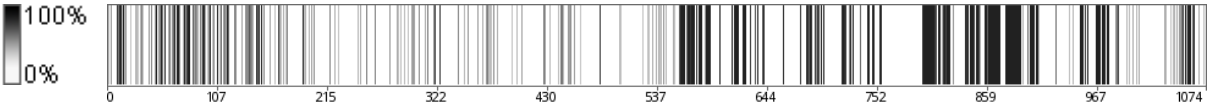
PaperPass旗舰版检测报告
简明打印版

比对结果(相似度):

总　体：35% (总体相似度是指本地库、互联网的综合对比结果)
本地库：29% (本地库相似度是指论文与学术期刊、学位论文、会议论文、图书数据库的对比结果)
期刊库：11% (期刊库相似度是指论文与学术期刊库的比对结果)
学位库：27% (学位库相似度是指论文与学位论文库的比对结果)
会议库： 3% (会议库相似度是指论文与会议论文库的比对结果)
图书库： 5% (图书库相似度是指论文与图书库的比对结果)
互联网：10% (互联网相似度是指论文与互联网资源的比对结果)

编　号：5AC630B8CFD39J0PO
版　本：旗舰版
标　题：基于Android平台的人脸识别身份认证系统的设计与实现
作　者：王志远
长　度：30595字符(不计空格)
句子数：1074 句
时　间：2018-4-5 22:20:40
比对库：学术期刊、学位论文、会议论文、书籍数据、互联网资源
查真伪：http://www.paperpass.com/check

句子相似度分布图:



本地库相似资源列表(学术期刊、学位论文、会议论文、书籍数据):

1.相似度：20% 篇名：《Android环境下基于人脸识别的手机解锁技术研究与软件开发》
来源：学位论文 东南大学 2016
2.相似度： 5% 篇名：《基于Android平台的人脸识别门禁系统设计与实现》
来源：学位论文 西北师范大学 2015
3.相似度： 1% 篇名：《人脸检测》
来源：书籍数据 上海交通大学出版社 2006-1-1

互联网相似资源列表：

1.相似度： 7% 标题：《基于android系统的人脸识别算法研究与实现 ...》
http://www.doc88.com/p-1718601709321.html
2.相似度： 3% 标题：《Java基础知识-Activity - CSDN...》
https://blog.csdn.net/erwugumo/article/details/79339624
3.相似度： 3% 标题：《Android Activity基础详解 - W...》
https://blog.csdn.net/wtoria/article/details/51940120
4.相似度： 3% 标题：《Activity - CSDN博客》
https://blog.csdn.net/qianxiangsen/article/details/51496453
5.相似度： 3% 标题：《Android之Activity系列总结（一）-...》

https://www.cnblogs.com/jycboy/p/6367282.html?utm_source=tuicool&utm_medium=referral
6.相似度： 3% 标题：《Activity - 普罗米修斯的博客 - CS...》
http://m.blog.csdn.net/mixiu888/article/details/78709871
7.相似度： 3% 标题：《Android 之Activity - 阿里云》
https://www.aliyun.com/jiaocheng/62453.html
8.相似度： 3% 标题：《 Activity Android Dev...》
https://developer.android.google.cn/guide/components/activities.html?hl=zh-cn
9.相似度： 3% 标题：《Android之Activity系列总结（一）-...》
https://www.cnblogs.com/jycboy/p/6367282.html
10.相似度： 3% 标题：《Android魔幻之旅（三）：Activity的...》
https://www.jianshu.com/p/8548c6914684
11.相似度： 2% 标题：《Activity - Android移动开发技术...》
https://www.2cto.com/kf/201608/541085.html
12.相似度： 2% 标题：《 Android Activit...》
http://www.codes51.com/article/detail_2773166.html
13.相似度： 2% 标题：《android developer tiny s...》
https://blog.csdn.net/zheng3stone/article/details/52440512
14.相似度： 2% 标题：《Activity详解 - CSDN博客》
https://blog.csdn.net/qq_34206198/article/details/52211074
15.相似度： 2% 标题：《Android-Activity所应该了解的大概...》
https://blog.csdn.net/icoollike/article/details/50239681
16.相似度： 2% 标题：《Activity生命周期总结 - CSDN博客》
https://blog.csdn.net/liuyi1207164339/article/details/51538842
17.相似度： 2% 标题：《Actvity 组件 - 简书》
https://www.jianshu.com/p/435ed3c04f89
18.相似度： 2% 标题：《1.2 Activity 的基本用法大全 - C...》
https://blog.csdn.net/comwill/article/details/78062589
19.相似度： 2% 标题：《1.2 Activity 的基本用法大全 - 阿...》
https://www.aliyun.com/jiaocheng/4142.html
20.相似度： 1% 标题：《Android应用组件之Activity介绍1 ...》
https://www.aliyun.com/jiaocheng/7283.html
21.相似度： 1% 标题：《SVM支持向量机算法的详细推导(详细到每个步骤_...》
http://www.doc88.com/p-1905946677891.html
22.相似度： 1% 标题：《SVM支持向量机算法的详细推导(详细到每个步骤, ...》
https://wenku.baidu.com/view/fa8e7f2eccbff121dd368336.html
23.相似度： 1% 标题：《Activity学习 - 简书》
https://www.jianshu.com/p/eeaa7da8c98d
24.相似度： 1% 标题：《Android环境下基于人脸识别的手机解锁技术研...》
http://www.doc88.com/p-8721303099916.html
25.相似度： 1% 标题：《支持向量机算法的详细推导》
http://www.docin.com/p-581243191.html

全文简明报告:

1 绪 论

1.1 课题研究背景及意义

{41%：当前，人工智能理论的研究已经应用于博弈、智能搜索、机器视觉、自然语言处理等多个方面，} 随着最近几年移动互联网的爆发，身份识别的需求也变得越来越普及，互联网相关业务帐号注册、网络银行业务办理、校园卡业务办理， {55%：甚至各大超市商场

的优惠卡的业务办理，无一例外的需要各种身份验证，由于这一类身份验证的需求量过大，}尤其表现在互联网上各种帐号的申请，原始的帐号密码申请呈现出了很大的弊端，各种不同的帐号密码让用户产生了混淆，可能一些帐号一段时间不使用就会忘记密码甚至帐号，给用户带来了极大的不便，增加了无意义的记忆负担；而所有帐号全部使用相同的密码存在很大的安全隐患，例如之前的CSDN、12306账号密码泄漏事件，一旦帐号密码被不法份子获取，会给用户的钱财和隐私带来威胁。因此寻找一种更简便、快捷、安全的身份验证方式成为了各大离校实验室、研究所、知名互联网研究院等研究机构的研究热点。{83%：生物识别由于具备稳定性、安全性、唯一性等特点，备受广大研究者们的关注。}{63%：相比其他几种生物识别方式，人脸识别具备其特有的优势。}{56%：首先，由于人脸识别所需的识别特征是人脸的图像信息，而人脸图像的采集是不需要接触的，}{87%：只需要摄像头拍摄照片即可，可以远距离实现身份确认，相比指纹、掌纹等等送类接触性的生物识别方式要更加的友好。}{100%：而相比步态识别、手势识别这些容易被模仿的生物识别方式，人脸识别则更加安全。}{40%：最后人脸图像的特征信息相比语音、步态、手势等更加方便存储。}{100%：基于这些方面的考虑，人脸识别技术更加受到用户的喜爱和认可。}

{41%：美国的苹果公司于2007年1月初推出了ios，2008年9月下旬谷歌公司发布了基于Linux系统的移动操作系统Android，}{47%：Android移动操作系统的推出刷新了人们对智能手机的观念，诺基亚和它的Symbian操作系统瞬间走下神坛，}{56%：当前，Android和ios这两大移动端的系统已经占据了移动端系统的几乎全部市场。}与此同时，硬件厂商也不断推出各种高性能硬件，由于Android操作系统的开放性，各大手机创业公司加入了这个浪潮之中，并对原生的Android操作系统的进行了部分重新设计，定制了具有自己独特风格的手机系统，例如小米科技的Miui，魅族公司的Flyme，一加手机氮OS，键子手机的SmartisanOS等等。{41%：相比传统PC，由于移动终端的具有便携、操作灵活的特点，通过几年的迅猛增长，移动互联网已占据整个互联网的大半江山。}Android操作系统目前已经发布至Android8.0，最新API版本号为27，每个大的版本都会有较大的改变，例如在Android5.0中发布了震惊世界的Material Design，根据谷歌公司官方数据统计，截至2018年1月8日，99.7%以上的基于Android操作系统的手机的Android版本在4.0以上，其中Android6.0占据28.1%的最商份额。当前Android和ios两大移动操作系统的竞争已经进入白热化的阶段，苹果公司和谷歌公司都希望自己的移动操作系统能够更多的占据市场上的份额，{45%：就目前阶段来看，由于Android系统对源代码进行了开放，市面上的各大手机厂商都开发了基于Android操作系统的定制系统，}{44%：而ios只存在与苹果公司的iPhone、iPad等产品，因此Android操作系统目前处于领先的地位。}根据数据研究公司IDC的数据统计报告，Android操作系统将持续保持领先，ios操作系统的份额将出现小幅度的下滑，{40%：预计至2019，Android将占据移动操作系统市场的82.6%比例，届时ios将占据移动操作系统的比例为14:}1。{45%：IDC还表示，2017年基于Android操作系统的手机出货量约为15.3亿部，这一数字在2021年将增至19亿部，}{62%：届时智能手机将占据整体手机市场92%的份额。}

{40%：从身份识别发展趋势以及移动互联网未来发展趋势考虑，基于Android OS的人脸识别系统的研究很有前景，}例如，这几年的时间，很多电商、O2O平台兴起，移动互联网上的身份验证应用场景越来越多，例如支付宝、微信支付、饿了么、美团、滴滴打车、大众点评等应用已经渗入到了人们衣食住行的方方面面，给人们生活带来了极大的方便，同时也带来了一些相应的安全隐患，因此，考虑通过人脸识别的方式来取代传统帐号密码的必要性不言而喻。同时这项工程也具有很大的挑战性，首先人脸识别一直存在拍摄时光线干扰、被拍摄者的表倩姿态、样本的残缺等难点问题需要解决；另一方面，基于Android的终端性能虽然在不断提高，但是相比传统PC的CPU，性能上仍然有不小差距，{55%：因此在识别效率上相比传统PC会慢一点，尤其是在大规模识别的应用场景下，例如警方对嫌疑人的照片进行识别，}以及一些人脸搜索系统等，效率会成为一个比较重要的参

考因素； {62%：最后，人脸识别具有很高的理论研究价值，它涉猎的知识范围到较广，有一定的深度。} {59%：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

