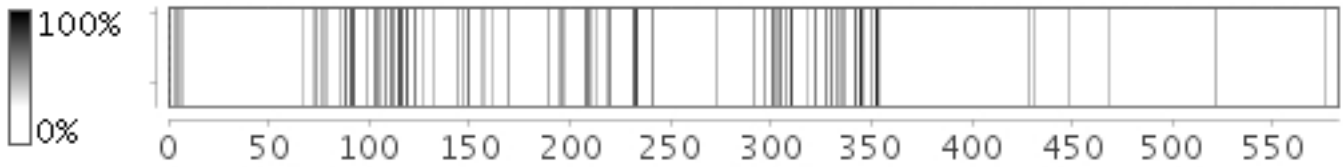
PaperPass检测报告简明打印版

比对结果（相似度）：
总　体：21 %　*（总体相似度是指本地库、互联网的综合比对结果）*
本地库：20 %　*（本地库相似度是指论文与学术期刊、学位论文、会议论文数据库的比对结果）*
互联网：3 %　*（互联网相似度是指论文与互联网资源的比对结果）*

编　号：5745DAFFF213E0MNG
标　题：人脸跟踪与识别系统实现
作　者：杨晨
长　度：17476 字符(不计空格)
句子数：582句
时　间：2016-5-26 1:03:59
比对库：学术期刊、学位论文（硕博库）、会议论文、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）：

1. 相似度：2 % 篇名:《基于上下文驱动多贝叶斯分类器的人脸检测定位系统》
   来源：学位论文 成都理工大学 2007 作者: 王曦
2. 相似度：1 % 篇名:《基于Gabor小波网络的人脸识别研究》
   来源：学位论文 北京交通大学 2007 作者: 谢竞
3. 相似度：1 % 篇名:《人脸检测与识别技术的研究》
   来源：学位论文 重庆大学 2010 作者: 张建慧
4. 相似度：1 % 篇名:《浅析人脸识别技术的现状和发展趋势》
   来源：学术期刊 《通讯世界》 2015年13期 作者: 陈奇毅
5. 相似度：1 % 篇名:《基于嵌入式隐Markov模型的人脸识别技术研究与实现》
   来源：学位论文 国防科学技术大学 2007 作者: 戴芬
6. 相似度：1 % 篇名:《基于动态反馈的特征抽取及人脸识别应用研究》
   来源：学位论文 扬州大学 2014 作者: 范冠杰
7. 相似度：1 % 篇名:《基于流形的特征抽取及人脸识别研究》
   来源：学位论文 扬州大学 2009 作者: 曹丽
8. 相似度：1 % 篇名:《基于小波变换人脸识别方法研究》
   来源：学位论文 电子科技大学 2014 作者: 李彦

9. 相似度：1％ 篇名:《基于ARM移动机器人中的人脸识别》
来源：学位论文 暨南大学 2014 作者: 张立峰
10. 相似度：1％ 篇名:《基于频谱脸和Fisherface的人脸识别系统的研究与实现》
来源：学位论文 东北大学 2005 作者: 韩凌
11. 相似度：1％ 篇名:《全国中小水库管理信息系统的设计与实现》
来源：学位论文 北京邮电大学 2009 作者: 李俊
12. 相似度：1％ 篇名:《基于统计的人脸识别方法研究——GLRAM与LPP的子空间法》
来源：学位论文 江南大学 2009 作者: 徐冬冬
13. 相似度：1％ 篇名:《高速公路防逃费中的人脸识别技术》
来源：学术期刊 《城市建设理论研究（电子版）》 2013年32期 作者: 汤永成
14. 相似度：1％ 篇名:《基于两级Adaboost的LBP快速人脸识别的实现与应用》
来源：学位论文 复旦大学 2008 作者: 褚力
15. 相似度：1％ 篇名:《基于决策融合与距离学习的人脸识别算法研究》
来源：学位论文 华中科技大学 2012 作者: 焦一正
16. 相似度：1％ 篇名:《人脸识别技术的研究现状与展望》
来源：学术期刊 《安防科技》 2011年10期 作者: 董琳 赵怀勋
17. 相似度：1％ 篇名:《情感神经网络及其在人脸识别中的应用研究》
来源：学位论文 江西理工大学 2012 作者: 张丽
18. 相似度：1％ 篇名:《自动人脸检测跟踪识别系统研究与实现》
来源：学位论文 东北大学 2008 作者: 涂平
19. 相似度：1％ 篇名:《人脸识别发展分析》
来源：学术期刊 《计算技术与自动化》 2006年4期 作者: 李祥宝
20. 相似度：1％ 篇名:《表情与光照变化下几种人脸识别方法的研究与改进》
来源：学位论文 哈尔滨工业大学 2009 作者: 陈昌凤
21. 相似度：1％ 篇名:《人脸图像检测与识别方法综述》
来源：学术期刊 《自动化技术与应用》 2004年12期 作者: 王科俊 姚向辉
22. 相似度：1％ 篇名:《GE-signa MRI训练系统主体框架设计及主要功能的实现》
来源：学位论文 山东中医药大学 2011 作者: 徐公明
23. 相似度：1％ 篇名:《金融中专教务信息管理系统的设计与实现》
来源：学位论文 大连理工大学 2009 作者: 宋琳
24. 相似度：1％ 篇名:《基于单视图的多姿态人脸识别》
来源：学位论文 江苏科技大学 2013 作者: 杨姝
25. 相似度：1％ 篇名:《基于人脸识别的图像考勤系统设计与实现》
来源：学术期刊 《无线互联科技》 2015年10期 作者: 王静
26. 相似度：1％ 篇名:《基于web的人脸检测与人脸识别》
来源：学位论文 西安电子科技大学 2009 作者: 卓永亮
27. 相似度：1％ 篇名:《基于ARM架构的嵌入式人脸识别技术研究》
来源：学位论文 华东师范大学 2008 作者: 李外云
28. 相似度：1％ 篇名:《视觉社交平台的研究与设计》
来源：学术期刊 《科学与财富》 2015年12期 作者: 瞿瑞坚 张琪
29. 相似度：1％ 篇名:《人脸自动识别技术综述》
来源：学术期刊 《苏州市职业大学学报》 2010年1期 作者: 尚丽 陈杰 张愉
30. 相似度：1％ 篇名:《基于肤色模型和AdaBoost算法的人脸检测研究》
来源：学位论文 长安大学 2014 作者: 王琳琳

