^n \zeta_i - \sum_{i=0}^n \mu_i \zeta_i - \sum_{i=1}^n \alpha_i [y_i(\omega \cdot x_i + b) - 1 + \zeta_i] \quad \text{式(3.30)}$$

经过公式变换，最终得到下面的对偶问题。

最大化 $L(\alpha)$ ：

$$L(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \quad \text{式 (3.31)}$$

约束条件为：

$$\sum_{i=0}^N \alpha_i y_i = 0, 0 \leq \alpha_i \leq C, i = 1, 2, \dots, N \quad \text{式 (3.32)}$$

由式(3.32)可以看出线性可分与不可分的目标函数表达式完全相同，只是约束条件 $\alpha_i \geq 0$ 不同，在线性不可分情况下约束条件为 $0 \leq \alpha_i \leq C$ ，所以线性可分可以看做线性不可分情况下的一种特例。

(3)核函数

核函数的思想为：通过非线性变换，将输入空间映射到高维特征空间。使其线性可分并且在该高维特征空间中寻找最优的分类面^[51]。通过加入核函数就将非线性问题转化为高维特征空间中的线性问题。

令 $\phi(x) = [\phi_1(x), \phi_2(x), \dots, \phi_M(x)]^T$ ，其中输入向量 x 表示 N 维的输入向量，在新空间中构建的分类超平面为：

$$\sum_{j=1}^M \omega_j \phi_j(x) + b = 0 \quad \text{式 (3.33)}$$

其中的 ω_j 为权值系数， b 为偏置或负阈值。令 $\phi(x)^*=1$, $\omega^*=b$,以上公式可以简化为

$$\sum_{j=1}^M \omega_j \phi_j(x) = 0 \quad \text{式 (3.34)}$$

也可以表示为:

$$\omega \cdot \phi(x) = 0 \quad \text{式 (3.35)}$$

根据适合线性可分模式输入空间的式，将需要 $\phi(x)$ 替换 x ，得到:

$$\omega = \sum_{i=1}^N \alpha_i y_i \phi(x_i) \quad \text{式 (3.36)}$$

将式(3.36)代入式(3.35)即可得出特征空间的分类超平面为:

$$\sum_{i=1}^N \alpha_i y_i \phi^T(x_i) \phi(x_i) = 0 \quad \text{式 (3.37)}$$

由式可得，仅使用内积即可构造出最优超平面。但同时需要一个函数 $K(x, x_i)$ ，使得构造超平面的时候避免考虑变换的形式。

$$K(x, x_i) = \phi^T(x) \phi(x_i) = \sum_{j=1}^M \phi_j(x) \phi_j(x_i) \quad \text{式 (3.38)}$$

该函数 $K(x, x_i)$ 被称为内积核函数。

这里得出核函数的定义： $K(x, y)$ 为连续的对称核， x 与 y 在闭区间 (a, b) 中。

核函数 $K(x, y)$ 可以展开为级数:

$$K(x, y) = \sum_{i=1}^{\infty} \lambda_i \phi_i(x) \phi_i(y) \quad \text{式 (3.39)}$$

其中所有的 $\lambda_i > 0$, 保证式一致收敛的充要条件是:

$$\int_b^a \int_b^a K(x, y) \phi(x) \phi(y) dx dy \geq 0 \quad \text{式 (3.40)}$$

对于所有满足 $\int_b^a \phi^2(x) dx < \infty$ 的 $\phi(x)$ 成立。

常用的核函数可分为如下几类

(1)线性核函数

$$K(x, y) = x \cdot y \quad \text{式 (3.41)}$$

(2)多项式核函数

$$K(x, y) = [(x \cdot y) + 1]^d, d = 1, 2, \dots \quad \text{式 (3.42)}$$

(3)径向基核函数

$$K(x, y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right) \quad \text{式 (3.43)}$$

其中， σ 是由用户决定的核宽度。

基于以上概念，得出支持向量机算法步骤为：

假设训练样本 $X\{(x_1, y_1), (x_2, y_2), \dots (x_n, y_n)\}$ ，其中， x_i 是 d 维实数空间中的向量，类标签 $y_i \in \{-1, +1\}$ 。

步骤 1：在约束条件

$$\sum_{i=1}^N \alpha_i y_i = 0, 0 \leq \alpha_i \leq C \quad \text{式 (3.44)}$$

下求解使目标函数：

$$L(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (x_i x_j) \quad \text{式 (3.45)}$$

最大化的 α^* 。

步骤 2：计算最优权值：

$$\omega^* = \sum_{i=1}^N \alpha^* y_i Y_i \quad \text{式 (3.46)}$$

其中 Y_i 为输出向量。

步骤 3：计算分类判别函数：

$$f(x) = \text{sgn}[\sum_{i=1}^N \alpha^* y_i K(x_i, x) + b^*] \quad \text{式 (3.47)}$$

根据 $f(x)$ 的值判定 X 的类别归属。

输出为 $f(x)$ 的值

3.3.3 最近邻算法和支持向量机算法相结合的分类器算法

由上一小节可知，支持向量机分类器原理是基于统计的训练模型，如果要分类的人脸图像在数据库中不存在时，支持向量机分类器很难进行拒识判断。而且由于手机性能的限制，无法训练大量的人脸图像，只能选择一些代表性的人脸模型作为训练的负样本。同时为了提高速度和增加拒识判断，在使用支持向量机分类器之前使用最近邻分类器再进行一次分类。最近邻分类器计算速度快，效率高，虽然不能提高识别的准确率，但是却可以有效地降低，误识率，适用于本系统。

本节提出了一种将运算速度较快的最近邻分类器和识别率高的支持向量机分类器相结合的方式。该算法首先使用最近邻算法进行分类并设定阈值，如果得到的分类结果小于设定的阈值，则继续用支持向量机分类器进行再次分类，否则，则拒识。

3.3.4 实验结果与分析

为了验证上一节中提到的分类器的效果，本节中分别对最近邻分类器、支持向量机分类器和两者相结合的分类器做了测试。样本为美国国防部高级研究项目署和美国陆军研究实验室共同建立的 FERET 人脸库，采用 PCA 特征提取方法，通过测试对比这几种分类器算法在分类识别过程中的准确率。

本文所采用的人脸库是 FERET 人脸库的一部分，包括 10 名志愿者的 100 幅人脸图像和一些测试图像。

每次训练选定同一个人的 10 张人脸图像作为正样本，其他的 90 张为负样本，共需要进行 10 次训练。每次训练完成后使用测试样本进行测试，结果取均值。测试结果见表 3.1：

表 3.1 基于 FERET 子库的实验结果

Table 3.1 Experimental results based on the FERET sub library

	最近邻算法	支持向量机算法	最近邻和支持向量 机相结合算法
正确识别率	76.7%	88.3%	90.4%
错误识别率	10.8%	17.6%	5.4%
拒绝识别率	23.3%	11.7%	9.6%
正确拒绝率	89.2%	82.4%	95.6%