{51%：人脸识别包括原始图像预处理、原始图像中人脸区域检测、对检测到的人脸区域进行特征提取、特征的分类识别这几个过程。} {97%：人脸识别技术的研究最早出现的时间出现在20世纪50年代左右， Chan和 Bledsoe首次发表了关于人脸识别理论研究的论文，} {83%：但是那个年代的硬件和软件的环境非常差，研究受到较大的限制，当时的算法的应用范围比较狭窄，} 因此当时这项技术的发展也比较缓慢。 {55%：直到1980年开始，计算机的硬件性能和软件系统开始极速发展，人脸识别技术的实验环境越来越好，} {55%：理论研究也有了极大的进展，这项技术也逐渐地被用于实际应用场景中。}

{100%：在人脸的检测方面，随着这方面需求的上升和工程技术方面的进展，大量的国内外学者涌入这方面的研究当中并取得了较好的成绩。} {63%：国外的耶鲁、麻省理工学院，国内的清华、浙大等等都研究出了检测效果显著的算法，作出了巨大的贡献。} {53%：目前人脸检测通常分为先验知识以及基于统计这两种方案。} 前者是以我们人类最直观的感受作为参考，即根据人脸的五官的位置、结构、形状、数量作为参考依据来判断是否为人脸。{98%：1994年G.Yang等人根据人脸的无关位置的分布，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。} 1997年Kotropoulos等基于标准人脸先验知识和人脸五官的灰度积分投影进行比对，提出了一种人脸检测算法，取得了较好的效果。 2001年，艾海舟等在基于一个人脸的模版， {49%：设定一个阈值，通过比较需要判断的图像区域与该人脸模版的相似度如果相似度在设定的阈值之上，} {100%：则说明待检测图像区域存在人脸，这个人脸模版的标准是基于人脸的一些特征并融合了肤色特征，} {67%：检测的效率和检测的精确度都较好。} 模版匹配方面， Miao等人将图像旋转，利用 Laplace算子对其进行边缘检测，将边缘检测提取出来的信息进行合成最终得到一个人脸模版， {55%：然后采用层级方法来进行检测并判断是否存在人脸。} {66%：送种方案虽然很直观，但是人脸毕竟是一种非刚性的物体，人脸经常会作出各种表情、姿态，} {100%：图像的采集也会遇到遮挡、光照等干扰因素，而这种方法对姿态、表情、光照等干扰因素的鲁棒性较差，} 1996年 John F提出的 Canny算子，1970年 Irwin提出的 Sobel算子等等就是为了一定程度上解决光照对人脸检测干扰、影响， 但是对于识别精准要求特别高的实际应用场合，这种方案显然还是不适合的。 {100%：相比较这种直观的、容易让人理解的基于先验知识的方法，基于统计的检测方法更加先进、准确率高。} {100%：对于这种方案，首先需要根据具体数据信息统计出大量的人脸和非人脸的数据信息，} {59%：根据这类统计的数据信息，设计出判断是否为人脸的二分类器，对于待检测的图像，} 只需要将图像作为输入，输出结果即是否为人脸的布尔值。 设计人脸分类器的过程比较复杂，为了提高分类器的准确率，必须要持续的调整各种参数来适应当前的应用场合，得到一个最优的分类器。{52%：2001年， Viola等人就提出了一种基于 Adaboost的人脸检测算法，它首先使用 Haar- Like特征来表示人脸，} {45%：通过积分图方法的计算，利用 Adaboost算法筛选出所需要的具备人脸特性 Haar特征，训练得到弱分类器，} {100%：将训练样本中分类出错的数据和新样本的数据进行混合，在这个混合样本下训练得到新的弱分类器，} {55%：再通过加权投票的方案把这些弱分类器构成一个强分类器，最终将这些训练得到的强分类器串联起来组合成一个级联分类器作为最终的分类器。} {79%：在人脸识别中，人脸图像常用矩阵表示，原始的人脸图像的维数会非常高，} {100%：不利于分类识别处理，需要对原始图像进行特征提取的操作。} {59%：这里又分为基于几何特征的方法和基于统计的特征提取方法，而基于统计的特征提取方法又分为线性方法和非线性方法。} {100%：和人脸检测的思想类似，基于几何特征的方案主要根据眼、耳、鼻送些脸部器官的相对位置来提取特征，} 这种方式的特点也和人脸检测中先验知识的特点类似，效率高但是对噪声的光照、姿态等问题

的鲁棒性不好，因此效果并不是很好。{72%：而在基于统计特征的方法这一方面则有很多经典的算法。}{50%：Kirby等人最早把PCA方法用于人脸识别应用中P7 K281，PCA是一种基于Karhunen-Loeve(K-L)变换的方法，}PCA方法的基本原理是利用离散K-L变换的方法把数据在一个新的空间上进行表示，该新的空间相对于原始空间，{62%：关联性的向量数据均已被去除，都是一些线性无关的向量数据，然后根据所需降到的维数，按照特征值由大到小进行选择，}{55%：这样的操作最终得到的是比较大的特征值所对应的特征向量，即可以达到降维的}{44%：效果，消除原始图像上的相关信息同时保留了原始图像的大部分的特征信息。}线性判别的方案里面除了PCA方法以外比较流行的还有FisherFace（LDA）方法，该方法通过选择一个向量让Fisher准则函数这到它的极值，{60%：以此来作为投影的方向，这样样本即可在这个方向上达到最大类间离散度和最小类内离散度，}理论上来说，相比PCA，LDA更加适合处理这种分类问题。{42%：而在高阶的情况下，无论是PCA还是LDA都不是很适合作处理，Bartlett等人最先提出将ICA应用到人脸识别领域的研究上，}{86%：考虑将整个图像当成多个图像的叠加，并实验证明了这种方案效果比PCA好。}{95%：非线性方法中，流行学习的方法出现于90年代，2000年Roweis和Tenenbaum等分别在著名的《Science》}{49%：上发表了关于局部线性嵌入（LLE）以及等距映射（IOSMAP）的论文造成了较大的影响。}2003年，Bekin基于谱理论发表了关于拉普拉斯特征映射（LE）的论文，该论文正是针对LLE的一些缺陷，例如对噪声敏感。{40%：He等人提出了的概念对LE进行了线性化推广并进一步改进得到近邻保持嵌入。}

{42%：特征提取完成之后，就需要构造分类器对这些特征进行分类识别，分类器作为人脸识别的最后一个环节，}{57%：也即"决策者"，分类器的构造显然至关重要。}{93%：在人脸分类识别方面，目前常用的分类器有KNN、BP神经网络、SVM、随机森林等等。}KNN最早是由Cover和Hart在1967年提出来的一种非参数方法，该算法是一种简洁易懂、效果显著的机器学习算法，{41%：KNN算法的规则就是将分类结果视为距离测试数据的最近的K个数据中出现类别个数最多的那个类别。}{64%：测试过程中以待测数据为中，渐渐往外扩散，逐渐包含训练样本点，直到包含的样本个数达到K，}{60%：其中训练样本和测试样本之间的距离通常情况下使用Euclidean距离来表示。}KNN算法比较直观易实现，但是由于需要通过不停计算距离来搜索到最近的K个训练样本，导致计算量非常大，效率很差。神经网络近几年最活跃的机器学习算法之一，神经网络最早出现在上世纪50年代末，直到BP神经网络的提出，神经网络正式实现了应用价值。{96%：为了对人脑进行更加精确的模拟，需要构造多个隐含层。}2006年左右，Geoffery Hinton等在《Science》发表了关于深度信念网络（DeepBeliefNetworks，简称DBNs）的论文；{40%：同一时期，Hinton的这篇论文对深度学习理论的研究产生了巨大的影响，例如几篇极具影响为的论文深度学习的研究开始越来越热，}深度学习是基于深度信念提出的算法，Hinton的论文中，是从受限波尔兹曼机中引出的概念，深度学习需要大量数据的支持，{70%：而在这个大数据时代，最不缺少的就是数据量，更加复杂且更加强大的深度模型能深刻揭示海量数据里所承载的复杂而丰富的信息。}当前，谷歌、微软、阿里、百度等世界顶尖互联网公司都投入巨大的资源在这方面的研究上，{55%：百度还创办了百度深度学习研究院（IDL），邀请了吴恩达作为研究院的首席科学家。}{63%：不过在实际人脸的分类识别场景中，人脸图像的数量还是十分有限的，相对于其他的人脸识别方法，}{43%：支持向量机这种适用于小规模数据的分类器还是非常有实用价值的。}支持向量机（SVM）最早是由Vapnik在1995年提出的来的，当时就在学术界和工程界引发了不小的轰动。{100%：支持向量机算法集合了多项技术、概念，如最大间隔的最优超平面、核函数、凸二次规划、松弛变量等等，}{68%：适用的场景非常多，且表现都比较好，常被作为人脸识别的分类器。}1997年，Osuna等人发表了关于一种分解算法的论文，且将该算法应用到了人脸检测中，Joachims在此基础上提取了一种SVMLight的算法优化了Osuna的分解算法，{42%：1998年，Platt提出了序贯最小化优化算法在变量、系数改进方面，1999年，Suykens等人提出了一种最小

二乘支持向量机算法，} 把标准的支持向量机线性不等式的约束转换成等式约束以此来转换训练为求解方程组，这种方案使得支持向量机更加广泛的运用到了工程技术中。

{62%：国内外也有不少的公司提供了人脸识别的各个平台下的API服务。} {56%：例如Face++、百度、科大讯飞、汉王云等公司都提供了相关的API。} Facebook、腾讯的QQ空间送些社交网络平台也都有人脸识别功能服务，帮助用户找到可能认识的人。 而对于Android本身的API中，Google就提供了人脸检测的API，小米公司的Miui系统、华为的部分机型均集成了人脸解锁功能。

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

{51%：本文主要研究了人脸识别的算法，以及研究的算法在Android平台下的具体实现。}

对原始图像的预处理方案进行了介绍，使得原始图像轮廓更加清晰分明，对比度更强，首先对彩色图像的进行灰度归一化处理， {100%：为了减少曝光的对图像带来的影响，对灰度化后的图像还要进行直方图均衡化的处理，通过直方图均衡化这种修正方式，} {57%：图像的会更加清晰柔和，最后介绍了图像平滑方案来减弱噪声。}

{58%：对人脸检测的概念和方法进行了介绍，} {57%：通过设计人脸分类器的角度来介绍了 Adaboost算法，} {74%：分类器的设计是多个强分类器构成的级联分类器，} {63%：并针对上一次的错误率调整参数，而每个强分类器是由多个弱分类器构成的，弱分类器则是由挑选的 Haar特征训练得到的。}