31. 相似度：1％ 篇名:《一种改进的AdaBoost人脸检测算法》
来源：学术期刊 《电视技术》 2014年15期 作者: 李文昊 陈泽华
32. 相似度：1％ 篇名:《人脸识别技术及其在视频监控系统中的应用》
来源：学位论文 南京理工大学 2009 作者: 郭晓平
33. 相似度：1％ 篇名:《基于fast-AdaBoost算法的人脸检测与识别方法研究》
来源：学位论文 太原理工大学 2014 作者: 齐光景
34. 相似度：1％ 篇名:《基于Adaboost算法训练分类器的研究及其在人脸检测中的应用》
来源：学位论文 天津工业大学 2010 作者: 谢欢
35. 相似度：1％ 篇名:《基于C++的人脸识别系统的设计与实现》
来源：学位论文 西安电子科技大学 2011 作者: 买买提江玉山
36. 相似度：1％ 篇名:《基于DM6446的嵌入式近红外人脸识别系统》
来源：学位论文 哈尔滨工业大学 2010 作者: 邵天双
37. 相似度：1％ 篇名:《基于分块DCT和双向2DPCA的人脸识别》
来源：学位论文 辽宁师范大学 2013 作者: 尉秀芹
38. 相似度：1％ 篇名:《人脸识别技术综述》
来源：学术期刊 《电子测试》 2015年10期 作者: 徐晓艳
39. 相似度：1％ 篇名:《基于openCV的人脸检测系统的设计》
来源：学术期刊 《电子设计工程》 2012年10期 作者: 陈志恒 姜明新
40. 相似度：1％ 篇名:《多姿态人脸识别系统的研究与实现》
来源：学位论文 东南大学 2012 作者: 韩小雪
41. 相似度：1％ 篇名:《基于块匹配的图像去噪和超分辨率重建算法研究》
来源：学位论文 大连理工大学 2013 作者: 王凯
42. 相似度：1％ 篇名:《关于AdaBoost有效性的分析》
来源：学术期刊 《城市建设理论研究（电子版）》 2013年22期 作者: 高宇宾
43. 相似度：1％ 篇名:《视频监控中基于人脸识别的身份验证》
来源：学位论文 河北工业大学 2011 作者: 张双科
44. 相似度：1％ 篇名:《人脸识别技术的研究与应用》
来源：学术期刊 《科技信息》 2010年14期 作者: 刘鹭 董立文 姜鹏 马月
45. 相似度：1％ 篇名:《基于工作流的企业信息管理系统的设计与实现》
来源：学位论文 电子科技大学 2014 作者: 郝洁明
46. 相似度：1％ 篇名:《基于肤色和AdaBoost算法人脸检测的研究》
来源：学位论文 燕山大学 2010 作者: 孔祥栋
47. 相似度：1％ 篇名:《基于上下文信息的运动目标跟踪算法的研究》
来源：学位论文 哈尔滨工业大学 2011 作者: 孙钟前
48. 相似度：1％ 篇名:《基于Adaboost算法的人脸检测方法研究》
来源：学位论文 南京邮电大学 2007 作者: 师雪丽
49. 相似度：1％ 篇名:《人脸识别研究综述》
来源：学术期刊 《城市建设理论研究（电子版）》 2013年23期 作者: 杨振福 张永立
50. 相似度：1％ 篇名:《Boosting分类算法的应用与研究》
来源：学位论文 兰州交通大学 2012 作者: 李想
……

互联网相似资源列表：

1. 相似度：3％ 标题：《目标检测的图像特征提取 - lucky_greenegg的专栏 ...》
   http://blog.csdn.net/lucky_greenegg/article/details/11713369
2. 相似度：1％ 标题：《android studio使用问题及说明 - chaoyue007...》
   http://blog.csdn.net/chaoyue0071/article/details/42123059/

全文简明报告：

摘要

{45％：在这个互联网飞速发展的时代，计算机技术已经深入到每个人的生活中。} {53％：而计算机图形图像技术在生活中被也越来越多的运用起来。} {50％：人脸跟踪与识别系统，涉及到人脸的检测与人脸的识别。} {43％：将其运用到Android系统之中，从而实现了通过手机摄像头识别人脸的功能。}

{43％：本文首先阐述了人脸检测Haar分类器、人脸识别LBP算法的相关内容，然后根据实际的Android开发内容编写了需求报告。} {40％：首先通过Android中Camera相关类获取视频流的一帧，然后使用Haar分类器进行人脸检测，获取人脸矩阵信息。} 对录入的人脸进行训练后再使用LBP算法进行人脸识别，最后在屏幕上绘制出人脸的矩阵与识别出的人的姓名。 如此循环，从而实现了对视频流的人脸识别。

关键词： 人脸识别； haar分类器； LBP算法； Android

Abstract

In this era of rapid development of Internet, computer technology has been deep into everyone ’ s life. The computer graphics technology is more and more used in our life. Face tracking and recognition system, related to face recognition and face detection. As it applied to Android system, enabling moblile phone camera face recognition function.

The paper describes the face detection Haar classifier, the content LBP face recognition algorithm, and then write a report based on the actual needs of the Android development content. Get through the first JavaCV Camera Class a video stream, and then use the Haar classifier for face detection, face matrix to obtain information. On the entry face training before using LBP face recognition algorithm, and finally draw the human face of the matrix and identified the person ’ s name on the screen. In the next frame, then the above operations, in order to achieve the recognition of the video stream.

keywords： face recognition； haar classifier； LBP algorithm； Android

目录

第1章 绪论

{80%： 1.1 人脸识别的研究背景及国内外研究现状}

在最近10年中，计算机科学技术取得了巨大的发展，人们的生活水平得到了很大的提升，人们越来越多的关注社会信息。 在很多行业中，对人员进行的信息进行确认与身份的辨别的需求越来越大。 {43%：例如公安机关设置摄像头，来监控地铁，商场，火车站等人流量打的地方，根据画面中出现的人脸来寻找犯罪嫌疑人。} 在以前这样的工作通常是由人工完成的，不仅消耗很多的人力，也消耗很多的时间； {43%：在机场，登机时，工作人员会对将人脸与身份证进行核对；} {41%：在银行取钱时，银行需要对取钱的用户进行人脸识别，判断其是否是本人；} {52%：在海关，同样也需要人脸识别技术来确认出入境人员是否为本人，这些在生活中切实存在的需求极大的推动了近些年人脸识别技术的飞速发展与应用[1]。}

{53%：人脸识别技术是通过计算机图形学，提取人脸的特征，通过这些特征，与训练好的样本进行对比，从而进行身份验证的一种技术。} 人脸特征与人身上其他的特征一样，例如指纹和虹膜，都具有唯一性和不容易被伪造的特性，这是作为身份鉴别依据的前提； {60%：对比其他生物特征的识别技术，人脸识别技术在操作上更简单，结果更直观，更加隐蔽。} {42%：因此，在刑事案件的侦破，信息安全领域等都有广泛的应用前景。}