如表 3.1 所示，正确识别率表示测试样本是本人且被正确识别的情况；错误识别表示测试样本不是本人但却被识别的情况；拒绝识别表示测试样本是本人但却没有被识别的情况；正确拒绝表示测试样本不是本人并且没有被识别的情况。

从上述测试中可以得出结论，本文提出的将最近邻算法和支持向量机算法相结合后实现的分类器算法的正确识别率和正确拒绝率是最高的。

3.4 本章小结

本小节主要介绍了三部分内容，一是基于 Adaboost 的人脸检测算法；二是基于 PCA 的人脸特征提取算法；三是几种人脸识别的训练算法，通过分析，提出了一种最近邻与支持向量机相结合的人脸识别算法，并通过实验证明了该算

法的可行性。

4 Android 环境搭建和开发技术

4.1 Android 系统概述与优势

Android 是一种面向智能手机的基于 Linux 内核的操作系统，它最初由 Andy Rubin 开发，Andy Rubin 开发这个系统的最初目的是为了创建一个智能的数码相机操作系统，但是当后来智能手机市场快速发展的时候，Android 被改造为主要面向智能手机的操作系统。后来 Android 系统被 Google 公司收购，成为 Google 旗下的一部分。在 2017 年 Google IO 开发者大会上，Google 宣布目前全球 Android 设备月活跃用户已经达到 20 亿。Android 系统以 Linux 为内核，采用多级分层的架构，包括 Linux kernel、Applications、Framework 以及 Libraries 组成。由于 Android 系统强大的可移植性和 Google 公司开源的性质，Android 系统理论上可以在任何电子产品上运行，这对以后产生更智能化的电子产品有着巨大的影响和推动作用。Android 系统得到了众多手机硬件厂商的支持，移动终端市场占有率年年升高。

和其他操作系统相比，Android 系统是开源的操作系统，拥有众多的开发者，更有着 Google 公司的支持。Android 系统拥有完整的开发生态，从底层运行到应用程序的开发和发布都有着强大的监管措施，极大地推动了移动产业的发展。因为 Android 系统的开放性和开源性允许任何移动设备制造厂商加入，对于众多的开发者来说，有着大量的资源可以利用。对于用户来说，不仅可以用到全世界的优秀开发者开发的应用程序，还可以用众多开放的源代码进行二次开发。开放的平台会带来了良性的竞争，能更好的服务用户，让用户们能够用到更优质的产品。

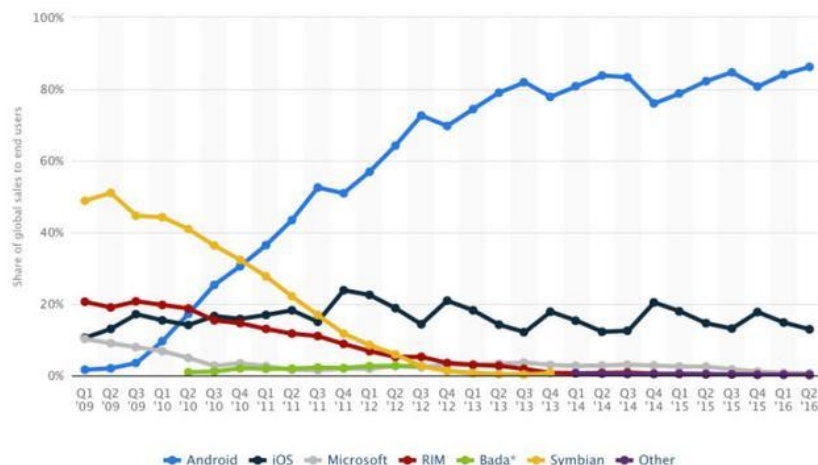


图 4.1 智能手机系统市场份额占比

Fig 4.1 Smartphone system market share

Google 公司为 Android 开发者提供了一个自由的开发平台,去除各种限制,激励应用开发者的创新。Android 系统中所有应用程序均运行 Java 虚拟机上,该虚拟机分配给应用程序需要的硬件资源和通信接口,不同应用程序之间的信息可以通过一定的方式共享。Android 平台提供的 SDK 使得开发者能够更好的和硬件结合,大大缩短了开发周期。

4.2 Android 应用组件

应用组件是 Android 应用的基本构建基块。Android 共有四种不同的应用组件类型,每种组件都服务于不同的目的,并且具有定义组件的创建和销毁方式的不同生命周期。这四种组件分别是: Activity 活动, Service 服务, BroadcastReceiver 广播, ContentProvider 内容提供者。

4.2.1 Activity 活动

Activity 是一个重要的应用组件,相当于一个窗口,用户可以通过触摸的方式与其进行交互,来执行打电话、发短信、发电子邮件等操作。创建一个 Activity 就会获得一个窗口。窗口可以被设置为充满屏幕或是以弹窗的形式出现。

生命周期描述的是该 Activity 被创建可见到不可见直至被销毁的过程,在这个过程中,Activity 一般以四种状态存在。运行状态:一个新的 Activity 创建

后，会显示在整个屏幕上，用户可以通过触摸的方式与其进行交互。这种状态下 Android 会尽最大可能保持该 Activity 运行，当内存不足时，会优先杀掉其它 Activity 来释放内存。暂停状态：当 Activity 处于此状态时，它已经失去焦点，不可与用户交互。停止状态：当 Activity 不可见时会处于这个状态，当这个状态触发时，一定要保存好当前的数据和 UI 状态，否则会丢失数据。死亡状态：当 Activity 被杀掉以后就处于死亡状态，这时 Activity 已从堆栈中移除。Activity 生命周期流程如图 4.2。