{57%：对特征提取方法进行了介绍，为了降低图像的维数，提取出关键特征信息，从而提高下一步分类识别的速度和准确率，} {47%：需要对人脸区域进行特征提取，着重介绍了基于 PCA的特征提取方案，对 K- L变换的原理进行了介绍，} 引出 PCA算法并描述了 PCA特征提取的过程。

{100%：对人脸的分类鉴别进行了介绍，介绍了人脸分类鉴别的概念，介绍了常见的几种分类器基本概念和原理。} {66%：重点研究了KNN分类器算法和支持向量机分类器算法。} {83%：为了进一步提高人脸识别率，研究了基于最近邻算法和支持向量机算法相结合的分类器算法。} {98%：实验表明，这种融合性分类器算法比单独使用KNN算法或单独使用SVM算法在人脸识别上具有更高的性能优势。}

{95%：顺序介绍了本文应用开发中所需要的几项技术：} Android核也技术、OpenCV函数库、JNI技术、NDK。 {60%：然后分析了应用的功能、应用的架构、应用的实现和实验。}

1.3.2 本文组织结构

本文的组织结构安排如下：

第1章绪论。 {51%：综述人脸识别的研究背景与意义、 Android操作系统的发展与现状，国内外研究现状，包括人脸检测技术研究现状、} {45%：特征提取技术研究现状、人脸分类识别技术研究现状、 Android平台下的人脸识别 API实际应用情况。}

第2章预处理。 介绍了图像灰度化的方法； {42%：对直方图的均衡化方法进行了详

细的论述，并说明该操作能够一定程度上解决人脸图像的光照问题：} 对图像平滑进行了介绍。

第3章人脸检测和特征提取。 {81%：本章主要人脸分类识别前的两块内容，} {49%：首先论述人脸检测概念，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的人脸检测算法的过程，} {42%：然后论文了人脸特征提取的概念，并由 K- L变换引出 PCA算法，阐述了 PCA提取人脸特征的具体过程。} 第4章人脸分类识别。 本章首先论述常用的几个人脸分类器。 {47%：重点介绍了KNN分类器算法和支持向量机分类器算法，并结合本系统提出了一种基于最近邻算法和支持向量机算法相结合的分类器算法。}

第5章Android人脸识别系统的实现。 {41%：本章首先论述Android端应用开发所需的一些核也技术，包括Android核心技术、JNI技术、OpenCV函数库、NDK；} {61%：分析了应用的功能、应用的架构、应用的编写实现，展示了结果。}

第6章总结与展望。 {98%：总结了全文的工作内容，对基于Android平台的人脸识别应用的下一步研究给予展望。}

人脸图像预处理

人脸图像预处理操作是人脸识别技术在实际应用中比较关键的一个步骤，因为手机摄像头拍摄到的图像与知名数据库中的图像情况不同， 由于光照、拍摄角度、拍摄距离和阴影等因素，手机摄像头拍摄到的图像可能的情况非常多， {45%：因此在人脸检测、特征提取以及分类鉴别之前，首先必须要对拍摄的图片进行预处理的操作。}

2.1 灰度归一化

对于手机摄像头拍摄的原始图像，由于拍摄的自然环境，拍摄者的拍摄角度等因素，样本可能存在各种各样的情况， 为了增强采集到的人脸图像的对比度， 使得人脸图像的细节以及关键的部位更加清晰、分明以降低由于采集过程中的一些不利因素如光线和关照强度带来的干扰， 方便后期算法的处理。 {98%：首先要对采集到的图像进行灰度化的处理，灰度化仅仅是去除了彩色信息，虽然丢失了一些颜色等级，} {41%：但是与原始图像的描述是一致的，不会影响图像的主要轮廓信息。}

{46%：颜色空间表示对图像的颜色相关信息的编码描述，不同的颜色空间对图像颜色信息的描述也有所不同，} 颜色空间的种类比较多，例如 RGB、 HSV、 YUV、 HLS等。

2.1.1 RGB颜色空间

{53%：对于RGB颜色空间的图像，R分量、G分量和B分量这三个分量共同决定了每个像素点的色彩。} {46%：这三个颜色通道可以取0~255之间的某个值，其中取0时最弱，取255时最亮，} 因此每个像素点的颜色的可能性可以有255*255*255，即1600多万种。

{52%：为了把采集到的彩色人脸图像灰度化从而转换为灰度图像，} 需要根据这三个分量计算灰度值，不同的灰度值计算方法会带来不一样效果的灰度图， {57%：常用的计算方法有分量平均值计算法，即直接求出图像中各像素点的 R、 G、 B三个分量的平均

值，} 然后以计算的结果作为该像素的这三个分量的值，但是这种简单计算平均值的方案忽视了这三个分量各自独立的特点， 效果不是很好，因此本文采用另外一种方案： 加权平均值法来进行计算。

D(x_ i， y_ i)=√(〖(r_ i- r_ j)〗^2+〖(g_ i- g_ j)〗^2+ 〖(b_ i- b_ j)〗^2)式(2.1)

D(x_ i， y_ i)=√(〖 w_ r(r_ i- r_ j)〗^2+〖 w_ g(g_ i- g_ j)〗^2+ w_ b〖(b_ i- b_ j)〗^2)式(2.2)

加权平均法根据应用场景不同取值也各异，(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）与灰度后的图（右）

2.1.2 HSV颜色空间

相比较RGB，HSV能够在视觉效果上更加直观，MAC操作系统以及Photoshop都是采用的该颜色空间。 {60%：该颜色空间由色相（Hue）、饱和度（Saturation）、明度（Value）三个分量组成。} 相比较 RGB的三个分量而言， HSV的三个分量的独立性更强， Hue用是用角度来度量的， 由绕 V轴的逆时针旋角度来确定，角度的取值范围是0°~360°，0°对应红色，120°对应绿色，240°对应蓝色； {41%：Saturation表示颜色接近于光谱的程度，即颜色的鲜艳程度，它的取值为是0到1；} {47%：Value的值表示颜色亮暗度，它的取值范围为0到1；} RGB颜色空间到HSV颜色空间的转换如下：

{41%：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={■(0°，(max= min)@60°*(g- b)/(max- min)+0°， (max= r， g≥ b)@60°*(g- b)/(max- min)+360°， (max= r， g≤ b)@60°*(b- r)/(max- min)+120°， (max= g)@60°*(r- g)/(max- min)+240°，(max= b))┤式(2.3)

S={■(0°， (max=0)@(max-min)/max，(max≠0))┤ 式 (2.4)

v=max 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：

h_i=⌊h/60⌋

f=h/60-h_i

p=v*(1-s) 式 (2.6)

q=v*(1-f*s)

t=v*(1-(1-f)*s)

2.2 直方图均衡化

采集人脸图像的过程中，由于拍摄时光线的影响极有可能导致采集到的图像曝光不足或者

曝光过度， 为了使得采集到的人脸图像的灰度级分布更加均匀，灰度级的数目增加，使得图像的较暗的部分更加明亮较亮的部分变暗些， 进一步减少光照给后面操作带来的影响，使得图像更加清晰、柔和，需要采取直方图均衡化的处理。

对于连续值图像，假设 r表示被增强图像的灰度，2.1中已经对采集到的人脸图像中所有的像素点进行了归一化处理， {45%：因此其最大值为1，对于处理之后的图像， S表示变换后图像的灰度，即当 r= s=0时表示黑色，} i= s= 1时候表示白色，则变换函数 T(r)的为：

s=T(r)=∫_0^r▒〖P(r)dr 〗 (0≤r≤1) 式 (2.7)

{48%：其中，P(r)表示概率密度函数，为了使得变换后的灰度保持由黑至白的的单一变化顺序，需要满足W下条件：}

(1) 当0≤r≤1时，T(r)的单调递增，且0≤T(r)≤1

(2) 反变换r=T^(-1) (s)； T^(-1) (s)同样单调递增，且0≤s≤1

对于离散值图，需要处理的则是其函数概率之和，例如一副图的像素和是 n， {62%：分为 L个灰度级别，则第 K个灰度级别出现的概率为；}

S_ k= T(r_ k)=Σ_(i=0)^ k▒〖 n_ j/ N=Σ_ (i=0)^ k▒〖 p_ r(r_ j)〗〗(0≤ r_ j≤1， k=0，1，2，… ，L-1) 式 (2.8)

该步骤实验过程：

{61%：(1) 对输入的每个灰度级的像素数目进行统计，其中i=0，1，2，…} ，L-1，L表示灰度级的总数目；

{60%：(2) 对原始输入图像的直方图进行计算，各灰度级的概率密度为：}

p_i (r_i)=n_i/n 式 (2.9)

其中n为原始图像的总像素数目；

(3) 计算累积分布函数：

s_k (r_k)=Σ_(i=0)^k▒〖P_i (r_i) 〗 (k=0，1，2，… ，L-1) 式 (2.10)

(4) 最后的输出灰度级进行计算；

g_k=INT[(L-1) S_k (r_k)+0.5/((L-1))] 式 (2.11)

{54%：(5) 使用原始图像灰度级函数f_k和g_k的映射关系，修改原始图像的灰度级得到输出图像，输出图像的直方图分布均匀。}

图2.2 直方图均衡化后的图

2.3 图像平滑

为了进一步减少图像的噪声，还需要对图像再进行平滑处理，对于空间域，图像的平滑也就是求解像素的平均值或者中值； 对于频域，则是用低通滤波来完成处理。

2.3.1 空间域

(1) 均值滤波器

{49%：均值滤波器的原理就是利用各像素点周边像素平均值，用该平均值来代替原先的像素点的值，} {41%：均值滤波器主要采取的方法是领域平均法。}

{43%：假定采集到的人脸图像 f(x， y)是由 N* N个像素组成，每个像素点使用其领域内的像素的平均值来取代，} 假设处理后的图像为 g(x， y)，则

g=1/M Σ_((m，n∈S))▒〖f(m，n)〗 式 (2.12)

其中，x，y=0，1，2，… {43%：，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）

{42%：通常领域选取的方式为4点领域和8点领域送两种，领域越大，图像的模糊度相应也会越大。}

(2) 中值滤波器

相比之下，中值滤波在在空域滤波中更加常用一些，因为它会产生较少的模数， {50%：更加适合处理图像中的孤立噪声点，中值滤波器原理和均值滤波器类似，} {53%：但是中值滤波器决定的像素是由领域像素的中值而非均值决定。}

{51%：假设采集到的人脸图像 f(x， y)，选定窗口大小 W，用选定的窗口像素中值取代后的输出结果为 g(x，} y)，则有：

g(x，y)=Med{f(x-k，y-l)，(k，l∈W)} 式 (2.13)

