



汕 頭 大 學

SHANTOU UNIVERSITY

硕 士 学 位 论 文

题 目: 手机人脸识别开发系统构建及 PCA 和
SVM 算法的 App 实现

英文题目: Construction of mobile phone face recognition
development system and App implementation
of PCA and SVM algorithm

姓 名 王继彪 学 号 51509012

所在学院 工学院 导师姓名 姜永权

专 业 电子与通信工程

入学日期 2015 年 9 月 答辩日期 2018 年 6 月

学位论文原创性声明

本论文是我个人在导师指导下进行的工作研究及取得的研究成果。论文中除了特别加以标注和致谢的地方外，不包含其他人或其他机构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，均已在论文中以明确方式标明。本人完全意识到本声明的法律责任由本人承担。

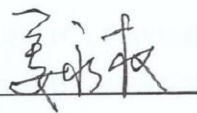
作者签名：_____ 日期：_____年____月____日

学位论文使用授权声明

本人授权汕头大学保存本学位论文的电子和纸质文档，允许论文被查阅和借阅；学校可将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其他复制手段保存和汇编论文；学校可以向国家有关部门或机构送交论文并授权其保存、借阅或上网公布本学位论文的全部或部分内容。对于保密的论文，按照保密的有关规定和程序处理。

作者签名：_____

导师签名：



日期：_____年____月____日

日期：2018年6月3日

摘 要

当今时代已是移动互联网泛在的时代，在国内移动终端（以下简称手机）的普及率已超过 96%。除了通话功能以外，手机上还可加载各种各样的应用程序，特别是一些与金融相关的应用程序，如支付宝、微信支付、银行手机 App 等对手机应用提出了更高的安全要求。传统的安全措施是采用账户密码进行保护，但每次都要输入账户密码使得应用不够方便。利用人体生物特征来代替账户密码，一方面使得应用更加方便，另一方面使得安全等级能够得到进一步提高。指纹识别技术已广泛应用于手机，近年来人脸识别技术也开始在一些高端机上得到了应用。

人脸识别技术是生物特征识别技术的一种，它能够提取人脸图像中所蕴涵的个体特征信息，并与已知的人脸特征进行比对，从而识别出使用者是否为合法用户。人脸识别技术具有用户信息采集方便、用户使用体验好、不易被伪造和盗用等特点。目前，手机人脸识别技术已成为生物特征识别领域的一个研究热点。结合专业硕士更加贴近工程应用的培养特点，本文构建了一种手机人脸识别的开发系统，给出了手机人脸识别技术的开发步骤和实现方法，并基于主成分分析法（PCA）和支持向量机（SVM）算法，研发了一种手机人脸识别 App 应用程序。本文构建的手机人脸识别开发系统是一个开放系统，能够将各种人脸识别算法移植到 App 应用程序中，为开发更加先进的 App 应用程序奠定了基础。构建的手机人脸识别开发系统包括以下三个模块：

1、App 前端：App 前端应用框架采用 MUI 框架开发，其功能模块是基于 H Builder 平台上开发的，界面模块是利用 Html5 语言编写代码的。

2、服务器端：首先利用 Eclipse 平台上配置 Tomcat 服务器，再使用 Ngrok 软件使得手机能够通过虚拟映射网址与服务器端进行通信。在服务器端开发 App 应用软件后再回传到手机端。

3、人脸识别算法：本文对 Eigenface、LBP 和 PCA 以及 PCA 和 SVM 的人脸识别算法进行了分析比较，选择了性能较优的 PCA 和 SVM 算法应用于研发的人脸识别 App 应用程序中。

本文利用构建的手机人脸识别开发系统，基于 PCA 和 SVM 算法开发了一种人脸识别 App 应用程序，并通过扩展的 ORL 人脸数据库、Yale 人脸数据库、IMM

人脸数据库、Face95 人脸数据库和自建的人脸数据库，分别从识别率和识别速度两项性能指标对 App 应用程序进行了测试，测试结果表明，该 App 应用程序能够有效判定出使用者是否为合法用户。

关键词： 手机人脸识别；开发系统；PCA；SVM；App 应用程序

Abstract

The current era is an era in which the mobile Internet is ubiquitous, and the penetration rate of domestic mobile terminals (hereinafter referred to as mobile phones) has exceeded 96%. Except for voice communications, various applications can be utilized in cellphone, especially some finance relevant applications, such as Alipay, WeChat Pay, Bank on cell, which require higher level of security. Traditional secure methods adopt account and passcode for protection, while entering account and passcode results in inconvenience of procedure. Employing human body biologic characteristic instead of account passcode can not only make the applications convenient, but also enhance the level of security. Fingerprint recognition technology has been widely used in the field of cellphone, and facial recognition technique has been applied in some high-end terminals.

Face recognition technique is one of the human body biologic characteristic technique, it can extract the human body biologic characteristic information for the facial images, and compare them with the known facial characteristic, which is able to identify whether the user is legal or not. Face recognition technology has the advantages of convenient user information collection, good user experience, and is not easy to be forged and embezzled. Nowadays, Face recognition on cellphone has become a hot point in the field of human body biologic characteristic recognition. Combined with professional master's training characteristics of being closer to engineering application, this paper constructs a Face recognition developing system on cellphone, and provides developing procedure as well as realization methods for Face recognition technique on cellphone. Moreover, a Face recognition application on cellphone has been developed, based on principle component (PCA) and supporting vector machine (SVM) algorithm. The Face recognition developing system on cellphone this paper built is an open system, which can transplant all kinds of Face recognition algorithm into applications, and therefor build up the foundation for advanced applications. The Face recognition developing system on cellphone we built consists of three modules:

1. App front end: the App front end architecture adopt MUI framework, whose functional modules are developed based on H Builder platform, and the interface modules are developed with Html5.
2. Server end: We first configure the Tomcat server with Eclipse platform, and then use Ngrok software to achieve the communications between the cellphone and server through virtual reflection website. The applications are developed on server end, and send back to mobile terminals.
3. Face recognition algorithm: this paper evaluates the difference among Eigenface algorithm, LBP and PCA algorithm as well as PCA and SVM algorithm, which have better performance, are chose for developing Face recognition applications.

This paper builds a Face recognition developing system on cellphone, and developing a Face recognition application based on PCA and SVM. We test the application with respect to two performance metrics, i.e. recognition rate and recognition speed, using the extended ORL face database, Yale face database, IMM face database, Face95 face database and self built face database. The results show that the application we built can determine whether user is legal or not efficiently.

Keywords: Face recognition of mobile phone; development system; PCA; SVM; Application

目 录

摘 要	I
Abstract.....	III
第一章 绪 论	1
1.1 人脸识别研究背景及意义	1
1.2 人脸识别技术国内外研究现状	2
1.3 人脸识别技术在移动终端的前景及意义	5
1.4 本文研究思路和主要内容安排	6
第二章 基于 Eigenface 和基于 PCA 和 SVM 的人脸识别算法介绍	9
2.1 数字图像处理基本知识	9
2.1.1 引言	9
2.1.2 灰度化及降采样	10
2.1.3 图像预处理基本操作	12
2.2 机器学习	14
2.2.1 最近邻算法	14
2.2.2 支持向量机(SVM)	15
2.2.3 主成分分析法 (PCA)	20
2.2.4 基于 Eigenface 的人脸识别算法介绍	21
2.2.5 基于 PCA 和 LBP 的人脸识别算法介绍	24
2.2.6 基于 PCA 和 SVM 的人脸识别算法介绍	26
2.3 本章小结	28
第三章 基于 PCA 和 SVM 算法的手机人脸识别系统设计与实现	29
3.1 概述	29
3.2 系统整体设计	29
3.3 App 前端设计	30
3.3.1 移动 Web 前端技术	30
3.3.2 Android 平台介绍	32
3.3.3 H Builder 开发平台介绍	33
3.3.4 基于 H Builder 平台的 App 前端设计与实现	34

3.4 服务器端设计	36
3.4.1 Tomcat 简介	36
3.4.2 Tomcat 的安装及在 Eclipse 环境下配置 Tomcat.....	37
3.4.3 HTTP 网络传输协议及在 App 人脸识别系统中的作用.....	38
3.4.4 Eclipse 平台介绍.....	39
3.4.5 Ngrok 软件使用介绍及其在 App 人脸识别系统中的作用	40
3.5 本章小结	41
第四章 基于 PCA 和 SVM 算法的手机人脸识别系统测试与分析	43
4.1 系统功能测试	43
4.1.1 实验设备	43
4.2.1 系统功能测试与分析	44
4.2 系统性能测试与分析	45
4.3 本章小结	47
第五章 总结与展望	48
5.1 总结	48
5.2 展望	49
参考文献	50
致 谢	53
个人简历	56

第一章 绪 论

1.1 人脸识别研究背景及意义

随着科技不断更新发展，人民的生活水平不断提高，我国政府部门也越来越重视人民的财产安全，2005 年，中共中央办公厅，国务院办公厅转发了《中共政治委员会，中央社会治安综合治理委员会关于深入开展平安建设的意见》。从 1996 年开始到目前为止，我国各大城市相继宣布建设平安城市，至此，平安城市建设的热潮在全国拉开帷幕^[1]。平安城市项目是旨在建设一个大型的、智能的、数字的、综合多功能的安全防控监管系统。同时它整合了城市中的所有监控资源，并连接了大部分社区和监管部门，因此它也具有综合治理、城市交通管理和消防指挥等职能^[2]。平安城市中重要的一个环节就是对城市中的流动人口进行监控，我们需要研究动态人脸检测与识别。人脸识别作为图像分析和理解中最成功的应用之一，近年来受到了广泛的关注，尤其是近几年来，经过将近 30 年的无数科研人员的不懈努力，人脸识别技术已经广泛应用到各领域，安防领域就是一个很好的例子。

我们首先介绍一下人脸识别技术的基本原理，人脸识别技术是通过用摄像机或摄像头等外部设备采集含有人脸的数字图像或者视频信息，然后经过图像处理等基本操作对图中或者视频中的人脸进行检测和跟踪，进而进行相应的脸部识别一系列技术。它也属于人工智能领域的一个分支，属于生物特征识别领域。可以

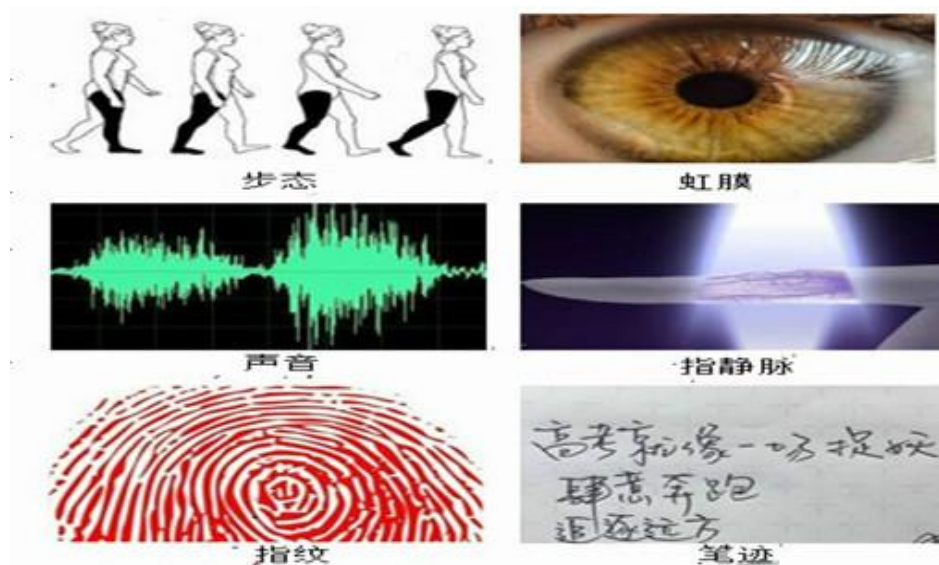


图 1-1 生物特征

用来识别的生物特征有很多，分两类。一类为生理特征，比如“虹膜”、“指纹”、“指静脉”等；另外一类为行为特征，比如“步态”、“声音”“笔迹”，如图 1-1 所示。人脸特征显然就是属于生理特征一类。

人脸识别技术无论是在理论研究上还是技术应用上都有着及其重要的意义。首先从理论研究上讲，人脸识别研究具有极高的学术价值，因为人脸由于外貌、表情、肤色、姿态、遮挡物这些变化因素，常常带给科研人员很大的挑战性。如果解决了这些变化因素，成功了突破了人脸识别技术的瓶颈，那将会给解决其他领域的问题带来很重要的提示和突破。

其次，从技术应用上讲，人脸识别技术是生物特征识别领域的一个相对比较成熟的应用，在其领域已经有很丰富的成果，人脸识别技术同时也是图像处理方向的一种分支应用，这种技术应用十分成功，该技术在很多方面都得到了充分的应用，尤其是在安全，身份识别认证等领域^[3]。人脸识别技术已经越来越火热，众多专家学者都投入到其中，并且它已经成为了科学研究中的一个重要的课题。在未来的时间里，人脸识别技术一定会在移动终端和各大网站、互联网上得到广泛应用，因为人脸识别技术给我们的生活带来了更多的安全性、便捷性。

1.2 人脸识别技术国内外研究现状

国内外科研人员开始从事人脸识别的研究最早可以追溯到 20 世纪 80 年代，经过近 30 年的几代科研人员的不懈努力，一点一滴的累积，此项技术已经趋近成熟，目前已经有相对成熟的商业应用，例如依图科技公司与阿里云合作搭建的“贵州公安交警大数据实时作战云平台”^[4]，北京商汤科技公司研制的一款用于控制人行通道闸的人脸识别设备，主要用于公共通道中的身份验证和智能化安全管理，可支持身份证、智能 IC 卡、二维码及人脸等多种验证方法^[5]，中国农业银行的超级柜台采用广州云从科技公司的人脸识别技术^[6]，北京旷视科技公司 Face ID 技术为洋钱罐提供智能数据服务、为小米金融实现快速放款，实时审批、为你我贷实现在线远程身份认证服务^[7]等等。

以下是国内外近三十年的人脸技术发展历程：

(a) 1976 年，Gerald J. Kaufman^[8]等人提出了一种能够从人脸轮廓中识别人脸的模式识别系统。它对每个轮廓进行预处理，去除噪声，平滑边缘，提取前缘。然后将处理后的轮廓用 12 维特征向量表示，其特征向量由循环自相关函数得到。

利用加权 k 近邻决策规则，在 110 类问题中，识别率可达 90%。还描述了一种用于设置权威文件的自适应训练过程。这个训练过程似乎是从一个信息内容的角度来看，或者最经常观察到的那些代表最重要的类的特征向量。最后，比较了循环自相关特征和不变矩特征的识别精度。

(b) 1991 年，Alan L. Yuille^[9]提出了一种从图像中提取面部特征并利用变形模板的概念来确定这些特征之间的空间组织的方法。参数化的几何模型在识别目标的时候可以度量与图像数据的匹配程度，参数的改变与识别对象的改变程度相关，同时参数也可以由概率模型指定。修改后的模板参数可以用于目标物体的描述和识别。

(c) 1993 年，程永清^[10]等人提出了基于投影图像的人脸特征提取，然后用距离分类器来分类的一种人脸识别方法。特征矢量对于姿态变化、面部表情变化、照片的模糊程度具有良好的稳定性和鲁棒性。同时由于特征矢量的稳定性也使得在分类的时候具有很高的识别率。

(d) 1999 年，S Eickeler^[11]等人提出了一种基于二维 DCT 特征和伪二维隐马尔可夫模型人脸识别系统，此系统在 ORL 人脸库的识别率为 99.5%。目前此系统的识别率远远超过目前的伪二维隐马尔可夫模型方法的识别率。同时 S Eickeler 还将论文中提出的方法与当前现有的方法做了对比分析，发现此方法在 ORL 人脸数据库上的识别率均要比现有的方法在 ORL 人脸数据库上的识别率要更好。

(e) 2002 年，MS Bartlett^[12]提出了一种基于独立分析法 Independent component analysis (ICA) 的人脸识别方法。独立分量分析法(ICA)是 Principal component analysis (PCA) 的一种推广。ICA 在 FERET 数据库上进行人脸图像识别时，有两种不同的体系结构，一种是把图像作为随机变量，像素作为输出，另一种方法是把像素作为随机变量，图像作为输出。第一种结构为人脸寻找空间局部基图像。第二种体系结构产生脸部特征码。两种 ICA 表示方法均优于基于 PCA 的人脸识别表示方法。结合这两种 ICA 方法可以有更好的分类效果。