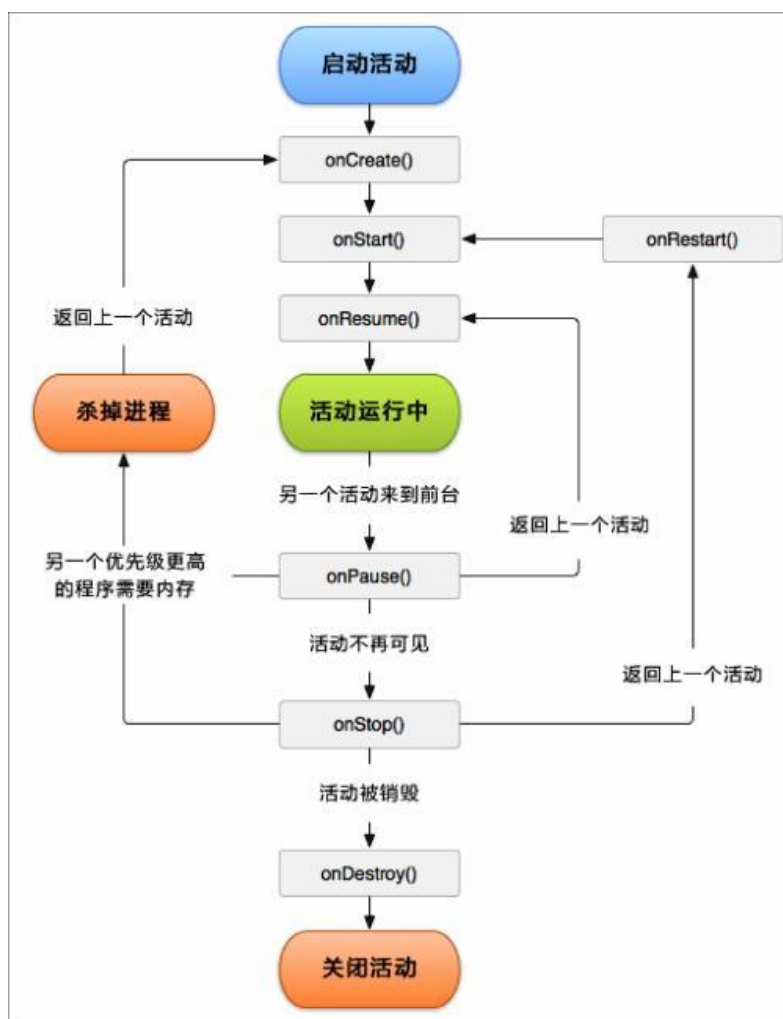


图 4.2 Activity 的生命周期

Fig 4.2 Activity life cycle

4.2.2 Service 服务

Service 是一个可以在后台执行长时间运行而不提供界面的应用组件，可以由其他应用组件启动，而且即使用户切换到其他应用，Service 仍将在后台继续

运行。服务一般分为两种形式：一种是通过其它应用组件直接启动，这一种服务启动后与其它应用组件没有交互，并在后台一直运行，即使启动服务的组件已被销毁也不受影响；另一种是其他应用组件通过将自己绑定的方式启动服务，这种方式允许组件与服务进行交互、发送请求、获取结果，当启动服务的组件被销毁时该服务也被销毁。这两种 Service 的生命周期不同，流程如图 4.3。

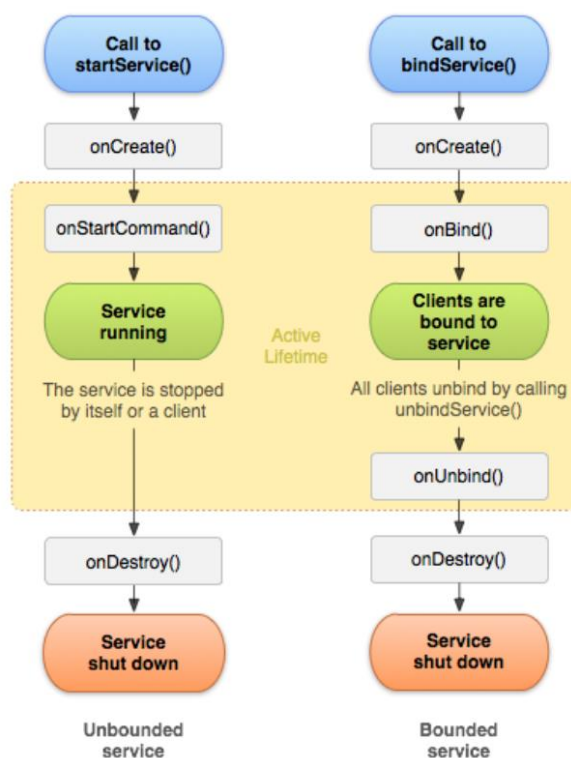


图 4.3 Service 的生命周期

Fig 4.3 Service life cycle

4.3 Android 开发准备

4.3.1 JDK

Android 开发首先要搭建 java 开发环境，JDK 是 Java 语言的软件开发工具包，JDK 是整个 java 开发的核心，它包含了 JAVA 的运行环境和 JAVA 调试工具。首先去官网下载 JDK，因为本人电脑是 Windows10 64 位系统，所以下载 JDK-Windows-X64，下载完成后将 JDK 运行包按照默认路径安装。安装完 JDK 之后配置系统环境变量，如图 4.4 所示。

