

分类号: TP391.4

单位代码: 10110

学 号: s20100865

中 北 大 学  
硕 士 学 位 论 文

基于 FPGA+ARM 的人脸实时  
检测系统研究

博士研究生\_\_\_\_\_张挺飞\_\_\_\_\_

指导教师\_\_\_\_\_樊永生\_\_\_\_\_

学科专业\_\_\_\_\_计算机应用技术\_\_\_\_\_

2013 年 05 月 日

图书分类号 TP391.4

密级 非 密

UDC <sup>注1</sup> 681.6

# 硕 士 学 位 论 文

## 基于 FPGA+ARM 的人脸实时检测系统研究

张挺飞

指导教师（姓名、职称） 樊永生 教授

申 请 学 位 级 别 工学硕士

专 业 名 称 计算机应用技术

论 文 提 交 日 期 2013 年 月 日

论 文 答 辩 日 期 2013 年 月 日

学 位 授 予 日 期 2013 年 月 日

论 文 评 阅 人

答 辩 委 员 会 主 席

2013 年 05 月 日

## 原创性声明

本人郑重声明：所呈交的学位论文，是本人在指导教师的指导下，独立进行研究所取得的成果。除文中已经注明引用的内容外，本论文不包含其他个人或集体已经发表或撰写过的科研成果。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。本声明的法律责任由本人承担。

论文作者签名：\_\_\_\_\_ 日期：\_\_\_\_\_

## 关于学位论文使用权的说明

本人完全了解中北大学有关保管、使用学位论文的规定，其中包括：  
①学校有权保管、并向有关部门送交学位论文的原件与复印件；②学校可以采用影印、缩印或其它复制手段复制并保存学位论文；③学校可允许学位论文被查阅或借阅；④学校可以学术交流为目的，复制赠送和交换学位论文；⑤学校可以公布学位论文的全部或部分内容（保密学位论文在解密后遵守此规定）。

签 名：\_\_\_\_\_ 日期：\_\_\_\_\_

导师签名：\_\_\_\_\_ 日期：\_\_\_\_\_

## 基于 FPGA+ARM 的人脸实时检测系统研究

### 摘 要

随着图像处理技术的快速发展,视觉检测技术已经应用到航天,工业,交通,家居,通信等各个领域,是当今最热门的技术领域之一。图像检测技术具有其它技术无法匹敌的优点——非接触式和无侵犯性。目前图像检测技术大多在 PC 平台上开发的,但其具有体积庞大、携带不便等缺点,在一些特殊场合并不适用。而嵌入式系统恰恰弥补了 PC 平台的缺点,并且具有功耗低、成本低等优点,因此设计一种基于嵌入式系统的图像检测系统有非常大的实用价值和广泛的发展前景。

本文设计了一种基于嵌入式系统的人脸检测系统,实现了图像采集、处理、检测及结果显示等功能。目前人脸图像检测技术一般采用 Adaboost 算法,本文详细介绍了 Adaboost 算法的 Haar 特征、积分图像、分类器以及级联分类器,系统的硬件平台使用 FPGA+ARM 组合的方式实现,有效的提高了系统的处理速度和兼容性,同时针对图像采集环节进行了优化处理,对图像的缓存进行乒乓操作,大大提高了图像采集的效率,提高了图像处理的实时性。linux 系统具有开源、体积小、可裁剪、功能齐全等优点,是目前嵌入式领域主流的操作系统,通过在 ARM 嵌入式系统中移植嵌入式 Linux 操作传统,使系统具有处理复杂情况的能力。图像处理技术软件的设计一般难度大、成本高、工作量大且开发周期长,而 OpenCV 的出现,为图像处理技术的广泛应用做出了重大的贡献,OpenCV 是图像处理技术爱好者开发的开源计算机视觉库,包含很多图像处理及计算机视觉算法,有效地提高开发效率,本文详细的阐述了 OpenCV 在 ARM 中的移植过程以及基于 OPenCV 的人脸检测程序的设计。