{43%：人脸识别技术在最早的时候限制很大，只能对单一背景的正面灰度图像进行识别，也就是只能在二维中进行人脸识别。} {42%：随着对多种姿态下人脸识别的研究，包括正面和侧面，人脸识别正在向三维方向发展，在维度提升的同时，识别率也在逐步提高。} 在这个过程中，人脸识别技术越来越成熟，但是还是存在着很多的问题，比如在复杂的背景下，系统很难识别出其中的人脸，跟踪也存在比较大的问题。 {44%：[2]人脸识别技术并非纯粹的数字技术，它需要研究人员拥有计算机图形学、计算机视觉、生物特征技术等学科的知识，} 是一个融合了多种学科的技术，因此对研究人员有非常高的要求。 环境和人脸本身都有很大的不确定性，在环境方面，光照和图像采集工具会对采集到的图像有比较大的影响， 而人脸本身由于表情，身体姿态和脸上戴的物件的不同，会对人脸识别造成影响。 {59%：综上所述，人脸识别是非常有挑战性的课题。}

人脸识别有悠久的研究历史。 {77%：高尔顿分别在1888年和1910年在《Nature》杂志上发表了两篇关于利用人脸来识别身份的文章。} {64%：这两篇文章对人类自身如何进行人脸的识别进行了分析。} 在当时的科学文化环境下，并没有涉及到人脸的自动识别。

{63%：1965年陈和布莱德索在Panoramic Research Inc.发表了关于自动人脸识别的研究论文。} {81%：布莱德索建立了一个半自动的人脸识别系统。} {71%：该系统识别人脸是以人脸的特征点的间距和比率等参数特征。} 这种思想也成为之后一段时间的主流。 这种人脸识别思想的特点有： {51%：①将人脸拆分成若干个部件，作为特征进行识别，主要利用的是各个部件的信息和部件之间的几何关系。} 这是一种很直观的方法，对人脸图像有很高的要求。 如果人脸图像中的人脸不是正面照或者表情有变化的话识别率就会很低。 {50%：为了弥补这种方法的缺点，后来出现了更好的人脸识别方法。} {42%：根据样本库中的样本和需要识别的人脸的灰度图的比较来对人脸进行鉴别，这种方法叫做模板匹配方法。} ②人脸识别对环境的要求非常高，在单一背景或者没有背景的情况下，人脸的位置非常容易获得，相反，在复杂的背景下，人脸的位置非常难获得。 在这段时间，人脸识别由于这个原因，并没有被应用到现实场景当中[3]。

{57%：1964年至1990年是人脸识别的第一个阶段，这一阶段人脸识别通常作为一个普通的模式识别问题来研究。} {64%：这个阶段采用的技术方案是基于人脸几何特征的方法。} {54%：研究者们花费了很多时间在对面部剪影曲线的结构特征的提取和分析方面。} {59%：而人工神经网络也曾被用于人脸识别的问题中。} 早期，除了布莱德索在进行自动人脸识别之外还有戈登斯泰因、金出武雄等。 {62%：金出武雄是人脸识别领域较活跃的人物，它在1973年，于京都大学完成了自动人脸识别方面的博士论文。} {57%：如今，他成为了卡内基-梅隆大学机器人研究院的教授。}

第二个阶段是1991年至1997年。 {52%：在短短的8年时间内，人脸识别的研究进入了高潮期，取得了很多的成就。} {70%：许多具有代表性的人脸识别算法在段时间内诞生。} {48%：比如著名的FERET人脸识别算法，麻省理工学院媒体实验室的特克和潘特提出的“特征脸”方法。} “特征脸”方法是这一时间段内的代表作，衍生出了之后许多的算法。 {94%：如今特征脸已经与归一化的协相关量方法一起成为人脸识别的性能测试基准算法。} {58%：麻省理工学院人工智能实验室的布鲁内里和波基奥进行了基于结构特征的方法与模板匹配方法的对比，} {77%：得出了模板匹配要优于基于特征的方法这个结论，促进了基于表现的线性子空间建模和基于统计模式识别技术的人脸识别方法的发展。} 这一时期主要的成果有： {69%：贝尔湖米尔等提出的Fisherface人脸识别方法；} {79%：麻省理工学院的马哈单提出的基于双子空间进行贝叶斯概率估计的人脸识别方法。}

第三个阶段是从1998年至现在。 为了解决在光照，姿态等非理想的情况下人脸识别鲁棒性较差的问题，研究者么注重研究了这一部分。 {57%：吉奥盖迪斯等人提出基于光照锥模型的多姿态、多光照条件人脸识别方法。} {68%：布兰兹和维特等提出了基于3D变形模型的多光照和姿态条件人脸图像分析与识别的方法。} 它是基于合成的分析技术。 {50%：这种方法使得人脸可以用简单的举证特征作为特征，将大量弱分类器组合成了强分类器，} 并且使用联级技术提高检测速度，为现在实时人脸识别打下了很好的基础。 {84%：沙苏哈在2001年提出了一种基于熵图像的人脸图像识别和绘制技术。} 为解决光照问题提供了一种重要的思路。

现在人脸识别的算法主要分成3种。

(1)基于肤色划分的人脸检测方法。 {49%：首先确定肤色模型，检测出其中的肤色区域，获得可能存在的人脸区域。} {40%：之后是对人脸区域的检测，通过区域特征的方法对人脸区域进行人脸检测，从而区分出具有类肤色的其他物体。} 这种方法在复杂背景的环境下会有比较大的误差。 在复杂背景下，人脸区域可能和其他肤色区域混杂在一起，从而造成混淆，在获得肤色区域后任然无法判断出此区域是人脸区域。 另一种情况是光照

和面部表情造成的影响。 光照和面部表情会对人脸检测造成影响，区域特征会受到影响，在这种情况下会使用聚类，归并，验证的方案来减少其带来的影响。 {45%：首先将人脸肤色像素按照严格的几何关系去拆分，再将其按照一定规则合并到一起，在合并过程中使用一些其他特征进行验证。}

(2)基于人工神经网的方法。 由于人脸轮廓的复杂性，人脸很难用数学模型表示，而神经网是一种可以表示复杂模型的方法。 人工神经网的优势是方便建模和较高的鲁棒性。 但是人工神经网在运算速度上有比较大的欠缺，这是制约它发展的一个因素。

(3)基于启发式模型的方法。 {53%：这种方法主要是通过抽取图像的若干特征进行人脸检测。} {48%：首先对人脸局部的特征进行判断，再对人脸整体布局进行判断。} 也就是从人的五官的检测到人的五官的相对位置的检测。 {90%：使用人脸五官分布特征的知识模型进行检测。} 由于抽取的特征较少，所以这种方法有比较快的检测速度。 {41%：这种方法的主要障碍也是由于抽取特征较少，在复杂环境下的人脸检测成功率较低[4]。}