(f) 2003 年，上海交通大学图像处理与模式识别实验室的 Zhang Yan Hun^[13]等人提出了基于一种基于支持向量机和最近邻分类器的人脸识别方法。支持向量机 (SVM) 作为一种新的模式识别方法，在人脸检测和人脸识别方面取得了良好的效果。该方法采用主成分分析 (PCA) 进行减少维数和提取特征，用 ORL 人脸数据库做训练集训练 SVM 分类器。该方法优于目前特征脸及其他方法。

(g) 2006 年 T Ahonen^[14]等人提出了一种基于局部二值模式 (LBP) 纹理特征的人脸图像高效表示方法。将人脸图像分割为若干区域, 从中提取 LBP 特征, 并将其拼接组合成增强的特征向量, 作为人脸描述符使用。T Ahonen 在对于不同复杂背景下的人脸识别问题中测试了该方法的性能, 测试结果显示该方法的识别效果较为理想, 识别率达到了 97%。

(h) 2007 年, Helin Dutağacı^[15]等人提出了一种利用掩模作为减轻表情失真的三维人脸的识别方法。Helin Dutağacı 等人使用各种不同的掩模进行试验, 例如椭圆二元掩模、高斯、超高斯和升余弦掩模等。并对掩模尺寸、中心、支撑区、尾翼衰减率等设计问题进行了研究, 接着对其识别性能进行了调整。

(i) 2010 年, Ying Ai Ju^[16]等人提出了一种基于梯度和相关局部统计量的人脸识别方法。大多数人脸识别方法通常使用原始图像本身作为特征向量。然而, 直接从原始图像中形成的特征向量似乎容易受到光照和面部表情的变化, 使用 BDIP (block difference of inverse probabilities) 作为局部统计梯度和使用两种 BVLC ((block variation of local correlation coefficients) 作为局部统计相关性, 当测试集照片开始测试时, 对测试集照片提取这三种特征向量, 然后融合这三种特征向量, 最后通过白化 PCA 和求余弦距离对测试图像分类。Ying Ai Ju 分别在 Yale 、Yale B、Weizmann 三种数据库中进行实验对比, 通过实验表明: BDIP 和两种 BVLC 特征向量融合后对光照和面部表情的变化具有较强的鲁棒性。

(j) 2015 年, SG Bhele^[17]提出了一种基于抽取局部二值模式 (LBP) 的判别统计特征, 并在归一化后将其提供给修改后的方向梯度直方图 (HOG) 描述算子, 形成了 LBPHOG 特征向量作为各种分类器的输入。利用判别分析和基于距离的分类器对人脸图像进行分类。SG Bhele 将提出的方法分别在 ORL、UMIST、Yale B、LFW 四个人脸数据库上进行实验, 实验结果表明: 在人脸识别中存在的光照、姿态变化、表情等等复杂问题下, 此方法都保持了良好的识别效果。

(k) 2016 年, Akhilesh Raj^[18]等人提出使用 RGB 和 YCbCr 颜色空间对图像中像素点进行边缘肤色分割, 去除噪声后, 利用人脸的一般几何特征和眼睛的 Haar 级联来识别人脸的方法。该方法是在 CMU 数据库测试, 测试结果显示具有较高的检测率。本文还提出了一种基于卷积神经网络提取特征, 然后使用 Softmax 分类器进行人脸识别的方法。

(l) 2017 年, Cemil Turan^[19]提出了基于稀疏重建的人脸识别方法。改进的稀

疏人脸识别方法在训练样本足够多的情况下，其识别率明显优于 PCA 方法的识别率。同年，Jun Ying Zeng^[20]提出了基于深层卷积神经网络的单样本人脸识别方法，由于单样本的人脸数据集，很难使用深度学习的方法进行训练识别，所以 Jun Ying Zeng 提出增加训练集数据，学习好深度卷积神经网络（DCNN 模型，用增加的训练数据来微调深度卷积神经网络 DCNN 模型。最后对微调后的深度卷积神经网络（DCNN）模型进行实验，实验表明经过微调后的 DCNN 模型取得了较好的识别效果。

1.3 人脸识别技术在移动终端的前景及意义

现在随着人们对手机使用的频率越来越高，每个人都是一个移动终端，如果能将人脸识别技术应用到手机终端，那将会使人们进行身份识别更加方便，更加快捷。目前国内手机移动平台主要有两种主流平台，一种是以苹果公司开发的 IOS（Iphone Operation System）操作系统的移动平台。另一种是以谷歌公司开发的 Android 操作系统的移动平台。目前由于苹果公司的操作系统具有相对的封闭性，而相比于 Android 操作系统更加具有开发性，可扩展性强，所以结合 Android 平台的优势，我们完全可以将两者结合起来。由于 Android 平台强大的市场占有率，我们基于 Android 平台开发的人脸识别系统将会有很大的研究价值和前景。目前各大主流公司也争相开始对将人脸识别技术应用在移动终端展开研究并取得了一定的成绩，如苹果公司推出的新款手机具有人脸识别解锁功能，阿里巴巴公司也宣布未来将会有更多智能支付方式，比如刷脸支付方式。但是当下的人脸识别技术应用在手机端还仅仅只是在少数的高端机上，并未普及到一般的智能手机上，这些都更加验证了人脸识别技术应用在手机端的广阔的前景和不菲的价值。

不仅是手机端可以结合人脸识别系统，人脸识别系统还可以应用在不同领域的移动平台上，例如在海关出入境，机场等领域，需要进行身份认证登记的地方，将人脸识别技术应用在移动终端的智能识别机器将会帮助海关人员，机场工作人员进行智能身份识别，大大缩减工作人员的工作量，也大大提高了机场和海关等地出行人员的检查登记时间。同时也能实时监控出入境的人员流动情况，对出入境、准备登机的人员进行智能分析，协助工作人员完成安全检查工作。

1.4 本文研究思路和主要内容安排

在阅读过人脸识别领域相关文献和该领域的研究现状以及研究背景后，对基于 PCA 和 SVM 算法的手机人脸识别系统进行了可行性分析，并对整个系统的流程图进行了构思和设计，整个系统的功能流程图如图 1-2 所示。对整个系统的进行了整体设计后，下面分别详细介绍如何实现系统的各个功能模块，以及如何将各个功能模块连通起来，使系统可以运作起来。

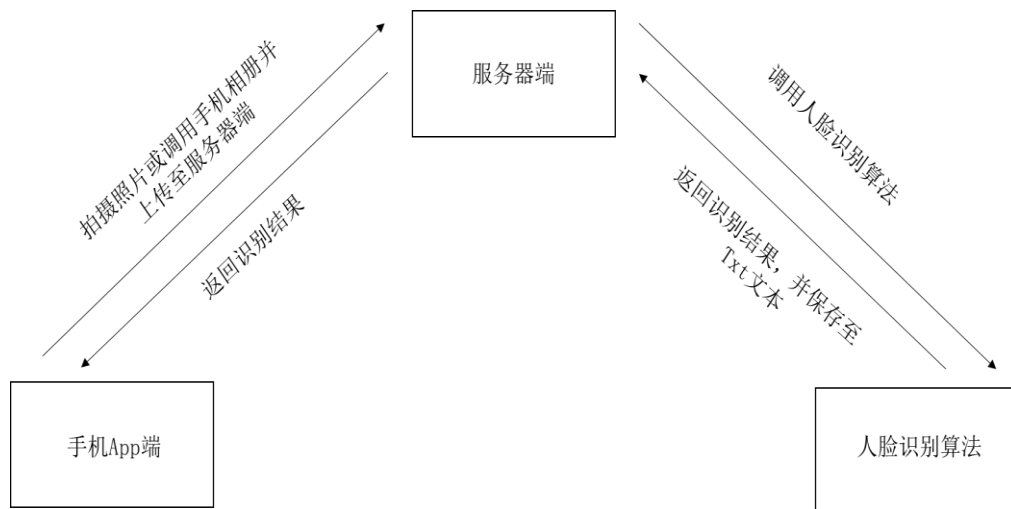


图 1-2 手机移动终端人脸识别系统架构

本系统 App 前端设计是利用 H Builder 开发工具所开发的。H Builder 开发工具是 D Cloud 公司旗下的一款完全免费的可视化前端页面制作工具，它具有良好的交互性，开发便捷，速度快等优点，通过完整的语法提示和代码输入法、代码模块，大大提升了 HTML、JS（JavaScript）、CSS（Cascading Style Sheets）的开发效率。基于 H Builder 平台，我们采用 html5 语言进行编写前端设计代码。H Builder 平台有非常完整的 API 接口，App 前端设计的功能模块基本上都是通过调用平台自动的 API 接口，例如前端 App 页面中的拍摄功能模块是通过调用 Camera 模块，再比如上传照片至服务器端功能模块是通过调用 Uploader 模块，Uploader 上传使用 HTTP 传输协议的 POST 方式提交数据，数据格式符合 Multipart/form-data 规范，即 rfc1867（Form-based File Upload in HTML）协议。前端设计的核心功能模块集成在 Uploader.html 文件中，在前端设计完成后，我们将上传的照片送至服务器端指定的文件夹下，在服务器端通过 VS（Visual Studio）平台搭配 Opencv 环境^[21]，调用基于 PCA 和 SVM 的人脸识别算法，其主要过程是对输入的预测照片进行预

处理（灰度化及归一化）并存储到一个高维向量中，对高维向量进行 PCA 降维后，通过训练好的 SVM 多分类器预测出待测照片的标签。最后将预测的标签存入指定文件夹下的 Txt 文件中，接着就是配置服务端模块，在服务器端配置的 Tomcat 服务器是在 Eclipse 开发环境下进行的，其核心配置文件为 PicService.java 文件，同时在配置好 Tomcat 服务器后，使用了 Ngrok 软件代理解决了手机 App 端与服务器端的数据通信，使得手机 App 端也可以在联网的情况下，将照片、文本文件等数据传输至服务器端，这样就可以通过 HTTP 传输协议将 Txt 文件中的信息作为识别结果传回到手机移动终端上。本文提出的系统在自建的人脸数据库和扩展的 ORL 人脸数据库、Yale 人脸数据库、IMM 人脸数据库和 Face95 人脸数据库上进行了验证，均取得的较好的识别率，同时系统速度也达到了一定水平。具体实验结果会在第四章详细介绍。

本论文的主要研究内容为搭建一个基于 PCA 和 SVM 算法的手机人脸识别系统，涉及到算法处理会在服务器端处理。本论文一共五个章节,分别是:

第一章主要从人脸识别技术的研究背景和意义、国内外研究现状、以及应用在移动终端的前景和意义等方面进行了详细的阐述。最后概括了本文的主要工作内容以及提出一种基于 PCA 和 SVM 算法的手机人脸识别系统，并介绍了此系统的工作原理和系统性能。

第二章首先介绍了数字图像处理相关基础知识，接着介绍了机器学习中经典的学习算法（最近邻算法、主成分分析法 PCA 算法、支持向量机 SVM 等）。最后介绍了基于 Eigenface 的人脸识别算法和基于 PCA 和 SVM 的人脸识别算法的原理。这些图像处理的基础知识、以及经典的机器学习算法、以及最后介绍的基于 PCA 和 SVM 的人脸识别算法都会在本文提出的基于 PCA 和 SVM 算法的手机人脸识别系统用到。

第三章介绍了基于 PCA 和 SVM 算法的手机人脸识别系统设计和实现。首先介绍了系统的整体设计，接着介绍移动 App 前端设计模块需要用的前端技术: Web 前端的 HTML、JavaScript、CSS 三大技术，然后介绍了 Android 操作平台架构，本文提出移动终端系统就是位于此平台架构中的最顶层的应用程序层，紧接着介绍了如何利用在 Android 平台上利用 Web 前端技术设计和实现本文中提出的移动终端人脸识别系统中的 App 前端。最后介绍了本文中提出的基于 PCA 和 SVM 算法的手机人脸识别系统中的服务器端的设计模块以及相关的理论知识。分别从

Tomcat 服务器的由来及如何配置相关环境变量、Eclipse 平台的相关理论知识和理论背景、HTTP 协议的理论知识以及在本文提出的系统中 HTTP 协议起到的作用、Ngrok 软件的相关知识和在本文提出的系统中的起到的作用这几个方面进行详细的阐述。

第四章是对本文提出的基于 PCA 和 SVM 算法的手机人脸识别系统进行整体功能测试和性能测试。首先介绍了整个系统设计与实现所需要的实验设备，最后对系统进行了功能测试和性能测试，并分析了实验结果。

第五章主要对本文中所做的工作内容进行了总结和展望。

第二章 基于 Eigenface 和基于 PCA 和 SVM 的人脸识别算法介绍

2.1 数字图像处理基本知识

2.1.1 引言

数字图像处理技术在很多领域有非常广泛的应用，比如在国民经济中的领域中。农林部门通过遥感图像、航拍高清图像对农林物生长情况进行实时观测并进行估计产值。各医疗部门利用各种数字图像技术对医学图像进行辅助分析。数字图像处理技术的定义分为两部分，第一部分是数字图像的定义，第二部分是怎样处理数字图像。首先一幅二维图像可以定义为一个二维函数 $f(x,y)$ ，其中 x 和 y 分别代表平面上的坐标，每一组 (x,y) 坐标对应一个 $f(x,y)$ ，我们把这个 $f(x,y)$ 称之为该图像在 (x,y) 处的灰度或者强度，当 $x, y, f(x,y)$ 都是有限的离散值的时候，我们称这幅图像为数字图像^[22]。这里的定义和我们定义数字信号的时候是一样道理，数字信号是一组离散的点，模拟信号是一组连续的点。接下来就是定义数字图像处理，我们把借助于数字计算机来处理数字图像称之为数字图像处理。数字图像处理的步骤主要是通过问题域输入相应的图像，等待计算机获取到图像后，根据输出的类别分为两类，一类为输出的数字图像，这一类的处理方法主要有：图像滤波与增强、图像复原、彩色图像处理、小波与多分辨率处理、图像压缩、形态学处理等方法；另一类为输出的数字图像的属性，这一类的处理方法有：形态学处理、图像分割、图像的表达和描述、目标识别等方法。其中形态学处理处理后输出的既可以是图像也可以是图像属性。另外我们常用于数字图像处理的图像主要有四类：二值图像、灰度图像、索引图像、真彩色 RGB 图像。其中二值图像主要用于文字、线条图的扫描识别 (Optical Character Recognition, OCR) 和掩膜图像的存储，灰度图像是图像矩阵中矩阵元素都是由取值范围为 $[0,255]$ 的灰度值组成，索引图像一般用于存放色彩要求比较简单的图像。索引图像由图像的二维矩阵和二维颜色索引矩阵组成。二维颜色索引矩阵的大小由存放图像的矩阵元素值域决定。RGB 彩色图像是直接将每个像素的颜色值直接存放到图像矩阵中，

每个像素的颜色值都由 R、G、B 三个分量表示。在输入端获取图像之后，往往不是直接对输入的图像进行处理，而是先对输入图像进行预处理，譬如归一化、图像缩放，在后面章节将会详细讲述。在本文中提出的基于 PCA 和 SVM 算法的手机人脸识别系统中，就是首先对手机 App 前端上传的照片进行预处理，再将处理好的图片存入待测的测试集中训练。图像增强是对一幅图像进行某种操作，在增强后得到的结果会在某些特定的环境下比原始图像更适合进行处理。在本文提出的人脸识别系统的预处理模块中，首先会对手机 App 前端上传的照片进行灰度化处理，然后再进行归一化，裁剪成统一尺寸 92*112，这就完成了预处理。下面会继续讲解如何将手机前端 App 上传的 RGB 彩色图像转换 gray 灰度图像及通过降采样裁剪图像尺寸。

2.1.2 灰度化及降采样

本文中提出的人脸识别系统是应用在 Android 智能手机上，而现在的智能手机拍摄的照片是 RGB 彩色图像，而在本文中提出的移动终端人脸识别系统的输入图像要求为灰度图像，所以我们需要将 RGB 彩色图像转换为灰度图像，转换灰度图像后我们接着对其进行降采样，最后得到统一大小的图像。