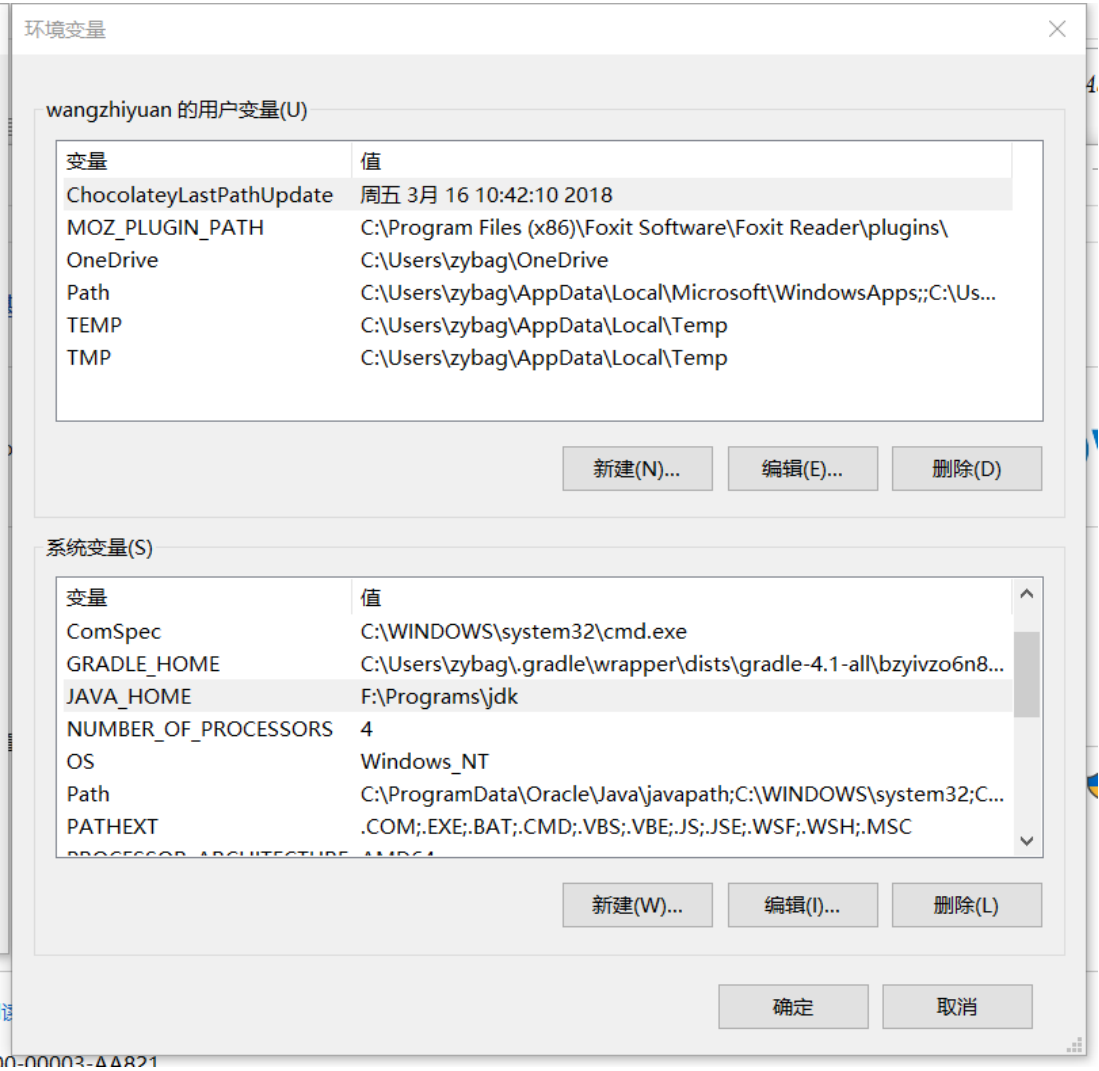
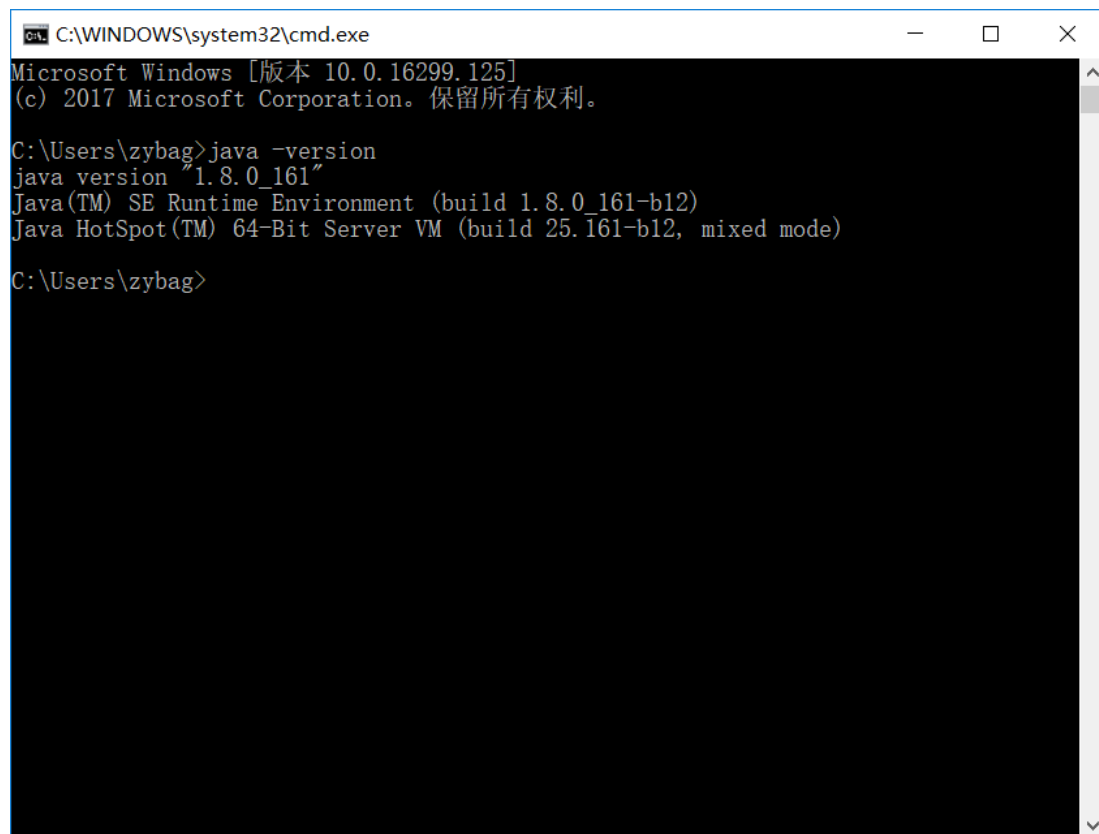


图 4.4 Java 环境变量配置

Fig 4.4 Java environment variable configuration

在我的电脑属性中找到环境变量配置选项，创建一个名称为 JAVA_HOME，值为 JDK 安装目录中 bin 文件夹绝对路径的系统变量。将该变量添加到 PATH 路径下。打开 cmd 终端，输入 java -version 命令测试，变量配置成功后如图 4.5 所示。



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.16299.125]
(c) 2017 Microsoft Corporation。保留所有权利。

C:\Users\zybag>java -version
java version "1.8.0_161"
Java(TM) SE Runtime Environment (build 1.8.0_161-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.161-b12, mixed mode)

