

代 号 10701

学 号 0922121551

分 类 号 TP311

密 级 公开

题 (中、英文) 目 基于 Linux 的人脸识别系统的设计与实现

The Design and Implementation of Face Recognition

System Base on Linux

作 者 姓 名 潘 超 指导教师姓名、职务 周 端 教授

学 科 门 类 工 学 学科、专业 计算机应用技术

提交论文日期 二〇一二年一月

西安电子科技大学

学位论文创新性声明

秉承学校严谨的学风和优良的科学道德，本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果；也不包含为获得西安电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中做了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切的法律责任。

本人签名：_____ 日期_____

西安电子科技大学

关于论文使用授权的说明

本人完全了解西安电子科技大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属西安电子科技大学。学校有权保留送交论文的复印件，允许查阅和借阅论文；学校可以公布论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存论文。同时本人保证，毕业后结合学位论文研究课题再撰写的文章一律署名为西安电子科技大学。（保密的论文在解密后遵守此规定）

本学位论文属于保密，在____年解密后适用本授权书。

本人签名：_____ 日期_____

导师签名：_____ 日期_____

摘 要

人脸识别技术作为生物识别技术之一，具有广阔的应用前景。人脸识别技术融合了计算机图像处理技术与生物统计学原理，广泛的应用到出入管理系统、门禁考勤系统、电脑安全防范、人脸照片搜索、访客登记、ATM 智能视频报警系统、监狱智能报警系统等等。

本文对 AdaBoost 算法的人脸检测方法和基于主成分分析方法的特征脸人脸识别方法进行了深入的研究。在 Linux 环境下，在嵌入式开发板上，完成了 Adaboost 算法和特征脸人脸识别方法，实现了实时的人脸检测和识别。

论文完成了嵌入式 Linux 平台的搭建，从交叉开发环境、Bootloader 移植、Linux 内核移植、根文件系统的制作和移植，一直到设备驱动程序的移植全过程，建立了嵌入式软件开发平台。

论文完成了摄像头的视频图像采集，在嵌入式开发板 Tiny6410 上实现了图像中的人脸检测和基于特征的人脸识别。论文分析了基于 Video4Linux 的摄像头视频采集方法，实现了图像采集模块，为人脸检测与识别提供图像来源；采用了 Adaboost 算法的人脸检测方法，实现了人脸检测模块，为人脸识别奠定了基础。分析了基于特征脸的人脸识别原理，实现了人脸识别模块。最后应用 QT 设计了人脸识别系统的人机界面。

实测结果表明，基于 ARM11 平台的嵌入式 Linux 人脸识别系统，在简单光照和背景条件下，正确率较高，实时性较强，方便携带。该工作对人脸检测与识别技术在嵌入式平台的应用有一定参考价值。

关键词：人脸识别 人脸检测 AdaBoost PCA 特征脸

Abstract

Face recognition technology ,as one of biometric technology has broad application prospects. Face recognition technology combines computer image processing technology and bio-statistical theory, widely applied to the access control system, Access Control systems, computer security, face photo search, visitor registration, ATM intelligent video alarm system, intelligent alarm prison system, and so on.

This paper studies the AdaBoost algorithm for face detection based on principal component analysis method for face recognition. In the Linux environment, in the embedded development board, has realized the Adaboost algorithm and face recognition method, realizes the real-time face detection and recognition.

This paper completes the embedded Linux platform. From the cross development environment, Bootloader Linux kernel transplantation, transplantation, root file system production and transplantation, until the device driver of the transplantation process, established the embedded software development platform.

The video camera image acquisition, in the embedded development board Tiny6410 to achieve the image of human face detection and feature based face recognition. This paper analyzes the Video4Linux camera based video acquisition method, realizes the image acquisition module, for face detection and recognition with image source; using Adaboost algorithm for face detection method, has realized face detection module, laid the foundation for face recognition. Analysis of the feature based face recognition principle, realize the face recognition module. Finally the application of QT design of face recognition system.

The measured results show that, ARM11 platform based on embedded Linux face recognition system, in the simple light and background. Correct rate is high. Real time sex is strong, easy to carry. The work has a certain reference value in face detection and recognition technology of embedded platform.

Keywords: Face Recognition Face Detection AdaBoost PCA Eigenface

目 录

摘 要.....	I
Abstract.....	III
目 录.....	V
第一章 绪论.....	1
1.1 课题背景.....	1
1.1.1 嵌入式系统发展.....	1
1.1.2 人脸识别技术简介.....	2
1.2 国内外研究现状.....	4
1.3 本文的工作.....	5
第二章 嵌入式人脸识别系统关键技术	7
2.1 嵌入式系统概述.....	7
2.1.1 嵌入式系统.....	7
2.1.2 嵌入式 Linux 软件系统	8
2.2 嵌入式系统开发流程.....	8
2.2.1 嵌入式软硬件平台选择.....	8
2.2.2 嵌入式 Linux 系统开发	9
2.3 人脸检测算法.....	10
2.3.1 人脸检测的特征依据.....	11
2.3.2 人脸检测的常见方法.....	12
2.4 典型人脸识别算法.....	14
2.4.1 基于几何特征的方法.....	14
2.4.2 基于肤色模型的方法.....	14
2.4.3 基于统计理论的方法.....	15
2.5 本章小结.....	16
第三章 嵌入式人脸识别系统方案	17
3.1 系统总体方案.....	17
3.1.1 系统软硬件平台选择.....	17
3.1.2 系统软件模块划分.....	18
3.2 基于 AdaBoost 算法的人脸检测.....	19
3.2.1 AdaBoost 算法原理描述	19
3.2.2 Haar-like 特征与积分图.....	20
3.2.3 级联分类器设计.....	22
3.3 人脸图像预处理.....	24
3.3.1 彩色图像转换为灰度图像.....	24
3.3.2 人脸图像噪声的消减.....	25
3.3.3 灰度图像直方图均衡化.....	26
3.4 基于 K-L 变换的特征脸人脸识别	27
3.4.1 K-L 变换	27
3.4.2 主成分分析法.....	28
3.4.3 特征脸算法.....	28

3.5 本章小节.....	30
第四章 嵌入式人脸识别系统的实现	31
4.1 嵌入式系统开发环境的搭建.....	31
4.1.1 建立交叉编译环境.....	31
4.1.2 在 Qt Creator 中部署 OpenCV	32
4.2 嵌入式 Linux 系统的编译与移植	33
4.2.1 Bootloader 的分析与选择.....	33
4.2.2 Linux 内核的配置和编译	34
4.2.3 根文件系统的设计.....	36
4.3 基于 Video4Linux 的视频采集.....	37
4.3.1 Video4Linux 数据结构.....	37
4.3.2 USB 摄像头视频采集过程.....	38
4.3.3 内存映射方式截取图像.....	39
4.4 基于 AdaBoost 算法的人脸检测.....	40
4.4.1 Haar 分类器的数据结构.....	40
4.4.2 人脸检测相关函数.....	41
4.4.3 人脸检测功能的实现.....	42
4.5 人脸识别功能的设计与实现.....	43
4.5.1 训练系统框架.....	43
4.5.2 人脸识别训练过程.....	45
4.5.3 基于特征脸人脸识别.....	46
4.6 本章小结.....	47
第五章 系统的测试结果与分析	49
5.1 人脸检测结果与分析.....	49
5.2 人脸识别结果与分析.....	50
5.3 本章小结.....	52
第六章 总结与展望.....	53
6.1 工作总结.....	53
6.2 进一步的展望.....	53
致谢.....	55
参考文献.....	57

第一章 绪论

1.1 课题背景

1.1.1 嵌入式系统发展

随着信息化, 智能化, 网络化的发展, 嵌入式系统技术也将获得广阔的发展空间。美国著名未来学家尼葛洛庞帝预言, 嵌入式系统将是 PC 和因特网之后最伟大的发明^[1]。

嵌入式技术已经全面展开, 目前已成为通信和消费类产品的共同发展方向。在通信领域, 数字技术正在全面取代模拟技术。在广播电视领域, 美国已开始由模拟电视向数字电视转变, 欧洲的 DVB (数字电视广播) 技术已在全球大多数国家推广。数字音频广播 (DAB) 也已进入商品化试播阶段。而软件、集成电路和新型元器件在产业发展中的作用日益重要。所有上述产品中, 都离不开嵌入式系统技术。前途无可计量的维纳斯计划生产机顶盒, 核心技术就是采用 32 位以上芯片级的嵌入式技术。在个人领域中, 嵌入式产品将主要是个人商用, 作为个人移动的数据处理和通讯软件。由于嵌入式设备具有自然的人机交互界面, GUI 屏幕为中心的多媒体界面给人很大的亲和力。手写文字输入、语音拨号上网、收发电子邮件以及彩色图形、图像已取得初步成效。

硬件方面, 不仅有各大公司的微处理器芯片, 还有用于学习和研发的各种配套开发包。目前低层系统和硬件平台经过若干年的研究, 已经相对比较成熟, 实现各种功能的芯片应有尽有。而且巨大的市场需求给我们提供了学习研发的资金和技术力量。

从软件方面讲, 也有相当部分的成熟软件系统。国外商品化的嵌入式实时操作系统, 已进入我国市场的有 WindRiver、Microsoft、QNX 和 Nuclear 等产品。我国自主开发的嵌入式系统软件产品如科银(CoreTek)公司的嵌入式软件开发平台 DeltaSystem, 中科院推出的 Hopen 嵌入式操作系统 (虽然还不够完善)。同时由于是研究热点, 所以我们可以网上找到各种各样的免费资源, 从各大厂商的开发文档, 到各种驱动, 程序源代码, 甚至很多厂商还提供微处理器的样片。这对于我们从事这方面的研发, 无疑是个资源宝库。对于软件设计来说, 不管是上手还是进一步开发, 都相对来说比较容易。这就使得很多生手能够比较快的进入研究状态, 利于发挥大家的积极创造性。

信息时代, 数字时代使得嵌入式产品获得了巨大的发展契机, 为嵌入式市场

展现了美好的前景，同时也对嵌入式生产厂商提出了新的挑战，从中我们可以看出未来嵌入式系统的几大发展趋势^[2]：

1. 嵌入式开发是一项系统工程，因此要求嵌入式系统厂商不仅要提供嵌入式软硬件系统本身，同时还需要提供强大的硬件开发工具和软件包支持。目前很多厂商已经充分考虑到这一点，在主推系统的同时，将开发环境也作为重点推广。当然，这也是市场竞争的结果。

2. 网络化、信息化的要求随着因特网技术的成熟、带宽的提高日益提高，使得以往单一功能的设备功能不再单一，结构更加复杂。这就要求芯片设计厂商在芯片上集成更多的功能来满足应用功能的升级。软件方面采用实时多任务编程技术和交叉开发工具技术来控制功能复杂性，简化应用程序设计、保障软件质量和缩短开发周期。

3. 网络互联成为必然趋势。未来的嵌入式设备为了适应网络发展的要求，必然要求硬件上提供各种网络通信接口。传统的单片机对于网络支持不足，而新一代的嵌入式处理器已经开始内嵌网络接口，同时也需要提供相应的通信组网协议软件和物理层驱动软件。软件方面系统系统内核支持网络模块，甚至可以在设备上嵌入 Web 浏览器，真正实现随时随地用各种设备上网。

4. 精简系统内核、算法，降低功耗和软硬件成本。未来的嵌入式产品是软硬件紧密结合的设备，为了减低功耗和成本，需要设计者尽量精简系统内核，只保留和系统功能紧密相关的软硬件，利用最低的资源实现最适当的功能，这就要求设计者选用最佳的编程模型和不断改进算法，优化编译器性能。

5. 提供友好的多媒体人机界面。嵌入式设备能与用户亲密接触，最重要的因素就是它能提供非常友好的用户界面。图像界面，灵活的控制方式，使得人们感觉嵌入式设备就象是一个熟悉的老朋友。手写文字输入、语音拨号上网、收发电子邮件以及彩色图形、图像都会使使用者获得自由的感觉。

1.1.2 人脸识别技术简介

嵌入式系统的发展为人脸识别的技术的广泛应用带来了便利。人脸识别技术是生物识别技术的一种。生物识别技术^[3-5]主要是指通过人类生物特征进行身份认证的一种技术，这里的生物特征通常具有唯一的（与他人不同）、可以测量或可自动识别和验证、遗传性或终身不变等特点。

生物识别的涵义很广，大致上可分为身体特征和行为特征两类^[6]，表 1.1 对六种常见的生物识别技术优缺点进行了比较。生物识别系统则对生物特征进行取样，提取其唯一的特征转化成数字代码，并进一步将这些代码组成特征模板，当人们同识别系统交互进行身份认证时，识别系统通过获取其特征与数据库中的特征模

板进行比对，以确定二者是否匹配，从而决定接受或拒绝该人。

表 1.1 常见的六种生物识别技术优缺点比较

	优点	缺点
虹膜	便于使用；最可靠的生物识别技术；无需物理的接触；	当前的虹膜识别只是用统计学原理进行小规模的试验；难将设备的尺寸小型化；昂贵的摄像头聚焦；镜头可能产生图像畸变而使可靠性降低；黑眼睛极难读取；需要较好光源。
人脸	面部识别是非接触的，不需要和设备直接的接触；	可使用桌面的视频摄像；面部的位置与周围的光环境都可能影响系统的精确性；改进有赖于提取特征与比对技术的提高；对于因人体面部的变化需要通过人工智能补偿。
签名	公认的身份识别的技术。	随着经验的生长、性情的变化而改变；用于签名的手写板结构复杂而且价格昂贵；很难将尺寸小型化。
声音	非接触，用户可以很自然地接受。	作为行为识别技术，声音变化的范围太大，很难精确的匹配；声音会随着音量、速度和音质的变化而影响结果。
指纹	独一无二的特征；复杂度足以鉴别；速度快，方便。	只是存储从指纹中得到的加密的指纹特征数据；每一次使用指纹时都会在指纹采集头上留下用户的指纹印痕，而这些指纹痕迹存在被用来复制指纹的可能性。
视网膜	极其固定的生物特征；最难欺骗。	未经测试；可能会影响使用者健康，这需要进一步的研究；对消费者而言，视网膜技术没有吸引力；难降低成本。

人脸识别特指利用分析比较人脸视觉特征信息进行身份鉴别的计算机技术。人脸识别是一项热门的计算机技术研究领域^[7,8]，它属于生物特征识别技术，是对生物体（一般特指人）本身的生物特征来区分生物体个体。

广义的人脸识别实际包括人脸图像采集、人脸定位、人脸识别预处理、身份确认以及身份查找等。系统输入一般是一张或者一系列含有未确定身份的人脸图像，以及人脸数据库中的若干已知身份的人脸图像或者相应的编码，而其输出则是一系列相似度得分，表明待识别的人脸的身份。

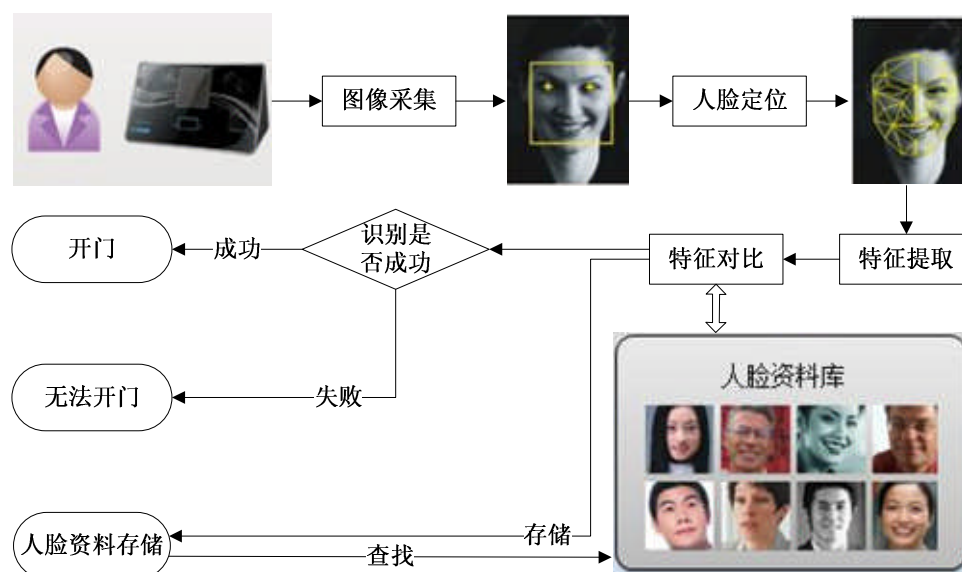


图 1.1 人脸识别在门禁系统中的应用

人脸识别主要用于身份识别，图 1.1 所示为人脸识别在门禁系统中的应用。由于视频监控正在快速普及，众多的视频监控应用迫切需要一种远距离、用户非配合状态下的快速身份识别技术，以求远距离快速确认人员身份，实现智能预警。人脸识别技术无疑是最佳的选择，采用快速人脸检测技术可以从监控视频图像中实时查找人脸，并与人脸数据库进行实时比对，从而实现快速身份识别。

人脸识别的优势在于其自然性和不被被测个体察觉的特点。所谓自然性，是指该识别方式同人类进行个体识别时所利用的生物特征相同。不被察觉的特点对于一种识别方法也很重要，这会使该识别方法不令人反感，并且因为不容易引起人的注意而不容易被欺骗。人脸识别具有这方面的特点，它完全利用可见光获取人脸图像信息，而不同于指纹识别或者虹膜识别，需要利用电子压力传感器采集指纹，或者利用红外线采集虹膜图像，这些特殊的采集方式很容易被人察觉，从而更有可能被伪装欺骗。

然而，人脸识别技术还未完全成熟，识别率受到许多因素地影响，包括背景、光照、姿态、表情、遮挡(头发、眼睛、装饰品以及其它遮挡物等)、人脸尺寸、噪声等等，高性能的人脸识别仍是当前模式识别和人工智能领域的一个热点^[9]。并且，传统的高性能自动识别系统大多基于 PC 平台联机应用，在实际应用中往往对自动识别系统要求有更高的便携性和易用性。

随着嵌入式系统的迅速发展，嵌入式处理器的速度越来越高，性能越来越稳定，功能越来越完善，已经能在很多高要求的环境中应用。基于此因，本文对基于嵌入式人脸识别系统进行了研究。在算法方面，研究并实现了基于 AdaBoost 算法的人脸检测和基于特征脸的人脸识别算法，以提高识别的精度和速度；在硬件方面，设计了基于 ARM(Advanced RISC Machines)的嵌入式系统硬件平台，提高了系统的便携性和易用性。该系统是将人脸识别技术与嵌入式技术相结合的系统，可以做到体积很小，因此可以作为客户端应用于许多场合，它也是适应当代身份认证要求的一种识别系统。

嵌入式系统以及人脸识别技术有着广阔的应用前景，本课题主要对嵌入式系统平台的搭建以及在人脸识别系统中的应用进行研究。

1.2 国内外研究现状

传统的人脸识别技术主要是基于可见光图像的人脸识别，这也是人们最熟悉的识别方式，已有 30 多年的研发历史。但这种方式有着难以克服的缺陷，尤其在环境光照发生变化时，识别效果会急剧下降，无法满足实际系统的需要。解决光照问题的方案有三维图像人脸识别和热成像人脸识别。但目前这两种技术还远不成熟，识别效果不尽人意。