如何将一幅 RGB 彩色图像转换为灰度图像呢？首先我们介绍 RGB 彩色图像和灰度图像。RGB 彩色图像是分别由红、绿、蓝三种颜色组合来表示图像中的每个像素的颜色。将 RGB 彩色图像灰度化，方法有多种，常见的有分量法、最大值法、平均值法、加权平均值法这几种。分量法是针对彩色图像中的每个像素点，取三原色中的其中一种。最大值法就是对于图像中的每一个像素点，取该点的 R、G、B 三元素中的最大值作为该点的灰度值，平均值法就是对于图像中的每个像素点，取该点的 R、G、B 三元素中的平均值。加权平均法就是对于彩色图像中的每一个像素点，根据人眼对红色光、绿色光、蓝色光的敏感程度设定对三元素的不同权重，最后将三者相加起来作为该点的灰度值。计算公式为(2-1)：

$$Gray = 0.30R + 0.59G + 0.11B \quad (2-1)$$

在本文中提出的基于 PCA 和 SVM 算法的手机人脸识别系统的预处理模块中，我们采用就是加权平均法来对手机前端 App 获取的彩色图像进行灰度化处理。如图 2-1 就是经过灰度化处理后的灰度图像，接下就是统一裁剪图像的大小，这

里就用到了图像的降采样方法。接下来我们就要讲述如何进行降采样了。



图 2-1 灰度化处理

本文采用的图像降采样的方法是对于一幅 $M \times N$ 的图像，设置降采样行比例系数 K ，列比例系数 J ，根据比例系数 K 、 J ，对图像的每行每列分别每隔 K 行和 J 列进行抽样，最后将采样的点按照顺序排列形成一幅新的 $Q \times W$ 的图像。 K 的计算公式为 $K = \frac{M}{Q}$ ， J 的计算公式为 $J = \frac{N}{W}$ 。这样就实现了统一了人脸图像的尺寸大小 92×112 。采样后的结果如图 2-2 所示。

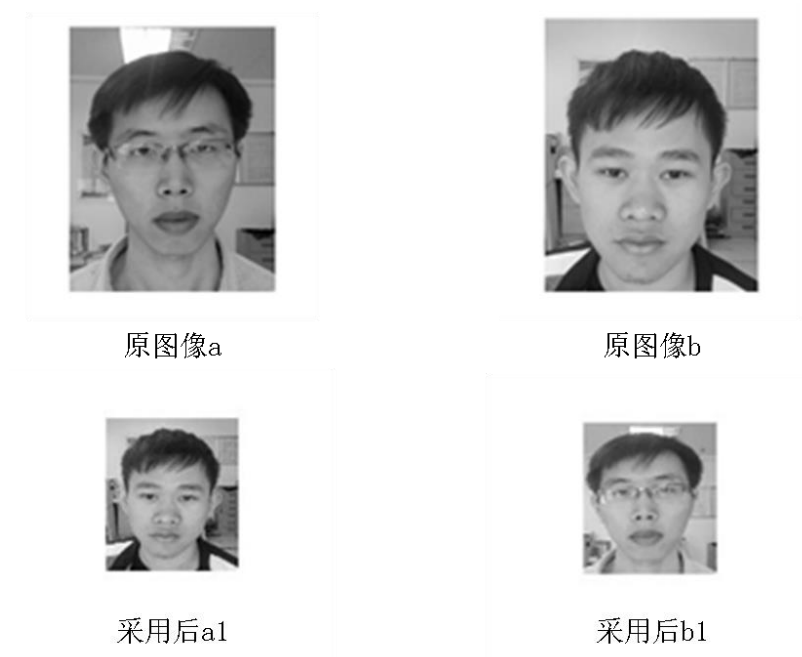


图 2-2 降采样的结果

经过灰度化和降采样后，就完成了对输入图像的预处理操作。

2.1.3 图像预处理基本操作

图像预处理就是在图像分析中，我们在对输入图像提取特征之前进行的处理。输入图像由于采集图像的设备性能的好坏与采集图像的环境的变化等原因大多数存在着各种噪声和对比度不强等问题。针对这些问题，需对输入图像进行相应的预处理，这样做的目的主要是消除输入图像中的无关信息，消除噪声，增强对比度，提高对输入图像的特征提取的准确度。

下面介绍几种常用的预处理方法，比如前两节中介绍的归一化、灰度化预处理方法，另外还有直方图均衡化（Histogram Equalization）、中值滤波（Median Filter）、顶帽变换（Top hat transform）、底帽变换（Bottom hat transform）等方法，接下来分别讲述这几种预处理的方法

直方图均衡化的工作原理是对灰度图像进行非线性拉伸，使得图像中的灰度级范围扩展到全部的灰度级范围，这样有效的解决了灰度图像中的像素在灰度级上分布不均匀等问题，图 2-3（a）是输入的原始图像，经过直方图均衡化后

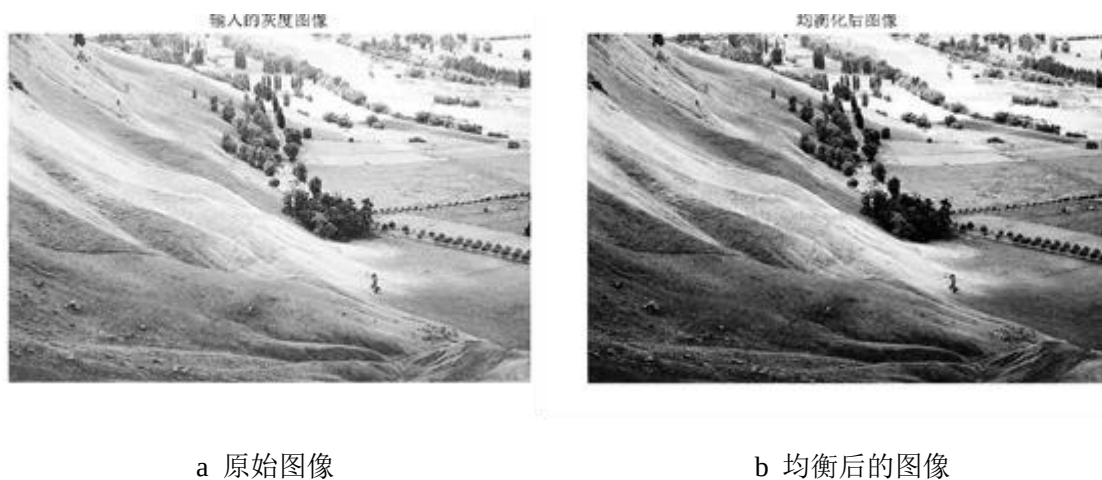


图 2-3 灰度图像均衡化

由图 2-3（b）可以看出，输出后的图像明显增强，灰度级分布更均匀。不过在灰度图像对比度增强的同时也存在着直方图均衡化的一些缺点，例如灰度图像的灰度级会减少，对应在图像的变化就是一些细节丢失了。

下面介绍另外两种增强对比度的方法：顶帽变换和底帽变换，很显然这是一对互补方法，首先说顶帽变换的算法原理它是灰度形态学的一种处理方法，通过

组合灰度形态学的基本图像处理操作而得到,例如灰度级开操作和灰度级闭操作。具体方法就是用输入的原图像减去被开操作后的原图像,其数学公式为:

$$T_{hat}(f) = f - (f \circ b) \quad (2-2)$$

对应的底帽变换也是由灰度形态学基础图像处理操作而得到,具体方法就是用被闭操作后的图像减去输入原图像,其数学公式为:

$$B_{hat}(f) = (f \bullet b) - f \quad (2-3)$$

下面我们举个例子,来看看顶帽变换的作用效果,如图 2-4 所示,经过顶帽变换



原图像

顶帽变换后的图像

图 2-4 顶帽变换

后的原图像对比度明显增强。

另外在实际应用中,拍摄的照片都会或多或少存在着各种噪声,例如椒盐噪声,它在灰度图像显示的就是各种黑白点,这些黑白点就是在灰度级两端的像素点,所以可以采用中值滤波保留灰度级中间部分,从而达到消除噪声目的。那么中值滤波又是怎么消除这些椒盐噪声的呢?首先设定一个邻域,这个领域是以图像中的某个像素为中心点,这个邻域也被称作模板,这个模板可以上下左右移动。通常情况下,设定的邻域为方形邻域比如 3*3 (一般默认方形邻域边长为奇数,方便取中间值),接着对邻域内的所有的像素值进行排序,找到排在中间的那个像素,最后将那个排在中间的像素赋值给对应的那个模板的中心像素,像这样让模板中心遍历到图像中的每一个元素。如图 4-4 所示,原始图像是一幅被椒盐噪声干扰的灰度图像,分别使用 3*3、5*5、7*7 模板进行中值滤波,从图 2-5 可以看出但是原图像中的一些细节也丢失了。所以我们在选取模板大小的时候也要考虑到

原图像的细节信息，应该在尽可能保持原图像的细节信息的同时消除噪声。

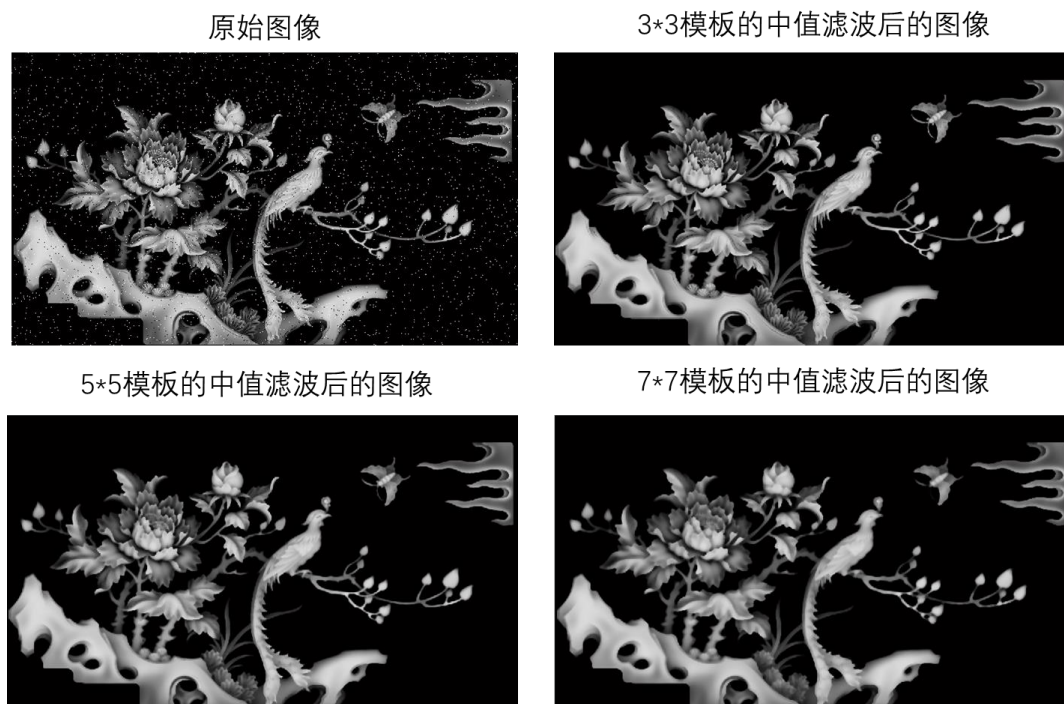


图 2-5 均值滤波降噪

2.2 机器学习

机器学习是人工智能领域非常重要的一个分支，它致力于研究如何通过计算的手段，利用先验知识来改善系统自身的性能^[23]。然而先验知识是以数据的形式存储在计算机系统中，机器学习主要研究内容就是如何找到利用经验数据产生出模型的算法，通过机器学习的学习算法，输入经验数据就可以生成模型，接着如果有新的数据输入到模型中，就可以对新的数据进行判断，得出结论。

下面介绍几种经典的机器学习算法，其中部分算法将会用在本文提出的基于 PCA 和 SVM 的人脸识别算法、基于 eigenface 的人脸识别算法和基于 LBP 和 PCA 的人脸识别算法中。

2.2.1 最近邻算法

首先我们介绍一下 K 近邻 (K-Nearest Neighbor, 简称 KNN) 学习算法是比较简单的分类算法^[24]。其算法核心思想是根据待测样本与训练样本之间的差异性，根据他们之间差异性的大小进行排序，最后从差异性最小的类开始依次选取前 K 个类，并将这 K 个类中出现次数最多的那个类作为预测结果输出。当 K=1 时，K

近邻算法就是最近邻算法，也即找到那个差异性最小的训练样本例，这个训练样本所属的类就是预测的类别。

如何计算待测样本和训练样本之间的差异性呢，我们通常采用计算欧式距离的方法，假设预测样本为 $p = (x_1, x_2, \dots, x_n)$ ，训练样本为 $Q = (y_1, y_2, \dots, y_n)$ ，则待测样本和训练样本之间的欧几里德距离（也称欧式距离）为式 2-4:

$$D = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2-4)$$

在用最近邻算法之后找出最近邻样本为 z ，假设测试样本 x ，则该算法的分类出现错误的概率为式 2-5:

$$P(err) = 1 - \sum_{c \in y} P(c|x)P(c|z) \quad (2-5)$$

有理论证明最近邻算法的泛化错误率小于等于贝叶斯最优分类器的错误率的两倍。由此看出，尽管最近邻算法是一个相对简单的分类算法，但是其泛化错误率还是相对较低的。

2.2.2 支持向量机(SVM)

支持向量机（Support Vector Machines，简称 SVM）是机器学习中比较经典分类的算法。支持向量机（SVM）可以认为是一种二分类模型，这个模型的定义一个在特征空间上的间隔最大的线性分类器，如何找出最大间隔呢？这里就用到了凸优化，求最大间隔问题最终转换为一个凸二次规划问题，通过求解凸二次规划问题可以学习到分离超平面，如式 2-6 所示：

$$w^T x + b = 0 \quad (2-6)$$

支持向量机是一种有监督的学习方法，多用于统计分类和回归分析中。支持向量机一般情况可以解决线性可分的分类问题、线性不可分的分类问题、非线性分类问题这三类问题。下面简单介绍一下这支持向量机是如何解决这三类问题的。

首先是线性可分分类问题，假设训练样本集 $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ ， $y_i \in \{-1, 1\}$ ，如何找到一个超平面（hyper plane）能将训练样本集 D 的两类训练样本分开并且使得离超平面最近的两个不同类之间的间隔最大呢？这里就用到了线性 SVM 分类器，通过线性 SVM 分类器模型学习后，确定超平面的参数 w 和 b

就可以找到这个超平面。超平面的表达式中 $w = (w_1; w_2; \dots; w_d)$ 为超平面的法向量， w 决定了超平面的方向， b 是超平面的位移项，它决定了超平面与原点之间的距离。假设超平面可以正确的将两类样本分类，那么根据超平面的表达式，对于 $(x_i, y_i) \in D$ ，如果 y_i 是“1”这一类，那么 $w^T x_i + b > 0$ ，如果 y_i 是“-1”这一类，那么 $w^T x_i + b < 0$ 。令

$$\begin{cases} w^T x_i + b \geq 1, y_i = 1 \\ w^T x_i + b \leq -1, y_i = -1 \end{cases} \quad (2-7)$$

当且仅当某些训练样本 x_i 在超平面上时，式 2-7 的等号成立。落在超平面上的训练样本被称之为“支持向量”（support vector），根据式 2-8 可以计算出两个不同类的训练样本到平面的距离之和，它被称之为间隔。

$$\text{Margin} = \frac{2}{\|w\|} \quad (2-8)$$

图 2-6 是支持向量机（SVM）解决二分类的例子，图中标注了间隔、支持向

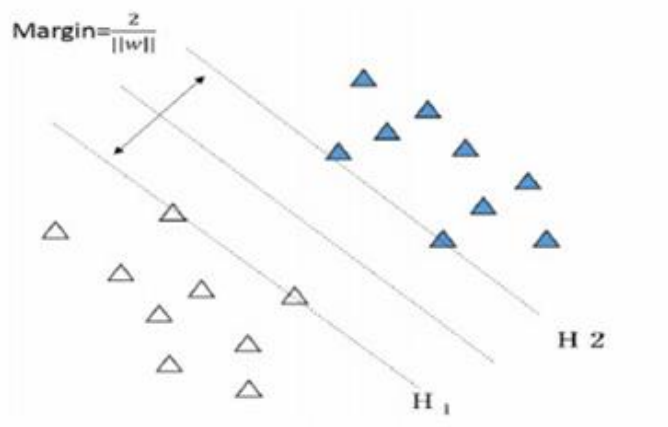


图 2-6 对于二分类的支持向量机

量。对于 $y_i = -1$ 这类点时，支持向量落在超平面 H_1 上，对于 $y_i = 1$ 这类点时，支持向量落在 H_2 上。

要使得 Margin 最大，同时又满足式 2-7，这样就转化为式 2-9:

$$\begin{cases} \max_{w, b} & \frac{2}{\|w\|} \\ \text{s.t.} & y_i(w^T x_i + b) \geq 1, \quad i=1, 2, \dots, m \end{cases} \quad (2-9)$$