C:\Users\zybag>
```

图 4.5 Java 环境测试

Fig 4.5 Java environment test

4.3.2 Android Studio

Android Studio 是 Google 于 2013 I/O 大会针对 Android 开发推出的新的开发工具。比起 Eclipse，速度更快，更加智能，整合了 Gradle 构建工具，内置终端，完美整合版本控制系统，使得越来越多的人倾向与使用 Android Studio 开发。这里便是使用 Android Studio 来开发整个系统。

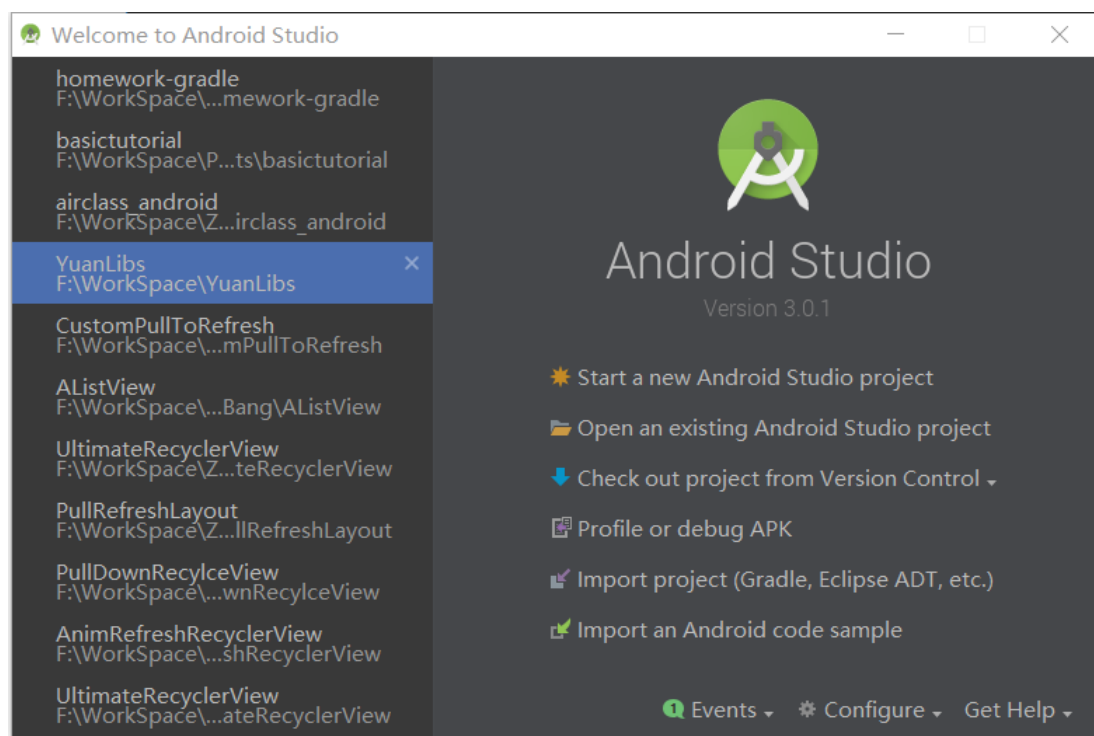


图 4.6 Android Studio 界面

Fig 4.6 Android Studio interface

4.3.3 JNI 技术与 Android NDK

JNI 是 Java Native Interface 的缩写，是 Java 生态的一部分。它提供了若干的 API 实现了 Java 和其他语言的通信。由于图像处理过程中计算量较大，为了提高系统执行效率，在较短的时间内完成人脸检测和识别，本文中核心人脸识别算法都是基于 C++ 语言进行开发的，然后再用 Java 语言调用 C++ 层的方法，所以这里必须使用 JNI 技术。

NDK 是 Native Development Kit 的缩写，是当开发者使用 C 语言等底层语言开发程序时，防止 Android 程序在 JVM 虚拟机中与 JAVA 语言冲突而开发的一套工具。NDK 支持在 Android 应用加入 C++ 语言开发。Android NDK 提供了一系列交叉编译工具，这些工具被用来支持对调用本地代码的 JNI 接口。NDK 中封装了一些底层的库函数供开发者使用。通过 NDK，Android 开发者将拥有更高的效率和灵活性。具体调用过程如图 4.7 所示。

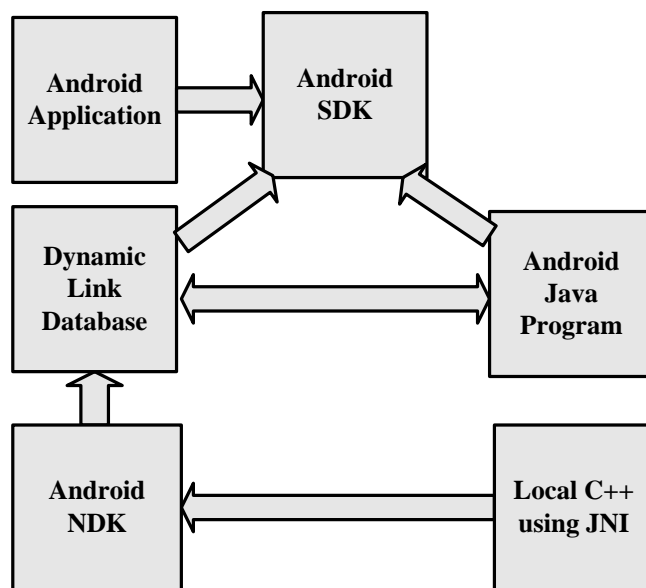


图 4.7 JNI 调用流程图

Fig 4.7 JNI call flow diagram

4.3.4 OpenCV

OpenCV 是一个开源的计算机视觉库，拥有超过 2500 种优化算法，其中包括一些经典的和最先进的图像处理和机器学习算法，这些算法可用于检测和识别人脸，识别物体，视频中的人的行为进行分类，跟踪摄像机运动，运动目标的跟踪，提取物体的三维模型等等。OpenCV 目前被广泛的应用在公司、个人和高校的研究和开发中。

OpenCV 主要包含下面几个模块：

- (1)core 模块：核心功能模块，定义了基本的数据结构和文件读写等操作。
- (2)imgproc 模块：图像处理模块，包括图像滤波、集合图像变换、直方图计算、形状描述子等
- (3)highgui 模块：高级图形界面，用来显示图像或视频到界面上，并提供了一些按钮和滑动条控件。
- (4) video 模块：视频分析模块。包括背景提取、光流跟踪、卡尔曼滤波等算法。
- (5) objdetect 模块：物体检测模块。包括 haar 分类器、SVM 检测器及文字检测。
- (6) ml 模块：即机器学习模块。包括统计模型、K 最近邻、支持向量机、

决策树、神经网络等经典的机器学习算法。

本文所有的图像处理算法是基于 OpenCV 的基础上开发的。首先从 OpenCV 官网下载源码，因为本文是基于 Android 平台开发的，所以我们选择 OpenCVForAndroid 这个版本。

4.4 本章小结

本节首先概述了 Android 系统的发展历程以及基于 Android 平台开发的优势，其次系统的介绍了 Android 系统开发的应用组件，最后介绍了 Android 开发需要做的准备工作，下一章将是整个系统的开发和测试。

5 基于 Android 平台的身份认证系统的设计与实现

5.1 系统功能分析

根据之前的分析，将整个身份认证系统分为图像采集、图像预处理、人脸检测、人脸特征提取与人脸识别这五个模块。具体流程如图 5.1。



图 5.1 人脸识别身份认证系统流程图

Fig 5.1 Face recognition authentication system flow chart

图像采集模块：利用 Android 平台摄像头进行图像采集，实现调用摄像头、对拍摄的物体进行自动对焦、连续拍照等功能，快速获取图像帧的信息。

人脸图像预处理模块：对采集到的图像帧进行灰度归一化、直方图均衡化和滤波去噪等处理。

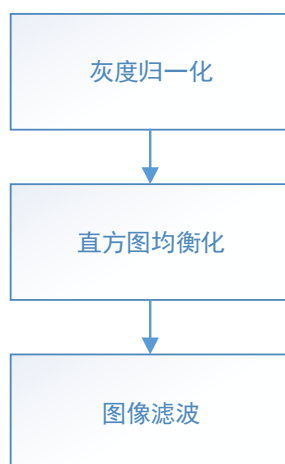


图 5.2 图像预处理流程图

Fig 5.2 Image preprocessing flow chart

人脸检测模块：经预处理的图像采用Adaboost 人脸检测方法获取人脸，并对裁剪出的人脸图像进行标记。

人脸特征提取模块：从人脸检测模块中获得人脸关键区域后，计算人脸的PCA 特征值并将特征值传递到人脸识别模块中。

人脸识别模块：根据测试者人脸图像计算人脸 PCA 特征，得到识别结果。如果测试者的人脸特征在我们设置的阈值范围内，则确认测试人的身份为手机持有者，否则提示该人不是手机持有者，请摆正人脸重新识别。

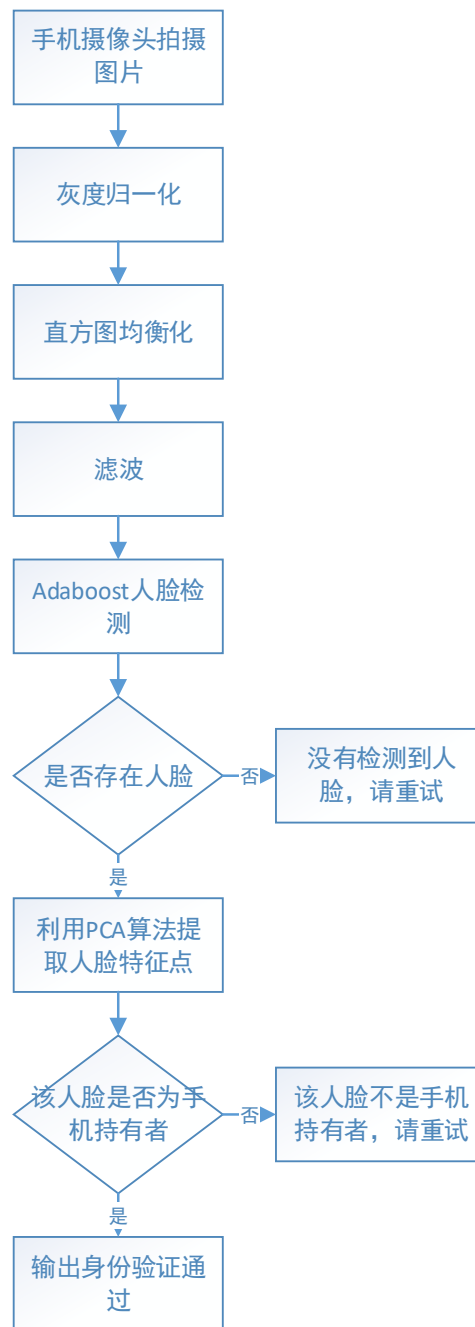


图 5.3 系统整体流程图

Fig 5.3 System overall flow chart

5.2 实验与结果分析

首先需要对手机持有者进行注册，为了有更好的识别效果，需要拍摄 10 张手机持有者不同表情下的照片。将手机持有者 10 张图像作为正样本，选取其他同学的 90 张图像作为负样本进行训练。

随机选取一个同学进行拍照，从手机摄像头采集到图像后经过预处理，再到人脸检测，识别出人脸关键区域，将人脸关键区域提取特征之后与数据库里存的手机持有者的人脸信息进去对比，特征相似度超过一定的阈值即可判定该人脸是手机持有者本人。识别效果与部分代码如下：