其中， f(m- k， n- l)为窗口 W的像素灰度值，为了确定中值， {57%：W的值一般取奇数，若 W值为偶数，则中值取中间两个值的平均值。}

2.3.2 频域

对于采集到的一副图像，会存在一些高频量，需要通过低通滤波来消除这些萄频分量。

假定采集到的图像 f(x， y)，经过 Fourier变换 f(u， v)， {48%：选择最优的滤波器函数 H(u， v)对 f(u， v)的频谱函数进行一定的调整，} 经过 Fourier反变换后得 g(x， y)。

{64%：不同的H(u，v)的选择，会产生不同的平滑效果，常见的低通滤波器有以下四种。}

理想低通滤波器

它的传递函数H(u，v)为；

H(　u，　v){■(1，　D(　u，　v)≤　D_0@0，　D(　u，　v)]　D_0)┤式(2.14)

{58%：其中，D_0是理想情况下的低通滤波器的截止频率。}

{83%：巴特沃斯低通滤波器　}

它的传递函数H(u，v)为:

H(u，v)=1/(1+〖[(D(u，v))/D_0　]〗^2n　)　式　(2.15)

{83%：其中n为巴特沃斯低通滤波器的阶数，D_0为截止频率。}

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　}　式　(2.16)

其中D_0为截止频率，n为阶数。

梯形低通滤波器

它的传递函数H(u，v)为：

H(　u，　v)={■(1，　D(　u，　v)[　D_0@(　D(　u，　v)-　D_0)/(D_0-　D_1)　D_0≤　D(　u，　v)≤　D_1@0　D(　u，　v)]　D_1)┤式(2.17)

其中D_0为截止频率，D_1为预先设定值，D_0≤D_1。

2.4　本章小结

本章主要介绍了预处理这个环节。　从不同颜色空间角度详细介绍了图像的灰度归一化方案，介绍了图像的直方图均衡化理论，从频域和空间域两个方面介绍了平滑滤波方案，对于基于　Android平台的实际的人脸识别应用，最初采集到的必然不是标准的灰度图像，而彩色的图像不利于下面人脸检测和特征提取的处理；　而直方图均衡化可以减少光线干扰因素；　{59%：图像的平街则可以增强图像的对比度，减少图像的噪声。}　该步骤处理的好坏影响着后面进一步的处理从而影响识别效率和识别精度。

人脸检测与特征提取

3.1　人脸检测技术

{43%：　人脸区域的检测是人脸分类鉴别必不可少的前提环节，要对人脸图像进行分类鉴别，}　首先需要得到图像中的脸部区域部分然后对这部分区域进行下一步的操作。{49%：而实际中拍摄人脸图像的过程中通常会由于拍摄距离，}　焦距等因素导致携带一些非人脸的特征，　{55%：此时需要对所拍摄图像的进行人脸区域检测来提取人脸区域，}　最后分类鉴别只会针对人脸部分来进行。　该环节实际上就是区分出人脸部位的特征与其他非

人脸物体的特征并将人脸区域部分提取出来。 人脸检测环节直接影响着人脸分类识别环节，该环节检测准确，后期人脸分类识别环节的效率和精度也会得到提高。

3.2 基于Adaboost的人脸检测方法

Adaboost算法是目前最常被使用到人脸部位检测上的机器学习方法之一，通过输入Haar特征，计算输出结果是否有人脸。 Adaboost人脸检测方法实际上就是把所指定的图像分成若干矩形区域，然后对每个矩形区域用 Haar特征输入并判断是否为人脸区域， 作用就是将整张图像的人脸部位进行提取，将非人脸部位的部分给丢弃。 {53%： Adaboost算法还采用了一种级联判断的方式，通过将多个强分类器串联起来组成一个级联分类器后，} 连续对输入进行判断，对于需要提取人脸的图像，如果判断出是人脸的矩形区域就让其通过当前的强分类器进入下一个强分类器， 否则直接将其丢弃掉。 由于串起来的各个分类器的精度要求是逐渐增加的，通过送些强分类器的层层过滤， 大大的加快了分类检测的速度，而且每一级的强分类器对上一级通过的矩形区域特征再一次进行分类判断， {48%：大大减少了伪正样本的通过率，即提高了检测的正确率。}

Adaboost还采用了一种迭代算法，并非简单地对训练样本进行权重不变的循环训练，最初的样本权重设置是一致的， 并在这种样本下训练出弱分类器，但是每次迭代中的样本比重都是由之前一次来确定的， 例如第 N次的比重由第 N-1次来确定。 {53%：每次把分类错误的样本的权重值提高，这样可以突出分类错误的样本同时得到新的样本分布并训练得到新的弱分类器。}

{63%：该方法的实现主要分为以下三个步骤：}

{40%：在大量的Haar特征中利用Adaboost来进行训练学习，然后选择一些最具备人脸部位特征的Haar特征，从而训练得到弱分类器h_1。}

{44%：将前一个训练样本中分类出错的数据和新样本数据混合在一起构成一个新的训练样本分布，并通过训练得到一个新的弱分类器 h_2，} {41%：再将本次分类过程中分类错误的样本再一次和新的样本混合并得到新的一个弱分类器，如此循环操作，} {58%：最终通过投票的方法将这几个弱分类器构成一个强分类器 h_ max。}

{57%：将这些通过训练得到的一系列强分类器逐一串起来，组成一个级联分类器。}

Haar特征

Haar特征是Adaboost算法的输入，也就是矩形特征。 {46%：通常情况下，人脸的样本的数量是非常少的，对于这种数据非常有限的情况，} {53%：使用矩形区域这种方式作为特征来检测要比基于像素的的速度快很多。} 矩形特征对边缘轮廓比较敏感，而人脸有着一些显著的边缘特征，最显著特征例如人脸的眼睛， 双眼的水平方向上的区域或者双眼上下垂直方向上的区域，眼睛的颜色要比周围的的颜色深很多， 再如嘴己的颜色也要比周围的颜色深很多，使用矩形特征很容易就可以锁定这些区域。

各矩形特征如图3-1：

图3.1常见矩形特征

可以看出，图3.1（B）矩形特征通常可用来表示边缘特征的情况，例如双眼和双眼下方的矩形区域； 图3.1（C）矩形特征可以用来表示线性的特征，例如嘴唇和两侧的水平区域可以用其来表示； 图3.1（D）矩形特征用来表示特定方向上的一些情形，它在人脸部位区域

的检测中应用的并不多。

3.2.2 积分图

{44%：Haar特征获取完成之后，需要对Haar特征值进行计算，常用的一种方案是使用积分图来计算该特征的值。} {41%：积分图对于原始图像而言，就是把原始图像的每一点(x，y)的像素值用该点的左方和上方像素的累加之和来表示，即：}

ii(x， y)=Σ_(xˆ’≤ x， yˆ’≤ y)▒〖 i(xˆ’， yˆ’)〗 式(3.1)

{44%：其中式子中的ii(x，y)表示初始图的积分图，ii(x，y)表示初始图。}

图3.2初始图及其积分图

对于像素值的求解，只要使用右下角端点处的值来进行计算即可得到。

图3.3区域像素值计算

{51%：根据积分图的方法可知，在图中， A区域的像素值可表示为 ii(1)，} A＋B区域的像素值可表示为 ii(2)， A＋ C区域的像素值可表示为 ii(3)， A＋ B＋ C＋ D区域的像素值可表示为 ii(4)，且根据积分图方法， D区像素值应该为A区域、 B区域、 C区域、 D区域的像素值之和＋ A区域像素值－ A、 B区域像素值的和－ A、 C区域像素值的和。 因此D区的像素值可表示为；

D区像素值＝ ii(4) +ii(1)- ii(2)- ii(3)

通过以上的推导可知，想要得到Haar的值，只需要考虑Harr积分图及区域端点的信息即可。通过端点积分图的求和差运算可以快速得到特征值，这也是检测效率高的原因。

3.2.3 分类器训练

Haar的特征数目通常会非常大，例如一个24*24的训练样本窗口，其 Haar特征数目在160000个以上， 这样对于 Adaboost算法来说负担会相当大，因此，必须要挑选出最合适的 Haar特征， {49%：然后对这些合适的 Haar恃征进行训练即可得到弱分类器。}

对于训练样本，可以用网上的一些公认的人脸数据库进行训练，使用最多的例如 MIT-CBCL，该图像库总共有10个人， 训练图像有把3200张还有大量的测试样本，且人脸图像有0度到75度的旋转，图像的大小为200 x200， 常用的还有例如 MITEx， AT& T，BioID等人脸数据库。

Haar特征最终是需要转化为弱分类器的， Haar特征的数目和特征值被确定下来之后，{66%：需要对这些 Haar特征 f进行训练得到对应的弱分类器 f(x，} {53%：f， p，θ)□可 W看出每一个弱分类器都是由它的相对应 Haar矩形特征的参数确定下来的，} 弱分类器的定义公式如下：

h(x，f，p，θ)={■(1 pf(x)[pθ@0 其他)┤ 式 (3.2)

{56%：其中， h(x， f， p，θ)表示一个弱分类器， x为一个特征检测窗口，f表示特征值，} {44%□θ为域值，训练的目标效果是使得 h(x， f， p，θ)的分

类误差值最低。}

训练流程如下；

将每一个训练样本灰度化以及归一化到相同尺寸，本文实验归一化到24*24大小，{52%：这样保证了样本的灰度和尺寸一致即可确定其 Haar在每一个样本中的出现。}

计算出每个f值，然后把所有计算出来的f值排序起来

{43%：对排序好的样本计算所有正例样本（即人脸样本）的权重之和T^+以及所有的负例样本（即非人脸样本）之和T^-}

计算该例先前所有正例的权值的和S^+和该例先前所有负例的权值的和S^-

{63%：计算阈值的分类误差，阈值分类误差为：}

e=min(S^++(T^--S^-)，S^-+(T^+-S^+)) 式 (3.3)

{65%：通过之前的T次迭代训练，已经得到了T个最优的弱分类器h_1 (x)，h_2 (x)，…} {44%：，h_T (x)，此时需要进一步对其进行组合从而得到一个强分类器，其中组合的公式为：}

C(x)={1(1Σ_(t=1)^ T〖α_ t h_ t(x) ≥1/2Σ_(t=1)^ Tα_ t〗 @0其他)┤ 式(3.4)

其中α_t=log 1/β_t =log (1-ε_t)/ε_t =-logε_t。

{54%：此时使用各个强分类器对图像来进行检测，只要通过它们也就相当于通过了组合成每个强分类器所有的弱分类器。} {50%：对于是否能够通过该强分类器，采取的方式是让组合成该强分类器的所有弱分类器进行投票，} {58%：按照这些弱分类器错误的分类率进行加权求和。}

3.2.4 级联分类器

{43%：通过3.2.3得到的强分类器较有比较不错的检测分类效果，} {51%：此时还需要将以上得到的强分类器串起来得到由强分类器组成的级联分类器，} 对每一个经过上一个强分类器的窗口进行再一次的检测分类，这样可以将伪人脸图像窗口进一步进行分类， 大大的提高了分类正确的概率。 它的流程图如图3-3：