{67%：1.2 基于Android平台人脸识别系统研究目的及意义}

{63%：人脸识别在生活中有广泛的应用场景。} 比较常见的有安检，监控，身份认证等。 {44%：随着人脸识别技术渐渐成熟，人脸识别会更多的出现在我们生活中的各个角落。} 人脸识别的优势是新颖的交互方法和人脸的不变性，劣势是复杂的环境下识别率低。 研究的目的便是实现人脸识别，并且提高人脸识别的准确率，并且将其运用到实际生活中[5]。

随着支付宝刷脸支付， FaceU检测人脸实时添加表情，腾讯QQ人脸登录等人脸识别技术在移动端的应用， {57%：人脸识别技术在移动端的应用也是越来越广泛。} {54%：学习人脸识别技术在移动端的应用具有重要的意义。}

1.3 基于Android平台人脸识别系统研究内容及目标

{43%：本文主要的研究内容是人脸检测与人脸识别在安卓终端的实现，其中主要内容有：}

(1)人脸检测算法，涉及到Haar分类器的研究。

{55%：(2)人脸识别算法，涉及到LBP人脸识别算法与样本训练。}

(3)学习使用基于opencv的JavaCV库。

(4)学习安卓客户端开发技术，。

项目的研究目标有：

(1)实现80%的人脸检测成功率。

(2)实现在非复杂背景下50%的人脸识别成功率。

{45%：(3)实现安卓客户端录入人脸样本并且完成训练。}

1.4 论文组织结构

{68%：本文一共分为7章，各章节具体介绍如下：}

第一章： 绪论。 {44%：主要介绍了人脸识别的发展前景与历史以及当前人脸识别主要使用的技术，讲述了研究人脸识别的目的和意义，最后列出了研究的具体内容与目标。}

第二章： 相关技术及开发工具简介。 简单介绍了人脸检测技术，人脸识别技术的定义，介绍了Android集成开发环境Android Studio，Java编程语言以及JavaCV图形图像库。

第三章： {51%：对基于安卓端的人脸识别系统进行了需求分析，分别进行了人脸检测模块分析，样本获取模块分析和人脸识别模块分析。}

第四章： 对算法进行了详细分析。 {51%：具体分析了Haar分类器算法和LBP人脸识别算法。}

第五章： 系统实现。 首先介绍了项目的整体结构，然后具体实现了每个子系统，并且最终将每个子系统联系到了一起。

第六章： 关键问题及解决方法。 {44%：主要讲述了开发过程中遇到的问题和相应的解决方法。}

第七章： 总结与展望。 {42%：对基于Android的人脸识别系统进行了总结，并且对系统可以提升和拓展的地方进行了展望。}

{67%：第2章 相关技术及开发工具简介}

2.1 人脸检测技术

{48%：人脸检测技术是指从图像或视频流中通过一定策略检测到人脸的位置，大小等信息的技术。} 起初人脸检测技术并不被关注，直到研究者们发现在复杂环境下人脸获取极其困难，只有解决了人脸在复杂环境下的检测，才能顺利地进行人脸识别。 {46%：比较常见的人脸检测方法有参考模型法、人脸规则法、样品学习法、肤色模型法、特征子脸法。} {43%：人脸检测的难点在于人脸内在的变化和人脸外部环境的变化。} {60%：本文使用了基于Haar分类器的人脸检测方法进行人脸检测。} {44%：Haar分类器在人脸识别中使用，实际上就是对样本进行人脸和非人脸的分类。} {43%：Haar分类器使用haar-like特征使人脸量化，通过Adaboost算法来进行机器学习，通过弱分类器的迭代可以快速准确地对非人脸进行过滤。} 通过摄像头可以获取到视频流的一帧图像。{48%：使用Haar分类器对这一帧图像进行分类，判断图像中是否有人脸存在，如果存在，则可以找到人脸的位置和大小信息，从而实现人脸的检测。}

2.2 人脸识别技术

{62%：人脸识别技术通常与人脸检测技术捆绑在一起。} 人脸识别技术一般分为3个过程： (1)通过一定方法获取人面信息，生成面纹并且存储到面纹库。 (2)通过摄像机或相机获取当前人脸信息，生成当前面纹。 (3)将获取到的当前面纹与面纹库中的面纹通过一定方法进行对比，找到最相近的面纹，从而获取到识别出的对象。 现在主要的人脸识别方法有： {85%：几何特征的人脸识别方法、基于特征脸的识别方法、神经网络的人脸识

别方法、弹性图匹配的｝｛54％：人脸识别方法、线段 Hausdorff距离的人脸识别方法、支持向量机(SVM)的人脸识别方法。｝｛85％：人脸识别技术具有非接触，非强制性，并发性等优点。｝ 人脸识别技术的缺点是人脸所在背景对识别的准确度影响较大，人脸本身的表情和穿戴的装饰对人脸识别的结果也影响较大。

本文使用了LBP算法进行人脸识别。 ｛41％：LBP算法是一种纹理算法，在人脸识别的领域，LBP算法会将人脸分成很多小块，并且进行局部的描述，最终形成一个整体的描述。｝｛42％：通过与样本库的比较，判断当前图像是否匹配到了样本库的人脸，从而实现了人脸识别。｝

2.3 开发工具

（1）Android Studio

Android Studio是谷歌推出的一个集成开发环境，提供了Android开发所需要的开发和调试工具。 Android Studio使用Gradle进行项目管理，它是在Intellij IDEA的基础上进行开发的。 ｛91％：Gradle是一种高级的构建工具，用于管理依赖性，允许自定义构建逻辑。｝｛51％：构建系统同时支持本地文件系统和远程存储库支持的依赖，这样就不用把依赖库下载到本地了。｝ Android Studio分成三个模块。 ｛59％：Java库模块，Android库模块，Android应用程序模块。｝ Android Studio 2.0提供了运行时修改代码，并且极大的提高了虚拟机的运行速度。使用Android Studio可以方便的进行Android开发。 相比于之前流行的Eclipse，Android Studio在代码提示，界面，功能上显得更加优秀。 Android Studio唯一的缺点是对JNI编程不够友好。 使用JavaCV可以避免JNI编程，完美解决了问题。

2.4 开发技术

（1）Java语言介绍

Android开发主要使用的Java语言。 Java语言是由Sun Micrososystems公司与1995年推出的。 ｛95％：Java是一种简单的、面向对象的、分布式的、解释型的、健壮安全的、结构中立的、可移植的、性能优异、多线程的动态语言。｝｛91％：Java语言的这些优良特性使得Java应用具有无比的健壮性和可靠性。｝｛54％：Java是编译性语言和解释型语言的合集。｝ 一个java文件首先被编译成class文件，然后再被解释成0和1组成的二进制指令并被执行。使用Java语言需要安装JDK和JRE。 JDK是Java开发包，是一个开发工具的合集。 JRE是Java的运行环境包含了JVM的标准实现及Java核心类库。