最后,实际搭建了图像采集的硬件平台并移植了设计好的软件程序,通过实验验证了系统的正确性和稳定性。

**关键词:** Adaboost, 图像处理, 嵌入式系统, OpenCV;

## Abstract

With the rapid development of image processing technology, visual inspection technology as one of today's hottest technical areas has been applied to the aerospace, industrial, transportation, home, communications and other fields. It has the merit of non-contact and non-offensive which are unable to compare by other technologies. Since most of the current image detection technology developed on the pc platform has the disadvantages of bulkiness and poor portability, It is not suitable on some special occasions. The embedded systems with the advantages of small size, low power consumption and low cost is just to make up for the disadvantages of PC platform, Therefore, design an image detecting system based on embedded system has great practical value and broader development prospects.

The paper designed a face detection system based on embedded system, and realized the imaging, processing, examination, result demonstration and other functions. Since the present face image inspecting technology uses the Adaboost algorithm generally, The paper first introduced the Haar characteristics, the integral image, sorter as well as the cascade sorter Adaboost algorithm in detail, then using a combination FPGA+ARM as the hardware platform of system which significantly improved the processing speed of the system, and compatibility, meanwhile, carried on the optimized processing of Image acquisition. Through the ping-pong operation to the buffer of image, the image acquisition efficiency and the timeliness of image processing was greatly improved. By transplants the embedded Linux operation tradition which has the merits of open source, small size and scalable to the ARM embedded system, Strengthened the processing complex situation ability of system. Generally, the design of image processing software was difficult, High costs, heavy and long development cycles. But with the emergence of OpenCV, has made the significant

contribution for the imagery processing technology widespread application. The OpenCV as an open source computer vision library developed by enthusiasts, Contains many image processing and computer vision algorithms, effectively improved the develop efficiency, This article detailed elaborated transplant process of OpenCV in ARM, and designed OpenCV based face detection procedures.

Finally, the image acquisition hardware platform was build and the software procedure was designed, the experimental results proved the correctness and stability of the system.

Keywords: Adaboost, image processing, embedded system, OpenCV;

## 目 录

第一章 绪 论.....	1
1.1 课题的背景及意义.....	1
1.2 国内外发展现状.....	2
1.3 论文的主要研究工作.....	4
1.4 论文的结构安排.....	4
第二章 基于 ADABOOST 算法的目标检测技术.....	6
2.1 经典方法概述与 ADABOOST 算法的选取.....	6
2.2 矩形特征及积分图.....	9
2.2.1 矩形特征.....	9
2.2.2 积分图.....	11
2.3 算法简介.....	12
2.4 样本的构建.....	14
2.5 分类器的训练.....	15
2.5.1 弱分类器.....	15
2.5.2 强分类器.....	17
2.5.3 级联分类器.....	19
2.6 本章小结.....	22
第三章 基于 FPGA+ARM 的高速图像数据采集处理平台.....	23
3.1 CMOS 图像传感器硬件电路设计及功能介绍.....	23
3.1.1 OV6620 图像传感器基本工作原理.....	23
3.1.2 OV6620 驱动时序.....	25
3.1.3 SCCB 总线协议的时序分析.....	25
3.1.3.1 SCCB 总线协议的时序分析.....	26
3.1.3.2 SCCB 总线协议的时序分析.....	27
3.2 FPGA 控制模块电路设计及功能介绍.....	29