{68%：图3.4级联分类器的检测流程图 }

{40%：级联分类器的设计是每一级逐渐严格的，下一级比上一级严格可以将上一级错误分类的伪人脸给排除掉，} 之前的控制级别宽松则可以很快的作出判断，将明显没有人脸特征的窗口给过滤。 因此级联分类器的错误判断具有两种倾向，靠前的分类器容易将负例通过，而靠后的分类器则容易将人脸的窗口给排除， 所以控制好各个分类器的复杂度获得一个平衡是研究重点和难点。 图3.5为训练和检测的流程图。

图3.5训练与检测图

图3.6检测效果图

人脸特征提取技术

在3.2中获取到人脸区域之后，还不能够直接对人脸图像进行分类操作，因为人脸图像的非常复杂，维数很高，这样直接进行分类的的话，分类的计算量将会非常大，导致识别速度很慢，从而不能保证人脸识别的实时性，而且因为在拍摄过程中存在的一些噪声会给分类过程带来不小的干扰，对识别的精确度也会有很大的影响，因此必须要人脸区域图像进行特征提取的操作，提取出分类过程中所需要的最有效的特征，然后针对提取到的这些最有效特征进行分类，这样在效率和精度上都会有所提升。

特征提取的方案包括基于以及基于知识这两种大的方面。其中基于统计的方法是提取图像中的抽象的统计数据的特征，而基于知识的方法则要更为直观一点，它统计的是人脸图像中我们现实生活中直接可以理解的内容，例如眼睛、眉毛、嘴巴等特征。其中基于统计的方法又包括线性方法和非线性方法，常见的线性方法有 PCA、 LDA、 ICA等，{82%：常见的非线性方法又包括基于核特征的提取方法和基于流形学习的特征提取方法。}本文实验以及系统使用的方案为PCA。

基于PCA的人脸特征提取方法

3.5.1 K-L变换

{48%： K- L变换是一种常用的数据压缩方法，它又被称为主成份变换，} 它消除了数据的相关性， K- L变换通过去除原有数据的相关性来减少特征量的个数，该变换是一个比较合理的、综合性的方案。

假定X是由n维的矢量构成的一个矢量的集合，其中矢量的个数为N，那么X ¯为：

X ¯=1/N Σ_(i=1)^n▒φ_i 式 (3.5)

其中X ¯为X的均值，φ_i表示矢量集合中的第i个矢量。

第i个矢量与X ¯的差值可以表示为：

d ¯=(x_i) ¯-X ¯ 式 (3.6)

则矢量协方差矩阵可以表示为：

P=1/N Σ_(i=1)^N▒〖(d_i) →(d_i) →^T 〗 式 (3.7)

{44%：矩阵P为N*N且实对称，矩阵P斜对角的元素为x_i的方差，其他元素则为x_i和x_j的协方差。} {41%：所谓K-L变换正是将矢量集义转换成成一个新的矢量集Y，其中变换的公式为：}

(y_i) ¯=S(d_i) ¯ (i=1，2，3… ，N) 式 (3.8)

{45%：其中，S为P正交归一化后的特征向量，且y_i的协方差矩阵Q和P满足关系：}

Q=1/ NΣ_(i=1)^ N▒〖(y_ i)→(y_ i) →^ T〗=1/ NΣ_(i=1)^ N▒〖 S(d_ i)→(S〖(d_ i)→)〗^ T=〗 S1/ NΣ_(i=1)^ N▒〖(d_ i)→(d_ i)→^ T S^ T=〗 SPS^ T式(3.9)

且Q矩阵对角上的值为λ_1 □… λ□_ i，非对角上的值为0，矢量集 Y的每一个向

量是不相关的，由此可见，通过K-L变换，将原有数据在一个新的空间进行了表示，并且，原有数据的相关性被去除了。

3.4.2 PCA提取特征过程

PCA主要是通过K-L变换来获取一个新的空间上的数据表示，原始的空间中会有很多具有关联性的向量数据，而变换后的新空间中将具有相关性的矢量数据以及次要数据去除，得到了一组线性无关的特征值对应的特征向量，这样既达到了特征的维数大幅度降低的效果，而且完全可以表达出原先图像的大部分主要特征。PCA特征提取的一般过程如下：

读取训练样本的数据

在人脸识别中，人脸常用矩阵来表示，首先需要用一个二维数组来保存测试人脸集的数据，{41%：假定归一化后的人脸库中的每一张图像的分辨率是i*j，选取库中的样本数量是m，}{42%：将这些所有的m个样本用一个矩阵A[m][n=i*j]来表示，其中每一个单独的样本作为一个行向量，}列向量则为这些单独样本在位置相同处的不同的像素的信息。

计算其协方差矩阵

{41%：步骤(1)中得到了矩阵A[m][i*j]，需要进一步求其大小为n*n的协方差矩阵S，}{43%：协方差表示矩阵中两个任意的行或列之间的线性关系，}对于人连图像而言即图像中的像素关系，协方差的值为大于或者小于零说明矢量之间有关联，相反等于零则两个矢量不相关即不能相互表示。{48%：通过以上协方差的计算，就可以获取人脸图像中的像素关联性。}

求解特征值和特征向量

{47%：经过步骤(2)中的计算之后就需要计算协方差矩阵S的特征值和特征向量了，因为特征向量的维数是和与之对应的原始图像一致的，}从而每一个特征向量可以被看作其对应的图像。

{48%：选择主成份对训练集进行降维}

{40%：经过步骤(3)中特征值和对应转征向量的计算，下面就是要根据实验所需的维数进行降维操作，}{44%：选取的维数是按照特征值的大小进行降序排列的，例如降到100维，则从大到小选取前100个特征值所对应的特征向量，}此时主要的人脸像素信息则被保留了下来。

{55%：对测试集进行和训练集一样的操作}

最后还需要对测试集进行和训练集想用的降维操作，例如训练集从原始的n维降到了P维，{45%：通过特征向量的选取构成了矩阵B，其中B为n*p，则测试集也降到P维，}此时图像的维数远远低于原始图像，下一步的分类操作的效率则会大幅度提高。

图3.7各维数下的重构效果图(左边p=50，中间p=100，右边p=150)

3.5 本章小结

本章主要介绍了人脸部位区域的检测和对检测到的人脸部位区域进行特征提取送个两个环节。 这两部分内容是人脸分类的前提，这两步操作执行结果的质量影响着最终的人脸分类识别。 {47%：通过矩形特征、积分图、弱分类器、强分类器、级联分类器这几个过程详细介绍了基于Adaboost的人脸检测算法的原理和流程；} {41%：详细介绍了基于PCA的特征提取方法流程，对上一步提取到的人脸区域进行检测，为下一步人脸的分类识别做好了准备。}

人脸识别算法研究

4.1 概述

{74%：Android环境下基于人脸识别的身份验证是以人脸检测、人脸特征提取和人脸识别算法技术为核心的。} {98%：此系统的运行速度和识别效果也是＂人脸检测、人脸特征提取和人脸识别算法的效率和成功率为主要依据。} {95%：人脸检测的算法已在第三章进行了详细的介绍，在通过第三章研究的算法获得低维人脸特征后，} {100%：最后一个任务就是设计一个有效的分类器对人脸特征进行分类从而达到人脸识别的效果，} {100%：即将待识别的人脸图像特征与训练图像集中的图像特征进行匹配。} {98%：分类器的好坏直接决定着识别率的高低，因此，分类器的设计是人脸识别的一个关键部分。}

{69%：目前被应用较广泛的分类器主要有最近邻、人工神经网络和支持向量机等。} {98%：最近邻分类器属于基于样本间距离的非参数决策的分类器，即通过相似性函数计算测试样本和训练样本的距离来决定类别的归属。} {100%：最近邻分类器原理简单且计算速度快，故本文设计的分类器方法中也利用到该分类器。} {100%：最近邻分类器是基于线性的分类器，对几乎无表情变化的人脸识别率较高，} {84%：但针对表情丰富的人脸图像，获得的特征往往是非线性向量，此时最近邻分类器将不再适用人工神经网络分类器不必进行复杂的特征提取工作，} {96%：而且此分类器具有准确率和鲁棒性等方面的优势，但是此方法需要大景的样本训练才能达到较好的识别效果，} {96%：且每个图像的数据量造成需要的神经元数目也会很多，因此对运算能力和存储空间等方面的要求比较高。} {96%： SVM是一种基于统计学习的分类方法，其优势在于能有效解决小样本问题、高维数问题和非线性问题，} {100%：并且广泛应用于包括人脸检测在内的诸多模式识别领域中。}

{85%：此外，本文研巧的是Android环境下基于人脸识别的身份认证系统，所研究的并不是1：} N的人脸识别过程，而是基于1: {91%：1识别验证的人脸巧别，是一个人脸确认的过程。} {100%：由于人脸确认技术在本质上是一个二分类问题，} 而 SVM优势在于对模式进行二分类， {100%：因此本文选用最近邻和支持向量机相结合的分类器进行人脸识别，} 并且通过实验验证其有效性。

4.2 最近邻分类器

{54%：最近邻分类（ Nearest Neighbor Classifier）是由 Cover和 Hart在1968提出的基于样本间距离的一种分类方法，} {98%：研究者对应种方法进行了很多深入的研究，使其现在仍然是模式识别领域当中的一个重要的算法。}

{100%：最近邻分类方法首先计算测试样本与训练样本在特征空间中的距离，然后选取最近邻者判为该测试样本的类归属。} {98%：最近邻分类方法是以特征空间中各点间的距离大小表示着测试样本与训练样本的相似性，特征空间中两个点之间的距离越近，} {100%：两个样本也就越相似，各个决策区就是由各类训练样本点的集合所构成的区域。}

假如有C个类别，每类有N(i=1，2，…　，C)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，…　，N　式　(4.1)

其中，〖X_i〗^k表示ω_i类的第k个样本，‖┤‖表示距离。

设X=(x_1，x_2，…　{76%：，x_n)表示一个待识别的人脸特征向量，Y=(y_1，y_2，…}{95%：，y_n)表示某一个标准样本的特征向量，则用于表示X和Y的距离的方法有以下几种：}

(1)　欧氏距离（Euclidean　Distance）

D(X，Y)=√(∑_(i=1)^n▒　x_i-y_i　^2　)　式　(4.2)

(2)　角度距离（Angle-based　Distance）

D(　X，　Y)=-　cos⁡(　X，　Y)=-(∑_(　i=1)^　n▒〖　x_　i　y_　i〗)/√(∑_(　i=1)^　n▒〖　x_　i〗^2∑_　(　i=1)^　n▒〖　y_　i〗^2)　式(4.3)