最近迅速发展起来的一种解决方案是基于主动近红外图像的多光源人脸识别技术。它可以克服光线变化的影响,已经取得了卓越的识别性能,在精度、稳定性和速度方面的整体系统性能超过三维图像人脸识别。这项技术在近两三年发展迅速,使人脸识别技术逐渐走向实用化。

目前生物识别技术已广泛用于政府、军队、银行、社会福利保障、电子商务、安全防务等领域^[10]。例如,一位储户走进了银行,他既没带银行卡,也没有回忆密码就径直提款,当他在提款机上提款时,一台摄像机对该用户的眼睛扫描,然后迅速而准确地完成了用户身份鉴定,办理完业务。这是美国德克萨斯州联合银行的一个营业部中发生的一个真实的镜头。而该营业部所使用的正是现代生物识别技术中的“虹膜识别系统”。此外,美国“9.11”事件后,反恐怖活动已成为各国政府的共识,加强机场的安全防务十分重要。美国维萨格公司的脸像识别技术在美国的两家机场大显神通,它能在拥挤的人群中挑出某一张面孔,判断他是不是通缉犯。随着技术的进一步成熟和社会认同度的提高,人脸识别技术将应用在更多的领域。

1、企业、住宅安全和管理。如人脸识别门禁考勤系统,人脸识别防盗门等。

2、电子护照及身份证。这或许是未来规模最大的应用,国际民航组织(ICAO)已确定,从2010年起,其118个成员国家和地区,必须使用机读护照,人脸识别技术为首推识别模式,该规定已经成为国际标准。中国的电子护照计划公安部一所正在加紧规划和实施。

3、公安、司法和刑侦。如利用人脸识别系统和网络,在全国范围内搜捕逃犯。

4、自助服务。如银行的自动提款机,如果用户卡片和密码被盗,就会被他人冒取现金。如果同时应用人脸识别就会避免这种情况的发生。

5、信息安全。如计算机登录、电子政务和电子商务。在电子商务中交易全部在网上完成,电子政务中的很多审批流程也都搬到了网上。而当前,交易或者审批的授权都是靠密码来实现,如果密码被盗,就无法保证安全。但是使用生物特征,就可以做到当事人在网上的数字身份和真实身份统一,从而大大增加电子商务和电子政务系统的可靠性。

1.3 本文的工作

本文的研究重点是基于Linux人脸识别系统,该论文在整体上可以划分为六个部分。

第一章为绪论,分析了本论文的课题背景,对嵌入式系统和人脸识别技术的发展现状进行了简短的介绍,并且明确了本文的主要工作。

第二章研究了嵌入式人脸识别系统涉及到的关键技术,包括嵌入式Linux系统

的软硬件组成，嵌入式 Linux 系统的开发流程，最后对常见的人脸检测和人脸识别算法进行了研究与分析。

第三章提出了嵌入式人脸识别系统的总体方案，包括本系统的软硬件平台的选择，系统软件模块的划分，在此基础上本文选择了 AdaBoost 算法进行人脸检测以及基于 K-L 变换的特征脸算法来进行人脸的识别。

第四章在前面的研究与分析的基础上设计并实现了嵌入式人脸识别系统，首先搭建了系统的交叉编译环境，然后进行了嵌入式 Linux 系统的编译与移植，最后按照人脸图像采集、人脸检测和人脸识别这三个部分详细的说明了系统的实现过程与采用的方法。

第五章对系统的功能进行了测试，并就所得到的结果进行了分析与评价。

第六章为全文的总结与展望。对本课题的内容进行了总结，并对进一步的研究工作进行了展望。

第二章 嵌入式人脸识别系统关键技术

2.1 嵌入式系统概述

嵌入式系统是软件和硬件的综合体，是专用计算机系统，本节介绍了嵌入式系统的特点，重点阐述了嵌入式 Linux 软件系统的组成。

2.1.1 嵌入式系统

根据 IEEE（电气和电子工程师协会）的定义，嵌入式系统是“控制、监视或者辅助装置、机器和设备运行的装置”（devices used to control, monitor, or assist the operation of equipment, machinery or plants）^[11]。从中可以看出嵌入式系统是软件和硬件的综合体，还可以涵盖机械等附属装置。目前国内一个普遍被认同的定义是：以应用为中心、以计算机技术为基础、软件硬件可裁剪、适应应用系统对功能、可靠性、成本、体积、功耗严格要求的专用计算机系统。

嵌入式系统一般由以下几部分组成：嵌入式微处理器、外围硬件设备、嵌入式操作系统和特定的应用程序。嵌入式系统拥有如下几个重要特征^[12,13]：

1. 系统内核小。由于嵌入式系统一般是应用于小型电子装置的，系统资源相对有限，所以内核较之传统的操作系统要小得多。
2. 专用性强。嵌入式系统的个性化很强，其中的软件系统和硬件的结合非常紧密，一般要针对硬件进行系统的移植。同时针对不同的任务，往往需要对系统进行较大更改，程序的编译下载要和系统相结合。
3. 系统精简。嵌入式系统一般没有系统软件和应用软件的明显区分，不要求其功能设计及实现上过于复杂，利于控制系统成本的同时也利于实现系统安全。
4. 高实时性的系统软件(OS)是嵌入式软件的基本要求。而且软件要求固态存储，以提高速度；软件代码要求高质量和高可靠性。
5. 嵌入式软件开发要想走向标准化，就必须使用多任务的操作系统。嵌入式系统的应用程序可以没有操作系统直接在芯片上运行，因而用户必须自行选配 RTOS 开发平台，这样才能保证程序执行的实时性、可靠性，并减少开发时间，保障软件质量。
6. 嵌入式系统开发需要开发工具和环境。由于其本身不具备自主开发能力，开发时往往有主机和目标机的概念。主机用于程序的开发，目标机作为最后的执行机，开发时需要交替结合进行。

2.1.2 嵌入式 Linux 软件系统

从软件的角度来看嵌入式 Linux 软件系统有以下 5 个部分构成：Bootloader、Linux 内核、根文件系统、硬件设备驱动程序以及应用程序^[14]。

1. 引导程序 Bootloader

Bootloader 引导加载程序是系统加电后运行的第一段软件代码。Bootloader 负责初始化硬件设备、建立内存空间的映射图，把系统的软硬件环境调整到合适的状态，为调用操作系统准备好环境。

2. Linux 内核

Linux 内核是 Linux 操作系统的核心程序，指的是一个提供硬件抽象层、磁盘及文件系统控制、多任务等功能的系统软件。Linux 内核向外部提供对系统设备的管理及调用并负责对系统资源分配及共享等，最终目的是为用户程序服务。

3. 文件系统

编译 Linux 内核之后，要想使系统启动起来，还需要有一个根文件系统。根文件系统的作用是存放了各种工具、应用程序、init 程序、必需的库文件等。

4. 设备驱动程序

嵌入式系统通常有许多设备用于与用户交互，像触摸屏、摄像头、LCD 等等。内核通过这些设备驱动程序来操作控制这些设备。

5. 应用程序

应用程序用来实现嵌入式系统预计实现的应用功能。

2.2 嵌入式系统开发流程

嵌入式系统的开发首先要做好软硬件平台的选择，在此基础上将详细介绍嵌入式 Linux 系统的开发流程。

2.2.1 嵌入式软硬件平台选择

按照常规的工程设计方法，嵌入式系统的设计^[15]可以划分为阶段：分析，设计和实现。在分析阶段结束后，开发者通常面临的一个棘手的问题就是硬件平台和软件平台的选择，因为它的好坏直接影响着实现阶段任务的完成。

一、硬件平台的选择

通常，硬件和软件的选择包括处理器、硬件部件、操作系统、编程语言、软件开发工具、硬件调试工具、软件组件等。在上述选择中，处理器往往是最重要的，操作系统和编程语言也是非常关键的。处理器的选择常常会限制操作系统的

选择，操作系统的选择又会限制开发工具的选择。

嵌入式系统的核心部件是各种类型的嵌入式处理器。由于嵌入式系统设计的差异性很大，因此选择是多样化的。**ARM** 是近年来在嵌入式系统有影响力的微处理器架构，**ARM** 的设计非常适用于小型的嵌入式设备。

设计者在选择处理器时要考虑的主要因素包括：处理性能、技术指标、功耗、软件支持工具、是否内置调试工具和供应商是否提供评估板。最后，只要可能，尽量选择使用普通的硬件。在 **CPU** 及架构的选择上，一个原则是：只要有可替代的方案，尽量不要选择 **Linux** 尚不支持的硬件平台。

二、软件平台的选择^[15]

嵌入式软件的开发流程，主要涉及代码编程、交叉编译、交叉连结、下载到目标板和调试等几个步骤，因此软件平台的选择也涉及操作系统、编程语言和集成开发环境三个方面，详细说明包括：操作系统提供的开发工具、向硬件接口移植的难度、内存要求、开发人员是否熟悉此操作系统及其提供的 **API**、是否提供硬件的驱动程序、可剪裁性和实时性能等等。

2.2.2 嵌入式 **Linux** 系统开发

当前，嵌入式开发已经逐步规范化，在遵循一般工程开发流程的基础上，嵌入式开发有其自身的一些特点，如图 2.1 所示为嵌入式系统开发的一般流程。主要包括系统需求分析(要求有严格规范的技术要求)、体系结构设计、软硬件及机械系统设计、系统集成和系统测试，最终得到最后的产品。

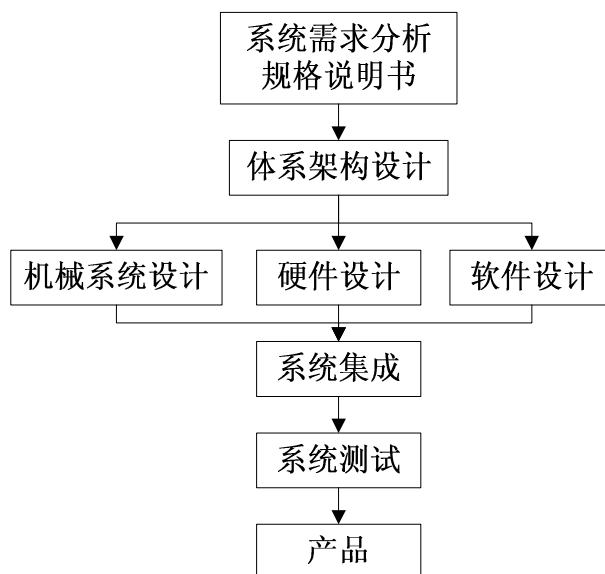


图 2.1 嵌入式系统开发流程

如果在一个嵌入式系统中使用 **Linux** 技术开发^[17]，根据应用需求的不同有不同的配置开发方法，但是，一般情况下都需要经过如下的过程。

(1) 建立开发环境, 操作系统一般使用 Redhat Linux, 选择定制安装或全部安装, 通过网络下载相应的 GCC 交叉编译器进行安装(比如: arm-linux-gcc、arm-linux-uclibc-gcc), 或者安装产品厂家提供的相关交叉编译工具链。

(2) 配置开发主机, 配置 MINICOM, 一般的参数为波特率 115200 Baud/s, 数据位 8 位, 停止位为 1, 无奇偶校验, 软件硬件流控制设为无。MINICOM 软件的作用是作为调试嵌入式开发板的信息输出的监视器和键盘输入的工具。

(3) 建立引导装载程序 BOOTLOADER, 从网络上下载一些开源代码的 BOOTLOADER, 再根据具体芯片进行移植修改。有些芯片没有内置引导装载程序, 但是如果用户购买了厂家的仿真器比较容易烧写 FLASH, 虽然无法了解其中的核心技术, 但对于需要迅速开发自己的应用的人来说可以极大提高开发速度。

(4) 下载已经移植好的 Linux 操作系统, 如 μ Linux、ARM-Linux、PPC-Linux 等, 如果有专门针对所使用的 CPU 移植好的 Linux 操作系统那是再好不过, 下载后再添加特定硬件的驱动程序, 然后进行调试修改, 对于带 MMU 的 CPU 可以使用模块方式调试驱动, 而对于 μ Linux 这样的系统只能编译内核进行调试。

(5) 建立根文件系统, 可以对 BUSYBOX 软件进行功能裁减, 产生一个最基本的根文件系统, 再根据自己的应用需要添加其他的程序。还要根据应用的需要修改根文件系统中的启动脚本, 它的存放位置位于/etc 目录下。根文件系统在嵌入式系统中一般设为只读, 需要使用 mkcramfs、genromfs 等工具产生烧写映像文件。

(6) 建立应用程序的 FLASH 磁盘分区, 一般使用 JFFS2 或 YAFFS 文件系统, 这需要在内核中提供这些文件系统的驱动, 有的系统使用一个线性 FLASH(NOR 型)512KB-32MB, 有的两个同时使用, 需要根据应用规划 FLASH 的分区方案。

(7) 开发应用程序, 可以放入根文件系统中, 也可以放入 YAFFS、JFFS2 文件系统中, 有的应用不使用根文件系统, 直接将应用程序和内核设计在一起, 这有点类似于 ucOS-II 的方式。

(8) 烧写内核、根文件系统和应用程序, 发布产品。

2.3 人脸检测算法

人脸检测^[18]在最近的十几年发展迅速。虽然人脸具有多样性和不规则性, 但是人脸所具有的一些共性特征为我们进行人脸检测提供了依据。人脸包含着丰富的特征信息, 人脸的主要的共性特征有: 几何特征、肤色特征、运动特征等。而依据采用的主要技术方法可将常见的人脸检测方法分为基于先验知识的方法、基于模板匹配的方法和基于机器学习的方法。

2.3.1 人脸检测的特征依据

一、几何特征

基于几何特征的人脸检测主要依靠提取人脸一些低层次的特征，然后对抽取的低层次的特征进行基于知识的分析从而进行检测人脸，人脸的几何图像如图 2.2 所示。

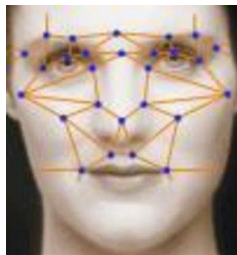


图 2.2 几何特征的人脸识别

按照抽取的特征的规则的不同可以分为：

(1) 对称性规则：绝大多数的物体都存在着对称性的特点，人的面部存在着一定的轴对称性，例如左半边脸和右半边脸；人脸的部分器官也存在着一定的对称性，例如人的左右眼。一种在图像的反对称正双交小分解数据域中，实现多尺度对称变化的算法并将它应用于脸部图像中的主要特征点的定位。

(2) 器官分布规则：虽然人脸各有所异、千变万化，但都遵循着一些普遍适用的规则。人的眼睛、耳、鼻、耳朵、嘴唇等器官的分布具有一定的规律性可言。首先需要对人脸的器官建立模板，然后检测图像中的各个器官所可能存在的位置，将得到的位置点与模板中的进行比较，从而筛选出可能存在的人脸。

(3) 轮廓特征：人脸具有典型的边沿和形状特征，人脸的整个轮廓可以近似的抽象成一个类椭圆，因此粗糙的检测人脸可以通过检测图像空间内是否存在类椭圆形来实现。人脸的各个器官都具有明显的轮廓特征，可以将这些轮廓特征抽象成为线段、弧线、椭圆等几何特征进行描述，采用模板匹配的方法去进行检测。

二、肤色特征

人的皮肤颜色是人脸表面最为显著的特征之一，它也是人脸的一种显著纹理。不同民族的人的面部肤色在颜色空间中的分布是相对集中的，颜色信息在一定程度上可以将人脸同大部分背景区分开。

肤色算法有如下的优点：第一，可以在普通工作站上以帧速率来实现人脸区域分割；第二，肤色分割算法没有使用特殊的脸部特征，因此头部方向和姿态的变化不会影响对于肤色区域的确定；第三，允许被跟踪对象自由活动，减少对环境的限制。但是由于光源的颜色以及光照的角度不同所造成的高亮和阴影等诸多因素的影响，通常需要将皮肤颜色信息与形状分析、运动信息等方法相结合，以

便能够更好地在图像序列中定位和跟踪人脸。

肤色模型出现过的彩色空间有很多种,如: RGB、归一化的 RGB、HIS、YCbCr 等。在去除掉等亮度信息后,肤色在色度空间上具有很好的聚类特性。不同的彩色空间去除亮度信息的方式不同,因此肤色在色度空间上的聚类效果也会有所不同。将其作为人脸检测的粗定位的方法,是相当合适的,具有直观、简单、快速等特点,通过肤色来检测人脸可以避免人脸姿态对检测的影响。

三、运动特征

运动信息是视频和图像序列的一个重要的特征,可以通过比较人脸或者人脸区域相对背景的位移就可以定位人脸,比如利用人脸简单的运动来定位人脸。在人脸在静止的背景下运动的时候,计算相邻帧之间的帧差是检测人脸运动的最简单有效的办法;但是当背景同时也发生变化时,这种方法的效果小好。运动区域的有效分割对于目标分类、跟踪和行为理解等后期处理非常重要。通常将运动特征和肤色模型结合起来运用,可以很好地检测人脸。

2.3.2 人脸检测的常见方法

依据采用的主要技术方法可将常见的人脸检测方法^[19]粗略地分为基于先验知识的方法、基于模板匹配的方法和基于机器学习的方法。

一、基于先验知识的人脸检测

基于先验知识的人脸检测方法是通过对人脸结构的认识,总结出一些描述人脸的特征规则。如人脸面部器官分布大致符合三庭五眼规则;正面人脸可近似为左右对称椭圆形或长方形;人的面部是由颜色较亮的面部皮肤和按一定规则分布在其上虽颜色较暗的器官(眉毛、眼睛、鼻子、嘴巴)构成。人们利用这些先验知识,设计出许多基于知识规则的人脸检测算法。

根据人脸图像的空间、亮度以及灰度分布具有近似的不变性经验规则,人们提出系列的人脸检测算法。**Kanadel**^[20]最早设计了利用人脸几何特征的人脸识别系统。基于几何特征进行人脸图像识别,首先必须自动提取人脸的几何特征。常用的方法有灰度的水平或垂直投影、基于边缘的特征提取等。由于对人脸器官的关键点进行水平或垂直投影后对应波峰或波谷,因此可以用来定位脸部主要器官。

人的皮肤颜色是人脸表面最为显著的特征之一,利用人脸肤色信息进行人脸检测曾经一度成为人脸检测领域的研究热点,**Akamatsu**^[21]利用彩色人脸图像的色度(YIQ 色度坐标)分布信息定位眼睛和嘴巴。**Lee** 等^[22]设计了肤色模型表征人脸颜色,利用感光模型进行复杂背景下人脸及器官的检测与分割。人脸肤色聚类在颜色空间中一个较小的区域,因此可以利用肤色模型有效地检测出图像中的人脸。与其他检测方法相比,利用颜色知识检测出的人脸区域可能不够准确,但如果在

整个系统实现中作为人脸检测的粗定位环节，它具有直观、实现简单、快速等特点，可以为后面进一步进行精确定位创造良好的条件，以达到最优的系统性能。

人脸具有一定的轴对称性，各器官也具有一定的对称性。Zabrodsky^[23]提出连续对称性检测方法，检测一个圆形区域的对称性，依据人脸的五官特性佐证判断是否为人脸。彭进业^[24]等人提出了一种在图像的反对称双正交小波分解数据域中，实现多尺度对称交换的方法，并将它应用于脸部图像中主要特征点的定位。

二、基于模板匹配的人脸检测

模板匹配法通过计算固定目标模板和候选图像区域之间的相关性或差异性，作为匹配准则来搜索人脸，这种方法实现起来比较简单，但由于简单的模板不适应尺寸、姿态和形状的变化，所以当人脸的模式过于复杂而不能用单一模板建模时，可以使用多分辨率、多尺度、多子模板和可变形模板实现模板匹配，以增强适应性和准确性。