3.2.1 电源模块.....	29
3.2.2 时钟模块.....	30
3.2.3 程序下载电路.....	30
3.2.4 OV6620 电路.....	31
3.2.5 FPGA 工作流程.....	32
3.1.5.1 FIFO 缓存探测器数据仿真.....	33
3.1.5.2 SRAM 读写模块仿真.....	34
3.1.5.3 灰度值预处理模块仿真.....	35
3.3 ARM 嵌入式系统的硬件结构.....	36
3.4 嵌入式系统软件平台的建立.....	38
3.4.1 嵌入式系统的介绍.....	38
3.4.2 交叉编译开发环境的建立.....	41
3.4.3 引导程序的移植.....	42
3.4.3.1 BOOTLOADER 引导程序.....	42
3.4.3.2 U-BOOT 的介绍.....	43
3.4.3.3 U-BOOT 的启动过程.....	44
3.4.3.4 U-BOOT 的移植.....	45
3.5 LINUX 操作系统的移植.....	45
3.5.1 LINUX 内核的移植 .....	46
3.5.2 根文件系统的移植.....	49
3.6 本章小结.....	50
第四章 基于 OPENCV 的实时人脸检测算法的实现.....	54
4.1 OPENCV 简介.....	54
4.2 人脸分类器的训练.....	56
4.2.1 样本的创建.....	56
4.2.2 分类器的训练.....	56
4.3 OPENCV 在 ARM 平台上的移植.....	58
4.4 基于 OPENCV 的人脸检测.....	59



4.5 实验结果及结论.....	60
4.6 本章小结.....	60
第五章 总结与展望.....	61
6.1 本论文的主要研究工作.....	61
6.2 工作的不足及下一步的工作.....	62
参考文献.....	63
攻读硕士学位期间发表的论文.....	68

## 第一章 绪论

### 1.1 课题的背景及意义

迄今为止,机器视觉已经经历了一个漫长的发展历程<sup>[1]</sup>,自 21 世纪起,伴随着半导体技术、计算机技术、微电子技术和嵌入式技术的飞速进步,智能识别技术更是取得了长足的发展<sup>[2-5]</sup>。现代社会已经步入一个信息化的时代,现代社会对人机交互系统、远程视频系统、计算机动画设计、视觉监视监控系统和自动身份验证等技术的需求也越来越广泛,与之对应的人工智能技术也开始受到广泛关注<sup>[6-8]</sup>。

生物特征识别技术(biometric identification technology)是人工智能领域研究的主要研究方向<sup>[9-11]</sup>,它是一种将信息技术与生物技术相结合的新型识别技术,

它的工作原理是利用人们本身具有的特质,通过处理器将各类光、声、电等传感器物理技术,还有数学统计学方法整体结合来识别这个人的身份。这样的方法的优势在于每个生物都具有自己本身独一无二,且无法被改变、遗忘和复制的特性,所以,这样的方法能在识别生物身份上的可靠性,安全性,便捷性上远远超过超越过去的识别技术。

之所以要进行人脸检测的实验研究<sup>[12]</sup>,不仅仅是为了让人的视觉控制系统在识别事物上有更好的发展,更是为了适应促进人工智能的发展。人脸自动匹配系统是利用人脸识别技术设计出来的,而人脸识别是利用计算机系统去自动进行的,其优点在于:(1) 非接触,无需被测对象与测试设备进行接触;(2) 无侵犯性,图像可以在被测对象未进行任何配合的条件下进行,避免了被测对象不配合或预先准备的情况发生;因而该技术的发展前景广阔。

另外,该技术还具有很高的学术意义,对人脸及器官的检测、定位和识别技术还不仅仅是追踪人的头部的运动,或是从身体形态、运动方式上获得信息,还有对手势、面部表情甚至唇语的辨别认知等等。根据这些不同的方式,去了解人类会在什么样的环境下有什么样的举动,从而设计出更加完美,人性化的人机交互方法,能够充分地满足用户们的要求,针对不同的人提供不一样的服务。

过去在设计利用人脸判别身份的技术上主要通过 PC 机完成，但其体积大、加个高昂的问题限制了其应用领域，随着以 ARM，FPGA 为代表的嵌入式处理器高速发展，其运算速度正在不断赶超 PC 机处理器的运算速度，因而利用嵌入式系统开发出具有实时检测能力的人脸识别系统，在体积和成本上将具有极大优势，应用空间更为广泛<sup>[13]</sup>。