由数学知识可以证明求 $\frac{2}{\|w\|}$ 的最大值等价于求 $\frac{1}{2} \|w\|^2$ 的最小值，所以式 2-8 转化

为式 2-10:

$$\begin{cases} \min_{w,b} & \frac{1}{2} \|w\|^2 \\ \text{s.t} & y_i(w^T x_i + b) \geq 1, \quad i=1,2,\dots,m \end{cases} \quad (2-10)$$

式 (2-10) 就是一个简单的支持向量机模型, 接下来就是如何求解出 w 和 b 呢? 这里利用拉格朗日对偶性, 我们引入拉格朗日乘子 $\alpha_i \geq 0, i=1,2,\dots,N$, 定义拉格朗日函数:

$$L(w,b,\alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^N \alpha_i y_i (w x_i + b) + \sum_{i=1}^N \alpha_i \quad (2-11)$$

根据拉格朗日对偶性, 我们将式 2-10 最优化约束问题就转化为式 2-12 的对偶优化问题。

$$\begin{cases} \min_a & \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (x_i * x_j) - \sum_{i=1}^N \alpha_i \\ \text{s.t} & \sum_{i=1}^N \alpha_i y_i = 0 \\ & \alpha_i \geq 0, \quad i=1,2,\dots,N \end{cases} \quad (2-12)$$

求解出 α_i 后, 就可以根据 2-13,2-14 式求解出 w 和 b 。

$$w = \sum_{i=1}^N \alpha_i y_i x_i \quad (2-13)$$

$$b = y_i - \sum_{i=1}^N \alpha_i y_i (x_i * x_i) \quad (2-14)$$

求出最优解 w 和 b 后, 就确定了分离超平面 H , 以及判断类别的分类决策函数^[25], 如式 2-15 所示:

$$f(x) = \text{sign}(w^T x + b) \quad (2-15)$$

求得分离超平面和分类决策函数成为线性可分支持向量机。利用线性可分支持向量机学习算法就解决了线性可分的分类问题。

接下来就是对于线性不可分分类问题, 对于这类问题我们采用线性支持向量机来解决。假设我们的训练样本集为 $T = \{(x_1, y_1)(x_2, y_2), \dots, (x_N, y_N)\}$, $y_i = \{1, -1\}$, 线性支持向量机的原理是引入一个松弛变量 β , 使得函数间隔大于等于 1, 这里实际上是通过加入松弛变量来允许分类器有部分分类错误, 这个允许分类器分类错

误被称为惩罚参数 C ，通过调节 C 让函数间隔尽可能大，同时又让错误分类的尽量少，我们把这种方法称为“软间隔最大化”。这样我们就在线性可分支持向量机模型之上进行改进，如式 2-16,2-17 所示：

$$\min_{w,b,\beta} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \beta \quad (2-16)$$

$$\begin{aligned} \text{s.t. } & y_i(w * x_i + b) \geq 1 - \beta, i = 1, 2, \dots, N \\ & \beta \geq 0, i = 1, 2, \dots, N \end{aligned} \quad (2-17)$$

这也是一个凸二次规划问题，同样我们可以用拉格朗日对偶性，对其进行求解。

最终转换为式 2-16,2-17，通过 2-18，2-19 式我们可以求出 α 。

$$\max_{\alpha} -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (x_i * x_j) + \sum_{i=1}^N \alpha \quad (2-18)$$

$$\text{s.t. } \sum_{i=1}^N \alpha_i y_i = 0 \quad (2-19)$$

$$0 \leq \alpha_i \leq C$$

求解出 α 后，就可以根据式 2-20,2-21 求解出 w 和 b ，进而求得分离超平面和分

$$w = \sum_{i=1}^N \alpha_i y_i x_i \quad (2-20)$$

$$b = y_i - \sum_{i=1}^N y_i \alpha_i (x_i x_j) \quad (2-21)$$

类决策函数。我们将求得的分类超平面和分类决策函数统称为线性支持向量机。这样就解决了线性不可分分类问题，实际生活中的问题常常是线性不可的，所以线性支持向量机比线性可分支持向量机应该更广泛。

最后的问题就是非线性分类问题，在实际生活中我们也会经常遇到非线性分类问题，这类问题就是不能用线性模型将两个不同类的数据正确分离开，这时候我们有什么解决办法呢，前面提到我们已经解决了线性分类问题，那怎么才能对非线性分类问题也可以由线性支持向量机解决呢，方法就是通过非线性变换将输入的非线性空间 χ 映射到线性空间，这个映射函数为式 2-22 所示：

$$\phi(x): \chi \rightarrow H \quad (2-22)$$

我们定义核函数为： $K(x, z) = \phi(x) * \phi(z)$ ，在前面小节中，我们可以看到无论是线

性可分还是线性不可分问题，我们求解的分离超平面和分类决策函数中都是只含有训练样本与训练样本之间的内积，所以同样如果我们利用核函数的定义来代替训练样本之间的内积。此时我们的目标函数和分类决策就可以改写成式 2-23,2-24

$$S(\alpha) = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^N \alpha_i \quad (2-23)$$

$$f(x) = \text{sign}(\sum_{i=1}^N \alpha_i y_i K(x_i, x) + b) \quad (2-24)$$

这样我们就可以通过核函数来使得本输入空间内积映射到新的线性空间做内积。我们通过非线性变换核函数就把非线性分类问题转换到一个线性可分问题，就可以利用线性支持向量机来解决了。在求取核函数的时候，我们通常不通过映射函数求取，而是直接求取。有效的核函数充分必要条件是核函数矩阵 K 是对称半正定的，常用的函数有多项式核函数、高斯核函数、字符串核函数、径向基核函数 RBF 等等。

在解决多分类问题时，SVM 是将多分类问题转换为多个二分类问题，通过训练生成多个二分类器，最后可以通过两类方式构造多分类器，一类是直接法，直接在单个二分类目标函数上进行修改，直接将多个二分类器的参数合并成一个最优化问题中，直接求解优化问题中的参数。这种方法显然会因为大量的参数求解而导致计算量非常大，所以除了少量的多分类问题，否则一般不采取这种方法。第二类方法就是间接法，间接法就是通过已经训练生成的多个二分类器进行组合构成多分类器，间接法分为两类方法，一类为“一对多法（One-Versus-Rest）”，一类为“一对一法（One-Versus-One）”。下面分别介绍这两种方法及其优缺点

首先一对多法，其方法原理是假设有 n 类训练样本需要分类，我们依次取这 n 类中的任意一类作为正类，剩余的类作为一个整体为负类，最终我们得到 n 个二分类器，训练这 n 个二分类器后，我们将预测样本分别输入到这 n 个二分类器中，取分类函数最大值对应的那个类作为预测样本的预测结果。下面评价一下这个方法的优缺点。优点是分类器数量较少，分类速度较快。缺点是：

- (a) 对于每个分类器训练时，都需要对整个训练集进行训练，这样导致训练时间增加。
- (b) 由于正类样本数远远小于负类样本数，所以导致出现了样本不对称情况。对识别准确率有影响。

- (c) 在增加新的类别分类时，要对之前训练好的分类模型重新训练，这样就造成训练时间过长。

接着讲述采用“一对一法”构造多分类器，方法原理是对于 n 类分类样本，我们任意取其中两类作为 SVM 二分类器，这样我们可以得到 $\frac{n \times (n-1)}{2}$ 个 SVM 二分类器，训练好这 $\frac{n \times (n-1)}{2}$ 个 SVM 二分类器后，将预测样本分别输入这些 SVM 二分类器后，采用投票机制，获得票数最多的类别作为预测的结果。这种方法的优点恰好就是改善了一对一法中再增加新的类别分类时，需要训练所有的 SVM 二分类器这一问题，同时此方法在训练单个 SVM 二分器时，速度较快。此方法的缺点也很明显，就是会随着 n 的增加，其构造的 SVM 二分器数量也会急剧增加，所以总的训练时间和测试时间还是相对较慢。

2.2.3 主成分分析法 (PCA)

在许多科学研究与应用领域中，经常会遇到对海量的数据进行数据分析，通过对海量的数据进行分析后得出有效的结果，在对海量的数据进行分析的时候常常会遇到因为数据量过大而导致分析难以进行，这时候我们就需要对海量的数据进行处理，将海量的数据提取出有效的信息，去除冗余的信息成分，我们在去除冗余的信息常常会遇到由于数据信息之间存在着一定的相关关系这样的问题，那么，如何找出一种合理的方法，它可以解决在数据信息之间的一定的相关关系的同时再去除冗余的信息、最大可能性的保留原数据中的有效信息。如果找到这样一种方法，就可以对海量的数据信息继续分析了。

主成分分析法(PCA)就是这样可以解决这类问题的方法，主成分分析法(PCA)是一种常用于数据降维的方法，是一种多元统计分析中用来分析数据的方法。其核心思想是找到一个 K 维的投影空间，然后利用这个 K 维的投影空间将高维特征数据降维至低维子空间上，实现数据降维。主成分分析法(PCA)的算法过程为：首先输入样本集为 $D = \{x_1, x_2, \dots, x_m\}$ ，并设定好低维空间的维数 k ，接着对输入的所有样本进行中心化，公式为 2-25 所示：

$$x_j = x_i - \frac{1}{m} \sum_{i=1}^m x_i \quad (2-25)$$

接着计算由 x_j 组成的协方差矩阵 XX^T ，计算完协方差矩阵 XX^T 后，对其进行特征值分解，接着将求得特征值从大到小排序，取前 k 个特征值，对应的特征向量为 $w_1, w_2, \dots, w_k$ ，得到这组特征向量后，由这组特征向量组成的矩阵就是我们得到的投影矩阵 $W = (w_1, w_2, \dots, w_k)$ ，得到这个投影矩阵后，就可以将 m 维特征降维至 k 维特征空间。整个过程就是实现了数据降维。主成分分析法（PCA）应用在人脸识别领域已经十分成熟了，主要应用该方法实现了人脸特征提取和数据降维。

2.2.4 基于 Eigenface 的人脸识别算法介绍

在基于主成分分析法（PCA）的基础之上，本文介绍一种基于特征 Eigenface 的人脸识别算法。图 2-7 为基于特征 Eigenface 的人脸识别算法流程图。该算法的大体流程分三个阶段，第一个阶段是预处理阶段，第二个阶段是特征提取阶段，第三个阶段是识别阶段。首先在预处理阶段对其输入图像进行灰度化、归一化等处理后，接着对预处理后的灰度图像进行 PCA 降维，提取特征处理，这一步是算法的核心，也即是找到一个 PCA 子空间，这个子空间是由 k 个最大特征值对应的特征向量组成的特征空间，那么如何确定这个 k 值呢？

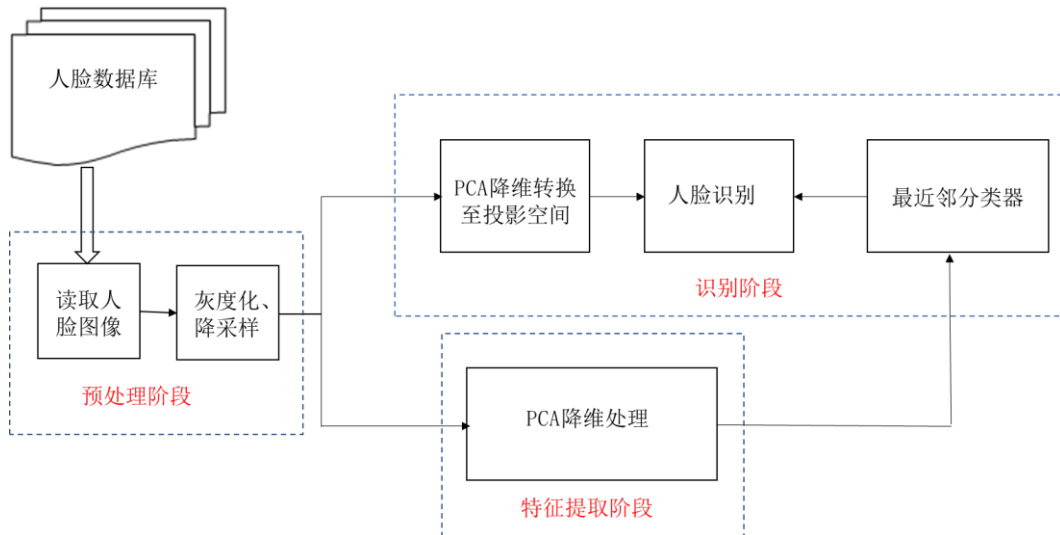


图 2-7 基于特征 Eigenface 的人脸识别算法流程图

这里就有用到了方差贡献率 β ，由上一节 2.2.3 的内容知道，假设训练样本 $D = \{x_1, x_2, \dots, x_m\}$ ，对所有样本进行中心化后，求中心化后的训练样本的协方差矩阵，最后求得相应的特征值。可以根据式 2-26 的判别式：

$$\frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^m \lambda_i} \geq \beta \quad (2-26)$$

可以确定 k 的值。最后就进入了最后的识别阶段了，接着将训练集中所有样本投影至用 PCA 方法获得的 PCA 子空间上，接着再将待识别的测试样本同样投影至 PCA 子空间上，最后计算测试样本和训练样本之间的欧式距离，再根据最近邻分类器的原理取测试样本与之最近的训练样本的类作为预测结果的类。Alogrithm2.1 算法介绍了此算法的实现过程。

Alogrithm2.1 基于 Eigenface 的人脸识别算法

输入：

D ：训练样本集 $\{(x_1, Y_1), \dots, (x_m, Y_m)\}$

预测样本 (y_i, Y_i)

低维空间维数 k

过程：

1：灰度化和降采样

2：对所有样本进行中心化： $x_j = x_i - \frac{1}{m} \sum_{i=1}^m x_i$

3：求中心化后的样本的协方差矩阵 XX^T

4：对协方差矩阵做特征值分解

5：取最大的 k 个特征值所对应的特征向量 $w_1, w_2, \dots, w_k$

6：将预测样本和训练样本都映射到由 k 个特征向量组成的特征空间上，得到在此特征空间的上预测样本 y_i' 和训练样本 x_i'

7：对预测样本和训练样本之间求欧式距离： $Dis = \sqrt{\sum_{i=1}^m (x_i' - y_i')^2}$

8：找出距离最小的训练样本对应的标签 Y_n

输出：

预测标签 Y_n

此算法在扩展的 ORL 人脸数据库取得了较好的识别结果，下面介绍扩展的 ORL 人脸数据库，并对此算法在扩展的 ORL 人脸数据库上进行算法验证和分析。

首先我们需要下载 ORL 人脸数据库，下载完成后存储路径为 E:\vs2010\project\faceRec_acculate\Debug，ORL 人脸数据库就存储在当前路径下的 att_faces 文件夹下，这个文件夹下有 40 个子文件夹，所有文件夹名是 1 到 40，每个文件夹分别是一类人脸图像，子文件中包含 10 张这类人不同姿态的人脸图像。

接着我们通过智能手机拍摄 5 副人脸图像，均采用同一位置，同一时间，每次拍摄时进行小幅度的姿态变化，每副人脸图像包含 10 张不同姿态的人脸图像，总计 50 张人脸图像，然后我们在训练样本之前会对新加入的图像进行降采样、灰度化处理，最后将每个类的人脸图像存储在 att_faces 文件夹下，文件夹名按照之前的文件夹依次取为 41 至 45。