Craw 等^[25]提出一种正面人脸的基于形状模板的定位方法，首先利用 Sobel 滤波器提取基于某些约束图像边界，将边界连通作为人脸模板定位头部轮廓，然后根据头部轮廓的位置，进一步定出眼睛、眉毛和嘴唇的轮廓。H.z.Ai^[26]提出了一种基于多相关联匹配的人脸检测方法，文献中使用了双眼模板和不同长宽比的人脸模板，先用双眼模板进行筛选，然后利用不同长宽比的人脸模板确定出人脸区域的位置和范围，最后利用人脸器官的边缘特征进行确认。模板匹配方法是模式识别的传统方法，其思想是：库中存储着已知人脸的若干模板，识别的时候，经过同样变换的输入图像的所有像素点位置与库中所有模板采用归一化相关度量进行匹配识别，来达到分类的目的，完成人脸的识别。

上述为静态模板匹配，但是它存在着对不同表情的人脸鲁棒性差的缺点，针对这一情况，人们提出了弹性模板匹配。Yuille 等^[27]用可变形模板对弹性人脸特征进行预先建模，构建一个能量函数。弹性模板匹配是根据待检测人脸特征的先验的形状信息，定义一个参数描述的形状模型，该模型的参数反映了对应特征形状的可变部分，如位置、大小、角度等，它们最终通过模型与图像的边缘、峰、谷和灰度分布特性的动态地交互适应来得以修正。

三、基于机器学习的人脸检测

在基于先验知识和基于模板匹配的人脸检测法中，人脸特征由专家预先定义；基于学习的方法一般依靠统计分析和机器学习技术来训练人脸样本和非人脸样本，将人脸检测转化为二类模式分类问题，从而达到区分人脸和非人脸目的。

Propp 等人^[28]是最早采用神经网络进行人脸检测的学者之一，他们通过构造一个 4 层(两个隐藏层)神经网络实现人脸检测。神经网络方法较其他人脸图像识别方法有着特有的优势，通过对神经网络的训练可以获得其他方法难以实现的关于人脸图像的规则和特征的隐性表示，避免了复杂的特征抽取工作，并有利于硬件的

实现。

Osuna 等人^[29]最早提出采用 SVM 分类器进行人脸检测，并提出了针对大规模数据集的 SVM 训练算法。支持向量机是统计模式识别领域的一个新的热点，它试图使得学习机在经验风险和泛化能力上达到一种妥协，从而提高学习机的性能。Nefian 等人^[30]采用隐马尔可夫模型检测人脸。HMM 用马尔可夫链来模拟信号统计特性的变化，而这种变化又是间接通过观察序列来描述的，因此马尔可夫过程是一个双重的随机过程。HMM 方法的鲁棒性较好，对表情、姿态变化不太敏感，识别率高。

2.4 典型人脸识别算法

2.4.1 基于几何特征的方法

人脸的几何特征指的是人脸面部器官在几何上体现的特征。对于该类方法，比较典型的有基于先验知识和基于模板的方法：

(1) 基于先验知识的方法

该方法是将人脸面部器官之间关系编码准则化的人脸检测方法。这是一种自顶而下的方法，依据人脸面部器官的对称性、灰度差异等先验知识，制定一系列的准则。当图像中的待测区域符合准则，则被检测为人脸。

(2) 基于模板的方法

基于模板的方法简单来说，就是预先将标准脸部模型通过函数预定义或参数化，分类时对输入图像计算检测区域(如脸部轮廓、眼睛、鼻子以及嘴等)与标准模板的相关值，相关值符合制定的准则就判断检测区域为人脸。这种方法的优点是实现容易，缺点是不能有效地处理搜索大小、姿态以及形状等问题。多分辨率、多重搜索、子模板以及变形模板等技术常被用以解决这些问题。

2.4.2 基于肤色模型的方法

肤色是人脸的重要信息，不依赖于面部的细节特征，对于旋转、表情等变化情况都能适用，具有相对的稳定性并且和大多数背景物体的颜色相区别，因此肤色特征在人脸检测中是最常用的一种特征。肤色特征主要由肤色模型描述，常用的肤色模型有高斯模型、混合高斯模型和直方图模型。

根据标定出肤色区域的近二万幅图片(包含约二十亿个像素)建立三维直方图，在此基础上比较了直方图模型和混合高斯模型，得出了前者检测精度和计算量两者性能上都优于后者的结论。

可以利用了肤色像素的连通性分割区域,使用椭圆拟合了各个区域,根据椭圆长短轴的比率判断其是否为人脸。根据肤色模型提供的像素的似然度,采用从局部最大值处逐渐扩展的方法得到肤色区域,然后使用灰度平均脸模板匹配的方法验证区域内是否有人脸;根据色度的一致性和空间距离将肤色像素聚类成区域,然后逐步归并直到得到符合一定先验知识的椭圆区域为止,最后检查区域中眼睛、嘴巴等特征形成的暗区或空洞,以确定是否为人脸。

2.4.3 基于统计理论的方法

基于统计的方法将人脸检测问题转化为模式识别的二分类问题,即利用统计分析与机器学习的方法来寻找出人脸样本与非人脸样本各自的统计特征,继而构建分类器,使用分类器完成人脸检测。主要包括:子空间方法、神经网络方法、支持向量机方法、隐马尔可夫模型方法以及 **Boosting** 方法。

(1) 子空间方法

子空间方法也称为基于特征空间的方法,此类方法将人脸区域图像变换到某一特征空间,根据其在特征空间中的分布规律划分“人脸”与“非人脸”两类模式。子空间方法主要包括两类:主分量分析(PCA)方法和线性判别分析(LDA)方法。PCA方法是根据图像的统计特性进行正交变换(KL 变换),以消除原来各分量之间的相关性,取变换后所得到的最大的若干个特征向量来表示原来的图像,保留了原图像差异的最大信息,这若干个特征向量就称为主分量。

(2) 人工神经网络方法

人工神经网络方法是通过训练一个网络结构,把模式的统计信息隐含在神经网络的结构和参数之中。在人工神经网络的方法中,系统分为两个阶段:第一阶段是基于神经网络的分类器。该分类器的输入是规定尺寸的检测区域,由此来判断检测区域是否是人脸;第二阶段是合并重复检测并判别。由于训练样本以及分类器等因素,在图像中以人脸为中心的一定区域内会产生重复检测。此阶段使用了一个单层的神经网络对重复检测进行合并,并使用一个多层的神经网络对检测结果进行判决。

(3) 支持向量机方法

支持向量机是基于结构风险最小化原理的统计学习理论上发展起来的。多数训练分类器的方法(如神经网络)是基于经验风险最小化(ERM),支持向量机采用的却是结构风险最小化(SRM),目的是为了减小期望泛化误差边界的上限。

(4) 隐马尔可夫模型方法

隐马尔可夫模型前提假设是模型可被定义为一个参数化的进程。该进程的参数可通过精确的方法估计出来。首先决定隐态并形成一个人脸 HMM 模型,然后通过

学习来自样本集的各个状态间的转移概率来训练 HMM，每个样例由一系列观测值表征。HMM 训练之后，观测值的输出概率就决定了它属于哪一类。

(5) Boosting 方法

该方法使用 Harr-like 特征表示图像，引入“积分图”概念，提高特征的计算速度，采用 AdaBoost 方法选择少量特征组成强分类器，使用“Cascade”策略提高人脸检测速度。在取得较好检测性能的同时，实现了人脸的实时检测。

2.5 本章小结

本章首先介绍了嵌入式系统的定义与特点，重点说明了嵌入式 Linux 软件系统的组成；然后介绍了嵌入式系统的开发流程，包括嵌入式系统软硬件平台的选择以及嵌入式 Linux 系统的开发流程；随后分析了人脸检测算法涉及的几何特征、肤色特征和运动特征等特征依据，研究了常见的几种人脸检测算法：基于先验规则的方法、基于模版匹配的方法、基于外观形状的方法和基于特征的方法；最后又分别介绍了常见的人脸识别算法，分别是基于几何特征、肤色模型和统计理论的人脸识别方法。

第三章 嵌入式人脸识别系统方案

3.1 系统总体方案

嵌入式人脸识别系统的总体方案包括系统软硬件平台的选择和系统软件模块的划分两个部分。本文选择了友善之臂的 Tiny6410 开发板作为硬件平台，将系统的软件部分按照功能划分为五个模块，现分别对两个部分进行详细的说明。

3.1.1 系统软硬件平台选择

嵌入式人脸识别系统的整理架构如图 3.1 所示，包括基于 ARM11 的 Tiny6410 开发板，选择的操作系统内核版本为 Linux2.6.36，基于 Qt4.7.0 的图形用户界面，最终实现嵌入式人脸识别系统。首先就系统的软硬件平台如何选择进行说明。

嵌入式人脸识别系统 (Application)		
FS文件系统	图像用户界面 (QtE-4.7.0)	系统管理接口
实时操作系统内核 (Linux2.6.36)		
板级支持包 (BSP)		
硬件层 (Tiny6410开发板)		

图 3.1 嵌入式人脸识别系统整体架构

本文采用的硬件平台是广州友善之臂公司的以 ARM11 芯片(三星 S3C6410)作为主处理器的嵌入式核心板的 Tiny6410 开发板，如图 3.2 所示。其 CPU 基于 ARM1176JZF-S 核设计，内部集成了强大的多媒体处理单元。

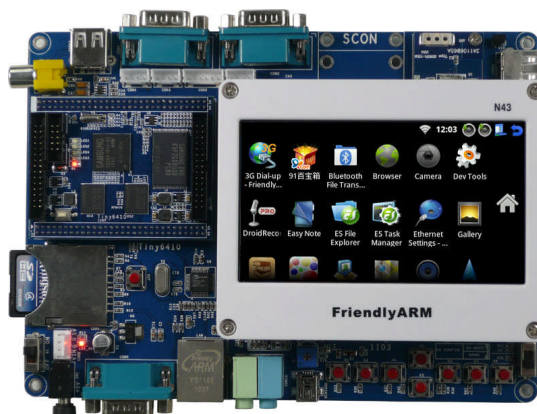


图 3.2 Tiny6410 开发板接 4.3 寸 LCD

Tiny6410 采用高密度 6 层板设计, 尺寸为 64 x 50mm, 它集成了 128M DDR RAM, 256M/1GB SLC Nand Flash 存储器, 采用 5V 供电, 在板实现 CPU 必需的各种核心电压转换, 还带有专业复位芯片, 通过 2.0mm 间距的排针, 引出各种常见的接口资源。Tiny6410 核心板资源特性如图 3.1 所示。

表 3.1 Tiny6410 核心板资源特性

CPU 处理器	Samsung S3C6410A, ARM1176JZF-S 核, 主频 533MHz 最高 667Mhz
DDR RAM 内存	在 128M/256M DDR RAM, 32bit 数据总线
FLASH 存储	在板 2GB MLC(三星 2 代)Nand Flash, 掉电非易失, (可选 256M/1GB SLC Nand Flash)
接口资源	2 x 60 pin 2.0mm space DIP connector 2 x 30 pin 2.0mm space DIP connector
在板资源	4 x User Leds (Green)、10 pin 2.0mm space Jtag connector Reset button on board、Supply Voltage from 2V to 6V
PCB 规格尺寸	6 层高密度电路板, 采用沉金工艺生产 64 x 50 x 12 (mm)
操作系统支持	Linux2.6.38 + Qtopia-2.2.0 + Qtopia4-Phone + QtE-4.7.0 WindowsCE.NET 6.0(R3) Android 2.3.2 Ubuntu-0910

嵌入式操作系统方面我们选择嵌入式 Linux, 内核版本为 Linux2.6.36, 该版本内核同时集成了 V4L, 提供了在 linux 内核中关于视频设备的 API 接口, 涉及开关视频设备、采集并处理视频图像信息; 配合的图形用户界面为 Qt4.7.0, QT 是完全面向对象的, 很容易扩展, 并且允许真正地组件编程, 具有优良的跨平台特性、面向对象、丰富的 API、大量的开发文档等优点。

3.1.2 系统软件模块划分

本嵌入式人脸识别系统按照功能可以划分为五个模块: 人脸图像采集模块、人脸图像预处理模块、人脸检测模块、人脸识别训练模块和人脸识别模块。

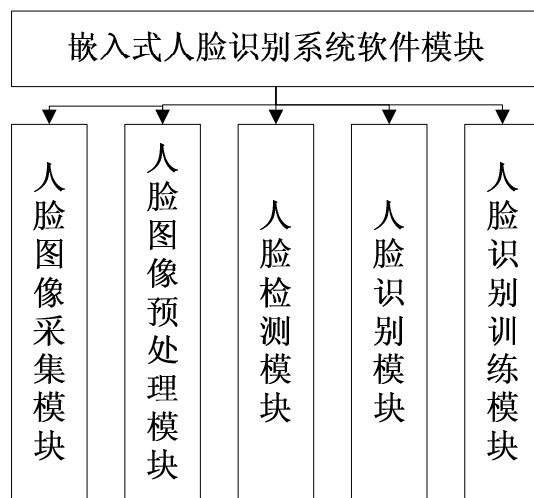


图 3.3 嵌入式人脸识别系统软件模块

现将五个模块的主要功能分别加以说明：

(1) 人脸图像采集模块：利用 USB 摄像头采集人脸视屏，利用内存映射的方式获取 YUV 格式的图片，转换成 QImage 格式并实时显示在 LCD 屏上，再将其转换成 IplImage 格式，以供下一步人脸检测或者人脸识别。摄像头可获取多幅图像以提高后面人脸识别准确度和可靠性。

(2) 人脸图像预处理模块：首先将彩色图像转换为灰度图像，经过人脸图像噪声消减，最后将得到的灰度图像直方图均衡化。

(3) 人脸检测模块：搜索经过预处理后的图像，寻找人脸区域，检测到人脸分类器返回为 1，否则返回为 0。为了检测整副图像，可以在图像中移动搜索窗口，检测每一个位置来确定可能的目标。为了在图像中检测未知大小的目标物体，用不同比例大小的搜索窗口对图片进行几次扫描。

(4) 人脸识别训练模块：对于检测到的人脸图片，由训练者判断是否加入训练集。当训练集图片足够后，点击训练按钮，即利用基于特征脸的主成分分析方法处理训练得到的人脸图片，生成对应的 XML 文件。

(5) 人脸识别模块：获取人脸区域图像后，提取人脸特征，读取前面人脸识别训练得到的 XML 文件，使用 PCA 方法将图像特征与已经存在的人脸数据库比对，人脸图像与数据库中的已知人脸进行比对，如果人脸图像存在并且可信度超过阈值，则在屏幕上显示出人名。

3.2 基于 AdaBoost 算法的人脸检测

在进行人脸识别之前，首先要检测人脸并且标定人脸的位置区域。近年来出现了大量的人脸检测方法，其中 Paul Viola 和 Michael Jones 于 2001 年提出的 Adaboost 算法^[31]是第一个实时的人脸检测算法，从根本上解决了检测的速度问题，同时具有较好的识别效果。

3.2.1 AdaBoost 算法原理描述

AdaBoost 算法是一种迭代方法。它的思想主要是把一系列的弱分类器级联组合成一个强分类器。算法的主要过程是：首先给出样本集合，然后对该样本集合进行循环操作，每次循环首先得到一个假设，然后计算该假设的错误率。根据该假设的错误率改变每个样本的权重进入下一个循环。在具体实现上，首先令每个样本的权重都相等，在此样本分布下训练出一弱分类器；然后根据这个分类器，来提高被它错误分类的样本点的权重，并降低被正确分类的样本的权重，这样分类错误的样本就被突出，我们得了一个新的样本分布。然后，权重更新后的样本

集被用来训练下一个分类器，依次类推，我们经过 T 次这样的循环，得到了 T 个弱分类器，把这 T 个弱分类器按一定权重累加起来，就得到了我们最终想要的强分类器。AdaBoost 算法针对两类问题的训练过程如下^[32]：

- (1) 我们可以假定 X 表示样本空间， Y 表示样本类别的标识集合，由于是二类的问题，可以假定 $Y=\{1, -1\}$ ，分别表示正样本和负样本；
- (2) 训练样本集合 $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ ， $x_{i \in X}, y_i \in \{-1, 1\}$ ， $i = 1, 2, 3, \dots, N$ ；
- (3) 循环次数 T ；
- (4) $D_j(i)$ 表示在第 j 次循环中样本 d_i 的权重， $D_i(i) = 1/N$ ；
- (5) 循环 for $t=1, 2, \dots, T$ 。
 - ①根据分类 $D_t(i)$ 训练分类器；
 - ②获得分类结果： $h_t \in - \rightarrow R$ ；
 - ③求出错误率： $\varepsilon_t = \sum D_t(i) |H_t(i) - Y_i| AC$ ；
 - ④更新： $D_{t+1}(i) = D_t(i) \beta_t^{1-e_i} / Z_t$ ，其中如果样本 x_i 被正确分类，则 $e_i = 0$ ，否则 $e_i = 1$ ； $\beta_t = \varepsilon_t / (1 - \varepsilon_t)$ ， Z_t 是归一化因子（使 D_{t+1} 之和为一，符合概率分布的要求）
- ⑤输出： $H(x) = \text{sign}(\sum_{t=1}^T a_t h_t(x))$ ， $a_t = \log(1/\beta_t)$ ， $\beta_t = \varepsilon_t / (1 - \varepsilon_t)$ ，循环结束。

3.2.2 Haar-like 特征与积分图

Haar-like 特征最早是由 Papageorgiou 等应用于人脸检测。Papageorgiou^[33]在针对正面人脸和人体检测问题的研究中使用 Haar 小波基函数，他们发现标准正交 Haar 小波基在应用上受到一定的限制，为了取得更好的空间分辨率，他们使用了 3 种类型的 3 种形式的特征。Viola^[34,35]等在此基础上作了扩展，使用 2 种类型 4 种形式的特征。3 种类型如图所示，A 和 B 为 2-矩形特征、C 为 3-矩形特征、D 为 4-矩形特征。为了提高人脸检测系统的性能，Rainer Lienhart 等^[36]将矩形特征进行了推广，引入了 45 度旋转矩形特征，使用了更多的 Haar 矩形特征作为分类的依据。

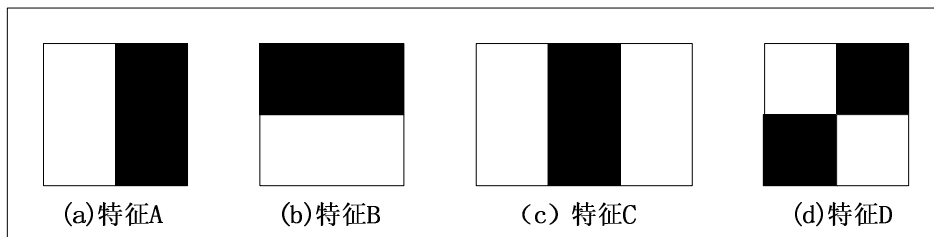


图 3.4 特征原型

对于图中的 A, B 和 D 这类特征, $Sum_{\text{白}}$ 表示白色区域像素和, $Sum_{\text{黑}}$ 表示黑色区域像素和, 特征数值计算公式为: $V = Sum_{\text{白}} - Sum_{\text{黑}}$; 而对于 C 来说, 计算公式如下: $V = Sum_{\text{白}} - 2 \times Sum_{\text{黑}}$; 之所以将黑色区域像素和乘以 2, 是为了使两种矩形区域中像素数目一致。