（2）JavaCV库介绍

JavaCV库是一个开源库，它是基于OpenCV的Java封装。 ｛78％：OpenCV是一个跨平台的计算机视觉库。｝在Windows，Linux和Mac OS上都可以使用。 OpenCV使用C++编写，它主要的接口是面向C++语言的。 但它也给其他的语言提供了接口，比如说: Python，Ruby，MATLAB。 ｛72％：实现了图片处理和计算机视觉方面的很多算法[8]。｝ 相较于其他计算机视觉库，例如： Libvideogfx、mVision，OpenCV的优点很突出。 OpenCV的开源性保证我们可以免费、放心的使用而不用担心法律问题，并且保证了代码的健壮性和高效性； OpenCV的跨平台性使得我们可以在Android平台上顺利的使用； ｛45％：OpenCV内置的分类器和人脸识别算法可以使我们方便的集成人脸识别功能；｝ OpenCV还有丰富的示例程序，可以降低学习成本。

第3章 人脸识别需求分析

3.1 功能需求

人脸识别系统划分为4个模块。 第一个是人脸样本获取模块，第二个是当前图像获取模块，第三个是人脸检测模块，第四个是人脸识别模块。 这4个模块相对独立，但是需要为彼此提供接口，最终实现从图像获取到人脸识别的功能。

3.1.1 人脸样本获取模块

人脸样本包含的要素有两个，第一个是人脸图片，第二个是人的姓名。 在存储时，需要把这两个要素结合起来。 一个人的人脸信息需要多张人脸图进行训练得到，因此在取样时需要同一个人拍摄多张人脸图像。

因为拍摄时人脸所处的环境可能比较复杂，所以需要在拍摄完成后对图片进行裁剪。 裁剪时需要统一图片的尺寸，质量，图片格式。

需要有给用户输入姓名的编辑框，在用户输入姓名后开始拍摄图片，再对拍摄好的图片进行裁剪。

3.1.2 当前单帧图像获取模块

当前图像是从视频流中获取的。 在Android设备上的表现是： 通过前置或后置摄像头获取视频流，再从视频流中获取单帧图像。

需要注意的是屏幕的方向以及最终单帧图像的数据结构。 应当与人脸检测模块的输入一致。

需要实现实时预览摄像头获取到的画面。

3.1.3 人脸检测功能需求

{43%： 人脸检测模块接收获取到的视频单帧图像，然后对图像中的人脸进行检测，最终将检测到的人脸位置大小显示在预览画面中。} {50%： 检测结果要求显示清晰，明了，一眼可以看出检测到的人脸位置和大小。}

每一帧要进行检测，如果检测到人脸及实时绘制出人脸所在位置和大小，如果未检测到人脸则不绘制任何东西。

{41%： 人脸检测的结果需要作为人脸识别模块的输入，因此要统一人脸数据的数据结构。}

3.1.4 人脸识别功能需求

{41%： 当人脸检测检测到人脸的时候，将人脸数据传递给人脸识别模块，人脸识别模块需要通过一定的算法与本地人脸面纹进行比对，得到人脸信息。} 如果匹配到了结果，则在预览页面显示出匹配到的人的姓名，如果未匹配到，则显示未匹配到身份。

3.2 非功能需求

{41%： 为了提高用户体验，人脸识别系统也需要有以下的非功能需求：}

（1）可用性： {41%：样本收集，人脸识别的过程应当减少用户的操作，通过尽量少的步骤完成整个人脸识别过程。} 检测和识别结果应当简单明了的显示出来。

（2）可靠性： {40%：人脸检测和人脸识别的成功率应当比较高，可以适应复杂环境下的人脸识别需求。}

（3）健壮性： 可以适配不同安卓版本，不同机型的终端，对异常有优秀的处理。

（4）可维护性： 人脸识别系统的每一个模块应当相对独立，减少相对之间的依赖，提取公共函数。 {50%：保证可以方便的更换人脸检测和人脸识别算法，可以方便的修改系统的各个部分。}

第4章 算法分析

4.1 Haar分类器

{61%：目前主要有两类人脸检测方法：} 基于知识和基于统计。 {53%：基于知识的人脸检测方法是利用先验知识，根据人脸器官的特征，以及他们之间的关系来检测人脸的。} {49%：基于统计的人脸检测是把人脸当成一个整体的模式，通过大量的人脸样本，使用统计的方法来构造人脸模式空间，利用相似度来判断是否存在人脸。} 目前很多方法是把基于知识和基于统计结合在一起使用的。

Haar分类器包含了Adaboot算法。 {76%：分类器是数据挖掘的一种重要概念。} {42%：分类器把有限数据分成若干个类，任何一个新的数据都可以映射到某一个类中，从而可以对数据进行预测。} {54%：比如在这里，分类器将所有的样本分成两类，第一类是人脸，第二类是非人脸。}

{44%：Haar分类器实际上是Boosting算法的一个应用，Haar分类器用到了Boosting算法中的Adaboot算法。} {61%：Haar分类器将AdaBoost算法训练出的强分类器进行了联级。} {65%：Haar分类器是由Haar-Like特征、积分图方法、AdaBoost组成的。}

4.1.1 Haar-Like特征

{68%：Haar-like特征在最早的时候是用于人脸表示的，由Papageorgiou等提出。} Haar特征分为3类： {90%：边缘特征、线性特征、中心特征和对角线特征，共同组合成特征模板。} {68%：在这个特征模板中分成了两种颜色的矩形，白色和黑色，定义模板的特征值为白色矩形像素减去黑色矩形像素和。} Haar特征值反映的是图像在灰度上的变化。 例如其在人脸上的表现为： 眼睛比鼻子的颜色深； 嘴巴颜色比周围颜色深。 {47%：由于特征是矩形的，所以其只能描述特定走向的边缘或线段结构。}

图4.1特征模板

{82%：对于图中A，B和D这几种特征，特征数值计算公式是：} v=白色的和-黑色的和。 但是对于C来说，计算公式为： v=白色的和-2*黑色的和。 {67%：黑色区域乘二是为了使两种矩形区域中的像素数量相同。}

{56%：矩形特征可以在图像的任意位置，可以设置任意大小，可以选定不同的矩形模板类别。} 因此，在

极小的检测窗口也可能会有大量的矩形特征。

4.1.2 AdaBoost

{64%： AdaBoost实际上是一种具有一般性的用来提升分类器的算法它并不局限使用某一特定的分类器。} AdaBoost可以帮助我们更好地选择矩阵特征的组合，矩阵特征组合将以二叉决策树的形式存储起来。