最后为了测试系统性能，将扩展的 ORL 人脸数据库分为测试集和训练集。我们将扩展的 ORL 人脸数据库中的每个人脸类中最后 2 张人脸图像作为测试集，训练集分别采用每个人脸类中 1 至 5 个人脸图像，最后我们分别在测试集和训练集的文件夹下进入 windows 系统终端下，分别输入命令行 `dir /b/s *pgm > att_faces_test.csv`、`dir /b/s *pgm > att_faces_train.csv` 生成对应的 CSV 文件。这样系统就可以通过读取 CVS 文件来获取训练集和测试集了，这样就省去了一张一张的载入系统中，在生成了 CSV 文件后，我们接着对训练集和测试集的 CSV 文件中的每个人脸图像标上标签，这里为了便于统计，我们给的标签是从阿拉伯数字 0 到 44。这样就完成了对 ORL 人脸数据库的改进工作。

我们在前面小节中介绍到在确定特征值的个数 k 是由方差贡献率 β 决定的，

表 2-1 是方差贡献率 β 与识别率关系

方差贡献率	正确识别人脸个数	测试集人脸总数	识别率
20%	51	225	22.67%
40%	80	225	35.56%
60%	118	225	52.44%
70%	136	225	60.44%
80%	155	225	68.89%
85%	157	225	69.78%
90%	163	225	72.44%
95%	154	225	68.44%
99%	151	225	67.11%

这里我们通过调整方差贡献率 β 来确定使得识别率最高时的 K 值，同时训练集采用扩展的 ORL 人脸数据库中每个人脸类中前 5 张作为训练，后 5 张作为测试

集。从表 2-1 中可以看出 $\beta=90\%$ 时, $K=160$ 维时, 算法的识别率最高。这里我们取 $\beta=90\%$, 再通过修改训练集的单个人脸类的训练样本的数量来测试算法采用最近邻分类器得到的识别率, 表 2-2 是使用最近邻分类器测试的系统识别率。从表 2-2 中可以看出: 随着单个类的训练样本个数的增加, 算法的识别率不断提高, 但是总体识别率还是偏低, 所以我们还需要再此基础之上改进, 我们对算法进行分析发现, 此算法在单个类训练样本数较少时, 识别效果欠佳。这里我们不妨先采

表 2-2 单个类的训练样本数目与基于 Eigenface 的算法识别率的关系

单个类的训练样本个数	正确识别人脸个数	测试集人脸总数	识别率
1	42	90	46.67%
2	51	90	56.67%
3	59	90	65.56%
4	71	90	78.89%
5	75	90	83.33%

用 LBP 算子提取特征, 再用 PCA 算法进行数据降维, 最后使用最近邻分类器, 这种方法就是我们要接下来介绍的基于 PCA 和 LBP 的人脸识别算法。

2.2.5 基于 PCA 和 LBP 的人脸识别算法介绍

本小节将介绍一种基于 PCA 和 LBP 的人脸识别算法, 此算法的原理是对数据进行预处理后, 使用 LBP 算子提取特征, 接着对提取特征后 LBP 图进行 PCA 降维, 最后使用最近邻分类器分类, 图 2-8 是 LBP 和 PCA 的人脸识别算法流程图。接下来具体讲述 LBP 算子是如何提取特征的, 首先对预处理后的图像进行分割为相同

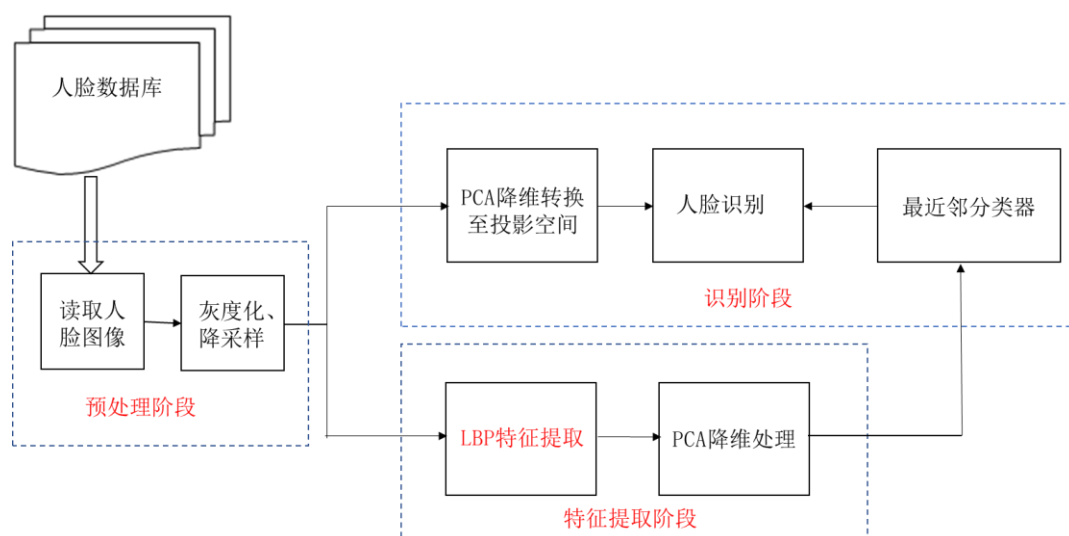


图 2-8 是 LBP 和 PCA 的人脸识别算法流程图

大小的方形模块，接着对每个方形模块中的每一个像素对其进行求取 LBP 值，我们是通过一个 3*3 的模板，通过模板中除去中心像素的像素点与中心像素进行比较像素值大小，比中心像素值大的像素点取“1”，比中心像素值小的取“0”，这样我们就可以得到 8 个不同的“0”，“1”值了，我们按照顺时针得到一组二进制数字，将这组二进制数字转换为十进制，就是我们的 LBP 值了。我们通过此方法对每一

Alogrithm2.2 基于 LBP 和 PCA 的人脸识别算法

输入：

D ：训练样本集 $\{(x_1, Y_1), \dots, (x_m, Y_m)\}$ 预测样本 (y_i, Y_i)

低维空间维数 k

过程：

1：灰度化和降采样

2：将训练样本分成 7*7 的方形模块

3：使用 3*3 的模板对每个方形模块中像素通过模板中除去中心像素的像素点与中心像素进行比较像素值大小，比中心像素值大的像素点取“1”，比中心像素值小的取“0”，这样我们就可以得到 8 个不同的“0”，“1”值了，按照顺时针得到一组二进制数字，将这组二进制数字转换为十进制，就是得到 LBP 值了，最终获得所有方形模块的 LBP 特征向量

4：对所有方形模块中 LBP 特征向量进行中心化：
$$z_j = z_i - \frac{1}{m} \sum_{i=1}^m z_i$$

5：求中心化后的样本的协方差矩阵 ZZ^T

6：对协方差矩阵做特征值分解

7：取最大的 k 个特征值所对应的特征向量 $w_1, w_2, \dots, w_k$

8：将预测样本和训练样本都映射到由 k 个特征向量组成的特征空间上，得到在此特征空间的上预测样本 y_i' 和训练样本 x_i'

9：对预测样本和训练样本之间求欧式距离：
$$Dis = \sqrt{\sum_{i=1}^m (x_i' - y_i')^2}$$

10：找出距离最小的训练样本对应的标签 Y_n

输出：

预测标签 Y_n

特征向量使用最近邻分类器识别分类。Alogrithm2.2 算法介绍了此算法的实现过程。

该算法同样在扩展的 ORL 人脸数据库中进行了算法验证，我们分别采用不同的方形模块来分割图像，将 LBP 特征向量进行 PCA 降维，降维至方差贡献率为 90%时，此时的 K 值为 300，即降维至 300 维时，最后用最近邻分类器进行分类识别。表 2-3 是使用不同大小的方形模块后在 ORL 人脸数据库上的识别效果。从表 2-3 中可以看出随着方形模块不断增大的时候，算法的识别率在不断提高，但是由于方形模块的不断增大，特征向量的维度也在急剧增加，导致计算时间较慢，需要综合考虑模块大小。我们还可以看出经过 LBP 特征提取后再经过 PCA 降维后的算法识别率有了一定的提升，但还未达到较理想的效果。

表 2-3 方形模块的大小与识别率的关系

方形模块边长	5	7	9	11
正确识别人脸个数	60	69	74	81
测试集人脸总数	90	90	90	90
识别率	66.67%	76.67%	82.22%	90.00%

2.2.6 基于 PCA 和 SVM 的人脸识别算法介绍

前面介绍的两个算法在识别时选用的最近邻分类器结构相对简单，分类效果欠佳，我们可以考虑使用结构较复杂 SVM 多分类器对 PCA 降维后的训练样本进行分类训练。这里我们就介绍一种基于 PCA 和 SVM 的人脸识别算法，其算法原理就是在 PCA 算法基础之上加入了 SVM 多分类器，图 2-9 是 PCA 和 SVM 的人

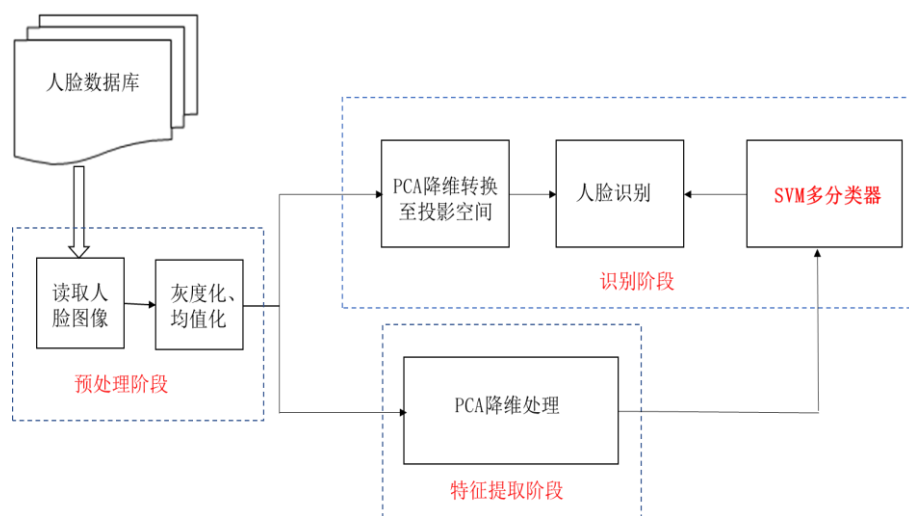


图 2-9 PCA 和 SVM 的人脸识别算法流程

脸识别算法流程图。接下来讲述一下这个 SVM 多分类器的工作原理，前面章节介绍了 SVM 多分类器一般由直接法和间接法两种方法得到，直接法一般对于分类类别较多时是不适合的，所以我们这里采用间接法，间接法中，我们采取一对一的投票策略，首先我们假设有 N 类样本需要划分，两两组合成训练集，通过训练样本集就得到了 $(N*(N-1))/2$ 个分类器。最后采用投票机制，将预测样本输入到这些分类器中，计算得票数最多的类别则为预测的预测结果类。这就是多分类器的采用投票机制实现将多分类问题转换为多个二分类的原理。Alogrithm2.3 算法介绍

Alogrithm2.3 基于 PCA 和 SVM 的人脸识别算法

输入：

D ：训练样本集 $\{(x_1, Y_1), \dots, (x_m, Y_m)\}$

预测样本 (y_i, Y_i)

低维空间维数 k

代价参数 C

径向基核函数 Gamma

过程：

1：灰度化和降采样

2：对所有样本进行中心化：
$$x_j = x_i - \frac{1}{m} \sum_{i=1}^m x_i$$

3：求中心化后的样本的协方差矩阵 XX^T

4：对协方差矩阵做特征值分解

5：取最大的 k 个特征值所对应的特征向量 $w_1, w_2, \dots, w_k$

6：将预测样本和训练样本都映射到由 k 个特征向量组成的特征空间上，得到在此特征空间的上预测样本 y_i' 和训练样本 x_i'

7：对特征空间上的训练样本使用多个 SVM 二分类器训练

8：将预测样本输入到多个 SVM 二分类器中，通过投票机制，统计出获票最多的那个人脸类作为预测结果类的标签

输出：

预测标签 Y_n

了此算法的实现过程。

该算法同样在扩展的 ORL 人脸数据库中进行了算法验证，同样在方差贡献率为 $\beta = 90\%$ ， $K=160$ 时，我们将构成 SVM 多分类器对映射在特征空间上的训练样本进行训练，在训练单个 SVM 分类器时，我们选用的核函数为径向基核函数，通过参数调优方法，我们取代价参数 $C=128$ ， $\text{Gamma}=0.01$ 识别率最高为 94%，得出训练模型，最后通过训练模型对预测样本进行预测，得出预测结果，我们这里同样采用改变训练集中单个类的训练样本数目来测试其算法识别率，表 2-4 是此算法与单个类的训练样本数的关系。

表 2-4 单个类的训练样本数与基于 PCA 和 SVM 算法识别率的关系

单个类的训练样本个数	正确识别人脸个数	测试集人脸总数	识别率
1	50	90	55.56%
2	66	90	73.33%
3	73	90	81.11%
4	77	90	85.56%
5	84	90	93.30%

从表 2-4 中可以看出，当单个类的训练样本增加的时候，此算法的识别率也在不断提高，可以确定的是算法的识别率和单个类的训练样本成正比关系。对比表 2-2 和表 2-3，我们可以看出使用 SVM 多分类器后，算法的识别率有了明显的提升，所以本文提出的移动终端系统采用 PCA 和 SVM 的人脸识别算法作为系统的算法调用模块。

2.3 本章小结

本章主要介绍了基于 Eigenface 的人脸识别算法、基于 LBP 和 PCA 的人脸识别算法和基于 PCA 和 SVM 的人脸识别算法三种算法，并在扩展的 ORL 人脸数据库分别从方差贡献率 β 、方形模块的大小和单个人脸类的训练样本数目来验证了其算法性能，比较了他们之间的优缺点，最后选出较优算法作为本文后面移动终端人脸识别系统的算法调用模块。还介绍了这两种算法用到的图像处理的基础知识和机器学习中经典分类算法。

第三章 基于 PCA 和 SVM 算法的手机人脸识别系统设计与实现

3.1 概述

在第二章中我们通过性能对比，最终选取基于 PCA 和 SVM 的人脸识别算法作为本文提出的基于 PCA 和 SVM 算法的手机人脸识别系统中的算法模块，接下来在本章中我们首先介绍系统整体的运行原理，然后接着详细介绍系统的另外两个模块，分别是 App 前端模块和服务器端模块。

3.2 系统整体设计

首先系统总体分三大模块，分别是人脸识别算法、App 前端、服务器端。系统框架流程图如图 3-1 所示

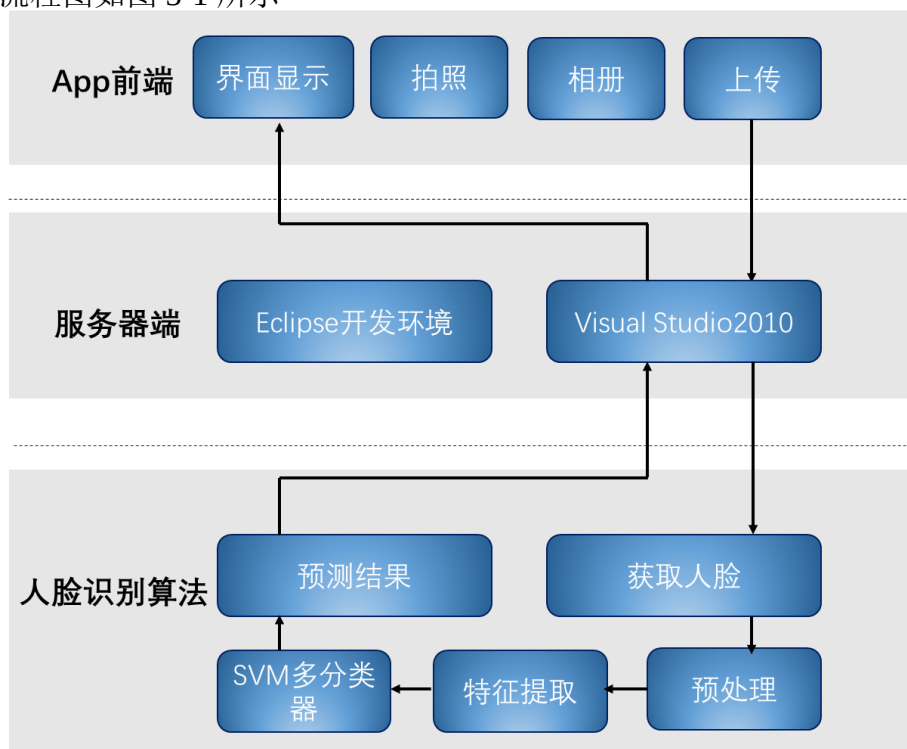


图 3-1 系统框架流程图

由图 3-1 可以看出，系统的工作原理是：首先从手机端通过调用手机摄像头或者调用相册两种方式获取人脸图像，接着将人脸图像上传至服务器端，服务器

端在指定文件夹中检测到上传的照片后，自动调用基于 PCA 和 SVM 的人脸识别算法，整个算法会经过预处理、特征提取、SVM 多分类器训练几个步骤，最后输出预测的结果，通过调用算法获取到预测结果后会以 Txt 文本保存至服务器端的指定文件夹中最后通过之前在 Eclipse 环境下配置 Tomcat 服务器检测到 Txt 文本后，自动将 Txt 文本中的内容通过 HTTP 协议传回至手机 App 端。