```
public MatonCameraFrame(CvCameraViewFrame inputFrame) {
    mRgba = inputFrame.rgba();
    mGray = inputFrame.gray();
    if (mAbsoluteFaceSize == 0) {
        int height = mGray.rows();
        if (Math.round(height * RELATIVE_FACE_SIZE) > 0) {
            mAbsoluteFaceSize = Math.round(height * RELATIVE_FACE_SIZE);
        }
    }

    if (mJavaDetector != null) {
        MatOfRect faces = new MatOfRect();
        mJavaDetector.detectMultiScale(mGray, faces, 1.1, 10,
            CV_HAAR_FIND_BIGGEST_OBJECT
            | CV_HAAR_SCALE_IMAGE
            | CV_HAAR_DO_ROUGH_SEARCH
            | CV_HAAR_DO_CANNY_PRUNING,
            new Size(mAbsoluteFaceSize, mAbsoluteFaceSize),
            new Size(mGray.width(), mGray.height()));

        Rect[] facesArray = faces.toArray();
        for (Rect aFacesArray : facesArray) {
            Core.rectangle(mRgba, aFacesArray.tl(), aFacesArray.br(), FACE_RECT_COLOR, 3);
            if (null != mOnFaceDetectorListener) {
                mOnFaceDetectorListener.onFace(mRgba, aFacesArray);
            }
        }
    }
    return mRgba;
}
```

图 5.4 人脸关键区域检测代码

Fig 5.4 The implementation of face detection