{79%： AdaBoost的核心思想是针对同一个训练集训练不同的若干个弱分类器，最终将这些弱分类器组合成一个强分类器。}

Adaboost算法是通过迭代实现的，其中重要的一步是更改数据的分布。 {43%： 根据每次训练集中样本是否正确来给样本设置权值，然后把划分权值的分类器传递给下层分类器进行训练，} {78%： 将每次训练得到的弱分类器融合，得到最终的强分类器。} 具体过程如下：

{57%： （1）对N个训练样本进行学习，得到第一个弱分类器。}

{40%： （2）使用测试样本进行测试，将分错的样本和其他的新样本构成一个新的样本数为N的训练样本，学习得到第二个弱分类器。}

{44%： （3）把（1）和（2）测试出来分错的样本加上若干新的样本组成N个训练样本，学习得到第三个弱分类器。}

{53%： （4）反复进行上面的步骤，学习得到若干个弱分类器，最终组合成一个强分类器。}

{44%： AdaBoost算法对分错的样本进行了加权，使得训练的焦点放在了比较难区分的训练样本上。} {60%： 该算法还对弱分类器进行了加权处理，使得分类效果较好的分类器具有比较高的权重，分类效果差的分类器具有较小的权重。} [9]

图4.2权重处理

4.1.3 积分图

{92%： 积分图是只遍历一次图像就可以求出图像中所有区域像素和的快速算法，大大提升了图像特征值计算的效率。} {44%： 在进行 haar-like分类器训练和检测的过程中，每当遇到图片样本的时候都会需要计算某一个窗口的特征值，} 这样的计算是非常大的，而积分图就可以解决 haar-like计算量过大的问题。 {91%： 积分图是一种可以描述全局信息的矩阵表示方法。} {93%： 积分图的构造方法是位置（i，j）处的值ii（i，j）是原图像（i，j）左上角方向所有像素的和：}

积分图的构建算法如下：

（1）用s（i，j）表示行方向的累加和，初始化s(i，-1)=0;

（2）用ii(i，j)表示一个积分图像，初始化ii(-1，i)=0；

{98%：（3）逐行扫描图像，递归计算每个像素(i， j)行方向的累加和s(i， j)和积分图像ii(i，j)的值}

（4）s(i，j)=s(i，j-1)+f(i， j)

（5）ii(i，j)=ii(i-1，j)+s(i，j)

{94%：（6）扫描图像一遍，当到达图像右下角像素时，积分图像ii就构造好了。}

{91%：（7）积分图构造好之后，图像中任何矩阵区域的像素累加和都可以通过简单运算得到如图所示。}

（8）设D的四个顶点分别为α、β、γ、δ，则D的像素和可以表示为

（9）Dsum＝ii(α)+ii(β)-(ii(γ)+ii(δ));

4.2 LBP算法

LBP算法，Local Binary Patterns，局部二值模式。 是一种使用局部特征作为判别依据的识别算法。 使用LBP时，图像必须为灰度图。 {47%：每张灰度图都是由一个一个像素点组成，每个像素点都有自己的灰度。}

LBP算法最初是定义在一个3×3的一个邻域内的，以中间的点作为阈值， {54%：将邻近的8个像素的灰度值与中间这个像素的灰度值进行对比，如果大于中心点的灰度值，} 则记为1，如果小于中心点的灰度值，则记为0。 这样在每个点对比过后，可以得到一个8位的2进制数，即这个邻域的LBP值。 这个值表现的是这个邻域的纹理信息[12]。

图4.3 LBP算法示意图

正式公式为：

其中 表示中间的那个点。 {54%：ic表示中间点的灰度值，ip表示邻域的点的灰度值。} S（x）是符号函数，定义如下：

最初的LBP算法存在着缺陷，它无法应对不同尺寸和不同频率纹理以及旋转的情况。 因此Ojala对LBP算法进行了改进。 他将3乘3的邻域改成了任意大小的邻域，并且并且由原来的正方形改成了圆形。 类似于下图：

假设圆的半径为R，其中有P个采样点，其中每个采样点的值可以用下面的公式表示：

其中 为中心点， 为某个采样点，通过上式可以计算出每个采样点的坐标值，但是坐标并不一定是整数值，可以通过双线性插值计算得到采样点的像素：

在获取到人脸的LBP特征后还需要进行特征匹配，例如下面这张人脸图像，将其划分

为7×7的子区域并计算得到每个邻域的LBP值并且统计其直方图。 {41%：这样做可以避免人脸没有完全对准的情况，也对LBP特征进行了降维处理。} 在得到直方图后，有多种方法可以判别其相似性。 可以使用：

（1）直方图交叉核算法

（2）卡方统计方法

其中Mi为已知人脸直方图，Si为待匹配人脸直方图。

第5章 系统实现

5.1 系统总体结构

系统总体结构如图5.1

图5-1 总体结构图

本系统使用Android Studio作为集成开发环境。 Android Studio使用Gradle对项目进行构建。

Project名字是FaceRecognizer，它代表了整个工作区。 faceRecognizer是Project中的一个Module。

manifests下的AndroidManifest.xml文件是Android应用的配置文件。

java文件夹下放置的是module下所有的java文件。

jniLibs文件夹下放置的是不同平台下的动态链接库。

{44%：res文件夹下放置的是各种资源文件，比如图片文件，布局文件等。}

resources存放的是本地文件，在项目中存放的是已经训练好的分类器文件。

Gradle Scripts存放的是Gradle脚本，如下图所示：

图5-2 gradle脚本

其中build.gradle表示Project或者Module的构建脚本。 Proguard-rules.pro表示混淆规则。 gradle.properties表示gradle特性，可以对构建过程进行设置。 setting.gradle可以对Gradle的Module进行管理。 可以通过local.properties对SDK或NDK进行管理。

5.2 视频预览实现

视频预览的实现涉及到Android的SurfaceView类和SurfaceView类。

SurfaceView继承于View，可以直接从内存或者DMA等硬件接口取得图像数据。 它的特点是： 通常在Android中，UI的绘制只能在主线程中进行，但是在绘图频繁进行的情况下，主线程会被阻塞，而SurfaceView可以实现在主线程以外的线程进行图片的绘制，从而提高了程序的反应速度。 在游戏开发中经常会使用到SurfaceView，可以用于摄像头预览。 使用SurfaceView需要继承SurfaceView，并且实现SurfaceView的Callback接口