3.3 App 前端设计

Web 前端的开发工具有很多，常见的 Web（Website）前端工具有 Bootstrap、Foundation、Notepad++、Cloud9 IDE、Visual Studio Code 等等，Bootstrap 是快速开发 Web 应用程序的前端工具包。这个工具包它既包括 CSS，还包括 HTML。Foundation 框架是一个非常便捷、有效、灵活的前端框架，它可以用来做很多 Web 端的应用开发。这个框架提供了很多 UI（User Interface）组件，比如 Button（按钮）、Icon（图标）等。Notepad++ 是一款开源的编辑器，做前端开发的必备软件，这个软件它支持很多常用的编程语言，例如 C 语言、C++、Java、C#、HTML、JavaScript。Visual Studio Code 是微软公司在 Build 开发者大会上宣布的一个项目，是一个运行与 Mac Operation System、Windows 和 Linux 之上的，针对于编写 Web 和云应用的跨平台编辑器^[26]。本系统的前端开发是在 H Builder 平台上基于 HTML5 语言开发的，最后在平台上生成 Apk（Android Package）安装包，最后通过 Apk 安装包解压安装在 Android 手机系统中。

3.3.1 移动 Web 前端技术

Web 前端技术包括 HTML、CSS、JavaScript 等技术，在进行网页布局时，HTML 进行元素定义，HTML 是网页内容的载体。网页的内容包含文字、图片、视频、音频等。CSS 对定义的元素进行布局，比如，标题字体、颜色变化，或为标题加入背景图片、边框等。这些用来改变内容外观的东西叫做表现。最终使用 JavaScript 完成对用户的交互设计，即（前端与服务器的通信），比如说鼠标滑过弹出下拉菜单。或者鼠标滑过表格的背景颜色改变等等与用户之间的交互。

下面分别详细讲述这三种技术。H Builder 开发平台支持 HTML、CSS、JavaScript 三种前端编程语言，在后面章节会介绍 H builder 平台的特点。首先我们

来介绍一下 HTML 技术, HTML 是一种超文本标记语言 (Hypertext Markup Language), 它最早是在 1993 年 6 月作为互联网工程工作小组 (IETF) 工作草案发布的^[27], 它是一种规范、一种标准, 它通过标记符号来标记要显示的网页中的各个部分。网页文件中的内容是文本文件的形式储存的, 我们通过 HTML5 标记要显示在网页上的内容, 我们编写好的 HTML 文件被 Web 浏览器根据标记符的解释按顺序解析文本文件中的内容, 对于标记错误的解释符, Web 浏览器也会继续按照错误的解释符解释文本内容, 开发人员只能通过 Web 浏览器显示出的效果来分析是否有错误的标记符。

接下来介绍 CSS, 全称为层叠样式表, 最早是 1996 年万维网联盟 (W3C) 推出的 CSS1 版本, CSS 是一种用来修饰 HTML 文件或 XML 文件中文本内容的计算机语言^[28], 通过 CSS 使得网页中的内容与其表现形式相互独立, 网页中的内容的表现形式通常放在 <head> 部分或者外部的 CSS 文件中。CSS 提供了各种各样的样式, 如修改颜色、字体大小、页面长宽、页面背景等。另外 CSS 还提供选 CSS 框模型, 它规定了元素框处理元素内容、内边距、边框和外边距的方式。CSS 定位允许你对元素进行定位、CSS 编程规则主要由选择器和一条或者多条声明组成, 选择器大多数情况下是我们要修改的 HTML 文本文件中元素, 每条声明都由一个属性和一个值组成, 这个属性就是你要修改 HTML 元素的样式属性。

最后介绍 JavaScript, JavaScript 是一种直译式脚本语言, 最早是 1995 年 Netscap 推出来最早的 JavaScript1.0 版本, JavaScript 刚推出就获得了巨大成功, JavaScript 主要作用是对用户做出的事件进行响应以及添加动态功能, 通常 JavaScript 脚本文件是通过嵌入 HTML 页面中, JavaScript 主要由三个部分组成: ECMAScript、文档对象模型 (Document Object Model, 简称 DOM)、浏览器对象模型 (Browser Object Model, 简称 BOM)。JavaScript 的数据类型主要分原始变量和引用类型变量两大类。原始类型的变量值存放在栈中, 访问变量时可以直接访问变量值, 原始类型变量主要由 Undefined、Null、Boolean、Number、String 五种。应用类型变量存放在的在栈 (heap) 中的是指针, 它的值实质上存放在堆 (stack) 中, 要通过存放在栈中的指针访问到堆中值, 所有的对象都是引用类型变量, 比如 Boolean、Number、String、Array、Date 等^[29]。JavaScript 正是由于具有面向对象的功能, 使得它具有直观、模块化等优点。

3.3.2 Android 平台介绍

Android 操作平台是由 Google 公司和开放手机联盟(Open Handset Alliance)合作开发,是一种基于 Linux 系统基础之上的开放源代码的、应用在绝大多数移动设备中(如智能手机、智能可穿戴设备、平板电脑等)。现在 Android 最新版本为 2017 年 8 月 21 日发布的 Android Oreo (8.0),最新 Android 操作系统加速了启动应用程序、增加了更多表情符号、增加了对系统字体更多的支持。Android 平台由应用程序层、应用框架层、系统运行层(系统运行时库和其他库)、Linux 内核层四层组成,如图 3-2 所示,由于这四层之间采用软件叠层的方式进行构建,所以相互独立,各层分工明确,这种构建方式有效的保证了层与层之间的低耦合,当下一层发生改变的时候,上一层不受影响。

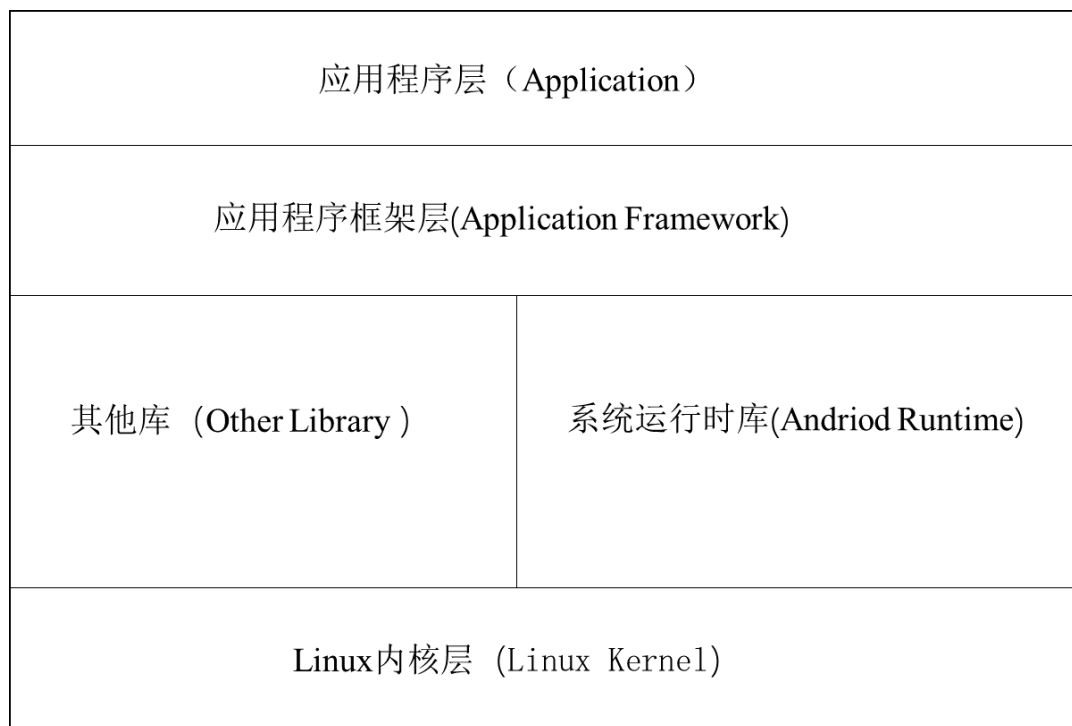


图 3-2 Android 平台架构

对于所有的手机应用程序都会安装在最顶层的应用程序层 (Application) 这些手机应用程序包括我们日常手机中各种应用程序(比如通信录、短信、邮件客户端、音乐播放器、浏览器等),这些日常的应用程序包括系统自带的核心程序和 Android 应用程序开发者开发的应用程序,两者相互独立,互不影响。第二层为应用程序框架层 (Application Framework) 提供了大量的 API (Application Interface),

这些 API 不仅可以供广大 Android 应用程序开发者使用,同时也为 Android 平台的部分核心程序提供了支持。应用程序框架层有活动管理器 (Activity Manager)、窗口管理器 (Window Manager)、内容提供者(Content Providers)、视图管理器(View System)、包管理器 (Package Manager)、资源管理器 (Resource Manager)、位置管理器 (Location Manager)、XMPP 服务这十个部分。任何一个应用程序都可以发布自身的功能模块,其他应用程序则可以使用那些已经发布的功能模块。基于这样的重用机制,用户就可以方便地替换平台本身的各种应用程序。第三层为系统库 (System Library) 和运行时库 (Android Runtime), 系统库包括九个子系统 Surface Manager、Media Framework、SQLite、Open GLE State、Free Type、Web Kit、SGL、SSL、LIBC。运行时库包括内核库和 Dalvik 虚拟机, 内核库中集成了 Java 开发者开发是所需要调用的功能函数及 Android 核心库, Dalvik 虚拟机主要功能是对生命周期的管理、堆栈的管理、各种异常管理及垃圾回收等^[30]。由于 Dalvik 虚拟机效率并不是最高的, Google 公司开发了一种新的运行环境即运行时 (Android Runtime), 新的运行环境较以往的 Dalvik 虚拟机表现了良好的性能, 新的运行环境显著缩短了大多数应用程序的执行时间, 一些应用运行更加流畅 (比如触控更加灵敏及时, 动画效果更加流畅), 另外电池的续航能力也有显著提升。最后一层是为 Linux 内核层, 这一层为 Android 系统的核心层, 该层集成了各种重要的驱动, 比如显示驱动 (Display Driver)、相机驱动 (Camera Driver)、WIFI 驱动 (WIFI Driver)、flash 存储器驱动 (Flash Memory Driver)、USB 驱动 (USB Driver)、电池管理 (Power Manger)、音频驱动(Audio Driver)等。

本文提出的基于 PCA 和 SVM 的人脸识别系统是应用在 Android 智能手机上, 系统的前端设计是 H Builder 平台上完成的, 最后打包成 Apk (Application Package) 安装在 Android 系统中的应用程序层。

3.3.3 H Builder 开发平台介绍

H Builder 开发平台是 D Cloud 公司即数字天堂网络技术有限公司推出的一款支持 HTML5 语言开发 Web 的集成开发环境 (Integrated Development Environment 简称 IDE)。H Builder 平台通过完整的语法提示和代码输入法、可编程代码块大大提高了 HTML、JavaScript、CSS 的开发效率。另外 H Builder 特有的边改边看模

式，可以在编写代码的同时看到实现的效果，H Builder 生态系统同时兼容 Eclipse 插件，这使得它具有十分丰富的 Web 集成开发环境。H Builder 系统平台的开发也使用了 Java、C、Web 和 Ruby 编程语言由于 Java 语言编译效率较低，所以采用了 C 语言启动 H Builder，同时 H Builder 支持手机数据线真机调试。而某些用户信息就使用了 Web 技术，代码块、快捷配置命令脚本这些是采用 Ruby 语言开发的^[31]。另外 H Builder 可以开发跨平台的移动端 App，目前支持 Android 和 IOS 两大主流系统。

基于 H Builder 良好的开发性能和兼容性，本文中提出的移动终端人脸识别系统的 App 前端设计就采用 H Builder 平台开发。

3.3.4 基于 H Builder 平台的 App 前端设计与实现

本文提出的人脸识别系统的 App 前端是在 H Builder 平台开源的 MUI 框架基础之上完成设计的，MUI 框架封装了各种功能模块，这些功能模块存放在 App 项目的各个文件夹中，例如 JS 文件夹中就存放着各种逻辑文件，用来实现用户与前端页面的交互和前端与后台服务器的逻辑实现、CSS 文件夹中存着各种样式变现方法等。我在开发这个人脸识别系统的前端功能模块就是通过调用这些已经存在文件夹中的各类文件。

首先整个前端页面内容是编写在 Uploader.html 文件中，这个 html 文件是用 Html5 语言编写的，HTML5 使用标记标签来描述本文要显示的界面内容^[32]，完整的 HTML 文件由<html></html>这对标签组成，这里要说到的是 HTML 文件中标签都是成对出现的，这对标签中的内容又由<head></head>和<body></body>两个部分组成，<head></head>中的内容可以引用脚本、指示浏览器在哪里可以找到样式表。<body></body>中的内容为整个 HTML 文件中的主体，一个完整的 HTML 文件是由 <html></html>、<head></head>、<body></body>三大部分组成的，其中<body></body>部分为此 HTML 文件的核心部分。

本文提出人脸识别系统的 App 前端功能模块都是在这里实现的。在文件中我们分别调用 mui 框架下已经生成 CSS、JavaScript 文件夹中的函数文件实现整个 App 前端功能实现。

下面分别介绍本文提出的人脸识别系统 App 前端功能模块的具体实现方法以

及整个系统界面的布局安排，App 前端功能模块主要有调用手机摄像头功能、调用手机相册功能、上传已加载的照片功能等功能，下面是实现这些功能的核心代码。

```
<table>
  <tbody>
    <tr>
      <td>
        //调用手机摄像头
        <div class="button button-select" id="anliuwen" onclick="appendByCamera()">
          <a class=" iconfont icon-xiangji_"></a>
        </div>
      </td>
      <td>
        //调用手机相册
        <div class="button button-select" id="anliuwen" onclick="appendByGallery()">
          <a class=" iconfont icon-xiangce2"></a>
        </div>
      </td>
      <td>
        //上传照片
        <div class="button" id="anliuwen" onclick="upload()">
          <a class=" iconfont icon-iconfontshangchuan"></a>
        </div>
      </td>
    </tr>
  </tbody>
</table>
```

接下来讲述本文提出的人脸识别系统 App 界面布局，首先页眉为系统 Logo 及系统名称部分，紧接着第二部分为 App 功能模块按键部分，然后就是显示调用

手机摄像头拍摄的照片或者手机相册中的照片部分，最后页尾为显示的识别结果部分,图 3-3 为系统界面的布局展示。



图 3-3 系统界面的布局展示

3.4 服务器端设计

3.4.1 Tomcat 简介

Tomcat 是源自 Apache 软件基金会 (Apache Software Foundation, 网址为 <http://tomcat.apache.org>) 的 Java servlet 容器 (container) 与 Web 服务器。虽然 Web 服务器是一种根据请求 (比如 Web 浏览用户的发出的某个请求) 而做出响应并给用户展现用户所需的网页的应用程序, 但是 Web 服务器不仅仅只局限于给用户提供静态的 HTML 网页, 它还可以通过运行程序来响应 Web 用户的请求, 然后把响应的动态结果传送到用户的浏览器上。Web 服务器拥有的这种功能也正是 Apache 的 Tomcat 擅长的地方。为什么这么说呢, 因为 Tomcat 不仅包含了 Java servlet 技术, 而且还包含了 Java Server Page (JSP) 技术, 另外, Tomcat 还包含了用各种编程语言编写的传统静态网页和外挂 CGI 程序^[33]。所以 Tomcat 用来做 Web 服务器是一个非常好的选择。