(3)　均方误差距离（Sum　Square　Error　Distance，SSE）

D(X，Y)=∑_(i=1)^n▒　x_i-y_i　^2　式　(4.4)

曼哈顿距离（Manhattan　Distance）

D(X，Y)=∑_(i=1)^n▒　x_i-y_i　式　(4.5)

{100%：从上述内容可以看出最近邻方法的原理很简单，用于人脸分类时更是便捷且分类速度快，}　{92%：但其在分类错误率上有很大不足，值得做深入的研巧。}

4.3　支持向量机

{96%：由于　Android平台在存储能力和运算能力上的局限性，本文所研巧的人脸识别系统无法实现训练大量的人脸图像样本，}　{100%：此外人脸特征也具有较高的维数，因此，Android平台下的人脸识别存在典型的小样本问题，}　{97%：支持向量机是一种专门针对小样本、高维的非线性分类算法，十分适合于人脸识别。}　{100%：支持向量则是指训练集中的某些最靠近分类决策面且最难分的训练点，支持向量机分类器是}　{80%：一种通过对已知类别的训练样本进行学习从而实现分类的有监督的方法。}

4.3.1　线性可分情况下的最优分类面

图4.1线性可分情况示意图

{98%：如图4-1所示，三角形和菱形分别代表着二维空间中属于不同类的训练样本，类似于图像样本在特征空间中的投影点。}　{98%：H分类线将不同类别的训练样本正确分开，H1、H2分类间隔线与H分类线平行且过各类样本中距离H分类线最近的点。}　{100%：最优分类线就是在保证正确分类前提下使分类间隔线之间的距离达到最大而得出的分类线。}　{94%：将此二维空间推广到商维空间，对最优分类面的定义类似于最优分类线的定义。}

假设一个线性可分的样本集为{(x_1，y_1　)，(x_2，y_2　)，…　，(x_n，y_n　)}。　{79%：其中，样本特征向量　x_　i∈　R^　d，即　X是　d维实数空间中的向量，}　{58%：类标签　y∈{-1，+1}，即只有两类样本，通常称类标签为+1的样本为正样本，

} 称类标签为-1的样本为负样本。 设分类面方程为：

ω·x+b=0 式 (4.6)

{74%：其中， 为向量运算符，ω为为权值向量（其数学本质即是分类平面的法向量），且ω=[ω_1□□} ω□_2□…□ω_ n]， b是分类平面的常数项。 当g(x)≥0，y_1取+1，反之，y_1取-1。 {88%：对判断函数g(x)进行归一化处理，以使正负样本满足 g(x)≥1。} 则有：

{■(ω· x_ i+ b=-1， y_ i=-1@ω· x_ i+ b=1， y_ i=+1)┤ 式(4.7)

{74%：如图所示的两个条件，若特殊数据点x_i满足此条件，则称y_i为支持向量。} {100%：SVM的主要思想是寻找最优分类面，且使分类间隔面的距离达到最大。} {100%：由解析几何可知，最优分类超平面的单位法向量为：}

ω_0=ω/‖ω‖ □ 式 (4.8)

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

Dist=(x_1-x_2)·ω/‖ω‖ 式 (4.9)

又因为x_1，x_2分为别平面H1和H2上的点，所以有：

{■(ω·x_i+b=+1 @ω·x_i+ b=-1)┤ 式 (4.10)

整理式得：

ω·(x_1-x_2)=2 式 (4.11)

由以上公式可知：

Dist=(x_1-x_2)·ω/‖ω‖ =2/‖ω‖ 式 (4.12)

有此可知，当‖ω‖最小时，也即是1/2 ‖ω‖^(2)最小时，间隔Dist达到最大。 其约束条件为：

y_i (ω·x_i+b)≧1，i∈{1，2，… ，N} 式 (4.13)

{100%：因此，寻找最优分类面使间隔最大化问题就转换为求解二次函数最优规划问题。} {84%：用α_i来表示拉格朗日乘子，则拉格朗日函数定义为：}

L(ω□ b， α)=1/2ω^ Tω-Σ_(i=1)^ n▒〖α[y_ i(ω· x_ i+ b)-1]， α_ i≧0〗式(4.14)

{100%：将函数两边求偏导得出：}

∂ L(ω□ b，α)/∂ω=0⟹ω=Σ_(i=1) ^ N▒〖α_ i y_ i x_ i〗式(4.15)

(∂ L(ω□ b，α))/∂ω=0 Σ_(i=1) ^ N▒〖α_ i y_ i〗=0式(4.16)

由以上两个公式，整理得：

L(α)=Σ_(i=0)^ N▒〖α_ i-1/2Σ_ (i=1)^ N▒Σ_(j=1)^ N▒〖α_ iα_ j y_ i y_ j x_ i^ t x_ j〗〗 式(4.17)

约束条件为： Σ_(i=1)^N▒〖α_i y_i=0，α_i≧0〗

{82%：这就变成了拉格朗日对偶问题，可根据相关优化技术来求解出式中最优解系数α^*□} ω^*， b^*随后便可得到最优的分类函数：

f(x)= sgn[(ω^*· x)+ b^*]= sgn{ Σ_(j=1)^ n▒〖α_ j y_ j(x_ j· x)+ b^*〗} 式(4.18)

其中， sgn为符号函数， x_ j为支持向量，α_ j为对应的 Lagrange系数，{87%： b^*为分类的阈值，可以通过两类中的任何一对支持微量求中值获得。}

{85%：4.3.2 线性不可分情况下的广义最优分类面 }

{100%：人脸表情变化丰富，其造成获得的人脸特征往往是一种非线性向量。} {69%：在样本非线性可分的情况下上述最优分类面的求解方法会失效，训练样本在被 H分类线错误分类，} {92%：并且通过线性方法无法找出可 W正确分类样本的分类面。} {82%：此时可以在约束条件中引入一个松弛变量ζ_i≧0，调整约束条件为：}

y_i (ω·x_i+b)≧1-ζ_i，ζ_i≧0，(i=1，2，… ，N) 式 (4.19)

{82%：图4.2线性不可分情况示意图}

{82%：(1)当ζ_ i=0，对应训练样本中存在的可以正确线性分类的样本，在图中，在分类间隔线外侧(H1左侧，} {58%： H2右侧)以及分类间隔线上(如图中样本)的被正确分类的那些样本。}

{82%：(2) 当0[ζ_i[1，对应训练样本中被允许落在分类间瞄之间但没有被错误分类的样本，比如图中标号为2的样本。}

{79%：(3) 当ζ_i]1，对应训练样本中被允许可W被错误分类的样本，如图中标号为3}

的样本。

{91%：图4-2中标号为“1，2，3”的样本均是在线性不可分情况下的支持向量。}

{100%：此时引入附加错误代价系数C后，目标函数变为：}

ψ(ω，x，b)=1/2 ‖ω‖^2+CΣ_(i=1)^N▒ζ_i 式 (4.20)

{95%：其中，目标函数Ψ为非线性变换函数，C为错误代价系数，调整拉格朗日函数为：}
L(ω□ x，α)=1/2‖ω‖^2+ CΣ_(i=0)^ n▒〖ζ_ i-Σ_(i=0)^ n▒〖μ_ iζ_ i-Σ_(i=1)^ n▒〖α_ i[y_ i(ω· x_ i+ b)-1+ζ_ i]〗〗〗

式 (4.21)

{87%：与图中的推导过程类似，最终得到下面的对偶问题。}

最大化L(α)

L(α)=Σ_(i=1)^ N▒〖α_ i-1/2〗Σ_ (i=1)^ N▒Σ_(j=1)^ N▒〖α_ iα_ j y_ i y_ j〗(x_(i) x_ j)式(4.22)

约束条件为：

Σ_(i=0)^N▒〖α_i y_i=0，0≦α_i 〗≦C，i=1，2，… ，N 式 (4.23)

{87%：由此可见，在上述目标函数中松弛函数ζ_i和他们的拉格朗日系数都没有出现。}{89%：因此线性可分的目标函数与非线性可分的目标函数表达式完全相同，不同的是线性可分情况下的约束条件α_ i≧0，} {86%：在非线性可分情况下被替换为约束更强的0≦α_ i≦ C，因此线性可} {82%：分情况下的约束条件α_ i≧0可看做非线性可分情况下的一种特例。}

4.3.3 SVM的核函数

SVM的基本思想可归纳为： {61%：通过非线性变换，将输入样本空间映射到某一高维特征空间使其线性可分并在这个新空间中寻找最优分类面。} {100%：这样就通过非线性映射将非线性问题转化为某个高维空间中的线性问题。} 令ϕ(x)=〖[ϕ_1 (x)，ϕ_2 (x)，… {82%ϕ_ M(x)]〗^ T，其中输入向量 x表示 N维的输入向量，表示映射到 M维高维空间的非线性变换，} {100%：在新空间中构建的分类超平面为：}

Σ_(j=1)^M▒ω_j ϕ_j (x)+b=0 式 (4.24)

其中的ω_j为权值系数，b为偏置或负阈值。 令〖ϕ(χ)〗^*=1 ω^*=b，以上公式可以简化为

Σ_(j=1)^M▒ω_j ϕ_j (x)=0 式 (4.25)

也可以表示为：

ω·ϕ(x)=0 式 (4.26)

{66%：根据适合线性可分模式输入空间的式，将需要ϕ(x)替换 x，得到：}

ω=Σ_(i=1)^N▒〖α_i y_i 〗 ϕ(x_i) 式 (4.27)

{77%：将式代入式可以得出特征空间的分类超平面为：}

Σ_(i=1)^ N▒〖α_ i y_ iϕ^ T(x_ i)〗ϕ(x_ i)=0 式(4.28)

{90%：由式可得，仅使用特征空间的内积即在此空间中构造出最优超平面。} {90%：需要寻找一个函数K(x，x_i)，使得在构建超平面时避免考虑变换的形式。}

K(x， x_ i)=ϕ^ T(x)ϕ(x_ i)=Σ_(j=1) ^ M▒〖ϕ_

j(x)ϕ_ j(x_ i)〗式(4.29)

这个函数K(x，x_i)被称为内积核函数。

{100%：根据泛函分析中的Mercer定理可得核函数的定义：} {87%：K(x，y)为连续的对称核，其中x定义在闭区间α≦x≦b，y类似。}

核函数K(x，y)可以展开为级数：

K(x，y)=Σ_(i=1)^∞▒〖λ_i ϕ_i 〗(x)ϕ_i (y) 式 (4.30)

其中所有的λ_i]0，保证式一致收敛的充要条件是：

∫_b^a▒∫_b^a▒〖K(x，y)ϕ(x)ϕ(y)dxdy≧0〗 式 (4.31)