由于训练样本通常有近万个, 并且矩形特征的数量非常庞大, 如果每次计算特征值都要统计矩形内所有像素之和, 将会大大降低训练和检测的速度。因此, 引入了一种新的图像表示方法-积分图像, 矩形特征的特征值计算, 只与此特征矩形的端点的积分图有关, 所以不管此特征矩形的尺度变换如何, 特征值的计算所消耗的时间都是常量。这样只要遍历图像一次, 就可以求得所有子窗体的特征值。

积分图的定义为:

$$ii(x, y) = \sum_{x' \leq x} \sum_{y' \leq y} I(x', y') \quad (3-1)$$

式(3-1)中 $I(x', y')$ 表示点 (x', y') 的像素值。

为了节约时间, 减少重复计算, 则图像 I 的积分图可按如下递推公式计算:

$$\begin{cases} s(x, y) = s(x, y-1) + ii(x, y) \\ ii(x, y) = ii(x-1, y) + s(x, y) \end{cases} \quad (3-2)$$

这样就可以进行 2 种运算:

(1) 任意矩形区域内像素积分。由图像的积分图可方便快速地计算图像中任意矩形内所有像素灰度积分。如下图所示, 点 1 的积分图像 ii_1 的值为(其中 Sum 为求和): $ii_1 = Sum(A)$ 。

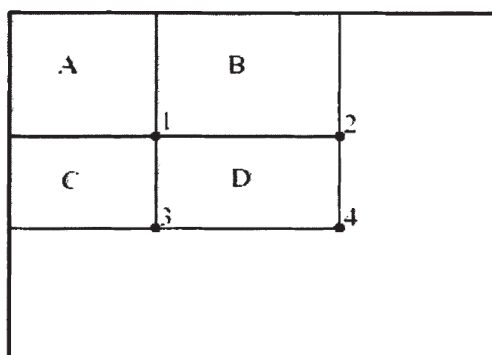


图 3.5 矩阵和计算

同理, 点 2、点 3、点 4 的积分图像分别为:

$$ii_2 = Sum(A) + Sum(B) \quad (3-3)$$

$$ii_3 = Sum(A) + Sum(C) \quad (3-4)$$

$$ii_4 = Sum(A) + Sum(B) + Sum(C) + Sum(D) \quad (3-5)$$

矩形区域 D 内的所有像素灰度积分可由矩形端点的积分图像得到:

$$Sum(D) = ii_4 - (ii_2 + ii_3) \quad (3-6)$$

(2) 特征值计算

矩形特征的特征值是两个不同的矩形区域像素和之差，可以计算任意矩形特征的特征值，一下图中特征原型 A 为例说明特征值的计算。

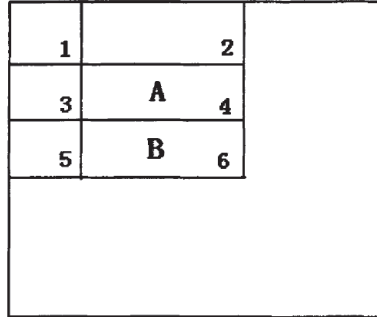


图 3.6 矩形特征值的计算

如图所示，该特征原型的特征值定义为： $Sum(A) - Sum(B)$ 。则有： $Sum(A) = ii_4 + ii_1 - (ii_2 + ii_3)$ ； $Sum(B) = ii_6 + ii_3 - (ii_4 + ii_5)$ ；所以此类特征原型的特征值为： $(ii_4 - ii_3) - (ii_2 - ii_1) + (ii_4 - ii_3) - (ii_6 - ii_5)$ 。由此可见，矩形特征的特征值计算只与此特征端点的积分图有关，而与图像坐标值无关。对于同一类型的矩形特征，不管特征的尺度和位置如何，特征值的计算所耗费的时间都是常量，而且都只是简单的加减运算。其它类型的特征值计算方法类似。

3.2.3 级联分类器设计

通过 Adaboost 算法训练得到的分类器。几乎可以检测出所有的人脸训练样本。但仍然会有一些非人脸样本(负样本)被误判为人脸。采用级联分类器，每层都可以排除一些非人脸样本。这样可以在保证检测率的基础上，大大降低误检率。

一个弱分类器 $h(x, f, p, \theta)$ 由一个特征 f ，阈值 θ 和指示不等号方向 p 组成：

$$h(x, f, p, \theta) = \begin{cases} 1 & pf(x) < p\theta \\ 0 & \text{其他} \end{cases} \quad (3-7)$$

其中 x 相当于某矩形的图像窗口。但是这样得到的单个分类器是一个弱分类器。算法采用 AdaBoost 把多个弱分类器级联合成一个强分类器^[37]。具体的学习算法如下所示：

(1) 设包含 n 个训练样本的训练样本集 $(x_1, y_1)(x_2, y_2), \dots, (x_n, y_n)$ ， x_i 属于是输入的样本， $y_i \in \{1, -1\}$ ， $y_i = 1$ 表示样本是人脸样本（正样本）， $y_i = 0$ ，表示其样本是非人脸样本(负样本)。

(2) 初始化权重 $w_{1,i} = 1/2m$ （对应正样本）或 $w_{1,i} = 1/2l$ （对应负样本），其中 m 为正样本数， l 为负样本数， $m + l = n$ 。

(3) For $t = 1, 2, 3, \dots, T$

①归一化权重, 使得 $w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^n w_{t,i}}$;

②选择一个最佳的分类器, 使其拥有最小的分类误差:

$$\varepsilon_t = \min_{f,p,\theta} \sum_i w_i |h(x_i, f, p, \theta) - y_i|$$

③定义 $h_t = h(x, f_t, p_t, \theta_t)$, 其中 f_t, p_t, θ_t 是 ε_t 取得的值;

④更新权值 $w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$, 其 $e_i = 1$ 表示 x_i 被正确分类, $e_i = 0$ 表示 x_i 被错误

分类; $\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$ 最终的强分类器:

$$c(x) = \begin{cases} 1 & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t, \alpha_t = \log \frac{1}{\beta_t} \\ 0 & \text{其他} \end{cases} \quad (3-8)$$

为了进一步提高系统检测的速度, 算法采用了一种瀑布式级联分类器^[38]的检测方式。对每一层使用 AdaBoost 训练直到得到最小的误检率。如果误检率高于目标, 则引入新的层。每一层在训练时都只针对通过上层分类器的样本进行训练。与前文提到的思想相似。整个框架的目的就是为了尽快缩小候选区域。瀑布式级联分类器^[39]的结构如图所示。

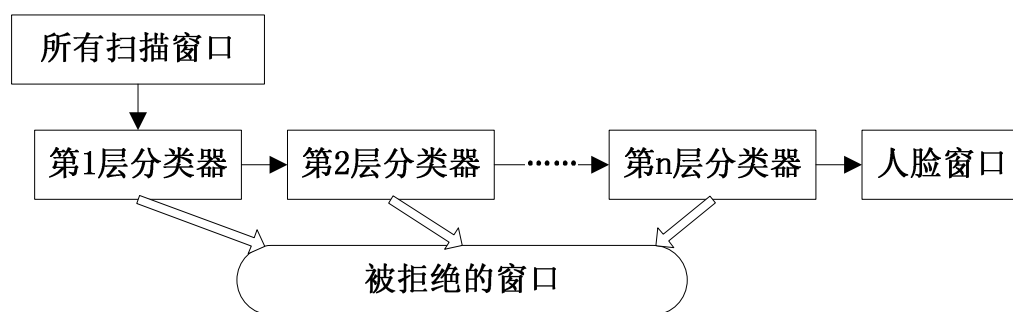


图 3.7 级联分类器检测流程图

级联分类器的输入是待检测图像子窗口。每个待测的窗口都要经过这种级联分类器的检验。对于一个待检测图像样本, 它首先将被送到第一层的分类器, 如要该图像被分类器判别为非人脸, 则检测过程结束: 否则将被送到下一层的分类器去继续进行判断。这样, 许多明显不是人脸的窗口在前几层就被判为非人脸。不需要经过后几层分类器的检验。考虑到进行人脸检测的时候大多数待检测图像为非人脸。因而这样就提高了分类器整体的检测速度。

3.3 人脸图像预处理

人脸图像的预处理分为三个阶段：彩色图像转化为灰度图像、人脸图像噪声消减和直方图均衡化。

3.3.1 彩色图像转换为灰度图像

人脸图像的特征提取和识别实在灰度图像上进行的，因此首先需要把彩色图像转换为灰度图像。

一般数据中获取到的图像为 RGB 彩色图像，RGB 三基色是加性三基色，通过对红(R)、绿(G)、蓝(B)三个颜色通道的变化以及它们相互之间的叠加来得到各式各样的颜色。RGB 颜色模型基于笛卡尔坐标系统，三维空间的三个轴分别对应 R、G、B 三基色。任何颜色都能用三维空间中的第一个象限的一个点表示，该点的三维坐标即为其 R、G、B 的颜色值。在 RGB 颜色空间的原点上，三基色均为 0（没有亮度），即原点对应黑色；而在离原点最远的顶点上，三基色都达到其最大值（最高亮度），表现为白色。因此，从黑到白的灰度值分布在从原点到离原点最远顶点的对角线上，而彩色立方体内的其余各点则对应着不同的颜色，可用从原点到该点的矢量表示。

灰度也可认为是亮度，简单的说就是色彩的深浅程度；灰度图是指是包含亮度信息，不包含色彩信息的图像。把黑-灰-白连续变化的灰度值量化为 256 个灰度级，灰度值的范围为 0~255，表示亮度从深到浅，对应图像中的颜色为从黑到白。

灰度图的使用具有其方便性：首先，图像数据即调色板索引值也就是实际的 RGB 亮度值；另外因为是 256 个级别，所以图像数据中的一个字节代表一个像素；如果是彩色的 256 色图，图像处理后可能产生不属于这 256 种颜色的新颜色，灰度图像则易于处理且大多数经典的图像处理方法都基于灰度图；再者，利用彩色图像的颜色信息进行识别往往受到复杂背景的影响，存在很多缺陷，因而人脸识别一般以灰度图像为研究对象。

下面给出彩色图像到灰度图像的转换公式：

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.148 & -0.289 & 0.437 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (3-9)$$

其中分量 Y 为颜色的明视度，即亮度，也就是图像的灰度值， U 和 V 是色调，处理时用不到这两个分量，后面处理的数据，都是针对灰度值分量 Y 进行处理的。本文中采用式(3-10)近似的来计算彩色图像灰度化后的灰度值：

$$gray = 0.3 \times R + 0.59 \times G + 0.11 \times B \quad (3-10)$$

3.3.2 人脸图像噪声的消减

在人脸图像采集过程中,图像常常被不同程度的强度随机信号所污染,而这些噪声对于人脸识别有较大的影响,这就要求在人脸识别前,对输入图像进行噪声消除处理,尽可能减小噪声的影响。一些常见的噪声有椒盐噪声、脉冲噪声、高斯噪声等。

中值滤波是最常用的滤波方法之一,是抑制噪声的非线性处理方法,人脸图像预处理中通常使用二维中值滤波方法。中值滤波方法主要功能是让与周围像素值的差较大的噪声值取为邻域像素的中间值,所以它可以去除孤立的噪声点。同时,由于中值滤波不是简单的取平均值,故产生的模糊比较小,所以它能在抑制噪声的同时能较好的保护边界,在一定的条件下可以克服线性滤波器带来的图像细节模糊。由于在实际运算过程中不需要图像的统计特性,因此这也带来不少方便。因此,中值滤波具有运算简单,实现方便,较好的保护边界的特点,可以消除孤立的噪声点,对滤除脉冲干扰及图像扫描噪声最为有效。

中值滤波的窗口尺寸对滤波器效果影响很大,因为随着窗口大小的不同,中值也有可能不同。二维中值滤波输出公式:

$$g(x,y) = \text{med}\{f(x-k, y-l), (k,l \in W)\} \quad (3-11)$$

其中 $f(x,y)$, $g(x,y)$ 分别为原始图像和处理后图像。 W 为二维模板,通常为 2×2 , 3×3 区域,也可以是不同的形状,如线状、圆形、十字形、圆环等。



图 3.8 原始图像及用 3×3 模板进行中值滤波后的图像。

考虑到运算速度与效果两方面,本文中采用的是大小为 3×3 的滑动模板。该方法实现的具体步骤为:首先,将模板在图像中滑动,并将模板中心与图像中某个像素的位置重合,读取模板下各个对应像素的灰度值。然后,将这些灰度值从小到大排成一行并找出这些值中排在中间的一个。最后,将这个中同值赋给对应模板中心位置的像素。

3.3.3 灰度图像直方图均衡化

直方图是一幅图像中各像素灰度值出现的次数或频率的统计结果，它反映图像中不同灰度值出现的频率。直方图均衡化通过构造图像灰度级变换，改造原图像的直方图，使原图像灰度级集中的区域拉开或使灰度分布均匀，从而增大反差，使图像的细节清晰，达到增强的目的。

对于 8 位灰度图像，有 256 个灰度级，直方图均衡处理后图像直方图分布在 0-255 的所有灰度级上，没有占绝对优势或者绝对劣势的灰度或灰度范围，从而有效地提高图像的对比度，改善图像的质量。此方法的计算步骤如下：

已知图像有 L 级灰度 ($0, 1, 2, \dots, L-1$)， n 为图像像素数， n_i 为第 i 级灰度的像素数，则第 k 级灰度的出现频度为：

$$P(k) = \frac{n_i}{n} \quad (3-12)$$

可得到直方图变换函数：

$$H(k) = \sum_{i=0}^k \frac{n_i}{n} = \sum_{i=0}^k P(i) \quad (3-13)$$

第 k 级灰度的像素点经直方图均衡后的灰度级为：

$$t_k = L \times H(k), \text{ 其中 } k = 0, 1, 2, \dots, L-1 \quad (3-14)$$

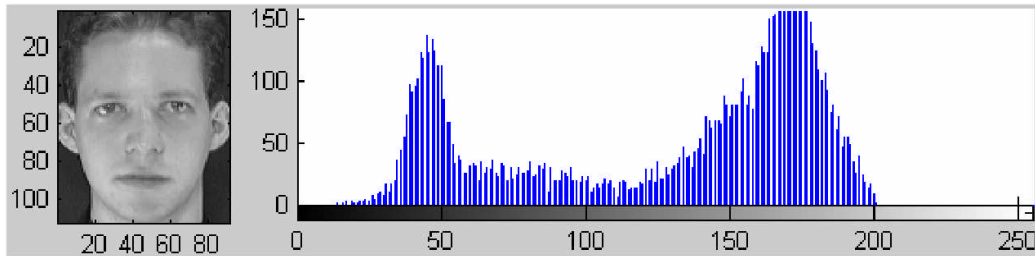


图 3.9 原始人脸图像及其直方图

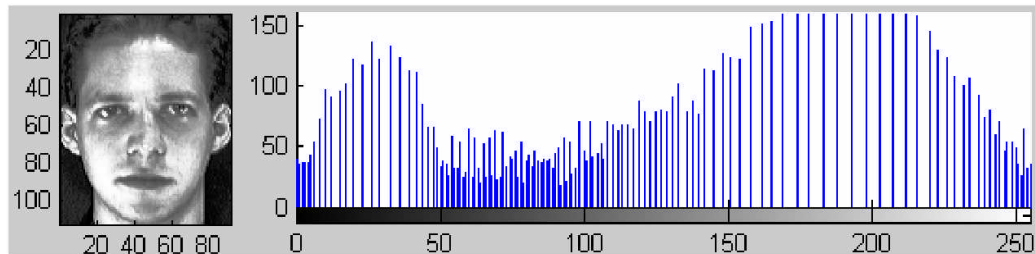


图 3.10 直方图均衡化处理后人脸图像及其直方图

直方图均衡化的处理结果如图 3.9 和图 3.10 所示。图 3.9 中，原始图像的对比度很弱，原因是灰度分布在一小区间内。而图 3.10 中直方图均衡化通过映射灰度值来逼近均匀分布，从而改善了对比度。

3.4 基于 K-L 变换的特征脸人脸识别

特征脸法是一种基于人脸全局特征的识别方法。所谓人脸全局特征是指所提取的特征与整幅人脸图像甚至与整个训练样本集相关，这种特征未必具有明确的物理意义，但却适合于分类。

主成分分析算法是一种特征提取方法，在许多领域得到了广泛应用。通过对原始数据的加工和处理，简化问题处理的难度并提高数据信息的信噪比，以改善抗干扰能力，保留了有效的信息。由于主成分分析算法是建立在由 Kirby 和 Sirovich^[40]提出的 K-L 变换^[41]基础上的，所以首先来介绍 K-L 变换。

3.4.1 K-L 变换

假设一个非周期性随机过程 $y(t)$ 在 $T_1 < T_2 < T_3$ 区间均匀采样，则可用向量 $y = [y(t_1)y(t_2)y(t_3)\cdots y(t_k)]^T$ 表示，对应的相关函数是一个 $k \times k$ 阶矩阵，它只有 k 个线性独立的特征向量，以此构成 Hilbert 空间中的一组基向量，则 y 可以用 k 个基向量的加权和表示，即

$$y = \sum_{i=1}^k a_i \varphi_i \quad (3-15)$$

用展开式中的有限项来估计 y ，即第 P 项以后被截断时，可得

$$\hat{y} = \sum_{i=1}^P a_i \varphi_i \quad (3-16)$$

当且仅当展开系数 $a_i = (y, \varphi_i), i = 1, 2, \cdots, p$ 时， \hat{y} 是 y 在这组基构成的子空间中的正交投影。由此引起得均方误差为

$$\varepsilon = E \left[\| y - \hat{y} \|^2 \right] \quad (3-17)$$

注意到 $a_i = (y, \varphi_i), i = 1, 2, \cdots, p$ 而且 $(\varphi_i, \varphi_j) = \delta(i - j)$ 上式等价于

$$\varepsilon = \sum_{i=p+1}^k \varphi_i^T E[yy^T] \varphi_i \quad (3-18)$$

令 $Q = E[yy^T]$ ，为使 $\varepsilon = \min$ ，即在均方误差最小准则下

$$\sum_{i=1}^P \varphi_i^T E[yy^T] \varphi_i = \sum_{i=1}^P \varphi_i^T Q \varphi_i = \max \quad (3-19)$$

由于 Q 是对称正定矩阵，所以上式等价于

$$\varphi_i^T Q \varphi_i = \lambda_i \geq 0, i = 1, 2, \cdots, p \quad (3-20)$$

均方误差最小准则等价于求相关矩阵的 p 个最大的特征值所对应的单位特征向量，这就是 **K-L** 变换。**K-L** 变换是一种最优正交变换，是子空间法模式识别的基础。若将 **K-L** 变换用于人脸识别，需假设人脸处于低维线性空间，且不同的人脸具有可分性。由于高维图像空间 **K-L** 变换后可得到一组新的正交基，因此可以通过保留部分正交基，以生成低维人脸空间。

3.4.2 主成分分析法

主成分分析(PCA)方法^[42-44]是依照系数方差最大准则确定 p 个最佳标准向量，即

$$\max\{E[\{(y, \varphi_i) - E[(y, \varphi_i)]\}^2], i = 1, 2, \dots, p\} \quad (3-21)$$

约束条件为

$$E[(y, \varphi_i)(y, \varphi_j)] = 0, i \neq j \quad (3-22)$$

显然，该公式等价于

$$E[(y - E[y])^2] = \varphi_i^T R \varphi_i = \max, i = 1, 2, \dots, p \quad (3-23)$$