FPGA 对实时任务有很强的支持能力，能完成多任务并且有较短的中断响应时间；同时具有功能很强的存储区保护功能，可扩展的处理器结构的特点，FPGA 算法可硬件化带来的高速性恰恰满足了人脸识别系统对速度的要求，同时，ARM 系统以其高速的运算能力，丰富的接口，便捷的开发环境以及更容易实现人机友好界面等功能深受开发者的欢迎，用户可以自定义功能强大的用户自定义模块对功能进行提速或者加强。。因此，本论文采用 FPGA+ARM 相结合的嵌入式系统，将对人脸识别系统性能和速度加以提升，满足应用需要。

## 1.2 国内外发展现状

世界上没有两片相同的叶子，而每个人的模样也没有完全一样的，因次利用人脸去辨别身份是一种最普通也最有效的视觉功能，但是在科技不够发达的情况过去，无法真正获取到人脸来验证人脸识别方法是否可行，而这样不够真实，不够具有科学性的算法在如今已被逐渐淘汰，随着对这方面的需求增大，人们要更加精确，更实用的人脸分析技术。因此，人脸检测从研究中分离了出来，被更专业地化研究<sup>[16-17]</sup>。

如何将人的面孔检测出来是进行人脸辨别的基础，即在一幅图像中，区别出何为背景，何为景物，何为我们所需要的目标以及其在图中的具体信息<sup>[18]</sup>。这一步检测分为两类，静态和动态。顾名思义，静态检测只用在一幅图中或者静止的录像中发现人脸的存在以及他的数量，每个脸部有多大，所在的位置等信息。相反的，动态检测的区别就在于这是在一连串的图像视频中完成这个任务。

这项工作大概是在 1970 年到 1980 年之间开始的，最初的研究并没有如今这么先进具体到人脸检测这个算法，而是比较简单的，如模版配对、子空间、变形模版配对等方法<sup>[19-22]</sup>，这些方法只能应用于背景单一无任何变动的情况下，而且实用性很差，十分笨拙。因此，在使用这些方法时，针对于不同的图像或者背景，都需要对全部的数据，系统设计上进行大篇幅的修改，非常不便于使用。因此，从那时起，科学家们都在不断研

究更加快捷且能适应各种不同情况环境的识别方法。例如，将整张图像进行多种方式的分割，根据不同的姿态动作，或者皮肤颜色，或是更加普遍的信息收集方式，还有新型的利用神经网络的方法，在很大程度上提高了对面部检测可能性和实用性。同时，人们也设计出了更加灵活的人脸的信息获取方式，弹性模版以及活动轮廓的设计，使得机器能够对面部的保持追踪并准确定位，再根据面部的特点去获取其面部特征这一技术成为可能。

如今科技迅猛发展，人的面部检测的研究成为许多研究人员的主要研究项目，特别是国外的一些大学研究所，如大众所知的美国的麻省理工学院，有自己的媒体研究室和人工智能实验室，还有南加州大学以及卡内基梅隆大学的卡内基—梅隆机器人研究机交互系统研究室等等<sup>[23-29]</sup>；因此，研究者们人脸探测上的能力和技术都有了很大的提升，这也表现在所发表的非常有价值，经典的国际论文，比如众所周知的图像处理国际会议、电气与电子工程师协会的人脸手势自动识别、电脑视觉及图形辨别等这些含金量非常高的国际会议组织，都会发表很多关于这方面的论文，而在关于人的面部研究的论文里，对于人脸检测的文章就有三分之一这么多。一般来说，在研究人脸识别的文献里都会提到人脸检测这一部分，因为这是一个基础，但大部分只是介绍了如何输入人脸识别这一系统，而没有对面部检测的一些具体算法和模型特征进行详细描述。