在本文研究的人脸识别系统中,我们将通过在 Eclipse 开发环境下配置 Tomcat 服务器,再通过使用 Ngrok 软件使得手机移动终端也可以通过指定网址访问服务器端进行数据通信。

3.4.2 Tomcat 的安装及在 Eclipse 环境下配置 Tomcat

首先我们需要去从 Tomcat 官网下载 apache-tomcat-8.5.23-windows-x64.zip 压缩包,等待下载完成后,解压到指定的目录下。解压完的内容如图 3-4 所示,

名称	修改日期	类型	大小
bin	2018/3/7 星期三 ...	文件夹	
conf	2018/3/7 星期三 ...	文件夹	
lib	2018/3/7 星期三 ...	文件夹	
logs	2014/2/13 星期...	文件夹	
temp	2018/3/7 星期三 ...	文件夹	
webapps	2018/3/7 星期三 ...	文件夹	
work	2014/2/13 星期...	文件夹	
LICENSE	2014/2/13 星期...	文件	57 KB
NOTICE	2014/2/13 星期...	文件	2 KB
RELEASE-NOTES	2014/2/13 星期...	文件	9 KB
RUNNING.txt	2014/2/13 星期...	文本文档	17 KB

图 3-4 Tomcat 的安装包中的文件

接下来就是配置 Tomcat 的环境变量,首先新建 TOMCAT_HOME 变量,右键计算机,在属性对话框中选择高级系统设置,接着单击高级按钮,选择环境变量,这是会弹出一个新建系统变量的对话框,在变量名那一栏填写 TOMCAT_HOME,在变量值那一栏填写我们下载的 Tomcat 压缩包解压的保存路径,接着新建 CATALINA_HOME 变量,新建 CATALINA_HOME 变量和新建 TOMCAT_HOME 变量是一样的方法,变量值也一样。然后修改变量 Path 路径,在系统变量中找到 Path 变量名,双击或者点击编辑,在末尾添加如下内容“;%TOMCAT_HOME%\bin;% CATALINA_HOME%\lib”,值得注意的是每个变量一定要用英文输入法下的分号隔开。这样我们就配置好了 TOMCAT 服务器。同时你也可以在 Eclipse 开发平台上配置环境变量,首先打开 Eclipse 开发环境,然后在 windows 中下拉菜单栏中点击 preference,接着在弹出的菜单栏中找到 Server 下发的 Runtime Environment ,单击右方的 Add 按钮,选择我们之前按照的 Tomcat

版本，单击 Next；设置 Tomcat 的安装目录；点击 ok 后就完成了设置。

3.4.3 HTTP 网络传输协议及在 App 人脸识别系统中的作用

在介绍 HTTP 传输协议的之前，先讲述一下计算机之间的信息交换的原理，计算机之间的信息交换是按照事先规定好的顺序进行。连接在互联网上的所有计算机都以事先规定好的基础，来处理请求和通信。按照事先按规定好的实质上就是共同签署的协议，协议一般是由称谓格式（Format）的信息构造和信息交换顺序的方法（procedure）所构成^[34]。接下来我们讲述应用在手机 App 中的 HTTP 协议，图是 3-5 是 HTTP 协议的功能流程图。

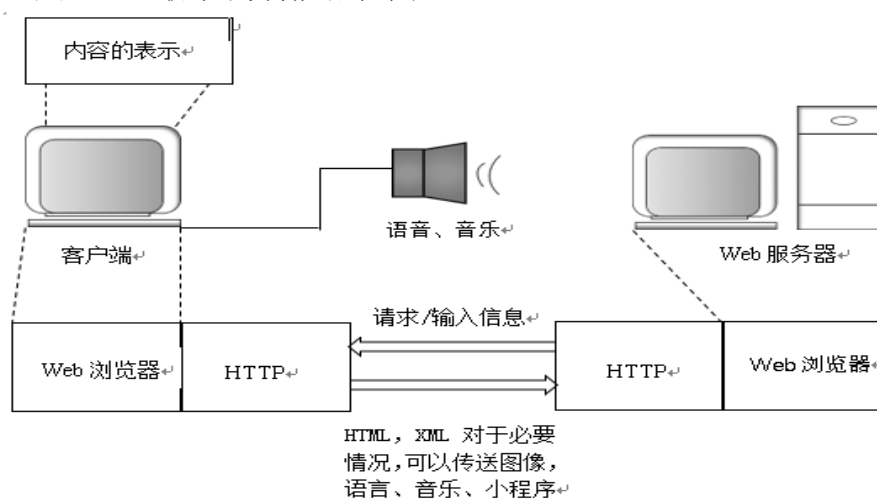


图 3-5 HTTP 传输协议流程图

通过 HTTP 协议来完成前端和后台服务器的数据通信。HTTP（Hyper Text Transfer Protocol）即是超文本传送协议，是针对在客户端和服务端之间的协议，也即是传送超文本的通信上的协议。超文本是文件和文件，或者文件和多媒体信息（比如语言、照片、动画），与此相关联的可能内容的总称。

客户端页面会显示内容，并根据输入的语音和音乐传送到 Web 浏览器页面，Web 浏览器再通过 HTTP 协议将输入的语言和音乐信息传送到 Web 服务器端，服务端处理好再通过 HTTP、XML 协议在传回相应的结果，例如图片、语言、音乐、小程序等。HTTP 协议是 TCP(Transmission Control Protocol)/IP(Internet protocol)协议中位于应用层。HTTP 协议包括客户端发出请求，和服务端端应答，客户端发出的请求信息是由方法和头部构成的，服务器端的应答消息也是由状态和头部构成。其中包含应答代码“200 OK”。应答代码“200 OK”是表示请求被正常处理返回的一

个状态码。如图 3-5 所示，常见的应答码 403 Forbidden、404 Not Found 、502 Bad Gateway 等等，403 Forbidden 应答代码表示服务理解用户的请求，但拒绝处理请求，通常是由于服务器权限问题儿导致的，404 Not Found 应答代码是比较常见的应答代码，表示无法找到用户指定位置的资源，502 Bad Gateway 应答代码表示服务器作为网关或代理的时候，服务器访问下一个服务器时返回非法应答。



图 3-6 请求正常完成在终端显示

图 3-6 为本文中提出的基于 PCA 和 SVM 算法的手机人脸识别系统在手机 App 前端发出识别请求，并在服务器终端完成识别结果，最终通过 HTTP 传输协议将识别结果传回手机前端这一过程完成后在 Ngrok 软件终端上显示的状态码，表示请求完成正常。关于 Ngrok 软件的介绍以及在本文提出的人脸识别系统中的它发挥的作用都会在 3.4 章节详细讲解到。

3.4.4 Eclipse 平台介绍

Eclipse 是一个开放源代码的，可供广大软件开发者开发使用，是基于 Java 语言的可扩展开发平台^[35]。Eclipse 就其本身而言，它只是一个框架和一组服务，可以通过插件组件（Eclipse 附带了一个标准的插件集，比如我们常用的 Java 开发工具（Java Development Kit，JDK））构成一个集成开发环境。Eclipse 在大多数用

户看来，它发挥更多的作用是一个 Java 集成开发环境（Integrated Development Environment，IDE），但是 Eclipse 不仅仅只是一个 Java 集成开发环境，它还可以支持 C/C++、PHP、Android 等编程语言的插件已经可用了。Eclipse 最初由 IBM 公司开发的替代商业软件（Visual Age for Java）的下一代集成开发环境（IDE），截止目前为止，eclipse 在 2015 年 6 月 25 日发布最新的代号为 Mars 的 4.5 版本。

在开发基于 PCA 和 SVM 算法的手机人脸识别系统时，是在 Eclipse 平台下实现了在配置的 Tomcat 服务器以及完成检测并读取手机前端传送的照片，并在调用识别算法识别完成后将识别结果通过 HTTP 协议传回至手机前端等操作。

3.4.5 Ngrok 软件使用介绍及其在 App 人脸识别系统中的作用

Ngrok 是一个反向代理，通过在公共的端点和本地运行的 Web 服务器之间建立一个安全的通道，由于我们常是使用个人电脑访问外网，这里通过 Ngrok 可以使得我们在使用外网时可以访问到个人电脑，也就是反向代理^[36]，图 3-7 是 Ngrok 软件首页。

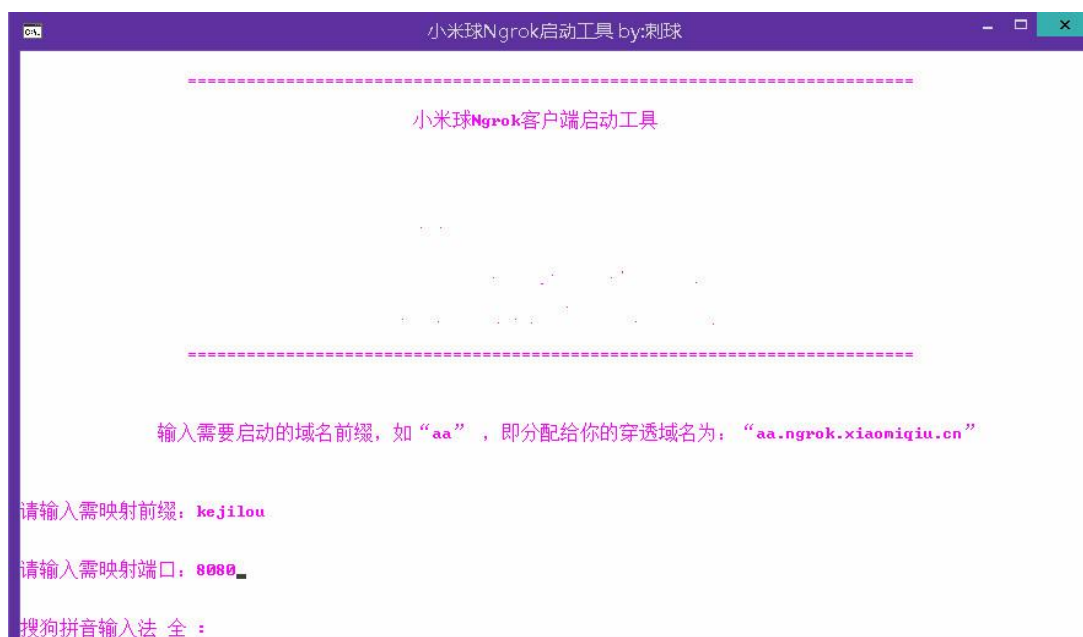


图 3-7 启动 Ngrok

在本文中提出的基于 PCA 和 SVM 算法的手机人脸识别系统中，Ngrok 软件主要起到了手机前端和服务器端的桥梁作用，使得我们手机前端在连接到外网的情况下访问到个人电脑。接下来我们具体介绍一下怎么通过 Ngrok 软件来实现手机前端和个人电脑之间的数据通信的。首先我们下载 Ngrok 客户端压缩包，解压后直接

运行脚本小米球 Ngrok 启动工具 bat, 启动小米球 Ngrok 后, 按照提示输入映射前缀, 这里的映射前缀取得“kejilou”, 这里取的映射前缀为任意取。映射端口为默认 8080。在保证网络通畅的情况下, 当输入完映射前缀和映射端口后, 将会在 Ngrok 的终端页面显示如图 3-8 内容。



图 3-8 成功启动 Ngrok

3.5 本章小结

本章首先介绍了系统的整体设计, 接着分两部分分别介绍了本文提出的基于 PCA 和 SVM 算法的手机人脸识别系统的 App 前端模块设计和服务器端模块设计, 算法调用部分在第二章已经详细介绍了。

首先介绍了设计 App 前端模块设计用到的移动 Web 前端的 Html、JavaScript、CSS 三大技术。然后介绍了 Android 平台的架构分布以及介绍将本文提出人脸识别系统应用在 Android 智能手机上的应用程序层。介绍了 App 前端设计是在 H Builder 开发环境下设计与实现的, 是通过 HTML5 语言编写核心 Uploader.html 文件, 文件中主要编写了实现 App 前端系统界面的布局和系统功能模块的实现。

接着介绍了设计系统的服务器端所需要的软件配置环境, 对 Tomcat 服务器进行了简介。介绍了配置的 Tomcat 服务器是在 Eclipse 开发环境下进行的, 其核心配置文件为 PicService.java 文件, 同时介绍了在配置好 Tomcat 服务器后, 需要使

用 Ngrok 软件代理来解决了 App 前端与服务器端的数据通信的 Ngrok 软件。

第四章 基于 PCA 和 SVM 算法的手机人脸识别系统测试与分析

4.1 系统功能测试

4.1.1 实验设备

一台搭载 Android 操作系统的魅族智能手机^[37]，本文提出的系统也是在 Android 系统上操作的。同时它搭载了 500W 前置摄像头和 1300W 后置摄像头，本文提出的移动终端人脸识别系统在采集人脸图像的时候将会调用手机前后置摄像头。本文提出的移动终端人脸识别系统就是运行在这台手机上、一台联想 G470 笔记本电脑^[38]，中央处理器(CPU)为 Intel(R)Core(TM)i5-2450M，主频 2.50GHz，主要给本文提出的系统提供服务器，同时本文介绍的基于 PCA 和 SVM 人脸识别算法也是在这台笔记本上运行的，该算法是采用 C 语言编写的。实验设备如图 4-1 所示：



图 4-1 实验设备展示

在系统中用于本文提出的基于 PCA 和 SVM 算法的手机人脸识别系统的测试和实现是在利用这些设备完成的。接下来会对本文提出的基于 PCA 和 SVM 算法的手机人脸识别系统的功能进行测试和分析。

4.2.1 系统功能测试与分析

首先验证本文提出的基于 PCA 和 SVM 算法的手机人脸识别系统的上传功能，图 4-2 所示的是在智能手机上系统成功上传人脸的界面和未上传人脸图像的界面。



图 4-2 成功上传和未上传的界面展示

接下来展示的是识别成功的界面，如图 4-3 所示,我们可以从图中可以看到系统界面的最后栏就是自动调用服务器端的人脸识别算法成功识别出扩展的 ORL 人脸数据库中的图像。



图 4-3 成功识别人脸图像

4.2 系统性能测试与分析

本小节首先本文提出的基于 PCA 和 SVM 算法的手机人脸识别系统在自己新建的人脸数据库上进行性能测试与分析。接着也在扩展的 ORL 人脸数据库、Yale 人脸数据库、IMM 人脸数据库和 Face95 人脸数据库上测试了其性能。Yale 人脸数据库是由 15 人脸类，每个人脸类 11 张照片，这 11 张人脸照片变化主要包括光照条件的变化，表情的变化等。IMM 人脸数据库是每个人脸类有 6 张照片，一共 40 个不同人脸类，共计 240 张，Face95 人脸数据库是每个人脸类有 20 张照片。一共 72 个不同人脸类，共计 1440 张。

下面介绍一下新建的人脸数据库，它是通过手机摄像头采集了实验室部分成员和本人的人脸图像，在采集每位成员的照片时保持在同一光照条件，适当改变姿态，在这种条件下，我们共采集 5 位成员人脸图像，每位人员 10 张不同姿态的人脸照片，在采集完成后保存至指定文件夹下，接着我们通过当前文件夹下进入 windows 终端，输入命令行 `dir /b/s *.pgm > att_faces_test.csv`、`dir /b/s *.pgm > att_faces_train.csv` 后，将采集的人脸数据自动生成对应的 CSV 文件，最后我们在 CSV 文件中分别给 5 类人脸图像打上标签，分别记为 0-4 作为本文提出系统测试时的训练集。

表 4-1 系统在自建数据库上测试结果

识别人脸类别		识别算法
		PCA 和 SVM 人脸识别算法
label : 0	识别正确的次数	9
	识别平均所需时间	9.5
label : 1	识别正确的次数	9
	识别平均所需时间	10.1
label : 2	识别正确的次数	9
	识别平均所需时间	9.9
label : 3	识别正确的次数	8
	识别平均所需时间	10.9
label : 4	识别正确的次数	10
	识别平均所需时间	9.7