，如下图：

在构造函数中首先为SurfaceView的Holder添加一个实现了Callback的回调。 这个Callback有3个回调函数：

在这3个回调函数里对Camera进行操作。

{46％： Camera类是Android中对摄像头进行操作的一个类。} 可以通过Camera.open()来获取一个Camera实例，然后通过setPreviewDisplay方法来为Camera设置一个SurfaceView对象作为预览容器。 {49％： 可以通过Camera.Parameters来对Camera的参数进行一些设置，比如预览的大小和摄像头预览的方向。} 通过为camera设置一个PreviewCallback，来获取每帧图像的回调。 关键代码如下：

至此，已经完成了预览的流程。 实现效果如下：

图5-3视频预览画面

5.3 人脸检测实现

在上一节中提到需要为Camera对象设置一个PreviewCallback回调。 创建了一个继承于View的FaceView来负责人脸信息的绘制。 FaceView实现了Camera的PreviewCallback，重写了onPreviewFrame方法。 OnPreviewFrame接收摄像头获取到的图像数据，每当获取到一帧，就回调一次。

{60％： FaceRecognizer类负责人脸检测和人脸识别。} 在FaceView的构造函数中，对FaceRecognizer进行了初始化。 {45％： 人脸检测的初始化是加载分类器，分类器保存在resources文件夹中。} {42％： 由于加载分类器是一个耗时操作，因此要放到一个新的线程中使用。} {46％： 主要使用JavaCV的CVHaarClassifierCascade类，在初始化时只需要传入分类器文件的路径。} 关键代码如下：

由于加载分类器是一个耗时的操作，因此会新开一个线程进行加载，在完成加载后调用主线程创建的Handler来发送一个消息给主线程， 表明加载完成，然后进行相应的 UI更新。

而当FaceView接收到预览图像一帧的回调时，将获取到的图像数据传递给FaceRecognizer对象。 关键代码如下：

在processImage方法中会先将获取到的图像转换成OpenCV中图片的数据结构，即IplImage。 直接调用IplImage的create方法即可。 在将其转换成IplImage之后再将其转换成灰度图。 具体实现代码如下：

进行人脸识别的方法为cvHaarDetectObject方法，传入待检测的图片，分类器等参数，最终可以获得识别到的人脸信息。 OpenCV中使用CVSeq数据结构进行存储。 使用CVSeq数组faces存储检测到的人脸。

在完成这个过程后，由于在onPreviewFrame中调用了postInvalidate方法进行view的重绘，因此会重新调用faceview的onDraw方法。 在onDraw方法中会从FaceRecognizer中取出检测到的人脸，根据人脸的信息来在屏幕上绘制矩阵。 只需要调用canvas的drawRect方法就可以方便的进行矩阵绘制。 {58％： 根据检测到的人脸数量绘制相应数量的人脸矩阵。} 实现的关键代码如下：

至此，人脸检测的过程就已经完成了。

5.4 人脸检测实现

{42%：人脸识别之前先要对所有的样本图片进行训练，在此之前要先进行样本的采集。} 首先在人脸采集页面添加EditText，使用户可以输入姓名。 {60%：在姓名输入完成后，点击保存按钮会跳转拍照页面。} 这里利用了Android的Intent机制，通过给Intent设置Action： ACTION_IMAGE_CAPTURE，可以调用系统的拍摄页面，只需要在onActivityResult中处理拍摄完成的结果即可。 关键代码如下：

在获取到人脸图像后，还需要对图像进行裁剪，来使得人脸占图像比例尽量大。 裁剪也是利用Intent机制，给Intent设置Action： com.android.camera.action.CROP。 并且通过Intent的putExtra方法来添加裁剪参数。

最后需要将拍摄的图片保存到本地。 使用Android的File类可以完成。 首先通过 File的 createTempFile方法创建一个图片文件，文件名使用姓名加上当前的时间戳，再将这个图片文件传入跳转裁剪页面的 Intent对象中，Android系统会在完成裁剪后将裁剪好的图像根据传入的图片文件保存到本地。

5.4人脸识别流程

人脸识别部分使用的OpenCV的FaceRecognizer类。 {41%：在完成人脸检测部分的初始化后会进行FaceRecognizer对象的初始化。} {44%：由于要对所有的人脸样本进行训练，所以首先要获取所有人脸样本。} 通过Android的文件系统，首先找到人脸样本的目录，通过File对象的listFiles可以获得所有人脸样本。 在获取到人脸样本后先要将其转换成灰度图，方法与人脸检测部分图片转换成灰度图相同。 通过createLBPFaceRecognizer方法可以新建一个FaceRecognizer对象。 训练图片只需要调用FaceRecognizer的train方法。 train方法要求传入MatVector对象和int数组。 也就是OpenCV图片向量与其对应的标签。 通过一个循环即可完成人脸样本到MatVector对象与对应标签的转换。 关键代码如下：

{43%：在人脸检测完成之后，如果检测到人脸就会调用predictFace方法来进行人脸识别，并且传入待检测的人脸。} 人脸识别时会调用FaceRecognizer的predict方法，这个方法会在识别完成后返回一个label，表示识别出的人脸在人脸数组中的位置。 通过遍历即可获得识别出的人的姓名。

在Faceview中，FaceRecognizer的ProcessImage方法回返回识别到的人的姓名。 在FaceView的onDraw方法中会判断是否有识别到的人脸，如果存在，则调用canvas的drawText方法把姓名显示到屏幕上。 效果图如下：

图5-4 人脸识别效果图

第6章 关键问题及解决方法

6.1 摄像头预览部分

在进行摄像头预览时在部分机型上未响应。 测试过后发现在部分Android6.0以上手机上未获取到摄像头权限。 Android M采用了不同于其他版本的动态权限管理，部分权限不会在安装时就获得，需要动态申请。 如果未获取到目标权限，相关操作不会有任何返回值。 需要将Target SDK设置为23或23以上，然后在代码判断是否获取到了权限，然后动态请求获取权限。

在进行摄像头预览时程序奔溃，提示setParameters failed。 在进行摄像头预览时需要设置预览时的宽度和高度

。 这时需要先获取摄像头支持的宽度和高度，然后与用于预览的 SurfaceView的宽度和高度进行比较，当宽高比小于阈值并且高度与某一支持的高度最相近的时候， 就采用这个宽高作为预览时 Camera的输出宽高。 如果设置了Camera不支持的宽高，则程序运行时就会报错。

6.2 人脸识别部分

从主页面跳转到视频预览页面存在长时间的卡顿。 在初始化 FaceRecognizer的时候从本地读取分类器文件初始化 Haar分类器，从本地读取人脸样本并进行训练来初始化 OpenCV的 FaceRecognizer类， {41％：这个过程很耗时，并且在主线程中完成，导致主线程阻塞，无法更新 UI。} Android中耗时任务一般不在主线程中执行，以防止主线程被阻塞无法更新UI，因此，需要把耗时任务放到子线程中执行。 首先新建一个线程，把初始化任务放到新的线程中执行，在完成初始化工作后再通过主线程创建的 Handler来告诉主线程初始化工作完成， 进行相应的UI更新。