```
public static double compare(Context context, String fileName1, String fileName2) {
    try {
        String pathFile1 = getFilePath(context, fileName1);
        String pathFile2 = getFilePath(context, fileName2);
        IplImage image1 = cvLoadImage(pathFile1, CV_LOAD_IMAGE_GRAYSCALE);
        IplImage image2 = cvLoadImage(pathFile2, CV_LOAD_IMAGE_GRAYSCALE);
        if (null == image1 || null == image2) {
            return -1;
        }
        int l_bins = 256;
        int hist_size[] = {l_bins};
        float v_ranges[] = {0, 255};
        float ranges[][] = {v_ranges};
        IplImage imageArr1[] = {image1};
        IplImage imageArr2[] = {image2};
        CvHistogram Histogram1 = CvHistogram.create(1, hist_size, CV_HIST_ARRAY, ranges, 1);
        CvHistogram Histogram2 = CvHistogram.create(1, hist_size, CV_HIST_ARRAY, ranges, 1);
        cvCalcHist(imageArr1, Histogram1, 0, null);
        cvCalcHist(imageArr2, Histogram2, 0, null);
        cvNormalizeHist(Histogram1, 100.0);
        cvNormalizeHist(Histogram2, 100.0);
        double c1 = cvCompareHist(Histogram1, Histogram2, CV_COMP_CORREL) * 100;
        double c2 = cvCompareHist(Histogram1, Histogram2, CV_COMP_INTERSECT);
        return (c1 + c2) / 2;
    } catch (Exception e) {
        e.printStackTrace();
        return -1;
    }
}
```

图 5.5 人脸识别代码

Fig 5.5 The implementation of face recognition

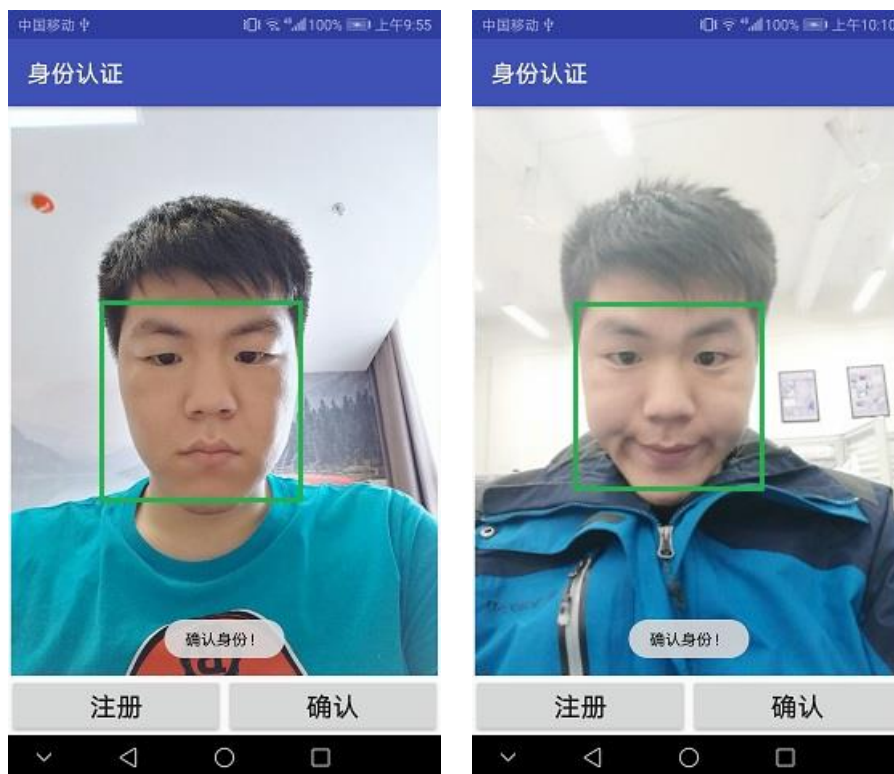




图 5.6 身份认证测试结果

Fig 5.6 Authentication test results

通过两种不同的智能手机来测试评估人脸检测和识别方法的准确性和可接受性。在第一种情况下，测试是在拥有 20 万像素摄像头的华为 Mate9 Android 7.0 智能手机和麒麟 960 CPU 上进行的。在第二种情况下，测试是在具有 1600 万像素摄像头的小米 5 Android 6.0 智能手机和高通骁龙 820CPU 上进行的。每次实验选取 5 个人（包括手机持有者）的 20 张图像，测试 10 次，结果取均值。

表 5.1 在 Huawei Mate9 上的测试结果

Table 5.1 Test results on Huawei Mate9				
	图像尺寸	准确率	耗时 (ms)	占用内存(MB)
人脸检测	1280*720	95%	87	26
人脸识别	1280*720	90%	181	30

表 5.2 在 XiaoMi5 上的测试结果

Table 5.2 Test results on XiaoMi5

	图像尺寸	准确率	耗时(ms)	占用内存(MB)
人脸检测	1280*720	95%	137	28
人脸识别	1280*720	90%	276	34

PC 环境: ThinkPad S2
PC 系统: Window 10 64 位
开发软件: Android Studio
手机: HuaWei Mate9、XiaoMi5

5.4 本章小结

本章首先分析了整个系统的需求，并对这些需求进行了功能划分，共分为图像采集、图像预处理、人脸检测、人脸特征提取与人脸识别这五个模块。然后介绍了各模块的实现。最后分别在 HuaWei Mate9 和 XiaoMi5 手机上测试了实验结果。

6 总结与展望

6.1 总结

随着移动互联网与人工智能技术的发展，基于个人特征的生物识别技术代替传统的身份验证方式已经是大势所趋。而人脸识别是生物识别技术的一个重要组成部分，拥有其他生物识别技术没有的独特优势。因此移动手机端下人脸识别的研究具有十分重要和深远的意义。本文针对 Android 平台下基于人脸识别的身份验证系统主要做了如下工作。

一、综述人脸检测和识别技术的研究背景与意义，人脸识别技术的发展历史与国内外研究现状，以及当前 Android 平台下的人脸识别的发展现状。

二、针对图像预处理介绍了三部分内容。首先，对采集的图像进行灰度归一化的操作，增强图像对比度，使图像的关键部分更清晰，让不同成像条件下拍摄的同一个人的图像保持一致。然后，对于灰度后的图像进行了直方图均衡化的操作，其作用是对图像进行非线性拉伸，重新分配图像灰度值，扩展像素个数多的灰度级，压缩像素个数少的灰度级，提高了对比度和灰度色调的变化。最后一步预处理工作是对图像进行滤波操作，消除噪点。

三、针对人脸检测与识别介绍了三部分内容，一是基于 Adaboost 的人脸检测算法；二是基于 PCA 的人脸特征提取算法；三是几种人脸识别的训练算法，通过分析，提出了一种最近邻与支持向量机相结合的人脸识别算法，并通过实验证明了该算法的可行性。

四、概述了 Android 系统的发展历程以及基于 Android 平台开发的优势，其次系统的介绍了 Android 系统开发的应用组件，最后介绍了 Android 开发需要做的准备工作。

五、分析了整个系统的需求，并对这些需求进行了功能划分，共分为图像采集、图像预处理、人脸检测、人脸特征提取与人脸识别这五个模块。然后介绍了各模块的实现。最后分别在 HuaWei Mate9 和 XiaoMi5 手机上测试了实验结果。

6.2 未来的展望

展望未来，本文中还有着许多不足的地方，本文的不足以及下一步工作主要有以下几个方面：

一、光照与阴影的干扰一直是人脸识别的难题，虽然本文通过灰度归一化和直方图均衡化两种算法对图像进行了预处理，但是效果不是特别理想。下一阶段可以尝试其他的预处理算法来进一步解决人脸图像的光照变化问题。

二、本文采用的用于的人脸样本比较少，对整个系统识别性能有一定的影响。下一步可以考虑从两个方面来解决这个问题。一是加大训练样本；二是寻找在小样本下人脸识别的优化技术。

参考文献

- [1] 刘鹭,董立文,姜鹏,马月. 人脸识别技术的研究与应用[J]. 科技信息. 2010(14).
- [2] 满江月. 浅析人脸识别的现状与未来[J]. 中国安防. 2015(15).
- [3] 郑展棒. 生物识别渐成门禁控制发展潜力点[J]. 中国公共安全(综合版), 2014,(14):65~69.
- [4] 杨晓飞. 人脸识别系统研究[D].北京化工大学 2005.
- [5] 人脸识别算法研究及实现[D]. 柯晓华.哈尔滨工程大学 2008.
- [6] 李瑛. 专利视角下的智能手机竞争态势研究[D]. 大连: 大连理工大学, 2013.
- [7] 张婷,张辉极. 智能手机安全问题及防护技术分析[J]. 信息网络安全. 2011(12)
- [8] 姚昱旻,刘卫国. Android 的架构与应用开发研究[J]. 计算机系统应用. 2008(11)
- [9] 门汝宁. 4G 手机快速发展 智能终端产业机遇和挑战并存[J].世界电信. 2014(12).
- [10] 陈勋. 基于 Android 平台的资源管理器设计[D].电子科技大学 2011.
- [11] 周勇,张嘉林,王桂珍,郑胜男.Android 平台下人脸识别系统的研究与实现[J]. 南京工程学院学报(自然科学版). 2013(01).
- [12] 蔡芷铃,林柏钢,姜青山. 基于人脸识别技术的 Android 平台隐私保护系统设计[J]. 信息网络安全. 2014(09).
- [13] 熊玉虎. 智能视觉物联网中 Android 人脸跟踪应用的研究[D].太原理工大学 2013.
- [14] 熊刚. 基于 Android 的智能手机的设计与实现[D].武汉理工大学 2010.
- [15] Galton F. Personal identification and description[J]. Journal of Anthropological Institute of Great Britain and Ireland. 1889: 177-191.
- [16] Galton F. Numeralised Profiles for Classification and Recognition[J]. Nature, 1910, 83(2109); 127-130.
- [17] Bledsoe W W. Man-machine facial recognition[J]. Rep. PRi, 1966, 22.