其中 $R = E[(y - E[y])(y - E[y])^T]$ 为 y 的协方差矩阵。所以只需找到与协方差矩阵的 p 个最大的特征值所对应的单位特征向量，就可以确定公式中所有 p 个标准向量组中最佳的一组，这就是主成分分析。比较 Q 和 R 的表达式可知，对于零均值随机模式向量，主成分分析与 **K-L** 变换等价。

3.4.3 特征脸算法

M.Turk 提出的特征脸^[45,46](即 **Eigenface**)方法的核心就是利用主成分分析方法提取人脸的特征向量，将每一张人脸图像看成是许多特征脸的线性组合。其思想是以训练样本集的总体散布矩阵为产生矩阵，经主成分分析变换后得到一组具有人脸的形的特征向量，该向量就是所谓的特征脸。识别的过程是把一副新的图像投影到特征脸子空间，并通过它的投影点在子空间的位置以及投影线的长度来进行判定和识别。特征脸识别人脸算法一般有四个步骤，下面分别介绍这四个步骤。

1. 构建特征脸空间

特征脸空间由人脸训练图像数据的协方差矩阵的特征向量构成。对应于协方差矩阵非零特征值的特征向量组成正交基，然后将 n 维空间中的原始图像投影到该特征空间中。特别说明，此时的原始图像 x 存成大小是 n 维的向量，即：

$$x^j = [x_1^j, x_2^j, \dots, x_n^j]^T \quad (3-24)$$

训练集为 x^1, x^2, \dots, x^p (p 为样本图像个数)，形成矩阵 $X[n][p]$ ，其中行代表像

素，列代表每张人脸图像。

将训练样本集中的人脸图像减去平均人脸图像，将训练图像中心化。

$$\bar{X}^i = [x^i - \mu], \text{ 此时 } \mu = \frac{1}{p} \sum_{i=1}^p x^i \quad (3-25)$$

将中心化之后图像组成一个大小为 $n \times p$ 的矩阵 \bar{X} ：

$$\bar{X} = [\bar{x}^1, \bar{x}^2, \dots, \bar{x}^p] \quad (3-26)$$

将中心化后的图像组成的矩阵 \bar{X} 乘以它的转置矩阵得到协方差矩阵：

$$\Omega = \bar{X} \bar{X}^T \quad (3-27)$$

求解协方差矩阵 Ω 的 k 个非零特征值 $\lambda_1, \lambda_2, \dots, \lambda_k$ ，以及所对应的特征向量 u_1, u_2, \dots, u_k 。一般来说，训练图像数量 p 远远小于一张图像的像素值 n ，所以协方差矩阵 Ω 最多有对应于非零特征值的 p 个特征向量，所以 $k \leq p$ 。特征向量按照相应的特征值大小从大到小排列，最大特征值对应的特征向量反映了训练图像间的最大差异，而特征值越小对应的特征向量，反映的图像间的差异越小。“特征脸”空间就是由所有的非零特征值对应的特征向量 u_1, u_2, \dots, u_k 组成的。

2. 训练图像投影得到特征脸子空间

获得降维特征子空间以后，任一中心化后的人脸图像都可以通过公式(3. 19)投影到特征脸子空间并获得一组坐标系数：

$$\bar{x}^i = U \bar{x}^i \quad (3-28)$$

这组系数表明该图像在子空间中的位置，从而可以作为人脸识别的依据。也就是说，任一张人脸图像都可以表示为这组“特征脸”的线性组合，也可称为该图像的代数特征。

3. 识别待识别样本图像

对于每一张待识别样本图像，首先进行中心化：

$$\bar{y}^i = y^i - \mu, \text{ 此时 } \mu = \frac{1}{p} \sum_{i=1}^p x^i \quad (3-29)$$

然后投影到特征脸子空间中，求出其系数向量：

$$\bar{y}^i = U^T \bar{y}^i \quad (3-30)$$

将待识别样本图像的投影与所有训练图像在特征空间的投影相比较，选择最接近的训练图像。

4. 距离测量

一旦待识别图像被投影到特征脸子空间中，剩下的任务就是如何判定这些图像之间的相似性。下面分别介绍四种测量性的方法：

(1) 1L 范式：1L 范式也称为和范式，它将像素间的绝对值相加。表达式为：

$$L_1(A, B) = \sum_{i=1}^N |A_i - B_i| \quad (3-31)$$

AB 为两个 N 维列向量，下同。

(2) 欧几里德距离：它将像素之差的平方相加。其表达式为：

$$L_1(A, B) = \sum_{i=1}^N (A_i - B_i)^2 \quad (3-32)$$

(3) 角度测距：它是计算两个归一化以后的向量间的角度。其表达式为：

$$\text{cov}(A, B) = \frac{A}{|A|} \bullet \frac{B}{|B|} \quad (3-33)$$

(4) Mahalanobis 距离：其表达式为：

$$\text{Mah}(A, B) = -\sum_{i=1}^N A_i B_i C_i, C_i = \frac{1}{\sqrt{\lambda_i}} \quad (3-34)$$

3.5 本章小节

本章首先提出了嵌入式人脸识别系统的总体方案，采用以 ARM11 芯片作为主处理器的嵌入式核心板的 Tiny6410 开发板作为硬件平台；然后将系统软件设计部分划分为人脸图像采集模块、人脸图像预处理模块、人脸检测模块、人脸识别训练模块和人脸识别模块五个模块；然后详细介绍了人脸检测、人脸图像预处理和人脸识别相关的基础理论。人脸检测是利用肤色特征，来实现复杂图像中人脸区域的分割。首先，在 YCbCr 空间建立了肤色高斯模型和椭圆模型，皮肤色彩在 YCbCr 色彩空间具有聚类特性，利用这种聚类特性进行肤色和非肤色区域的划分，从而找出候选人脸区域。通常，候选肤色区域并非全部是人脸区域。如存在边缘的噪声、手、胳膊等非人脸的肤色区域，本文采用基于形态学滤波的方法去除一些边缘噪声，结合图像中的人脸形状、大小等特征，最终确定人脸位置。在人脸区域检测基础上，进一步介绍人脸识别理论。首先分析了主成分分析的基础 K-L 变换，接着介绍了主成分分析方法。最后详细分析了人脸识别经典算法--特征脸算法，包括如何构建特征脸空间，构建训练图像投影得到特征脸予空间，识别测试样本图像，通过计算特征向量的距离来实现人脸识别。

第四章 嵌入式人脸识别系统的实现

4.1 嵌入式系统开发环境的搭建

4.1.1 建立交叉编译环境

在进行嵌入式系统的开发时，运行程序的目标平台通常具有有限的存储空间和运算能力。这种情况下，在 ARM 平台上进行本机编译就不太可能了。为了解决这个问题，交叉编译工具就应运而生了。交叉编译^[47] (cross-compilation)是指，在某个主机平台上用交叉编译器编译出可在其他平台上运行的代码的过程。

通过交叉编译工具，我们就可以在 CPU 能力很强、存储控件足够的主机平台上编译出针对其他平台的可执行程序。我们在安装配置交叉编译工具链之前先选择 Linux 宿主机系统并且安装配置 Linux 下的超级终端 minicom。

一、Linux 宿主机系统的选择

宿主机 (host) 是编辑和编译程序的平台，一般是基于 X86 的 PC 机，通常也被称为主机。本文选择 ubuntu-10.10-desktop-i386 作为宿主机的操作系统。Ubuntu 注重系统的可用性，其设计为在标准安装完成后即可以让使用者投入使用的操作系统。

二、minicom 的安装配置

minicom 是 Linux 操作系统下的串行通讯接口，功能跟 windows 中的超级终端类似，在嵌入式系统开发中常用它做控制台。在使用 minicom 串口终端前，需要先配置参数，minicom 的配置界面是菜单方式。安装超级终端 minicom；执行命令启动超级终端，选择“Serial port setup”选项。

```
+-----+
| A -   Serial Device       : /dev/ttyS0
| B - Lockfile Location    : /var/lock
| C -   Callin Program     :
| D -   Callout Program    :
| E -   Bps/Par/Bits       : 115200 8N1
| F - Hardware Flow Control : No
| G - Software Flow Control : No
|
| Change which setting?
+-----+
| Screen and keyboard
| Save setup as dfl
| Save setup as..
| Exit
| Exit from Minicom
+-----+
```

图 4.1 配置 Linux 超级终端 minicom

按照图 4.1 所示, 配置 Linux 超级终端 minicom, “Serial Device” 设置为 “/dev/ttyS0”, “Bps/Par/Bits” 设置为 “115200 8N1”, “Hardware Flow Control” 选择 “No”, 最后选择 “Save setup as df1” 保存即可。

三、交叉编译工具链的安装配置

在本文搭建的平台下, 要为开发板编译内核, 图形界面 Qtopia/Qt4, bootloader, 还有其他一些应用程序, 均需要交叉编译工具链, 使用的是 arm-linux-gcc-4.5.1, 它默认采用 armv6 指令集, 支持硬浮点运算, 下面是安装该交叉编译工具链的详细步骤。

将光盘 Linux 目录中的 arm-linux-gcc-4.5.1-v6-vfp-20101103.tgz 复制到 ubuntu10.10 的某个目录下如 tmp/, 然后进入到该目录, 执行命令: #tar xvfz arm-linux-gcc-4.5.1-v6-vfp-20101103.tgz -C /, 把 arm-linux-gcc-4.5.1 安装到/opt/FriendlyARM/toolschain/4.5.1 目录。

把编译器路径加入系统环境变量, 运行命令#gedit /root/.bashrc 编辑/root/.bashrc 文件, 注意 “bashrc” 前面有一个 “.”, 修改最后一行为 export PATH=\$PATH:/opt/FriendlyARM/toolschain/4.5.1/bin。

```

panchao@ubuntu:~$ arm-linux-gcc -v
Using built-in specs.
COLLECT_GCC=arm-linux-gcc
COLLECT_LTO_WRAPPER=/opt/FriendlyARM/toolschain/4.5.1/libexec/gcc/arm-none-linux-gnueabi/4.5.1/lto-wrapper
Target: arm-none-linux-gnueabi
Configured with: /work/toolchain/build/src/gcc-4.5.1/configure --build=i686-build_pc-linux-gnu --host=i686-build_pc-linux-gnu --target=arm-none-linux-gnueabi --prefix=/opt/FriendlyARM/toolschain/4.5.1 --with-sysroot=/opt/FriendlyARM/toolschain/4.5.1/arm-none-linux-gnueabi/sys-root --enable-languages=c,c++ --disable-multilib --with-cpu=arm1176jzf-s --with-tune=arm1176jzf-s --with-fpu=vfp --with-float=softfp --with-pkgversion=ctng-1.8.1-FA --with-bugurl=http://www.arm9.net/ --disable-sjlj-exceptions --enable-__cxa_atexit --disable-libmudflap --with-host-libstdcxx=-static-libgcc -Wl,-Bstatic,-lstdc++,-Bdynamic -lm' --with-gmp=/work/toolchain/build/arm-none-linux-gnueabi/build/static --with-mpfr=/work/toolchain/build/arm-none-linux-gnueabi/build/static --with-ppl=/work/toolchain/build/arm-none-linux-gnueabi/build/static --with-cloog=/work/toolchain/build/arm-none-linux-gnueabi/build/static --with-mpc=/work/toolchain/build/arm-none-linux-gnueabi/build/static --with-libelf=/work/toolchain/build/arm-none-linux-gnueabi/build/static --enable-threads=posix --with-local-prefix=/opt/FriendlyARM/toolschain/4.5.1/arm-none-linux-gnueabi/sys-root --disable-nls --enable-symvers=gnu --enable-c99 --enable-long-long
Thread model: posix
gcc version 4.5.1 (ctng-1.8.1-FA)

```

图 4.2 arm-linux-gcc 交叉编译工具链安装成功

重新登录系统(不必重启机器, 开始->logout 即可), 使以上设置生效, 在命令行输入 arm-linux-gcc -v, 会出现如图信息, 这说明交叉编译环境已经成功安装。

4.1.2 在 Qt Creator 中部署 OpenCV

本文选择 Qt Creator 作为系统的集成开发与调试环境。Qt Creator 是跨平台的 Qt IDE, 此 IDE 能够跨平台运行, 支持的系统包括 Linux (32 位及 64 位)、Mac OS X 以及 Windows。根据官方描述, Qt Creator 的设计目标是使开发人员能够利用 Qt 这个应用程序框架更加快速及轻易的完成开发任务。

OpenCV^[48,49]是一个跨平台计算机视觉库,可以运行在 Linux、Windows 和 Mac OS 操作系统上。它轻量级而且高效—由一系列 C 函数和少量 C++类构成,同时提供了 Python、Ruby、MATLAB 等语言的接口,实现了图像处理 and 计算机视觉方面的很多通用算法。使用之前首先要在在 Qt Creator 中部署 OpenCV。

安装 QtCreator 后编辑 qmake.conf 文件,如图 4.3,在配置文件内添加语句:

QMAKE_INCDIR = /usr/include/opencv

QMAKE_LIBS = -lcvaux -lcx -lcxcore -lhighgui -lml

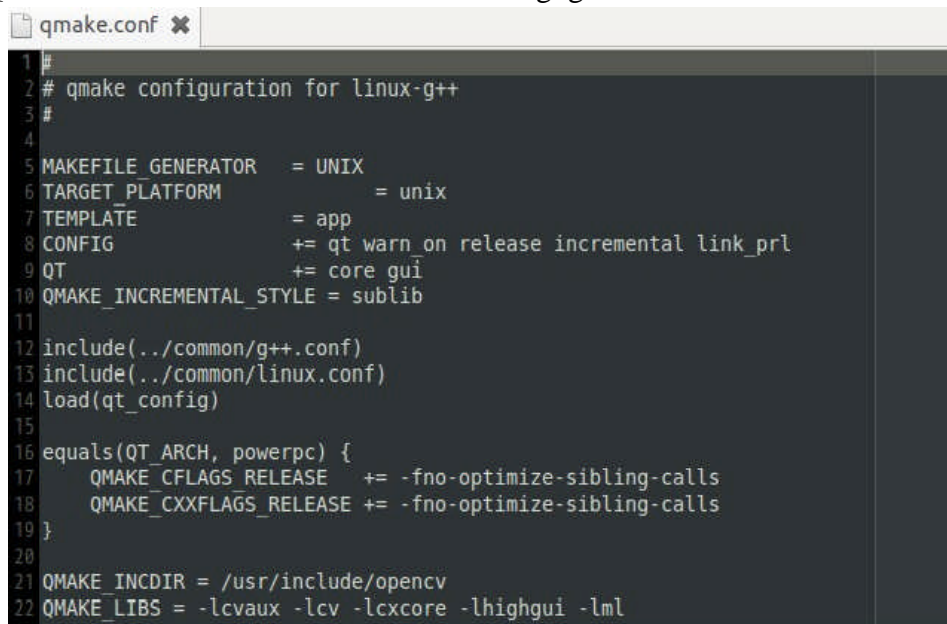


图 4.3 修改 QtCreator 的 qmake.conf 文件

还要打开对应的工程 XXX.pro 配置文件,向其中添加以下内容:

INCLUDEPATH += /usr/include/opencv

LIBS += /usr/lib/libcv.so \

/usr/lib/libcvaux.so \

/usr/lib/libcxcore.so \

/usr/lib/libhighgui.so \

/usr/lib/libml.so

4.2 嵌入式 Linux 系统的编译与移植

4.2.1 Bootloader 的分析与选择

Bootloader 是引导加载程序,是系统加电后运行的第一段代码。Bootloader 有硬件相关性高的特点,选用不同体系结构处理器的系统,需选择不同的 Bootloader。有多种不同 Bootloader,下面列出了几种比较流行并且支持 Linux 操作系统的

Bootloader^[50]。

表 4.1 几种支持 Linux 操作系统的 Bootloader

BootLoader	支持的体系结构				
	ARM	X86	PPC	MIPS	M68K
GRUB		√			
RedBoot	√	√	√	√	√
Blob	√				
VIVI	√				
U-Boot	√	√	√	√	√

U-Boot^[51]，全称 Universal Boot Loader，它是一个由在德国 DENX 软件工程中工作的 Wolfgang Denk 所维护，由 SourceForge 所主导的开放源代码开机加载程序，可以支持多种不同的计算机系统结构，包括 PPC、ARM、AVR32、MIPS、x86、68k、Nios 与 MicroBlaze。U-Boot 不仅支持引导 Linux，还支持引导 WinCE、VxWorks、QNX 等嵌入式操作系统。下表为 U-Boot 主要目录结构。

表 4.2 U-Boot 的主要目录结构说明

目录	文件说明
board/	目标板相关文件
cpu/	与处理器相关的文件
driver/	通用设备驱动
include/	U-Boot 头文件目录
lib-xxx/	与处理器体系相关的目录，如 lib-arm、lib-ppc 目录分别包含与 ARM、PowerPC 体系结构相关的文件
post/	上电自检文件的目录

本文选择 U-Boot 的理由有：开放源码；支持多种嵌入式操作系统，有 Linux、WinCE 等；支持多种体系结构的处理器，如 ARM、PowerPC、MIPS、Xscale；高度灵活的功能设置；丰富的开发文档他强大的网络技术支持。

4.2.2 Linux 内核的配置和编译

Linux 内核主要由 5 个子系统组成：进程调度、内存管理、虚拟文件系统、网络接口和进程间通信。Linux 内核本身并不是操作系统，它是一个完整操作系统的组成部分。Red Hat、红旗、Novell、Debian 和 SUSE 等 Linux 发行商都采用 Linux 内核，然后加入更多的工具、库和应用程序来构建一个完整的操作系统。

最新版本内核可以从 Linux 内核官方发布网站 <http://www.kernel.org> 获得。本文采用的是 Linux2.6.36 版本的内核。

Linux 内核源代码中的每一个目录或子目录可以看作一个模块，下表是对 Linux 内核主要目录的简单描述：

表 4.3 Linux 内核主要目录的说明

目录	描述
arch	子目录包含了与 CPU 体系架构相关的代码，Linux 支持 i386、anti 等架构
include	子目录包含了 Linux 内核代码的头文件
init	子目录包含了内核的初始化代码，这是内核开始工作的起点
drivers	目录包含了内核中设备驱动程序，如块设备，scsi 设备驱动程序等
fs	目录包含了所有文件系统的代码，如：ext2、vfat、JFFS2 模块的代码等等
net	目录包含了内核中网络相关的代码
ram	目录包含了所有的内存管理代码
ipc	目录包含了进程间通信的代码
kernel	目录包含了主内核代码，包括进程、程序执行、信号、模块功能的代码

获取内核源码之后，接着就要对内核进行配置编译^[52]，从而得到内核镜像。

首先对内核的进行裁剪和配置。分别执行以下命令：

```
$make ARCH=arm CROSS_COMPILE=arm-linux-gcc
```

```
$make s3c6410_config
```

```
$make menuconfig
```

其中，ARCH 变量用来指明为哪种架构建立内核。CROSS_COMPILE 变量表示使用的交叉编译工具。本系统所采用的开发板是以 S3C6410 为原型的，所以先载入 S3C6410 的默认设置。

在 Linux 终端中，用 make menuconfig 进入配置界面，如图 4.4 所示，根据裁剪需要进行设置，配置完毕后，对配置结果进行保存，保存为.config 文件，图 2.3 为 Linux 内核配置界面。