对于所有满足 ∫_b^a▒〖ϕ^2 (x)dx[∞〗的ϕ(x)成立。

{89%：可以看出式对于内积核函数K(x，y)的展开是Mercer定理的一种特殊情况。} {86%：Mercer定理指出如何确定一个候选是不是某个空间的内积核，但是没有指出如何构造函数ϕ_i (x)。}

常用的核函数可分为如下几类

线性核函数

K(x，y)=x·y 式 (4.32)

多项式核函数

K(x，y)=〖[(x·y)+1]〗^d，d=1，2，… 式 (4.33)

径向基核函数(Radio Basic Function，RBF)

K(x，y)=exp(-‖x-y‖/〖2σ□^2) 式 (4.34)

其中，σ是由用户决定的核宽度。

Sigmoid核函数

K(x，y)=tanh[V(x·y)+c] 式 (4.35)

基于以上内容，支持向量机算法步骤为：

算法4-1： 支持向量机(SVM)

输入： 训练样本X{(x_(1，) y_1)，(x_2，y_2)，… (x_(n，) y_n)}，其中，x_i是d维实数空间中的

向量，类标签义y_i ϵ{-1□+1}□

步骤1： 在约束条件

Σ_(i=1)^N▒〖α_i y_i 〗=0，0≦α_i≦C 式 (4.36)

下求解使目标函数：

L(α)=Σ_(i=1)^ N▒〖α_ i-1/2〗Σ_ (i=1)^ N▒Σ_(j=1)^ N▒〖α_ iα_ j y_ i y_ j〗(x_(i) x_ j)式(4.37)

最大化的α^*□

步骤2： 计算最优权值：

ω^*=Σ_(i=1)^N▒〖α^* y_i 〗 Y_i 式 (4.38)

其中Y_i为输出向量。

步骤3： {95%：对于待分类模式x，计算分类判别函数：}

f (x) = sgn[Σ_(i=0)^ N▒〖α^* y_ i〗 K(x_ i， x)+ b^*]式(4.39)

根据f(x)为+1或者-1的情况判定X的类别归属。

输出： f(x)的值

{87%：4.4 最近邻和支持向量机相结合的分类算法 }

{97%：由4.3节的 SVM分类器原理可知其是基于统计学习的训练模型，} {100%：该方法存在一个问题，假如给定一幅在数据库中不存在的人脸图像时，} {95%：基于 SVM的分类器很难进行拒识判断。} {100%：解决该问题的一种方法是在数据库中加入该人脸模式而进行学习，} {94%：但手机用户繁多做到将人脸图像种类都录入数据库作为训练样本不符会实际。} {100%：另一种方法是选择有代表性的人脸模式作为训练的负样本，但这些负样本的选取准则目前还没有相应的标准或者理论研究被提出。} {91%：本文所研究的系统是Android环境下基于人脸识别的身份验证系统，将手机机主本人的人验图像样本作为类内对象，} {100%：暂时随机选取非手机机主的其他人的人脸图像样本作为类外对象来训练 SVM，为了提高速度和增加拒识判断，} {100%：在进入 SVM分类器之前需进行一次初级分类。} {92%：这种将分类器组合起来的方式虽然不能提离识别率，} 但是却可以有效地降低误识率，{83%：这样的分类器适用在对准确度要求高的场合上，} {76%：本文设计的身份验证系统就是这样的一个应用场合。}

{98%：基于以上内容，本节提出了将运算速度快的最近邻分类器和识别率高的支持向量机相结合进行分类器的方法，} {96%：该算法先使用最近邻算法进行第一次分类，若结果小于既定的阈值，则用支持向量机分类器进行精确分类，} 否则，则拒识。

4.5 实验结果与分析

{47%：为了验证本章中提到的最近邻分类器、支持向量机分类器和两者相结合的分类器的效果，本节将在FERET人脸库上做实验。} {84%：本节采用的是第三章介绍的PCA特征提取方法，在此基础上对比分析这几种分类器算法的分类识别的性能。}

{90%：FERET人脸库是由美国国防部高级研究项目署和美国陆军研究实验室创建的，是目前最权威的用于检测人脸识别算法性能的人脸数据库。} {78%：本文所采用的人脸库是其中

的一部分，包括20名志愿者的200幅人脸图像（每人10张图像），} {100%：主要用于姿态、光照条件、面部饰物和表情等建模分析。}

{98%：针对FERET人脸库进行实验，选用每个人的10幅人脸图像，故共有200幅样本图像。} {95%：每次实验选定一人的人脸图像作为测试图像，则200幅样本图像中有10幅图像是正样本图像，} {89%：而其他的190幅图像则为负样本图像，故共进行200次实验。} 实验结果见表4.1：

{54%：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83.33%88.33%92.50%

错误识别率

拒绝识别率

正确拒绝率11.20%

16.67%

88.80%8.53%

11.67%

91.47%5.40%

7.50%

95.60%

表4.1基于FERET子库的实验结果

{100%：其中，正确识别表示样本图像集中正样本图像是本人的图像且被识别的情况；} {100%：错误识别表示样本图像集中正样本图像不是本人的图像但却被识别的情况；} {82%：拒绝识别表示样本图像集中正样本图像是本人的图像但却不被识别的情况；} {100%：正确拒绝表示样本图像集中正样本图像不是本人的图像且不被识别的情况。}

{100%：从上述实验中可以看出，在正确识别率和正确拒绝率上基于最近邻和支持向量机相结合的分类器在三者中是最高的，} {100%：因为在支持向量机进行分类之前，最近邻已经将不符合阈值的人脸排除了，这样就大大缩减了支持向量机分类的时间。} {98%：此外最近邻的错误识别率和拒绝识别率是三个方法中最低的，这也验证了在支持向量机分类器之前通过最近邻分类器的初次排除，} {100%：可以有效地减少拒识和错识的情况。} {83%：由此表明，基于最近邻和支持向量机相结合的算法相比于单独使用最近邻算法或支持向量机算法在人脸识别上有较好的效果}

4.6 本章小结

{100%：本章首先介绍最近邻分类器的基本原理。} {100%：然后讨论了支持向量机在线性可分情况下的最优分类面和线性不可分情况下的广义最优分类面，} {100%：采用合适的

核函数可以实现将非线性问题转化为某个高维空间中的线性问题。} {82%：最后研究了最近邻和支持向量机相结合的分类器算法的基本思想。} {48%：实验证明，本章研究的最近邻和支持向量机相结合的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率上有所提高。}

5 Android环境搭建与人脸识别系统的实现

5.1 Android系统概述与优势

{98%: Android是一种基于 Linux内核开放源码的智能操作系统，最初由 Andy Rubin开发，安迪·鲁宾开发这个系统的最初目的是创建一个先进的数码相机操作系统，} {100%：但是当智能手机市场快速发展时， Android后来被改造为主要面向智能手机的OS，后来被 Google公司收购。} {71%：Android 公司成为 Google 旗下的一部分，在 2017 年 Google 开发者大会上Google 宣布目前全球Android月活跃用户已经超过20亿。} {98%：Android系统是以 Linux 为内核采用分层架构，包括底层 Linux 内核、库、应用程序框架以及应用程序层组成。} {100%：由于其开放性，理论上可以在任何电子电器产品上运行，这对未来产生更智能化的电子产品有着巨大的推动作用。} {100%：得到了手机硬件厂商和软件应用开发商的支持，一经推出移动终端市场占有率就节节攀升。}

{100%：和其他OS相比，Android系统是一个完全的开放性平台，Google发布了Android的源代码。} {98%：Android的完整开发生态环境能够使得从底层运行到应用程序的开发和发布都能在监管下自由的进行，对移动产业发展有巨大的推进作用。} {100%：开发的平台允许任何移动设备厂商加入到Android联盟中来，对开发者来说可以利用丰富的资源进行开发。} {100%：对于用户来说，不仅可以使用全世界的优秀开发着开发的应用程序，还可以用开放的源代码进行手机移植。} {100%：开放的平台会带来竞争，用户们会从中选取优质的产品。}

{100%：Android系统提供开发者一个自由的开发环境，去除各种限制性的条条框框，鼓励开发者的创新性。} {100%：所有应用程序均运行虚拟机上，该虚拟机分配给所有应用程序所需的硬件资源和通信接口，} {100%：第三方应用程序在 Android平台下被平等的对待，不同应用程序之间的信息可以方便的共享。} {100%：Android平台提供的开发插件API以及开发库使得开发者在开发工作中能够很好的调用已有的底层内核，缩短了开发周期。}

5.2 Android应用组件

Android的应用组件分别是： Activity活动，BroadcastReceiver广播，Service服务，ContentProvider内容提供者。 这里主要用到的是Activity。

{97%：Activity 是一个主要应用组件，用户可与其提供的屏幕进行交互，以执行拨打电话、拍摄照片、发送电子邮件或查看地图等操作。} {100%： 每个 Activity 都会获得一个用于绘制其用户界面的窗口。} {100%：窗口通常会充满屏幕，但也可小于屏幕并浮动在其他窗口之上。}

{100%：一个应用通常由多个彼此松散联系的 Activity 组成。} {100%： 一般会指定应用中的某个 Activity 为主Activity，即首次启动应用时呈现给用户的那个Activity。} {100%： 而且每个 Activity 均可启动另一个 Activity，以便执行不同的操作。} {100%： 每次新 Activity 启动时，前一 Activity 便会停止，但系统会在堆栈（返回栈）中保留该 Activity。} {100%： 当新 Activity 启动时，系统会将其推送到返回栈上，并取得用户焦点。} {100%：返回栈遵循基本的“后进先

出”堆栈机制，因此，当用户完成当前 Activity并按“返回”按钮时，} {100%：系统会从堆栈中将其弹出（并销毁），然后恢复前一 Activity。}

{100%：当一个 Activity 因某个新 Activity 启动而停止时，系统会通过该 Activity 的生命周期回调方法通知其这一状态变化。} {100%: Activity因状态变化—系统是创建 Activity、停止 Activity、恢复 Activity还是销毁 Activity—而收到的回调方法可能有若干种，} {100%：每一种回调都会为您提供执行与该状态变化相应的特定操作的机会。} {100%： 例如，停止时，您的 Activity 应释放任何大型对象，例如网络或数据库连接。} {100%： 当 Activity 恢复时，您可以重新获取所需资源，并恢复执行中断的操作。} 这些状态转变都是 Activity 生命周期的一部分。

{47%：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，} 会在不同的情形下会调用几个可重写的方法，这些方法被称为生命周期方法。深入理解Activity的生命周期对于开发者来说非常重要，只有这样才能在使用Activity的时候更加游刃有余，并给用户带来更好的体验。 Activity基本上以三种状态存在：