在国内，一些比较著名的重点大学或者理工类院校也有这方面的研究，比如清华大学、交大、北京工业大学以及中科院的计算、自动化研究所等<sup>[30-36]</sup>。但是必须承认的是，我国在人脸检测这方面的研究确实还不够深入，对面部探测系统的最主要的技术，最需要解决的问题上所获得的成果还很少，虽然我国的许多研究人员都不断在人脸的检测，面部的追踪，个人特征的识别获取上进行探讨，调查，研究，也取得了一些收获，但是像是在标准人脸模版的建立，如何评测识别的准确度等根本问题上需要更多的探索。国家多媒体内容描述接口标准组织也为此设立了小组进行了人脸检测算法的征集和研究。

1995 年左右，人们在人脸识别检测的研究上已取得一定程度的成功，由于这方面的技术已有一定程度的成熟度和可靠性，许多技术公司开始提供基于人脸检测识别这一技术和系统，如 Identix 公司、Eyematic、Viisage、LAU、eTrue、BioID、VisionSphere 等<sup>[37-40]</sup>，同时，为了有更好的面部识别系统，满足用户使用需求，这些公司都对其系统，

技术采取许多改进策略，在算法上进行了大量的修改，使其更加工程化系统化。不过由于这已经涉及到商业秘密，后期这些技术和算法不得公开发表，因此很难得到这些人脸识别检测的具体过程的信息。

目前在嵌入式上实现人脸检测基本上都是采用软件移植的方法<sup>[41-46]</sup>，通过软件实现的方式来进行人脸的检测，但是当前的 PC 机处理能力还无法达到实时处理的要求，以在速度与精度的综合性能上表现最好的 Adaboost 模型为例<sup>[47]</sup>，以软件运用为主的 Adaboost 算法若要进行一次图像的分析运算时，基本上 CPU 的全部计算能力都要要用掉它上面，即使如此，软件实现 Adaboost 算法的处理能力还是达不到实时性要求。由于这个原因，产生能过成功运行进行人脸检测算法的嵌入式硬件成为当务之急。

综上所述，虽然如今的人脸检测技术还未完全成熟，但从一开始的初步入门到现在的可运用技术，人脸检测的研究的发展已经有了质的飞跃。根据一些国际上具有绝对的发言权的评估结果，以及我国自身一些比较完善准确的评估系统的结果来看，如今在人脸检测技术这一块的发展已经能达到一个比较完善的阶段，可以在一些比较理想的情况能够有非常实际，准确的应用。而将嵌入式技术与人脸检测技术相结合实现的应用应该是未来最有发展前景的应用，因为人脸识别这一技术会在未来有更广泛的运用，更大的发展空间，但是这方面的研究还很匮乏，未来人们应该投入更多精力在这一技术的研究上。

### 1.3 论文的主要研究工作

本文的主要研究目的在于将图像的收集，分析运算，以及最终定位的结果放在集 FPGA 和型号为 UT6410BV04 的 ARM 为一体的平台上进行，以便于更好的，更全面地发挥该系统所具有的能力，这是一种新的挑战和创新，让人脸检测的简单化，实用化成为可能。本文所运用的结构是利用 FPGA 和 ARM 技术，进一步达到时刻进行图像处理分析的目的。另外，为了使操作系统能具备更加全面，智能性的处理问题的能力而采用了 Linux 系统，因为这一款操作系统不仅具有可移植性，而且能随意裁剪且保持稳定的性能。这样所生产的系统实用性非常强，不仅仅在技术上能广泛运用于各种复杂多样的问题，而且实际生产运用所需要的成本小，消耗少。

根据文章内容，本文大致分为以下几大块：

(1)从时刻人眼定位获取系统所需要的功能上，对软件和硬件进行全面的挑选，例如性能的好坏，价格是否合理，实验室是否具备所需的实验条件等等，并依据嵌入式系统开发平台设计要求进行了系统设计。