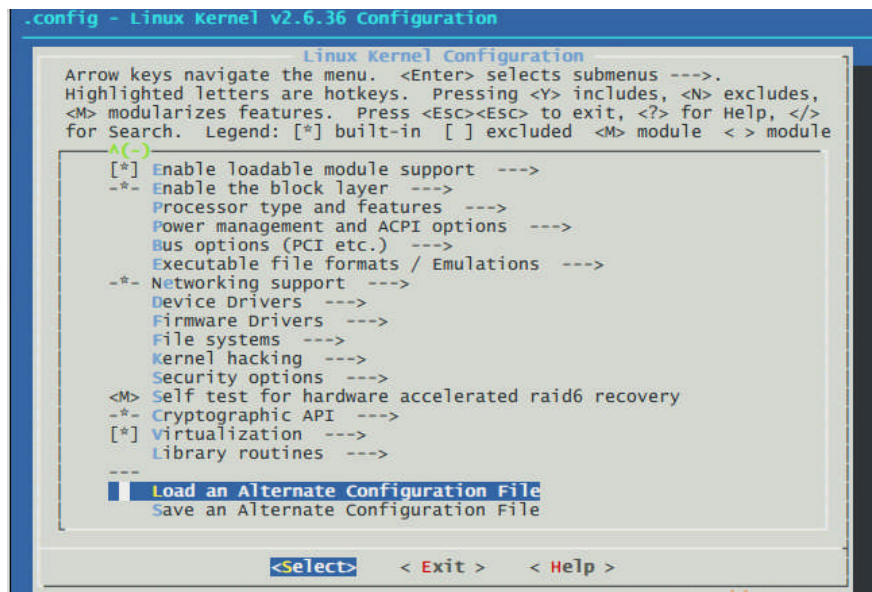


图 4.4 配置编译 Linux2.6.36 内核

最后成内核映像。配置好 Linux 内核后，需要用“\$make”命令编译 Linux 内

核；编译完成之后，将在 arch/arm/boot 目录下生成的 Linux 内核映像 zImage，该映像可直接由 U-Boot 来引导 Linux 系统。

4.2.3 根文件系统的设计

根文件系统是系统在启动过程中挂载的第一个文件系统，Linux 的根文件系统具有非常独特的特点，就其基本组成来说，Linux 的根文件系统应该包括支持 Linux 系统正常运行的基本内容，系统使用的软件和库，以及所有用来为用户提供支持架构和用户使用的应用软件。

由于嵌入式设备中 Flash 资源很紧张，如何有效地使用有限的存储空间是嵌入式开发者必须考虑的，合适的文件系统格式是解决这一问题的关键所在。

如何选择根文件系统，需要综合考虑文件系统的能力以及目标板的用途。目前，可用于嵌入式 Linux 系统的文件系统有很多种，表 4.4 列出了各种文件系统的特性。

表 4.4 各种嵌入式文件系统特性比较

文件系统	被写入	永久性	断电可靠性	经过压缩	存在于 RAM 中
Cramfs	否	不适用	不适用	是	否
JFFS2	是	是	是	是	否
JFFS	是	是	是	否	否
NFTL 上的 Ext2	是	是	否	否	否
NFTL 上的 Ext3	是	是	是	否	否
RAMdisk 上的 Ext2	是	否	否	否	是

在本文中，采用了目前使用较多也较简单的 Cramfs 文件系统，并在此基础上根据项目需要定制了自己的根文件系统。Cramfs 是针对 Linux 内核 2.4 之后的版本所设计的一种新型只读文件系统，采用了 zlib 压缩。因为 Cramfs 不会影响系统的读取文件的速度，而且是一个高度压缩的文件系统。

在嵌入式的环境之下，内存和外存资源都需要节约使用。Cramfs 是一个压缩格式的文件系统，它并不需要一次性地将文件系统中的所有内容都解压缩到内存之中，而只是在系统需要访问某个位置的数据的时候，马上计算出该数据在 Cramfs 中的位置，将它实时地解压缩到内存之中，然后通过对内存的访问来获取文件系统中需要读取的数据。

制作 Cramfs 格式的根文件系统的步骤：在内核中加入对 Cramfs 文件系统的支持；建立相应的各级目录，如/bm、/home、/usr 等；安装链接库，这里的链接库也就是之前安装交叉编译器时生成的/arm、linux/lib 目录中的库文件；编译安装 Busybox；配置 etc 目录下的配置文件；将已经配置好的根文件系统制作成所需要的 Cramfs 镜像；最后/mkcramfs 源目录目标文件。

4.3 基于 Video4Linux 的视频采集

Video4Linux^[53] (简 V4L)是 Linux 中关于视频设备的内核驱动, 它为针对视频设备的应用程序编程提供一系列接口函数, 这些视频设备包括现今市场上流行的 TV 卡、视频捕捉卡和 USB 摄像头等。对于 USB 口摄像头, 其驱动程序中需要提供基本的 I/O 操作接口函数 open、read、write、close 的实现。对中断的处理实现, 内存映射功能以及对 I/O 通道的控制接口函数 ioctl 的实现等, 并把它们定义在 struct file_operations 中。这样当应用程序对设备文件进行诸如 open、close、read、write 等系统调用操作时, Linux 内核将通过 file_operations 结构访问驱动程序提供的函数。

4.3.1 Video4Linux 数据结构

Linux 内核提供 Video4Linux 应用程序接口, 在程序开发时, 首先是基于 Video4Linux API 函数来设计程序, 使用的一些主要参数和函数定义在 Linux 系统的 /include/linux/videodev.h 和 videodev.c 文件中。现将本系统用到的 V4L 结构中包含的数据结构及其用途在表 4.5 中作简要说明:

表 4.5 Video4Linux 数据结构介绍

数据结构	功能用途	分量描述
video_capability	描述设备的基本信息, 如设备名称、支持的最大最小分辨率、信号源信息	设备的名称、功能标准、支持的信号源个数、采集图像的大小限制等
video_buffer	对最底层 buffer 进行描述	帧缓存的基地址、高宽度、深度等
video_window	表示采集的图像的显示方法	x 坐标、y 坐标、图像宽度高度、色度值、裁剪的矩形数目等
video_channel	描述各个信号源的属性	信号源编号、名称、类型、制式等
video_picture	设置摄像头采集的图像的各种属性, 据此来设置图像缓冲区的大小等参数	图像的亮度、色调、色度、对比度、灰度和深度以及调色板的格式
video_mbuf	描述利用 mmap 进行映射的帧的信息, 即是摄像头存储器缓冲区中的帧信息	每帧的大小、最左支持的帧数、每帧相对基址的偏移量
video_mmap	用来进行内存映射	

ioctl 是设备驱动程序中对设备的 I/O 通道进行管理的函数。所谓对 I/O 通道进行管理。对它的调用如: int ioctl(int fd, ind cmd, ……), 其中 fd 就是用户程序打开设备时使用 open 函数返回的文件标示符, cmd 就是用户程序对设备的控制命令, Video4Linux 中对 cmd 重要的宏定义^[54]如表 4.6 所示。

表 4.6 Video4Linux 重要的宏定义

宏定义	命令功能
VIDIOCGCAP	获取图像设备的基本信息，保存到 video_capability 结构体中
VIDIOCSCAP	根据 video_capability 结构体中的值来设置图像设备初始参数
VIDIOCGPICT	获取图像的基本信息，保存在 video_picture 中，包括图像的亮度、色度、对比度等
VIDIOCSPICT	根据 video_picture 结构体中的值来设置采集图像的初始参数
VIDIOCGCHAN	获取信号源的信息，存放在 video channel 中
VDIOCGMBUF	获取摄像头存储缓冲区的帧信息
VIDIOCAMCAPTURE	截取视频图像
VIDIOSYNC	判断视频图像是否截取成功

4.3.2 USB 摄像头视频采集过程

在本系统平台上要实现对 USB 摄像头驱动，首先把 USB 控制器驱动模块静态编译进内核，使平台中支持 USB 接口，再在需要使用摄像头采集时，使用 insmode 动态加载其驱动模块，这样摄像头就可正常工作了。接下来实现对视频流的采集过程，如图 4.5 所示。

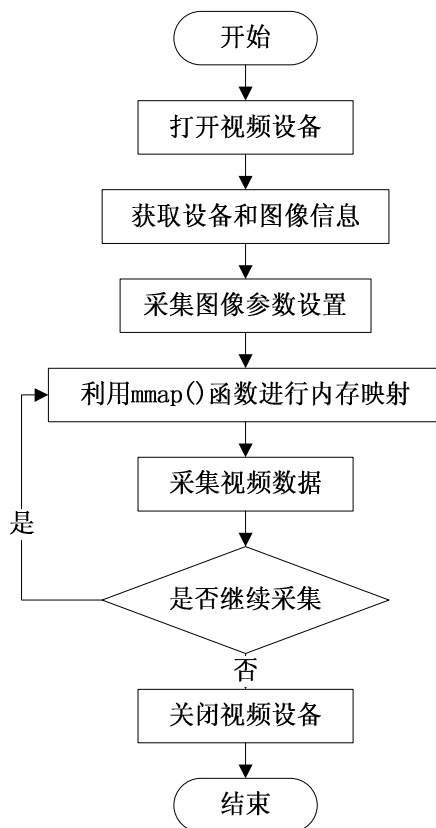


图 4.5 视频采集流程图

打开视频设备，摄像头在系统中对应的设备文件为/dev/video0，采用系统调用函数 grab_fd=open("/dev/video0", O_RDWR)，grab_fd 是设备打开后返回的文件描

述符（打开错误返回-1），以后的系统调用函数就可使用它来对设备文件进行操作了。接着，利用 `ioctl (grab_fd, VIDIOCGCAP, &grab_cap)` 函数读取 `struct video_capability` 中有关摄像头的信息。该函数成功返回后，这些信息从内核空间拷贝到用户程序空间 `grab_cap` 各成员分量中。

4.3.3 内存映射方式截取图像

设备初始化工作完成后，就可以对视频图像截取了，有两种方法：一种是 `read()` 直接读取；另外一种 `mmap()` 内存映射。综合表 4.7 内容对比两种视频图像截取方式，后者具有速度快和进城之间实现了内存共享等特点，所以在程序实现中采用了内存映射方式，即 `mmap()` 方式。

表 4.7 两种视频图像截取方式对比

截取方式	工作原理	速度	特点
<code>mmap()</code>	把设备文件映射到内存中，直接内存读取	内存访问，速度明显更快	进城之间通过映射同一文件实现内存共享，各个进程可以像访问普通内存一样对文件进行访问，而且访问时仅用到指针
<code>read()</code>	通过内核缓冲区读取数据	用到磁盘访问，速度慢	需要通过其他方式实现内存共享，访问时调用文件操作函数

利用 `mmap()` 内存映射方式对视频进行采集的流程^[55]如下：

1. 先使用 `ioctl(vd->fd, VIDIOCGMBUF, &(Vd->mbuf))` 函数获得摄像头存储缓冲区的帧信息，然后重新设置图像帧的垂直及水平分辨率、彩色显示格式以及当前帧的状态等信息。

2. 接着把摄像头设备缓冲区的设备文件映射到内存空间，其实现语句为 `vd->map=(unsigned char*)mmap(0, vd->mbuf.size, PROT_READ|PROT_WRITE, MAP_SHARED, vd->fd, 0)`，该函数成功时返回映射内存区的指针。这样设备文件的数据就映射到内存区，该映射内存区可读可写且不同进程间可共享，由语句 `vd->map+vd->mbuf.offsets[vd->frame]` 确定每一帧在内存中的地址，其中 `frame` 是当前截取的帧的序号。

表 4.8 内存映射 `mmap()` 函数参数说明

参数	含义说明
0	共享内存的起始地址，表示由系统自动分配
<code>vd->mbuf.size</code>	表映射到调用进程地址空间的字节数
<code>PROT_READ PROT_WRITE</code>	指定共享内存的访问权限 <code>PROT_READ</code> 可读， <code>PROT_WRITE</code> 可写， <code>PROT_EXEC</code> 可执行
<code>MAP_SHARED</code>	共享内存的属性，从 <code>MAP_SHARED</code> 和 <code>MAP_PRIVATE</code> 中选用一个
<code>vd->fd</code>	有效地摄像头设备的文件描述符
0	被映射对象在内存相对起始地址的偏移

3. 视频截取。把视频从摄像头设备缓冲区映射到内存后就可以进行视频的截取，实现语句为 `ioctl(vd->fd, VIDEOMCAPTURE, &(vd->mmap))`，若调用成功，开始一帧截取，该操作是非阻塞的，是否截取完毕留给 `VIDIOCSYNC` 来判断。

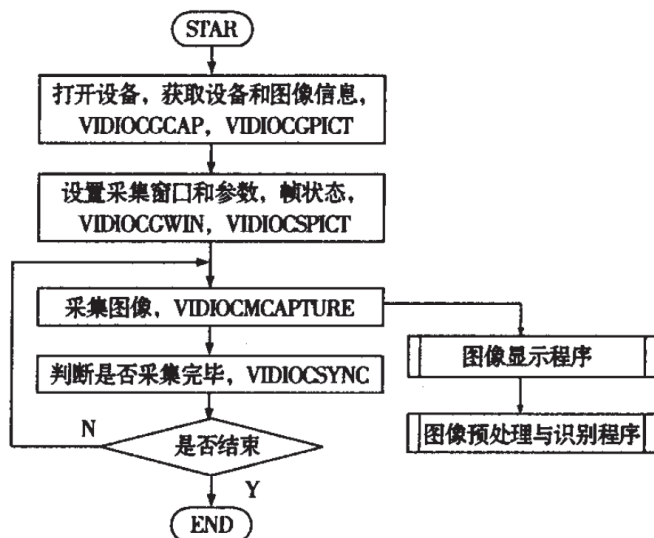


图 4.6 基于 Video4Linux 的图像采集的程序流程图

4. 调用 `VIDIOCSYNC` 等待一帧截取结束。实现语句为 `ioctl(vd->fd, VIDIOCSYNC, &(vd->frame))` 函数调用成功，表明一帧图像截取完成，可以开始进行下一次视频截取。本系统设置 `vd->mbuf.frames=2`，所以可支持一次采集两帧图像，一帧图像被处理时可以采集另一帧图像。利用 `vd->frame=(vd->frame+1)%2` 和 `vd->map+vd->mbuf.offsets[vd->frame]` 这两条语句，可以循环采集两帧图像。

4.4 基于 AdaBoost 算法的人脸检测

要实现基于 AdaBoost 算法的人脸检测，先要了解 Haar 分类器的数据结构，在此基础上利用相关人脸检测函数实现本系统的人脸检测功能。

4.4.1 Haar 分类器的数据结构

OpenCV 用 4 个结构体定义了 BoostHaar 分类器结构，其中比较重要的包括 `CvHaarStageClassifie` 与 `CvHaarClassifierCascade` 结构，现将两个结构体的重要数据成员在表 4.9 和表 4.10 中分别说明。

表 4.9 `CvHaarStageClassifie` 结构体数据成员

数据成员	类型	说明
count	int	强分类器中弱分类器的数量
threshold	float	叠加分类器的阈值
classifier	<code>CvHaarClassifier *</code>	定义分类器数组

表 4.10 CvHaarClassifierCascade 结构体数据成员

数据成员	类型	说明
flags	int	标志位
count	int	分级分类器中强分类器数量
orig_window_size	CvSize	训练中原始目标的大小
real_window_size	CvSize	待检测物体的大小
scale	double	Haar 块缩放的大小
stage_classifier	CvHaarStageClassifier*	定义强分类器数组

4.4.2 人脸检测相关函数

基于 Adaboost 的人脸检测器主要包含训练和检测 2 个模块，现对 OpenCV 中的检测模块进行简要介绍，检测模块的流程如图 4.7 所示。

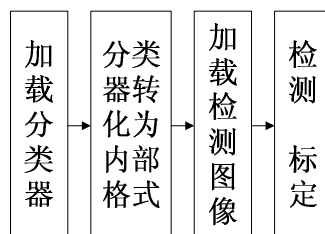


图 4.7 检测程序流程图

下面就人脸检测过程中用到的相关函数^[56]作简要的说明。

1) 函数 LoadHaarClassifierCascade。装载训练好的级联分类器，其格式为
 CvHaarClassifierCascade *cv LoadHaarClassifierCascade(const char *directory,
 CvSize orig_window_size),

其中 directory 为包含训练好的级联分类器文件的路径。orig_window_size 为级联分类器在训练中采用的检测目标的尺寸。

2) 将分类器转化为 OpenCV 的内部格式，这一步骤是必不可少的。这是因为 OpenCV 只能处理内部优化格式的分类器。

3) 检测要用到函数 HaarDetectObjects，该函数使用针对某目标物体训练的级联分类器在图像中找到包含目标物体的矩形区域，并将这些区域作为一系列的矩形框返回。其格式为 CvSeq *cvHaarDetectObjects (const CvArr *image ,
 CvHaarClassifierCascade *cascade, CvMemStorage *storage, double scale_factor=1.1,
 int min_neighbors=3, int flags=0, CvSize min_size=cvSize(0,0));

其中 image 是被检图像，cascade haar 是级联分类器内部标志形式，storage 用来存储检测到的一系列候选目标矩形框的内存区域，scale_factor 是在前后两次相继的扫描中，搜索窗口的比例系数。min_neighbors 是构成检测目标的相邻矩形的

最小个数(默认为-1)。flags 是操作方式。rain_size 是检测窗口的最小尺寸。默认情况下被设为分类器训练时采用的样本尺寸。

4) 标定要用到函数 `RunHaarClassifierCascade`。该函数在给定位置的图像中运行级联分类器,用于对单幅图片进行检测。其格式为 `int cvRunHaarClassifierCascade (CvHaarClassifierCascade *cascade, CvPoint pt, int start_stage=0)`;其中 `cascade Haar` 是级联分类器, `pt` 是待检测区域的左上角坐标, `start_stage` 是级联层的初始下标值。

4.4.3 人脸检测功能的实现

图像的人脸检测流程如下图所示,大致可以分为四个过程:(1) 加载训练好的级联分类器;(2) 将其转化为内部格式,以便于被计算机识别处理;(3) 加载待检测的图像;(4) 检测出图像中的人脸区域并且标记处人脸位置。

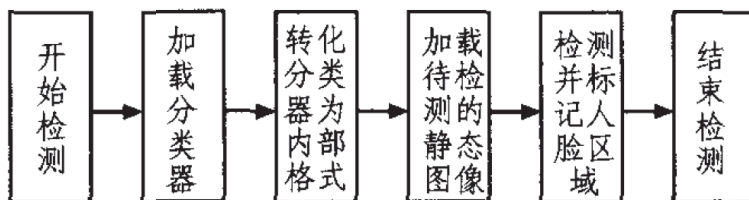


图 4.8 图像的人脸检测流程图

摄像头采集来的视频的人脸检测流程如下图所示,主要过程:(1) 检测并启动摄像头,如果没有安装摄像头会发出警告并且结束检测;(2) 加载已经训练好的级联分类器并且将其转化为内部格式;(3) 判断标志位,若为真,首先捕获当前一帧并创建该帧的图像,然后载入当前一帧的图像,并对图像中的人脸区域进行检测和标记,最后判断保存标志位,此时若为真则保存标记的人脸区域图像并进入下一帧的检测,否则直接进入下一帧的检测;若检测标志位为假则结束检测。