- [18] Brunelli R, Poggio T. Face recognition: Features versus templates[J]. IEEE

- Transactions on Pattern Analysis & Machine Intelligence. 1993 (10); 1042-1052.
- [19] Yang G, Huang T S. Human face detection in a complex back ground[J]. Pattern recognition, 1994, 27(1): 53-63.
- [20] Kotropoulos C, Pitas I. Rule-based face detection in frontal views[C]. Acoustics, Speech, and Signal Processing, 1997.ICASSP-97, 1997 IEEE International Conference on. IEEE, 1997, 4: 2537-2540.
- [21] Viola P, Jones M J. Robust real-time face detection[J]. International journal of computer vision, 2004, 57(2): 137-154.
- [22] Viola P, Jones M. Rapid object detection using a boosted cascade of simple features[C]. Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on IEEE, 2001, 1(1); 511-518.
- [23] Poultney C, Chopra S, Cun Y L. Efficient learning of sparse representations with an energy-based Model[C]. Advances in neural information processing systems. 2006; 1137-1144.
- [24] Hinton G. A practical guide to training restricted Boltzmann machines[J]. Momentum, 2010, 9(1):926.
- [25] 汪海波. 基于深度学习的人脸识别方法研究[D].合肥工业大学 2014.
- [26] Sermanet P, Chintala S, Lecun Y. Convolutional neu-ral networks applied to house numbers digit classification. International Conference on Pattern Recognition . 2012.
- [27] 卢世军. 生物特征识别技术发展与应用综述[J]. 计算机安全. 2013(01).
- [28] 薛冰, 郭晓松, 蒲鹏程. 人脸识别技术综述[J]. 四川兵工学报. 2010(07).
- [29] King, Davis E. "Dlib-ml: A machine learning toolkit." The Journal of Machine Learning Research 10 (2009): 1755-1758.
- [30] The OpenCV Library Gary Bradski in Dr. Dobbs Journal, 2000.
- [31] 葛微,程宇奇,刘春香,陈秋萍. 基于子空间分析的人脸识别方法研究[J]. 中国光学与应用光学. 2009(05).
- [32] 孙鑫,刘兵,刘本永. 基于分块 PCA 的人脸识别[J]. 计算机工程与应用. 2005(27)
- [33] T Mandla,Q WU. Face Recognition using Curvelet based on PCA.Pattern Recognition. ICPR,19th international conference . 2008
- [34] 徐秀秀. 基于流形学习的人脸识别方法研究[D].江南大学 2015.

- [35] 白万荣. 人脸识别中基于流形学习的特征提取方法研究[D].兰州理工大学 2012.
- [36] Mikhail Belkin, Partha Niyogi. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation[J]. Neural Computation. 2003 (6)
- [37] He Xiaofe, Niyogi P. Locality preserving projections. Proc of the Conference on Neural Information Processing Systems. 2003.
- [38] Zhang Jiu-long, Li Peng. Facial feature extraction by Curvelet and LDA. Journal of Computational Information Systems. 2008.
- [39] Khalid M Hosny. New set of rotationally legendre moment invariants. International Journal of Electrical and Electronics Engineering. 2010.
- [40] Michael Collins, Robert E. Schapire, Yoram Singer. Logistic Regression, AdaBoost and Bregman Distances[J]. Machine Learning. 2002 (1).
- [41] 李鹏, 耿国华, 周明全. 一种基于神经网络和贝叶斯决策的人脸检测方法[J]. 计算机应用研究. 2007(08).
- [42] Simon H. Neural network and learning machines. 2011.
- [43] Geoffrey E. Hinton, Simon Osindero, Yee-Whye Teh. A Fast Learning Algorithm for Deep Belief Nets[J]. Neural Computation. 2006 (7).
- [44] VAPNIK V N. The Nature of Statistical Learning Theory. 1995.
- [45] Hinton G. Momentum. A practical guide to training restricted Boltzmann machines. 2010.
- [46] Joachims T. Making large-scale support vector machine practical. Advances in Kernel Methods-Support Vector Learning. 1999.
- [47] 王宏漫, 欧宗瑛. 支持向量机在人脸识别中的应用[J]. 计算机工程与应用. 2003(11)
- [48] Support vector machines for face recognition[J]. Guodong Guo, Stan Z. Li, Kap Luk Chan. Image and Vision Computing. 2001 (9).
- [49] J.A.K. Suykens, J. Vandewalle. Least Squares Support Vector Machine Classifiers[J]. Neural Processing Letters. 1999 (3).
- [50] Tony van Gestel, Johan A.K. Suykens, Bart Baesens, Stijn Viaene, Jan Vanthienen, Guido Dedene, Bart de Moor, Joos Vandewalle. Benchmarking Least Squares Support Vector Machine Classifiers[J]. Machine Learning. 2004 (1).
- [51] 李红莲, 王春花, 袁保宗. 一种改进的支持向量机 NN-SVM[J]. 计算机学

报. 2003(08).

致谢

时光荏苒，岁月如梭，转眼间三年研究生生活即将结束。在毕业论文完成之际，我对所有帮助、关心过我的人们表示衷心的感谢！

首先向我的导师刘山教授致以最真诚的感谢。刘山老师渊博的专业知识，严以律己、宽以待人的崇高风范，平易近人的人格魅力对我影响深远。三年来，刘山老师不仅在学业上对我精心指导，严格要求，同时还在思想、生活上给我无微不至的关怀和教导：在我学习态度松懈的时候谆谆教导，耳提面命；在我论文写作出现问题的时候耐心讲解，不吝赐教；在我论文投稿屡屡受挫的时候鼓励我再接再厉，向其他同学学习；在此谨向老师致以衷心的感谢和崇高的敬意。

感谢孙农亮教授、赵增顺教授、白培瑞老师在我硕士期间提供的学习和生活中的帮助及指导。

感谢所有陪我度过研究生生活的同学们。感谢我的好朋友陈欢和王虎，在我遇到问题和困难的时候，总是无条件的给予我最大的帮助。感谢李红、盛春梅、刘伟健同学帮助和指导。

非常感谢我的父母对我的支持和照顾。感谢他们一直以来的默默付出，无微不至的关爱和呵护，愿他们健康长寿，开心幸福！

最后感谢在百忙之中抽时间对本文进行评阅的专家、教授。由于本人水平有限，文中难免有不足之处，恳请专家和读者批评指正。

攻读硕士期间主要研究成果