(2) 将 Adaboost 算法应用于人脸检测中，构建了一个实时人脸检测系统。

(3) 系统的分析了 OPenCV 的结构，并将带有开源视觉处理库的 OPenCV 移植到嵌入式 ARM 平台上。

## 1.4 论文的结构安排

第一章以介绍该课题的研究背景为主，明确研究该课题的目的及意义，当前的技术发展程度和本文的核心内容。

第二章介绍了基于 Adaboost 算法的人脸检测的原理，对该学习算法的原理做了详细的分析和研究。其中包括如何利用积分图像快速地计算 Haar 特征、弱分类器、强分类器以及级联分类器的训练方法。

第三章从基于所需要达到的功能，在嵌入式操作系统，微处理器的型号和硬件模版的选用上进行了分析，搭建了系统所需的软硬件平台。

第四章介绍了 OpenCV 库的功能及优势，重点研究了如何移植 OpenCV 到 ARM 嵌入式系统，以及使用 OpenCV 来实现实时人脸检测。

第五章总结全文并提出了本文还存在的一些不足之处。

## 第二章 基于 Adaboost 算法的目标检测技术

### 2.1 经典方法概述与 Adaboost 算法的选取

人脸检测本质上是关于如何区分二值的问题，这二值即指一幅图中的“人脸”和其余物体。人脸检测就是为了去判断人脸部分和非人脸的部分，如今，人脸检测的技术逐渐发展成熟，对此的研究也更深刻，也创造了更多实用的人脸辨别方法。基本上这些方法可分为两大类，根据知识理论辨别的方法<sup>[48]</sup>和根据统计理论来辨别的方法。

第一类通过知识的办法通常是根据基本的形状特征，如灰度、纹理、几何图形和这些基本形状知识特征综合来判断一幅图中是否有人脸的存在。知识规则可根据以前的知识概念得到，也可以从过去的经验总结获得。通常，人的脸部特点以及自身之间的关系可以通过十分简单普遍的规则来定义，当检测一副图像时，首先要获得面部的特征，一般是根据编码规则而选定的人脸区域。该方法的问题为，不能明确地将知识定义为确定的规则。另外，由于将全部情况都罗列出来是不可能的，所以这样的方法在进行不同的位置姿势的检测时，是很难执行的。

杨等人因此发现了一种以镶嵌图即马赛克图（Mosaic）为基础的人脸检测技术<sup>[49]</sup>，并创造了一种以知识为基础的三层检测系统。首先，要将人脸的区域分割为  $4 \times 4$  个马赛克块，每一块中各个像素的灰度平均值成了每块的灰度值，并根据一组规则来检测图像以获取有可能为人脸的区域。接着，再将获取的人脸区域分割为  $8 \times 8$  个马赛克块，再一次构建马赛克图，利用另外的规则来检验是否为人脸，最终根据上述所说的二值法将人脸区域分割，再通过边缘的特性最终确定。

统计方法中最主要的且应用最多的人脸检测方法是以下几种：基于子空间的、神经网络、支持向量机、隐马尔可夫模型和 Adaboost 算法：

基于子空间的方法<sup>[50]</sup>是把图像做 KL 变换达到消去各自分量之间存在的相关性的目的，再将这一步的结果中最大值的特征向量用来代替表示之前的图像，这样的做法只用保存原本图像具有最大差异的信息量，我们称这些分量为主成分。Turk 和 Peniland 利用

主成分分析分析来面部图像集合，同时使之变成很小的特征向量集合，称为“特征脸”，这些特征向量集合是原始的用来训练的图像集合基础部分，对用来训练非人类图像集合做同样的处理。在这个过程中，人的脸部的改变很少，然而其他区域的改变非常大，所以利用这个特征，可以检测图像在所有人脸的位置，再计算得到这些潜在人脸图像与人脸空间间的距离，真正的人脸所在点便是这些距离的最小值的位置。杨等提出的 Fisher 线性判别方法（FLD）是所有线性判别方法中的最被认可的方法。首先，根据自组织映射（SOM）原理同时把人脸和非人类集合分为 25 类，再计算出每一个类间的离散度和每一个类内的离散度，并通过这两种计算结果的比值得到投影矩阵。和本征脸方法比较来看，Fisher 线性判别方法结合了类内、类间的离散度得到的投影方向更加精准，效果更好，所以，Fisher 线性判别方法获得了比较好的检测成果。