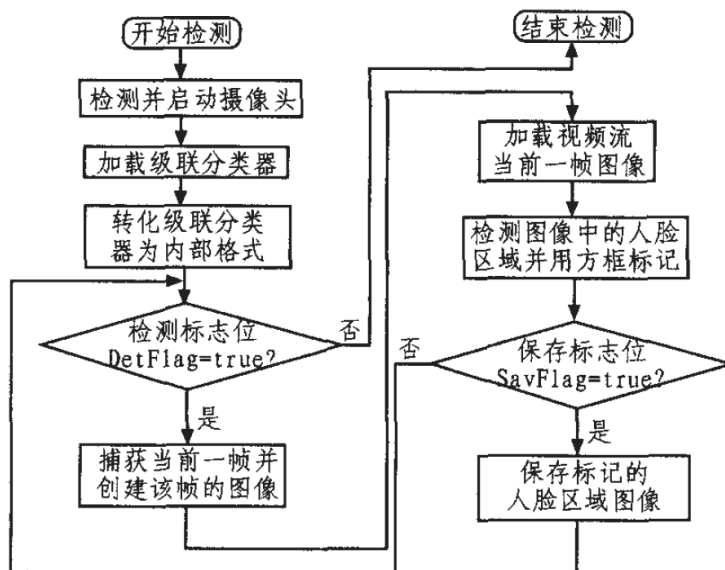


图 4.9 摄像头视频的人脸检测流程图

视频可以看成是一帧一帧的图像在时间上的排列, 这样视频人脸检测可看作是对一组图像序列的连续检测, 因此, 基于 avi 视频的人脸检测和基于摄像头视频的人脸检测的流程基本上是一样的, 不同之处是其不需要检测启动摄像头这一步, 而是进行加载 avi 视频, 所以其流程不再赘述。

利用计算机视觉类库 OpenCV 中用于人脸检测^[57]的详细过程:

1) `cascade=(CvHaarClassifierCascade)cvLoad(cascade_name, 0, 0, 0);` 该函数实现载入级联分类器的功能, 其中 `cascade_name` 为级联分类器的路径。

2) `cvGrabFrame (capture);` 和 `frame= cvRetrieveFrame(capture);` 这两个函数一起调用实现快速获取 avi 视频帧。其中 `capture=cvCaptureFromAVI(path);` 读取捕获 avi 视频。

3) `CvMemStorage storage=cvCreateMemStorage(0);` 该函数创建一个默认大小为 64 kB 的内存块并返回指向块首的指针。

4) `CvSeq* faces=cvHaarDetectObjects(small_img, cascade, storage, 1.1, 2, 0, cvSize(30, 30));` 该函数的功能是使用训练好的级联分类器在图像中找到包含人脸目标的矩形区域, 并将这些区域作为一序列的矩形框返回。

5) `cvRectangle(img, LeftPoint, RightPoint, color, 3, 8, 0);` 该函数实现对检测到的人脸区域用方框标记出来。

6) `cvSaveImage(savepath+treat(strCount, "bmp"), OutImage);` 该函数实现把标记的人脸区域图像保存为 bmp 格式的图像, 其中 `OutImage=cvCreateImage(cvSize(120, 120), 8, 3);` 创建头并分配 120x120 大小的图像数据空间。

7) 获取到被检测图像之后需要进行包括图像色彩转换、图像大小变换和直方图均衡化等相关处理后方可进行后面的人脸检测。相关处理函数如下:
`cvCvtColor(img, gray, CV_BGR2GRAY);` 色彩转换。`cvResize(gray, small_img, CV_INTER_LINEAR);` 图像大小变换。`cvEqualizeHist(small_img, small_img);` 灰度图像直方图均衡化。

4.5 人脸识别功能的设计与实现

4.5.1 训练系统框架

每个 Haar 特征对应看一个弱分类器, 但并不是任何一个 Haar 特征都能较好的描述人脸灰度分布的某一特点, 如何从大量的 Haar 特征中挑选出最优的 Haar 特征并制作成分类器用于人脸检测, 这是 AdaBoost 算法训练过程所要解决的关键问题。

Paul Viola 和 Michael Jones 于 2001 年将 Adaboost 算法应用于人脸检测中, 其基本思想是针对不同的训练集训练同一个分类器, 把这些不同训练集上得到的

分类器联合起来构成一个最终的强分类器。Adaboost 算法中不同的训练集是通过调整每个样本对应的权重来实现的。开始时，每个样本对应的权重是相同的，对于分类错误的样本，加大其对应的权重；而对于分类正确的样本，降低其权重，这样分错的样本就被突出出来，从而得到一个新的样本分布。在新的样本分布下，再次对弱分类器进行训练，得到弱分类器。依次类推，经过 T 次循环，得到 T 个弱分类器，把这 T 个弱分类器按一定的权重叠加起来，得到最终的强分类器。

训练系统总体框架，由“训练部分”和“补充部分”构成。依据系统框架，本文的训练系统可分为以下几个模块：

- (1) 以样本集为输入，在给定的矩形特征原型下，计算并获得矩形特征集；
- (2) 以特征集为输入，根据给定的弱学习算法，确定阈值，将特征与弱分类器一一对应，获得弱分类器集；
- (3) 以弱分类器集为输入，在训练检出率和误判率限制下，使用 AdaBoost 算法挑选最优的弱分类器构成强分类器；
- (4) 以强分类器集为输入，将其组合为级联分类器；
- (5) 以非人脸图片集为输入，组合强分类器为临时的级联分类器，筛选并补充非人脸样本。

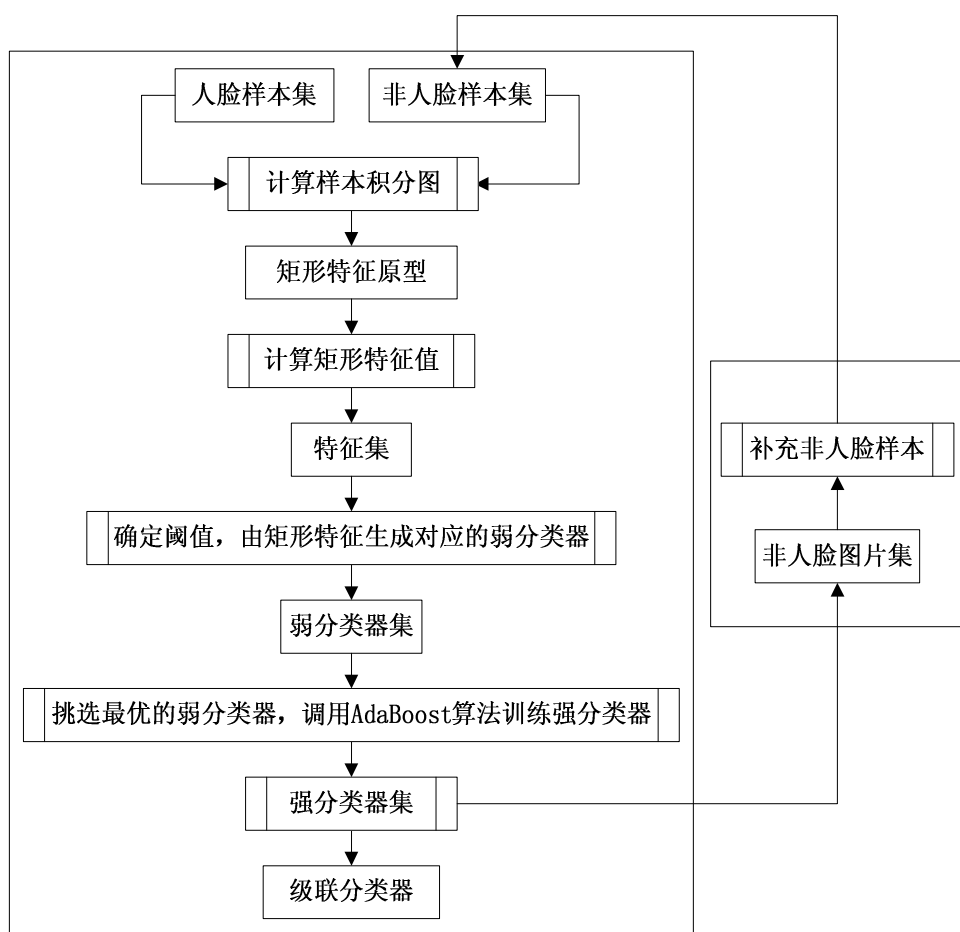


图 4.10 级联分类器训练系统框架

训练样本要求是面部特写图像，大小被归一化为 24×24 像素，其中正训练样本要求是面部特写图像，但是人脸形态千差万别，所以训练样本选取过程中要考虑到样本的多样性。负训练样本，大小被归一化为 24×24 像素，其中各样本不完全相同，分别具有一定的代表性。

4.5.2 人脸识别训练过程

训练过程分为 3 个步骤：首先需要提取 Haar 特征；然后将 Haar 特征转化成对应的弱分类器；最后从大量的弱分类器中迭代选择出最优弱分类器。

(1) 提取 Haar 特征

常用的 Haar 特征有 4 种，当然也可以在这 4 种特征的基础上设计出更多、更复杂的特征。以大小为 24×24 像素的训练样本为例，上述 4 种特征的总个数超过了 160000 个。这样庞大的数字给后续的迭代训练工作带来了庞大的计算量，直接导致 AdaBoost 算法训练过程极为费时，这恰恰是算法需要改进的关键问题之一。

(2) 生成弱分类器

每一个 Haar 特征都对应着一个弱分类器，每一个弱分类器都是根据它所对应的 Haar 特征的参数来定义的。利用上述 Haar 特征的位置信息，对训练样本进行统计就可以得到对应的特征参数。AdaBoost 算法中所训练的弱分类器是任何分类器，包括决策树，神经网络，隐马尔科夫模型，如果弱分类器是线性神经网络，那么 AdaBoost 算法每次将构造多层感知器的一个节点。

(3) 采用 AdaBoost 算法选取优化的弱分类器

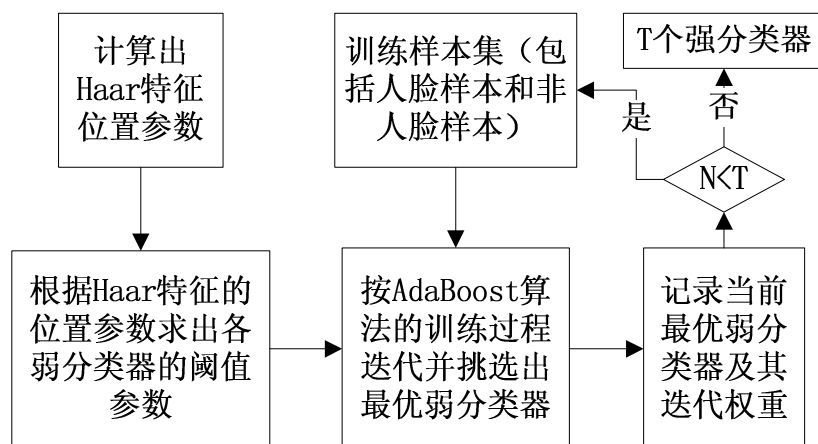


图 4.11 AdaBoost 算法训练过程示意图

AdaBoost 算法训练过程就是挑选最优弱分类器，并赋予权重过程，图 4.11 是 AdaBoost 算法训练示意图。

4.5.3 基于特征脸人脸识别

本文采用特征脸方法^[58]进行人脸识别。人脸检测标定人脸区域图像后，提取人脸图像的人脸特征值，读取存储的人脸数据库 XML 文件，使用 PCA 方法进行比对，若可信度超过阈值，则在屏幕上显示对应人脸的身份信息。

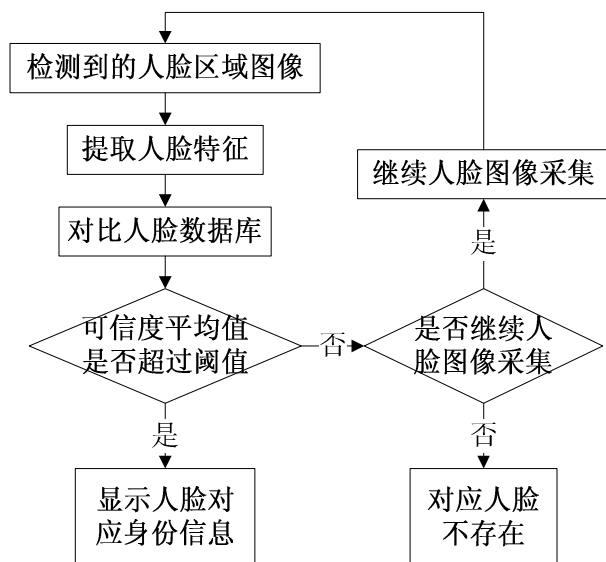


图 4.12 人脸识别过程示意图

摄像头可获取多幅图像以提高准确度和可靠性，可以采集多幅图像求可信度平均值与阈值比较。也就是说，当可信度平均值小于阈值时，可以选择继续采集人脸图像。

相对于欧式距离，利用马氏距离作为人脸图像间的距离识别效果较好，但由于本作品主要用于判断待测人脸是否为数据库中人脸，可信度的计算方法极为关键，而目前最好的可信度的计算方程是基于欧式距离的。

现将有关人脸识别的主要函数介绍如下：

(1) `fileStorage=cvOpenFileStorage("facedata.xml",0,CV_STORAGE_READ);` 以 `CV_STORAGE_READ` 方式导入人脸数据库中的 XML，并将返回值赋给 `fileStorage`。

(2) `findNearestNeighbor(float*projectedTestFace)` 将提取到的人脸特征与人脸数据库对比，查找特征值最接近的人脸。其中 $fConfidence=1.0f-\sqrt{leastDistSq/(float)(nTrainFaces * nEigens)}/255.0f$; `fConfidence` 为提取的人脸特征值可信度平均值，当 `fConfidence` 大于设定的阈值 `threshold` 时，人脸识别成功，返回对应人脸身份信息。

(3) `cvEigenDecomposite(IplImage* obj, int nEigObjs, void* eigInput,int ioFlags, void* userData, IplImage* avg, float* coeffs)`，函数作用是将人脸图像通过 Eigenface 变换矩阵，投射到子空间中。其中 `obj` 为输入目标，`nEigObjs` 为自选 eigen 数量，

eigInput 为 eigenface 向量组成的变换矩阵, 数据结构属于 IplImage 队列, 或者根据 ioFlag 参数值变化的回调函数, coeffs 是唯一一个输出, 即人脸在子空间的投影。

(4) cvCalcEigenObject()只需要在训练时使用, 依据训练样本, 计算 Eigenface 的变换矩阵。而 cvEigenDecomposite(), 既需要产生训练样本的投影, 又需要产生测试脸的投影。训练时得到训练样本投影脸的函数调用方法如下: cvEigenDecomposite(faceImgArr[i], nEigens, eigenVectArr, 0, 0, pAvgTrainImg, projectedTrainFaceMat->data.fl + i*offset), 其中 i 从 0 到 nTrainFaces。

4.6 本章小结

本章主要包括两方面的内容: 嵌入式人脸识别系统软硬件平台的搭建和人脸识别软件的实现。

嵌入式人脸识别系统软硬件平台的搭建: 首先就是以 Ubuntu 系统为宿主机建立起了交叉编译环境, 安装配置了交叉编译工具链 arm-linux-gcc, 然后配置 linux 下的超级终端 minicom, 本文选择 Qt Creator 作为集成开发环境, 最后就如何在其中部署 OpenCV 作了详细的说明。

人脸识别软件的实现: 首先是人脸图像的获取, 本文采用了基于 Video4Linux 的摄像头视频采集, 以内存映射 mmap()的方式截取人脸图像; 然后介绍了 Haar 分类器的数据结构, 重点说明了 OpenCV 中人脸识别的 HaarDetectObjects()等相关函数, 在此基础上实现了人脸识别功能; 最后在研究了级联分类器训练系统框架的基础上实现了人脸识别的训练过程; 实现了基于特征脸算法的人脸识别。

第五章 系统的测试结果与分析

5.1 人脸检测结果与分析

本文用 OpenCV 中的检测器，使用其自带的一个 XML 文件执行检测，该分类器具有良好的检测效果，能够适用于多种情况下的正面人脸检测，如人脸小角度的旋转，不同的表情，是否佩戴眼镜等等。

人脸图像检测以英国剑桥 Olivetti 实验室 ORL 人脸库的 400 幅人脸图像为样本，用该分类器进行测试。在函数 cvHaarDetectObjects 中设置 scale_factor 为 1.2，min_neighbors 为 2.0，flags 为 CV_HAAR_DO_CANNY_PRUNING，min_size 为 40x40 的检测窗口。ORL 人脸库只包含人脸检测图像，只考虑了人脸小角度的旋转和不同的面部表情等因素，忽视人脸背景是否复杂、光照不均匀等情况。



图 5.1 对于小角度旋转的人脸的检测结果



图 5.2 不同人戴眼镜的人脸检测结果

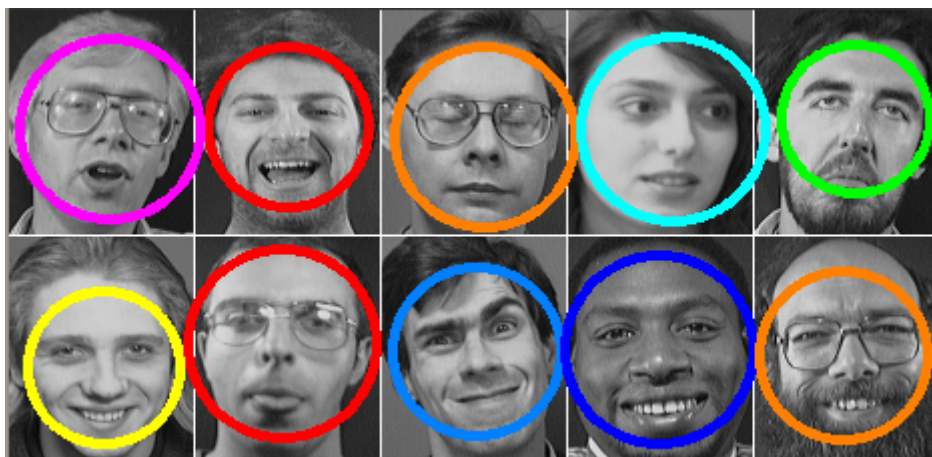


图 5.3 不同面部表情的人脸检测结果

对检测结果的分析如下：

- (1) 本检测样本只包含人脸图像，共检测人脸图片 400 幅，结果正确的为 383 幅，漏检为 17 幅，检测正确率为 95.75%。
- (2) 对于小角度旋转的人脸，本系统能准确的检测到人脸，但是当人脸旋转的角度过大的时候，将会有漏检的情况出现，如图 5.1 所示。
- (3) 对于佩戴眼镜等饰物以及带有面部表情的人脸，如图 5.2 和图 5.3 所示，本系统仍能准确的检测出人脸。

5.2 人脸识别结果与分析

人脸识别过程中，系统从 USB 摄像头获取 YUV 格式的图片，将其转换成 IplImage 格式，利用 AdaBoost 算法进行人脸检测，得到一个矩形区域，截取该矩形区域图像进行直方图均衡化处理，进行人脸识别。如图 5.4 和图 5.5 所示为本系统的人脸图像采集显示结果。