(1) 运行状态： {68%：此 Activity 位于屏幕前台并具有用户焦点，处于运行状态的Activity最不容易被系统回收。}

(2) 暂停状态： {100%：另一个 Activity 位于屏幕前台并具有用户焦点，但此 Activity 仍可见。} {100%：也就是说，另一个 Activity 显示在此 Activity 上方，并且该 Activity 部分透明或未覆盖整个屏幕。} {100%：暂停的 Activity处于完全活动状态（ Activity对象保留在内存中，它保留了所有状态和成员信息，} {100%：并与窗口管理器保持连接），但在内存极度不足的情况下，可能会被系统终止。}

(3) 停止状态

该 Activity 被另一个 Activity 完全遮盖（该 Activity 目前位于“后台”）。 {100%： 已停止的 Activity 同样仍处于活动状态（Activity对象保留在内存中，它保留了所有状态和成员信息，但未与窗口管理器连接）。} {100%： 不过，它对用户不再可见，在他处需要内存时可能会被系统终止。}

{44%：Activity最常用的生命周期方法一共有七个，这七个生命周期方法对应Activity的每一个不同的状态。}

(1)onCreate()： 在活动第一次被创建的时候调用，初始化操作都在onCreate()方法中完麻例如布局的设置，事件的绑定。

(2)onStart()： 在活动由不可见变为可见的时候调用。

(3)onResume()： 当活动进入返回栈顶并且此时它处于运行状态的时候便会调用该方法。

(4)onPause()： 当活动不处于栈顶但并没有不可见时调用该方法，例如弹出对话框。

(5) onStop()； 当活动不处于栈顶且完全不可见的时候调用该方法，例如有一个新的全覆盖的Activity进入栈顶。

(6) onDestroy()； {66%：在某个Activity被销毁之前调用该方法，调用onDestroy()方法后该Activity会被销毁。}

(7)onRestart()； 当活动由不可见的状态重新变为可见状态时调用该方法。Activity生命周期流程如图5-1。

图5.1 Activity的生命周期

Activity启动模式有四种； Standard、SingleTop、SingleTask、SingleInstance。

(1) Standard: 默认模式，无论是否已经存在该Activity，都只是继续创建，所以会创建相同的活动。

(2) SingleTop: {41%： 如果某个Activity使用了这种启动找模式，再次后动该Activity的时候会看找顶是否存在送个Activity，如果存在则不会继续创建。}

(3) SingleTask: {45%： 如果某个Activity配置了送种启动找模式，那么该Activity在整个App中只会存在一个实例。}

(4) SingleInstance: 设置LaunchMode为SingleInstance，它会给其创建单独的任务栈。

5.3 Android开发环境搭建

5.3.1 JDK

本人的开发平台是基于 windows10环境下搭建的，首先安装 JDK(Java Development Kit)， {93%： JDK是 Java开发的基础环境和核心部分，包括 Java工具和开发基础类库文件。} {100%：将在网上下载完成的JDK软件安装包打开按照默认路径安装。} {91%：安装完JDK后配置环境变量，如图5.2所示。}

图5.2 Java环境变量配置

{81%：鼠标右键点击我的电脑，在属性里找到高级系统设置，在高级选项卡里鼠标点击环境变量，系统环境变量里选择新建，} {100%：名称填写 PATH，值为 JDK安装目录中bin文件夹路径，用同样的方法添加 CLASSPATH变量，值为 JDK安装目录中 lib文件夹和 demo文件夹的路径，} 其中安装目录可以使用 Windows系统规定的% JAVA_HOME%作为引用。 {93%：完成后如图5.3，点击开始，在运行输入 CMD，确定。}

图5.3 Java环境测试

5.3.2 Android Studio

Android Studio是Google于2013I/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5.4 Android Studio界面

5.3.3 JNI技术与Android NDK

{98%：JNI(Java Native Interface)标准是Java平台的一部分，其最大的特色就是允许Java代码和其他语言，例如C语言写的代码互相调用。} {100%：平台的可移植性会受到使用Java与本地已编译的代码交互的影响。} {100%：本文所采用的人脸识别算法都是基于C语

言进行研发的，为了完成人脸图像识别的特殊任务，提高系统执行效率，必须使用JNI技术。}

Android NDK(Android Native Development Kit)是当开发者使用类似 C 语言原生代码语言执行部分程序时， {100%：为了让 Android程序运行在 Dalvik虚拟机中不与 JAVA语言冲突的一种技术。} NDK将是Android开发支持C语言开发的开端。{100%：Android NDK更新了交叉编译功能支持对调用本地代码的JNI接口和一些其他的库文件。} {100%：Google官方允许应用程序源代码通过JNI调用在本地进行实现。}{100%：Android应用程序开发将具有更高的灵活性。}

图5.5 JNI调用流程图

5.3.4 OpenCV

{98%：Open CV是一个同Android系统类似的具有开放性特征的跨平台计算机视觉库。}{97%：很多图像处理和计算机视觉方面的通用都在Open CV上得到了实现。}

Open CV的出现改善了在计算机视觉市场上并没有标准API的现状，目前的计算机视觉不足主要有：

目前的研究代码运行速度慢且不稳定，而且兼容也很差。

MATLAB和Simulink开发成本较高。

解决方案过于依赖硬件。

5.4 系统设计与实现

5.4.1 功能分析

{56%： Android平台上的人脸检测与识别系统主要由图像采集模块、人脸图像预处理模块、} {63%：人脸检测模块、人脸注册模块和人脸识别模块等共五个模块组成。} 具体流程如图5.6。

{80%：图5.6 人脸识别身份认证系统流程图}

图像采集模块： 利用 Android平台摄像头进行图像采集， 调用 OpenCV库，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光照补偿、滤波去噪处理和几何归一化的处理等处理。

人脸检测模块： {45%：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。}

人脸识别模块： {43%：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主人，否则提示该人不是手机主人，请摆正人脸重新识别。

5.4.2 具体实现

首先需要对手机主人的人脸进行注册，为了确定注册的人脸图像有效，首先需要对注册时

候的人脸进行检测，{49%：防止在人脸检测环节出现问题，导致下一步人脸分类识别步骤无法进行。} 检测到人脸后可以对该人脸样本进行注册，输入该样本的信息，保存特征信息至xml文件中，保存该样本的相关信息至数据库中。

从手机摄像头采集到数据后经过预处理，再到人脸检测，识别出人脸关键区域，{44%：将人脸关键区域提取特征之后与数据库里存的手机主人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主人本人。 识别效果与部分代码如下：

{49%：代码5.1 人脸关键区域检测代码 代码5-2 人脸识别代码}

(a)正常表情下的测试(b)微笑表情下的测试

(c)皱眉表情下测试(d)其他同学测试

图5.6 身份认证测试结果

5.4.3 结论

PC环境： ThinkPad S2

PC系统： Window 10 64位

开发软件： Android Studio

手机： HuaWei Mate9、XiaoMi5

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。

图像尺寸准确率耗时 (ms)占用内存(MB)

人脸检测1280*72095%1526

人脸识别1280*72080%18130

表5.1 在Huawei Mate9上的测试结果

图像尺寸准确率耗时(ms)占用内存(MB)

人脸检测1280*72095%3728

人脸识别1280*72080%27634

表5.2 在XiaoMi5上的测试结果

5.5 本章小结

{50%：本章先是介绍了Android系统的优势和前景，选择使用Android系统的好处，然后介绍了Android的核心技术和组件；} {44%：通过介绍在Android中使用OpenCV函数库的方法介绍了JNI技术，OpenCV函数，NDK；} 分析了本文系统的功能、系统的架构并对系统进行了

实现，然后分别在HuaWei Mate9和XiaoMi5上测试实验结果。

6 总结与展望

6.1 本章小结

{41%：本文在大量的国内外文献阅读的情况下结合了当前基于Android平台的人脸识别现状进行了相关算法的研究和系统的实现。} 信息时代已经到来，生物识别代替传统的身份验证方式已经是大势所趋，而移动互联网也占据了整个互联网的大半江山， {42%：移动端下人脸识别系统的研究具有重要的价值意义，本文基于此主要做了如下工作；}

{46%：一、分析了本文课题的研究背景及意义，介绍了人脸识别过程各个环节的研究现状及相关算法，引出本课题研究内容。}

二、对图像预处理模块进行了研究，为了增强图像的一些主要细节，一定程度上解决光照带来的干扰， {48%：增强算法的鲁棒性，对原始图像进行了灰度化、直方图均衡化、图像平消的操作。}

{53%：三、对人脸识别过程中的人脸检测环节和特征提取环节进行了深入的研究，} {51%：详细研究了 Adaboost算法的实现原理及过程并使用该算法进行了人脸检测的操作；} {42%：详细研究了PCA算法的实现原理及过程并使用该算法进行特征提取操作。}

{63%：四、对人脸识别算法进行了研究。} {83%：本文设计的基于人脸识别的手机身份认证系统并不是1：} N的人脸识别过程，而是基于1: {100%：1识别验证的人脸识别，是一个人脸确认的过程。} {42%：因为手机主人只有一个，所以所要做的工作只是确认摄像头拍的人脸是不是手机主人，在本质上是一个二分类问题，} {96%：而 SVM优势在于对问题进行二分类，又因为最近邻分类器运算速度快和能有效的做拒识判断的优点，} {97%：因此本文选用最近邻和支持向量机相结合的分类器进行人脸识别。} {100%：该算法先使用最近邻算法进行第一次分类，若结果小于既定的阈值，则用支持向量机分类器进行精确分类，否则，则拒识。} {100%：实验验证了该分类器比单独使用SVM算法在人脸识别性能优势。}

{71%：五、Android手机身份认证系统设计与实现。} {100%：对系统进行了需求分析，然后结合Android平台特性，根据软件工程的思想对系统进行设计与实现。} {94%：在Android平台上完成了系统框架的搭建，并用Java实现了人脸检测、人脸特征提取、人脸识别、身份认证等核心模块的编写。}

6.2 下一步工作

本文对人脸识别相关的算法和 Android系统进行了大量的文献的阅读和仿真实验，但是由于作者水平有限， {49%：仍然有很多不足，本文的不足以及下一步工作主要有以下几个方面：}

一、随着近几年移动端硬件的持续升级，手机的硬件性能已经相当高，但是相对 PC端来说，还是非常的不足， 识别速度相对 PC端还是相对较慢，对于数据库非常大以及算法比较复杂的情况比较难保证实时性的需求。

{43%：二、光照一直是人脸识别一个难题，虽然本文对图像进行了预处理提高了图像对于光照问题的鲁棒性，} 但是仍然难以满足实际应用中的离要求，在后续的工作中，需要研究一些对光照鲁棒性较好的算法并对其进行改进。

检测报告由PaperPass文献相似度检测系统生成