神经网络(Neural Network)方法<sup>[51]</sup>是一种基于大脑神经突触互相连接的结构的方法，是一种用于处理信息的模型。这个模型中又许多被称为“神经元”、“单元”的节点，它们互相连接构成“神经网络”，利用这个网络来进行数据分析处理，并且能快速地进行大量运算。神经网络利用并行分布的结构，无限地近似于各种复杂的非线性结构，并能够运算大量数据。同时，神经网络能自我学习，通过这个能力，在进行图像识别的过程中，只要将各类图像的原始样本和其得到的辨别结论同时输入到神经网络中，它就能学会辨识这些图像。神经网络的优势在于成功训练一个可以检测人脸的系统时非常简单的。但是它的缺陷在于要通过非常大程度地调整各个层数、节点数、学习速率等方面的问题才能有得到人们对其性能上的要求。在基于神经网络的人脸检测上，罗利等人曾在 199 年是最早的成功做出一定成果的工作。他们利用局部连接的神经网络，将图像分割为多个标准几何区域来更好表现人脸的局部特点，各个区域有着一个隐含的神经网络单元，判断神经网络的分级的结果的而减少神经网络的错误率。

支持向量机(SVM)<sup>[52]</sup>：支持向量机是 Vapnik 等提出的基于结构风险最小化和统计学原理的理论，支持向量机的理论依据是先将输入的空间在非线性变换后转换到高维度的空间，并从这个空间里计算得出具有最优线性的分类面。最优分类面的概念是能令结构风险最小，而分类间隔最大的分类面。基于结构风险最小化原理令 VC(Vapnik Chervonenkis)维数有最小化上限，因此，比起基于经验风险最小化的神经网络，支持向量机有着更优秀的泛化方法。而奥苏娜等人把支持向量机用在了人脸检测方式上，同时实验的结果也令人满意。基于支持向量机的检测方法时利用支持向量机将所有  $19 \times 19$



像素的检测窗口分门别类，以划分而值窗口。但是这个方法被限制于需要高的计算能力去解决的二次优化问题。因此，普莱特利用 SMO 来解决这个困难。而 Bemd 等人设计了一个将 4 个线性支持向量机和 1 个非线性支持向量机串联起来的，具有层次性的支持向量机分类器。在进行人脸检测时，其中的支持向量机分别根据  $3 \times 3$ ， $4 \times 4$ ， $11 \times 11$ ， $19 \times 19$  的窗口去全面搜索检测输入的图像，迅速地将大部分的图像背景非人脸区域排除，留下少数的区域用非线性的支持向量机利用  $19 \times 19$  的窗口来做最后的判断。

隐马尔可夫模型<sup>[53]</sup>是一组经常用来对信号统计进行描述模型(HMM)。它的工作原理是根据信号统计特征的变化、通过观察序列可以使得马尔可夫链被间接地描述出来，由此可见，马尔可夫过程本身具有双重工作过程。一般情况下节点代表事物的发展状态，而有向边则表示状态之间的转移情况，在一定程度上可以表现出来某个特定空间的任意特征，在这个相同的特征里，不一样的状态则会表现出不同的特征。它们发生的概率也都不尽相同，前面提到 HMM 是一个统计模型，所以它对于相同特征的序列可能会用许多不同状态的序列来表示。由于 HMM 在运算过程中相对于状态序列，他们是无法看出来的，所以我们将之称为隐马尔可夫模型。

基于 Adaboost 的人脸检测方法<sup>[54]</sup>在检测时根本无需提供人脸的显著特殊，此方法会从许多典型的数据信息中获取并进行特定的分析，得出人脸图像，这也是人们普遍使用它进行人脸识别的最根本原因。