图 5.4 人脸图像采集（一）

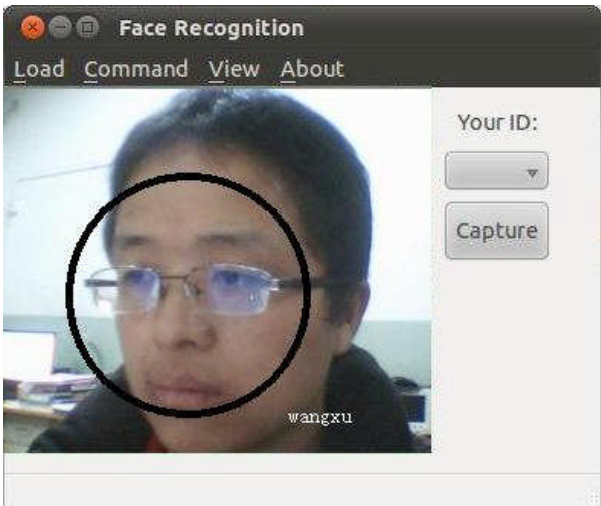


图 5.5 人脸图像采集识别（二）

提取人脸图像对应的特征脸是人脸识别过程中的一个重要环节，图 5.6 为对 ORL 人脸数据库进行人脸识别训练过程中所提取的部分特征脸。

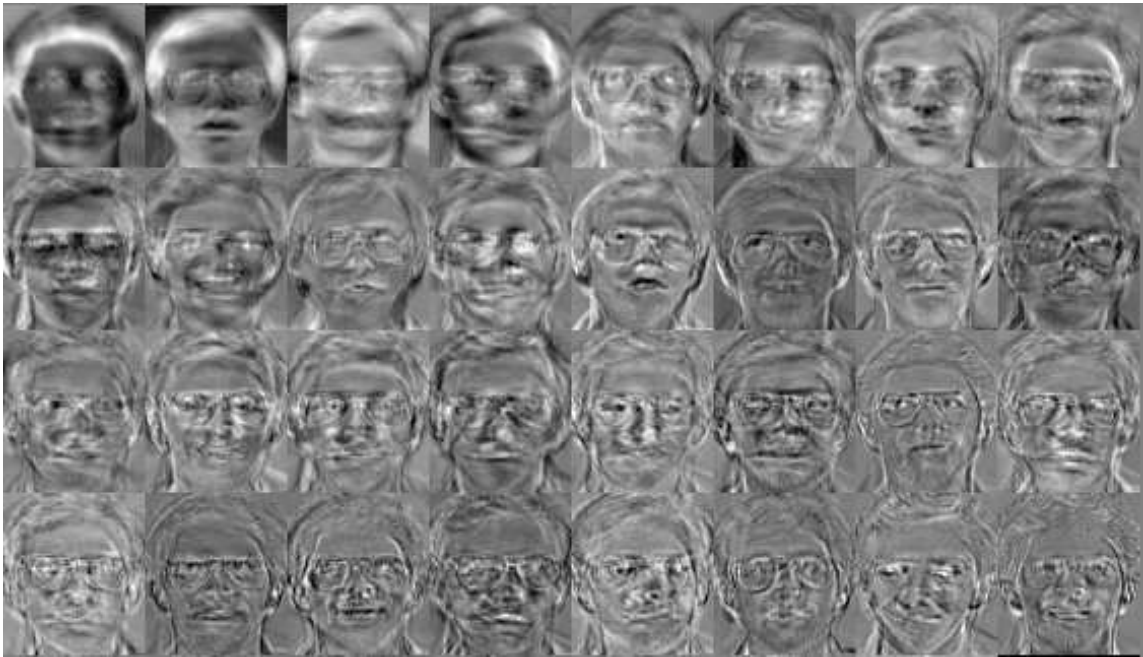


图 5.6 人脸识别训练过程中部分提取的特征脸

本系统定义阈值 $\text{threshold} = 0.85f$, $f\text{Confidence} = 1.0f - \sqrt{(\text{leastDistSq} / (\text{float})(n\text{TrainFaces} * n\text{Eigens}))} / 255.0f$, 当 $f\text{Confidence} > \text{threshold}$ 时, 人脸识别成功。本文选取 ORL 人脸数据库分两组实验对系统人脸识别功能进行测试。

第一组实验：选取 ORL 人脸数据库中 8 个人的脸，每人 10 张人脸图片，其中前 5 张进行人脸识别训练，后面 5 张进行人脸识别的测试。如表 5.1 所示为人脸识别结果，共测试 40 张人脸，识别成功率为 92.5%，识别失败的可能原因分别为配戴眼镜、比较夸张的面部表情和较大的人脸旋转角度。

表 5.1 第一组实验人脸识别结果

人脸识别 对象编号	人脸编号				
	face01	face02	face03	face04	face05
personA	0.999928	0.999948	0.999954	0.999935	0.999950
personB	0.999920	失败	失败	0.999919	0.999938
personC	0.999912	0.999950	0.999950	0.999930	0.999938
personD	0.999924	0.999948	0.999905	0.999914	0.999954
personE	0.999928	0.999917	0.999925	0.999923	0.999937
personF	0.999911	0.999939	0.999923	0.999929	0.999915
personG	0.999905	0.999924	0.999923	0.999913	0.999923
personH	0.999939	0.999924	0.999902	0.999904	失败

第二组实验：选取 ORL 人脸数据库中 8 个人的人脸，每人 10 张人脸图片，其中前 8 张进行人脸识别训练，后面 2 张进行人脸识别的测试。如表 5.2 所示为人脸识别结果，共测试 16 张人脸，识别成功率为 100%。

表 5.2 第二组实验人脸识别结果

人脸识别 对象编号	人脸编号	
	face01	face02
personA	1.000000	1.000000
personB	1.000000	1.000000
personC	1.000000	1.000000
personD	1.000000	1.000000
personE	1.000000	1.000000
personF	1.000000	1.000000
personG	1.000000	1.000000
personH	1.000000	1.000000

对比实验结果发现：获取多幅人脸图像可以提高人脸识别的准确度和可靠性，在人脸识别训练过程中可以增加采集的人脸图像数目提高人脸识别的成功率和可靠性；另外也可以采集多幅图像求可信度平均值与阈值比较，当可信度平均值大于阈值时，显示识别成功的人脸姓名。

5.3 本章小结

本章首先利用 ORL 人脸数据库测试了本系统的人脸检测功能，表明系统对小角度的旋转人脸和不同表情的人脸均能够准确的检测到人脸区域；然后在利用 ORL 人脸数据库测试了系统的人脸识别功能，实验结果基本达到了系统设计的功能。

第六章 总结与展望

6.1 工作总结

本文就一种基于 Linux 的人脸识别系统的设计与实现过程进行了详细的论述。其中包括软硬件平台的选择,交叉编译环境的搭建,嵌入式 Linux 操作系统的编译与移植,基于 USB 摄像头的视频采集,人脸检测和人脸识别算法研究及应用程序开发,图形用户界面的开发等。本文所做的工作主要有以下四个方面:

(1) 实现嵌入式 Linux 系统的开发环境的搭建。使用基于 ARM11 的 Tiny6410 开发板作为系统的硬件平台,选择嵌入式 Linux 作为软件平台。在此基础上定制 U-Boot 作为系统的 Bootloader,完成 Linux2.6.36 内核的编译和移植,建立 Cramfs 根文件系统,在宿主机上配置安装了 arm-linux-gcc-4.5.1 交叉编译工具链。

(2) 实现基于 Video4Linux 的视频采集和人脸图像的预处理。使用普通的 USB 摄像头进行视频采集,利用 mmap()内存映射的方式将人脸图像映射到内存空间。人脸图像的特征提取和识别是在灰度图像上进行的,因此首先要把彩色图像转换为灰度图像。然后采用大小为 3×3 的滑动模板对图像进行中值滤波,对输入图像进行噪声消除处理。最后将图像直方图均衡化,增大反差,使图像的细节清晰,增强人脸识别的准确性。

(3) 实现基于 AdaBoost 算法的人脸检测。AdaBoost 算法是一种迭代方法,把一系列的弱分类器级联组合成一个强分类器,从根本上解决了检测的速度问题,同时具有较好的识别效果。采用积分图计算人脸图像的 Haar-like 特征,利用瀑布式级联分类器判别人脸,由于大多数待检测图像为非人脸,因而提高了分类器整体的检测速度。

(4) 实现基于特征脸算法的人脸识别。特征脸法是一种基于人脸全局特征的识别方法。它的核心是利用主成分分析方法提取人脸的特征向量,将每一张人脸图像看成是许多特征脸的线性组合。识别的过程是把一副新的图像投影到特征脸子空间,并通过它的投影点在子空间的位置以及投影线的长度来进行判定和识别,用最近邻分类器实现特征匹配。

6.2 进一步的展望

到目前为止,本文虽然搭建了基于 Linux 的嵌入式人脸识别系统,但它还有很多方面需要改进,才能走出实验室,形成产品。

首先，在算法选择方面。本系统选取经典 AdaBoost 算法和基于 K-L 变换的特征脸算法分别作为人脸检测和识别的方法，但在多视角，多光照和复杂背景等情况下的人脸检测和识别效果不理想，仍需要在算法上进行改进。

其次，在系统硬件配置方面。基于成本和实验复杂性的考虑，采用了普通的 USB 摄像头作为视频捕获设备。但在实际的人脸图像采集效果上并不理想，影响了实验结果的准确性，在今后的工作中考虑采用更加精确视频图像采集设备；从实验结果的效率上判断，为达到实时人脸检测识别的效果，应该为系统添加图形处理芯片。

最后，在系统的设计方面。本系统的核心部分是人脸识别。人脸识别作为生物识别技术中的一种，面临着诸多挑战，系统的实际应用受到很多条件的约束。例如，背景的复杂多变、人脸的姿态和表情、光照条件等等情况都对人脸的识别有较大的影响。因此，本系统只能在约束条件满足的情况下使用，随着新技术新方法的出现和使用，本系统的性能也有待提高。

致谢

在此论文完成之际，我要向很多人表示由衷的感谢，正是他们对我无私的帮助，才使我的论文可以顺利完成。

首先，我要衷心地感谢我的导师周端教授，在研究生这几年学习生活中，我不仅从老师这里学习到了丰富的专业知识，更是从老师这里学到了为人处事、对待生活的态度，使我的生活观有了一定的改善。在这篇论文的完成过程中，老师一直悉心指导我的论文结构、改善我的论文内容，让我感受到了我的态度似乎不够踏实，做事还应更加地踏实认真。同时，周老师在生活上平易近人、呵护备至令我意外感到了研究生生活的温暖，而且老师还经常教导我们要劳逸结合，培养热爱生活的精神，这将使我受用无穷。

同时也要感谢实验室里的每一位同学，在实验室大家庭里大家在生活上互相关心照顾，学习上互相探讨与勉励，形成了良好的氛围。在此，我要向赵培立、赵国亮、刘文博、王旭、陆阳、豆宣宣、曹雪芬、戴琪、白乐等同学表示感谢。

感谢在校期间每一个给我们讲课，照顾我的老师们，感谢我的朋友和同学们，在这里是你们陪我度过了我人生最重要的几年，使我成长为一个有能力有素质有知识的人，我取得的这些进步和你们的帮助是息息相关的，在这里祝你们身体健康，生活幸福，工作顺利。

感谢我的家人，是他们无私的奉献，支持和鼓励才能使我得以顺利完成学业，是我一直向前的动力。

参考文献

- [1] 邹思轶.嵌入式Linux设计与应用[M].北京:清华大学出版社,2002,5-18.
- [2] 吴国伟,毕玲,陈庆.嵌入式操作系统原理及应用开发[M].北京:北京航空航天大学出版社,2007,5-23.
- [3] David D.Zhang.Automated biometrics:technologies and systems[M].Boston: Kluwer Academic Publishers,2000,10-20.
- [4] Anil K.Jain.Biometrics Personal Identification in Networked Society[M].Boston: Kluwer Academic Publishers,2001,20-56.
- [5] Wildes,Richard P.Iris recognition:an emerging biometric technology[J],Proc.IEEE 1997,85(9):1347-1363.
- [6] 张铎. 生物识别技术基础[M] .武汉:武汉大学出版社,2009,2-10.
- [7] A.Pentland,T.Choudhury,Face recognition for smart environments,IEEE Computer,2000:50-55.
- [8] A.Samal,EA.yengar,Automatic recognition and analysis of human faces and facial expressions:a survey.Pattern Recognition,1992,25(1):65-77.
- [9] 周激流,张晔.人脸识别理论研究进展[J].计算机辅助设计与图形学学报,2001,11(2):180-184.
- [10] 田捷,杨鑫. 生物特征识别理论与应用[M]. 北京:清华大学出版社, 2009.5-15.
- [11] 林永仁著.嵌入式系统一项目分析入门与实践[M].北京:中国铁道出版社,2004,25-40.
- [12] 李善平,刘文峰,王焕龙.Linux与嵌入式系统[M].北京:清华大学出版社, 2002, 2-10.
- [13] 韦东山.嵌入式Linux应用开发完全手册[M].北京:人民邮电出版社,2006.
- [14] 康维新.嵌入式Linux系统开发与应用[M]. 北京:机械工业出版社, 2011.
- [15] 陈艳华,侯安华,刘盼盼.基于ARM的嵌入式系统开发与实例[M].北京:人民邮电出版社,2008.
- [16] 罗菀棠. 嵌入式Linux应用系统开发实例精讲[M]. 北京:电子工业出版社, 2005
- [17] 梁路宏,艾海舟.人脸检测研究综述[J].计算机学报,2002,25(5):449-458.
- [18] 孙纪坤,张小全.嵌入式Linux系统开发技术详解[M].北京:人民邮电出版社, 2006.
- [19] 许录平.数字图像处理[M],北京:科学出版社,2007:278-298.
- [20] Takeo Kanade.Picture Processing System by Computer Complex and Recognition

- of HumanFaces[J].PhD thesis,Department of Information Science,Kyoto University,1973.11.
- [21] Akamatsu:The IEEE Third International Conference on Automatic Face and Gesture Recognition,Nara,Japan,April 14-16,1998.
- [22] Lee,C.H.Kim,J.S.,Park. K.H.Automatic Human Face Location in a Complex Background Using Motion and Color Information[J].Pattern Recognition,1996, 29(11):1877-1889.
- [23] H.Zabrodsky et al.IEEE Trans.PAMI,17:1154-1166,1995.
- [24] PENG Jinye,YU Bianhang,WANG Dakai,LI Nan.Multi — Scale Symmetry Transform with Appl ication to Location of Feature Points on Human Face Image[J].Acta Electronica Sinica,2002,30(3):363-366.
- [25] I.Craw, H.Ellis,J.Lishman.Finding Face Features[C],Proc.Second European Conf.Computer Vision,1992:92-96.
- [26] H.Z.Ai,L.H.Liang and B.Zhang. Face Detection Based on Multiple Related Template Matching[C],Proc.of Japan-China Symposium on Advanced information technology, Tokyo Japan 1999, 283-294.
- [27] A.Yuille,P.Hallinan,and D.Cohen.Feature Extraction from Faces Using DefoFinable Templates[J].International Journal of Computer Vision,1992,8(2):99-111.
- [28] Propp M,SamaI A.Artificial Neural Network Architectures for Human Face Detection[J],Intelligent Eng.Systems through Artificial Neural Networks,1992, 2.
- [29] Osuna E.,Freund R and Girosi F. Training Support Vector Machines:An Application to Face Detection[C]. Pro.IEEE Conf. Computer Vision and Pattern Recognition,1997, 130-136.
- [30] Nefian A.,Hayes M. Hidden Markov Models for Face Detection[C].IEEE International Conf.on Acoustics,Speech and Signals Processing. Seattle,Washington,1998, 2721-2724.
- [31] VIOLA P.JONES M.Robust Real Time Object Detection[C]. 8th IEEE International Conference on Computer Vision.Vancouver.2001, 151-155.
- [32] 崔晓青.基于AdaBoost算法的人脸检测系统的研究与实现[D].长春:吉林大学计算机科学与技术学院,2008.
- [33] Papagcorgiou C, Oren M,Poggio T.A General Framework for Object Detection[C], International Conference on Computer Vision, 1998,555-562.
- [34] Viola P,Jones M.J.Robust Real-Time Face Detection[C]. International Journal of Computer Vision 2004, 57(2),137-154.
- [35] P Viola, M Jones.Rapid Object Detection Using a Boosted Cascade of Simple

- Features[C].In:Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001, 1: 511-518.
- [36] Lienhart R, Maydt J. An Extended Set of Haar-like Features for Rapid Object Detection[J], IEEE Image Processing, 2002, 900-903.
- [37] FREUND Y, SCHAPIRE R. A Short Introduction to Boosting[J]. Journal of Japanese Society for Artificial Intelligence, 1999, 14(5): 771-780.
- [38] 凡芝, 张兴周, 谢耀菊. 基于Adaboost的人脸检测技术[J]. 应用科技, 2005, 32(6): 7-9.
- [39] P. Viola, M. Jones, Robust real time object detection[J], 8th IEEE International Conference on Computer Vision (ICCV), 2001, Vancouver, British Columbia.
- [40] Kirby M, Sirovich L. Application of the Karhunen-Loeve procedure for the characterization of human faces[J]. IEEE Trans. Patt. Anal. Mach. Intell. 12.
- [41] 肖健华. 智能模式识别方法[M]. 广州: 华南理工大学出版社, 2006: 235-260.
- [42] 张尧庭, 方开泰. 多元统计分析引论[M]. 北京: 科学出版社, 1982: 175-203.
- [43] 边肇祺, 张学工. 模式识别, 北京: 清华大学出版社, 2000, 1: 223-228.
- [44] I. T. Jolliffe. Principal Component Analysis[M]. Springer-Verlag, New York, 1986.
- [45] M. Turk, A. Pentland. Eigenfaces for recognition[J]. Cogn. Neurosci, 1991, 3(1): 71-86.
- [46] 田浩. 数字图像处理原理与应用[M], 北京: 清华大学出版社, 2007: 202-230.
- [47] 邹思轶. 嵌入式Linux设计与应用. 北京: 清华大学出版社, 2002.
- [48] 刘瑞祯, 于仕琪. OpenCV教程-基础篇[M]. 北京: 航空航天大学出版社, 2007.
- [49] Mathew Price. Using the Intel OpenCV Library[M]. Ebisawa: Pupits and Nostrils, 2004.
- [50] 韩存兵, 龚波. 构建嵌入式Linux系统. 北京: 中国电力出版社, 2004.
- [51] 李驹光, 聂雪媛, 江泽明. ARM应用系统开发详解. 北京: 清华大学出版社, 2003.
- [52] 陈建, 宋健建. Linux程序设计[M], 北京: 人民邮电出版社, 2003.
- [53] 李侃, 廖启征. 基于S3C2410平台与嵌入式Linux的图像采集应用[J]. ARM开发与应用, 2006, 22(8): 125-168.
- [54] 周立功, 陈明计, 陈渝. ARM嵌入式Linux系统构建与驱动开发范例[M]. 北京: 北京航空航天大学, 2005: 211-236.
- [55] 黎冰, 吴松, 曾凡涛. 人脸识别在智能手机中的实现[J]. 计算机工程, 2006, 32(7): 272-274.
- [56] 陈胜勇, 刘盛. 基于OpenCV的计算机视觉技术实现[M]. 北京: 科学出版社, 2008.
- [57] 郭磊, 王秋光. AdaBoost人脸检测算法研究及OpenCV实现[J]. 哈尔滨理工大学学报, 2009, 14(5): 123-126.
- [58] Matthew Turk, Alex Penland. Eigenfaces for Recognition. Journal of Cognitive Neuroscience, 1991, 3(1): 71-86.