首先我们对本文提出的基于 SVM 和 PCA 的移动终端人脸识别系统在自己新

建的人脸数据库上进行了系统测试，前面介绍了我们将自己建立的数据库作为训练集，另外我们测试集照片采用实时采集，每位成员测试 10 次，共 50 个测试样本，我们对测试的识别率和平均时间进行了统计，表 4-1 中分别计数了每个成员识别成功的次数和每个成员识别平均所需时间。

根据表 4-1 我们可以计算出系统的识别精度为 90%，系统总的识别平均速度为 10.02s。

接着我们在扩展的 ORL 人脸数据库和 Yale 数据库、IMM 人脸数据库和 Face95 人脸数据库上进行测试，我们在方差贡献率 $\beta=90\%$, $k=160$ 维时，扩展的 ORL 数据库每个类中的前 8 张作为训练集，后 2 张作为测试集，Yale 数据库中每个人脸类中前 5 张作为训练集，后 6 张作为测试集，IMM 人脸数据库中前 3 张作为训练集，最后 3 张作为测试集，Face95 人脸数据库中前 10 张作为训练集，后 10 张作为测试集。

表 4-2 是测试在扩展的 ORL、Yale、IMM、Face95 数据库上系统的识别准确率和平均所需时间。从表 4-2 中可以看出系统在扩展的 ORL 人脸数据库和 Yale 人脸

表 4-2 系统在扩展的 ORL、Yale、IMM、Face95 数据库上的测试结果

人脸数据库	识别方法	
	基于 PCA 和 SVM 的人脸识别算法	
扩展的 ORL	测试集数目	90
	识别正确数目	84
	识别率	93.34%
	平均所需时间	11.6s
Yale	测试集数目	90
	识别正确数目	81
	识别数	90.00%
	平均所需时间	12.4s
IMM	测试集数目	120
	识别正确数目	111
	识别率	92.50%
	平均所需时间	13.6s
Face95	测试集数目	720
	识别正确数目	664
	识别率	92.22%
	平均所需时间	12.8s

数据库、IMM 人脸数据库、Face95 人脸数据库均表现出较好的识别率，在识别时间上在 Face95 人脸数据库上识别速度相对于其他几个数据库较慢，系统整体识别速度在 11.6s 至 13.6s 之间。

综合表 4-1、4-2 可以得出：本文中提出的基于 PCA 和 SVM 算法的手机人脸识别系统在自建的人脸数据库和扩展的 ORL 人脸数据库和 Yale 人脸数据库、IMM 人脸数据库、Face95 人脸数据库上都表现出了良好的性能，初步形成一定的实用性。同时此手机人脸识别系统也可以有效判定上传照片的人物身份。

4.3 本章小结

本章首先介绍了本文提出的基于 PCA 和 SVM 算法的手机人脸识别系统，接着介绍了设计与实现这个移动终端人脸识别系统所需的实验设备，最后测试了此系统的性能（识别准确率和识别平均所需时间）和各个功能模块。

第五章 总结与展望

5.1 总结

本章将对本文的工作内容进行总结与展望，本文在比较基于 Eigenface 的人脸识别算法、基于 LBP 和 PCA 的人脸识别算法和基于 PCA 和 SVM 的人脸识别算法三种算法后，提出了一种基于 PCA 和 SVM 算法的手机人脸识别系统，并在 Android 智能手机和联想 G470 笔记本电脑等实验设备上实现了这种基于 PCA 和 SVM 算法的手机人脸识别系统。本文的工作内容主要由以下几点：

（1）介绍了三大移动 Web 前端技术 JavaScript、CSS、Html，同时也讲述了利用这三大移动 Web 前端技术开发了本文提出的系统的前端功能模块。其次对 Android 平台的系统架构进行了讲解以及在如何在本文提出的系统中应用。还介绍了一款开发移动 Web 前端的开发环境 H Builder。

（2）介绍了 Eclipse 开发环境及其特点，介绍了在 Eclipse 环境下如何配置 Tomcat 服务器。还介绍了使用 Ngrok 反向代理软件，并通过 HTTP 网络传输协议，使得手机 App 端可以在联网的情况下访问服务器端，将 App 前端上传的数据传入到服务器端。

（3）介绍了数字图像处理技术有哪些以及应用，并详细介绍灰度化，降采样图像处理方法，还介绍了其在本文中的系统的应用。接着举例讲述了图像预处理的方法，比如直方图均衡化、均值滤波，还介绍了机器学习中的经典算法主成分分析法（PCA）、支持向量机（SVM）、最近邻学习算法，最后介绍了基于 Eigenface 的人脸识别算法、基于 LBP 和 PCA 的人脸识别算法和基于 PCA 和 SVM 的人脸识别算法。

（4）在比较基于 Eigenface 的人脸识别算法、基于 LBP 和 PCA 的人脸识别算法和基于 PCA 和 SVM 的人脸识别算法三种算法后，设计了一种基于 PCA 和 SVM 算法的手机人脸识别系统，并在智能手机、笔记本电脑等实验设备下进行了设计和实现，并对其系统进行了性能测试和功能测试，测试结果发现识别精度较好，识别平均速度可以达到 10s 左右。

5.2 展望

本论文提出的基于 PCA 和 SVM 算法的手机人脸识别系统虽然在性能上已经达到一定的水平，但还有很大的提升空间，本系统还存在一些不足之处，还需要进一步的改善和提高：

- （1）系统的识别速度和识别精度仍有待提升，还未达到商用级别，还未具备一定的实用性，通过改变算法性能和提升硬件设备来解决系统识别速度和识别精度是下一步需要考虑的工作内容。
- （2）本系统的识别算法对人脸光照变化不均匀鲁棒性不强，这也是后面工作中待解决的问题。
- （3）系统前端模块设计还缺乏更多细节的考虑，比如界面设计不是非常的美观。后期可以通过更专业的前端设计软件来完善，比如柚子（北京）科技有限公司的 API Cloud 软件^[39]、Jetbrains 公司旗下一款 JavaScript 开发工具 Webstorm^[40]。
- （4）本文提出的系统如果想有更多的应用价值，需要结合其他系统进行结合，例如通过可以通过将此系统内嵌至智能手机解锁系统中，实现利用人脸识别来解锁手机屏幕，再例如将此系统内嵌入安检口的闸机上，通过调用本文提出的系统来对出行人员进行安全检查，这样会具有更大的实用意义。
- （5）此系统可以扩展到 Web 端，实现在 Web 端实现人脸识别，也具有一定实用价值。

参考文献

- [1] <http://www.afzhan.com/news/detail/36848.html>.
- [2] 李雪菲, 李志民. 平安城市接入方案研究[J]. 数码世界, 2016(8):39-40.
- [3] 吴巾一, 周德龙. 人脸识别方法综述[J]. 计算机应用研究, 2009, 26(9):3205-3209.
- [4] <http://www.yitutech.com/>
- [5] <https://www.sensetime.com/>
- [6] <http://www.cloudwalk.cn/>
- [7] <https://www.megvii.com/>
- [8] Kaufman G J, Breeding K J. The automatic recognition of human faces from profile silhouettes.[J]. IEEE Trans Smc, 1976, 6(2):113-121.
- [9] Yuille A L. Deformable templates for face recognition[M]. MIT Press, 1991.
- [10] 程永清, 庄永明, 汪华峰,等. 一种有效的人脸识别方法[J]. 自动化学报, 1993, 19(1):54-62.
- [11] Eickeler S, Muller S, Rigoll G. High performance face recognition using pseudo 2-D hidden Markov models[C] Control Conference. IEEE, 2015:3023-3028.
- [12] Bartlett M S, Movellan J R, Sejnowski T J. Face Recognition by Independent Component Analysis[C]// IEEE Transactions on Neural Networks. 2002:1450-1464.
- [13] Zhang Yankun, Liu Chongqing. Face Recognition Based on Support Vector Machine and Nearest Neighbor Classifier[J]. 系统工程与电子技术(英文版), 2003, 14(3):73-76.
- [14] Ahonen T, Hadid A, Pietikainen M. Face Description with Local Binary Patterns: Application to Face Recognition[J]. IEEE Trans Pattern Anal Mach Intell, 2006, 28(12):2037-2041.
- [15] Dutagaci H, Gokberk B, Sankur B, et al. A study on region-based recognition of 3D faces with expression variations[C] Signal Processing Conference, 2007, European. IEEE, 2010:753-757.
- [16] Ying A J, So H J, Kim N C, et al. Face recognition using local statistics of gradients and correlations[C]// Signal Processing Conference, 2010, European. IEEE, 2011:1169-1173.

-
- [17] Bhele S G, Mankar V H. Recognition of Faces Using Discriminative Features of LBP and HOG Descriptor in Varying Environment[C]// International Conference on Computational Intelligence and Communication Networks. IEEE, 2016:426-432.
 - [18] Raj A, Gupta S, Verma N K. Face detection and recognition based on skin segmentation and CNN[C]// International Conference on Industrial and Information Systems. 2016:54-59.
 - [19] Cemil Turan. Robust face recognition via sparse reconstruction vector[C]// International Conference on Electronics, Computer and Computation.2017
 - [20] J. Zeng, X. Zhao, C. Qin and Z. Lin. Single sample per person face recognition based on deep convolutional neural network[C]// International Conference on Computer and Communications (ICCC).IEEE,2017 , pp. 1647-1651.
 - [21] 毛星云, 冷雪飞.OpenCV3 编程入门[M].电子工业出版社, 2015.
 - [22] 冈萨雷斯(美), 阮秋琪(译).数字图像处理(第三版)[M].电子工业, 2011.
 - [23] 周志华.机器学习[M].清华大学出版社, 2016.
 - [24] 哈林顿(美), 李锐(译).机器学习实战[M].人民邮电出版社, 2013.
 - [25] 李航.统计学习方法[M].清华大学出版社, 2012
 - [26] <https://zhidao.baidu.com/question/1898614048624712380.html>.
 - [27] <https://www.cnblogs.com/feng9exe/p/6126236.html>
 - [28] 黄雪琴,耿强,陈显军.基于 CSS+DIV 的自适应宽度网页布局方法[J].计算机与现代化,2014(6).
 - [29] <https://www.2cto.com/kf/201401/275070.html>
 - [30] <https://blog.csdn.net/yue31313/article/details/51894666#t1>
 - [31] <https://www.oschina.net/p/hbuilder>
 - [32] 唐俊开.HTML5 移动 Web 开发指南[M].电子工业出版社.2012.
 - [33] Jason Brittain , 吴豪(译).Tomcat 权威指南(第二版)[M].中国电力出版社.2009.
 - [34] 小泉 修, 王浩(译).Web 技术[M].科学出版, 2004.
 - [35] <https://baike.baidu.com/item/Eclipse/61703>
 - [36] <https://baike.baidu.com/item/ngrok/13986278>
 - [37] <https://www.meizu.com/>

- [38] http://detail.zol.com.cn/series/16/8546_1.html
- [39] <https://app.apicloud.com/>
- [40] <https://baike.baidu.com/item/WebStorm/9392448?fr=aladdin>

致 谢

光阴似箭，又到了学校里盛开木棉花的季节，看着盛开的木棉花，仿佛回到了刚进学校那会儿的情景，那情景历历在目。不知不觉中，今年已经是我在汕头大学的第三个年头，三年的研究生生涯如弹指一挥间。校训碑前的凤凰花树即将绽放出惊艳的凤凰花，它是那样鲜艳夺目，它也预示着我的硕士研究室学习生涯即将结束，开始走入社会，参加工作，为建设美丽的祖国出一份力量，同时仿佛也在告诉我即将要告别这个我生活了三年之久的可爱的母校，要告别我亲爱的老师们和同学们。想到这里，内心充满满满的不舍之情，想到老师们和同学们对我的关心和照顾，内心充满着深深的感激之情，在这里衷心表示感谢，谢谢你们，我敬爱的老师们，我亲爱的同学们。

每每想起母校对我的栽培之恩，心中总是不胜感激和敬畏之情，汕大以“有志、有识、有恒、有为”的校训，时时刻刻提醒着我们要做一个有志、有识、有恒、有为的人，学校注重对学生个人动手能力的培养，为广大学子提供了丰富多样的学习平台。学校组织各种各样的知识竞赛，鼓励学生积极动手将理论与实际结合起来。学校丰富的图书馆馆藏资源让同学们能找到自己想了解的知识，同时安静、舒适、人性化的图书馆建筑也给广大师生们提供了学习的良好空间，学校一流的国内外师资队伍保障了学校高水平的教育质量，同时也给国家输送了一大批优秀人才。学校静谧的水库、充满鸟语花香的荷花池、富有创意的涂鸦墙、有创新性的教学实践大楼这些地方都流下了我们的足迹，学校美丽的环境让我们也更加专心学习。

难忘老师们三年的谆谆教诲，难忘老师们的栽培之恩，在这里，我要特别感谢姜永权教授和范衡教授，是他们引领我走向了学术的殿堂，给了我学术上的专业的指导，让我接受更多的专业知识，拓宽了我的学术视野。在两位老师的谆谆教诲之下，我努力的学习专业知识，积极参加学校组织的学术报告，并参与相应的教学工作，同时我在思想上，积极向上，积极向党组织靠拢。两位老师老不仅是我学术上的老师，更是我人生的老师，他们严谨治学，他们工作

实事求是、求真务实，他们为人和蔼可亲。看到老师们的优良作风，时刻鼓舞着我积极向二位老师靠拢，向他们学习，学习他们的对学术的严谨、对工作的认真。以后进入工作岗位后，我也要牢记老师们的优良作风。

同时在这里我也感谢与我一起度过三年的研究生生涯的实验室的成员们，他们对学术研究充满了激情，在遇到学术难点问题时，他们敢于迎难而上，敢于求真问实，同时孜孜不倦的学习专业知识，在每周的小组讨论和平时的学术探讨沙龙，他们都让我学到了收获颇多，给了我更多的灵感，让我对我面临的难点问题有了更进一步的认识，在私下里，我也和我的实验室成员们打成一片，大家团结合作。在实验室形成了一种积极探讨学术的良好学术氛围。在这里衷心感谢你们，陪我度过最开心、最难忘三年研究生学生生涯，衷心的感谢你们！

在这里还要感谢的是我们电子工程系的全体同学们，在这个大家庭里，大家相亲相爱、互帮互助，相处非常地融洽，同时各种班级活动、学术沙龙让大家的感情更加深入。还记得刚刚进入学校后，对研究生活的懵懵懂懂、不知如何有意义的度过这三年研究生生涯时，是师兄们的细心指导让我慢慢走出迷茫期，是他们传授了做学术研究的经验，是他们指导我如何规划三年学术生活。现如今师兄师姐们已经步入社会或者继续深造，但他们的细心指导和关怀我记忆犹新，在这里祝师兄师姐们工作顺利，心想事成！还有和我一起即将面临着毕业，走入社会的同班同学们，我们在美丽的母校一起朝夕相处三年，在学习和生活上互相帮助、相亲相爱，结下了难忘的友谊，也给我留下来三年难忘的回忆。现如今即将要分别，心中是满满的不舍之情！在这里也祝你们前程似锦，心想事成！

在这里我更要深深的感谢我父母的养育之恩，谢谢你们把我带到这个世界上，谢谢你们理解我的想法，不顾一切的支持我，谢谢你们在生活上给我的无限照顾，谢谢你们不辞辛苦的把我养大成人，让我学习知识，接受教育。谢谢你们，我最爱的爸爸妈妈，每每想到你们对我的爱，我都对未来充满信心，对未来的挑战充满勇气。因为我知道，我有你们二老做我坚强的后盾！

最后我怀着对母校的敬畏之情，对所有关心我的同学、老师、朋友们深深的感激之情，在这里真诚的感谢你们！是你们让我度过了这意义的三年研究生

生涯，是你们给了我无限的勇气，去面对所有的困难和挑战。再次谢谢你们！
最后祝愿汕头大学，我的母校越来越好，为我国输送更多的高精尖人才。

王继彪

2018 年 5 月 汕头

个人简历

姓名：王继彪，于 1993 年 2 月 2 日出生于湖北省孝感市孝昌县，2011 年 9 月至 2015 年 6 月于湖北省武汉市武汉工程科技学院（原中国地质大学江城学院）完成电子信息工程本科学业，2015 年 9 月至今于广东省汕头市汕头大学电子与通信工程专业攻读硕士研究生，研究方向为数字图像处理、模式识别。