图像数据位深度不一致导致程序报错。 在OpenCV中存在多种图像位深度，使用分类器时需要相同位深度的图像深度。 通常从摄像头获取到的图像位深度都是8U的，因此，在进行cvCreateImage操作时统一使用8U这个深度。

第7章 总结与展望

随着移动互联网的发展和人们对新颖交互方式日趋强烈的需求，人脸识别在移动客户端上的应用愈来愈广泛。{45％：本文利用Android开发技术与OpenCV图形图像库完成了在Android设备上的人脸识别功能。} 现在对本文的工作进行一个总结：

本文首先对人脸识别的现状，发展历程以及生活场景中的应用进行了概括，然后总结对需要实现的系统进行了分析。 随后决定在Android客户端实现系统。 {43％：使用Haar分类器算法实现人脸检测部分，使用LBP算法实现人脸识别部分。} 并且使用JavaCV库来实现这两个算法。 {53％：最后对人脸识别系统具体的开发流程进行了介绍。}

由于时间和精力的以及研究深度上的限制，无法对算法进行深层次的探索，只是理解了算法的基本原理和库实现的方法。 在实现Android人脸识别系统后还有很大的改进空间：

1.没有使用运动跟踪算法。 系统对视频流中的每一帧进行人脸识别，从而实现了在视频中实时的人脸识别，但是这样效率较低，使用运动跟踪算法可以提高系统的效率。

2.人脸样本的获取方法局限于使用摄像头获取，样本数量较少，应当可以通过其他方式导入人脸样本，从而提高系统的易用性和识别成功率。

{41％：3.人脸识别成功率较低，直接使用 OpenCV中的人脸识别算法人脸识别识别错误率较高， } {52％：对获取到的人脸图片进行预处理，以提高识别成功率。}

4.可以对人脸样本库的人脸图片进行预览，从而方便用户进行比对，提高用户体验。

总而言之，人脸识别在Android端的应用会越来越多，为了提升用户体验，需要研究如何提高人脸识别的速度与成功率，以及如何简便的获取人脸样本。 只有沉下心来对算法进行深入的研究和实践，才能提高算法的效率。

这些都是我今后要去完成的。

参考文献

[1]严严，章毓晋，基于视频的人脸研究进展，北京： 清华大学信息科学与技术国家实验室，2009.5.

[2]王智飞，苗振江，低分辨率人脸识别算法研究，北京： 北京交通大学，2013.

[3]杨利平，辜小花，重庆： 重庆大学光电技术及系统教育部重点实验室，重庆： 重庆科技学院，2014.

[4]吴巾一，周德龙，人脸识别方法综述，浙江： 浙江工业大学计算机科学与技术学院，2009.9.

[5]王华君，李荣，徐燕华，孟德建，基于局部相位纹理表示的光照变化人脸识别算法，无锡： 无锡太湖学院工学院，2015.

[6]范哲意，曾亚军，蒋姣，翁澍沁，刘志文，视频人脸识别实验平台的设计与实现，北京： 北京理工大学信息与电子学院，2016.

[7]杨恢先，翟云龙，蔡勇勇，奉俊鹏，李球球，纹理与边缘相结合的单样本人脸识别，湘潭： 湘潭大学，2014.10.

[8]张俊，李鑫，赵莎莎，邓硕辰，基于嵌入式平台和OpenCV的人脸识吧系统设计，太原： 太原理工大学信息工程学院，2016.4.

[9]袁华，钟欢虹，欧阳宁，莫建文，基于分块LBP和鲁棒核编码的人脸识别，桂林： 桂林科技大学信息与通信学院，2016.2.

[10]张三友，王磊，基于Android的人脸识别系统设计与实现，苏州： 苏州市吴江区公安局科技信息化大队；公安部第三研究所物联网中心，2016.4

[11]耿冬冬，胡友彬，万友，欧静华，基于视频流的人脸检测与跟踪关键技术研究，解放军理工大学气象海洋学院，2016.3.

[12]盖健，刘小华，李锐杰，宁尚军，基于LBP的图像集人脸识别算法，吉林： 吉林大学计算机科学与技术学院，2016.4.

[13] Minyoung Kim Dept. of Computer Science, Rutgers University, Piscataway, NJ 08854, USA, 2008.

[14] Shang-Hung Lin, Ph.D. IC Media Corporation. An Introduction to Face Recognition Technology, IC Media Corporation, 2000,

[15] J. J. Weng Dept. of Comput. Sci., Michigan State Univ., East Lansing, MI, USA , Toward automation of learning: the state self-organization problem for a face recognizer, 1998.

致谢

{42%：光阴似箭，四年大学时光即将结束，首先我要向母校武汉理工大学致以真诚的感谢。} 我在这里度过了美好又充实的4年。 {46%：论文的完成，离不开我的指导老师岑丽副教授的指点，从论文的选题，撰写，} 到完成，从系统的架构，实现，调试与测试岑老师都对我进行了精心的指导， {46%：提出了宝贵的意见，能够顺利完成毕业设计，我非常感谢岑丽老师。} 除此之外也要感谢身边的同学和老师对我的帮助以及鼓励，感谢他们的陪伴。

经过这一学期的努力，我的论文终于成功完成。 论文完成过程中，遇到问题时，岑老师都会抽空为我解答。{42%：岑老师学识渊博，治学态度严谨，工作作风优良，为人谦逊，非常值得我学习。} 此外，岑丽老师对我的生活也很关心，在生活上也为我提供了帮助，很平易近人。 {46%：能成为岑老师的一名学生是很幸运的一件事。}

{58%：我也要感谢计算机科学与技术学院的领导和老师们。} 作为软件工程的学生，我感受到了学院的老师们上课时认真的态度和解答问题的耐心，他们传授的知识是我能顺利完成论文的基础。 此外，学院丰富的教学资源能够帮助我更好的学习，接触最宝贵优秀的资源。 我还要感谢四年里一起学习生活的同学们，我们一起学习，一起生活，一起玩耍，他们在各方面都给我提供了帮助，一起讨论的过程中慢慢建立起了友谊，非常感谢他们。

{41%：我还要感谢我的父母，他们无私的奉献不求回报，为我提供他们能提供的最好的资源。} 在我心情低落的时候会鼓励我，在我遇到挫折的时候会给我温暖的拥抱，没有他们，就无法如此顺利地完成四年的学业。

即将毕业步入工作，只有努力工作，踏实学习才可以不辜负亲人，在社会中创造自己的价值，才能不辜负师长和朋友们对我的期望和谆谆教诲。