## 2.2 矩形特征及积分图

人脸检测从研究走向实用的一次质的飞跃是在 2001 年间，实时人脸检测系统的方法速度非常快可以做到每秒 15 帧，其在坚持速度及精确性上的表现让大家惊叹，当时计算矩形特征的方法是积分图方法，简单说就是用计算矩形特征得到的结果来训练分类器。

### 2.2.1 矩形特征

#### (1) Harr-like 的定义阐述

简单矩形特征也就是统称的 Harr-like，学者们将其概述为黑白两色的矩形在分窗口相应的交叉范围合数的差就是 Harr-like，2001 年时用到的特征总共分为三大类，如下图

第一行的矩形属于第二个矩形的基本特征，第二行左边的第一个矩形又属于右边第三个矩形的特征。据此类推每个图像都能从 Harr-like 特征中得到超过 18 万个特征，咋一看数量实在巨大，但是实际上不会需要这么多特征，在选择特征时可根据实际操作进行减少。

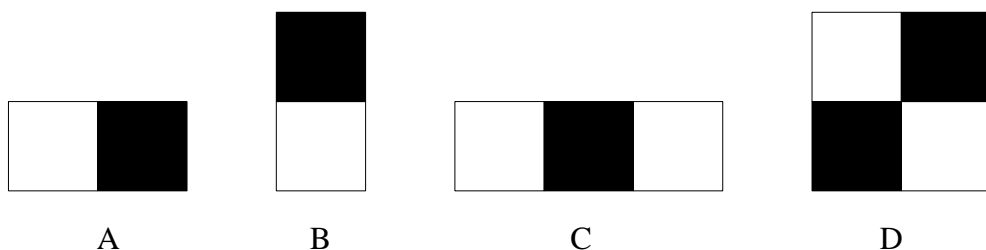


图 2.1 四种矩形 Haar-like 特征

## (2) 离散沃尔什变换的基本特征

在数字信号进行处理的时候用到的正交变换方法就是离散沃尔什变换 (Walsh)，离散沃尔什变换与离散傅立叶法有相同的原理，其对数字信号频率的反应非常灵敏，它的变换矩阵只会是非常简单的正负数字 1 组合而成，这样的好处是大大减少了计算量。它的结构为：

① 设离散沃尔什变换特征算子的高度、宽度分别为  $x$ 、 $y$ （注意  $x$ 、 $y$  必须是 2 的整数次方）

② 先假设  $y \times y$ 、后假设  $x \times x$  维的沃尔什变换矩阵为  $H_x$ 、 $H_y$

③ 得到 Walsh 特征算子： $W_{i,j} = (H_m)_i^T (H_n)_j$

由上图可以看出沃尔什在提取图像的某个局部特征时不会产生冗余，因为所有的  $x \times y$  维的沃尔什特征算子都出现相互正交的情况，而且形状要比 Harr-like 特征算子更为丰富。另外，沃尔什特征算子中要求  $x$ 、 $y$  都必须是偶数 2 的整数次方，所以它的全部沃尔什特征数就是有限的，Walsh 特征的所有量一定比 Harr-like 特征量小<sup>[32]</sup>。而 Harr-like 特征在整个提取的过程中会出现很多相近的特征，用起来相当麻烦而且冗余也相当大，所以采用沃尔什特征来代替 Harr-like 是非常好的方法。

## (3) Gabor 的特征阐述

几何特征、PCA 特征、小波特征、边缘特征、Gabor 特征等许多特征都是从人脸图

像中提取得特征类型，Gabor 特征和其它特征相比因其与哺乳动物对图像信号的响应非常的相似<sup>[33][34]</sup>，所以在刻画细节与局部结构能力上表现的更胜一筹，可以多元化得对人脸图像做出更多的描述。Gabor 变换通常也用作对图像的多分辨率分析，人的视觉系统具有对数频程的特征，因此在通常采用非均匀采样方法，由于人的视觉系统具有对数频程能力，因此 Gabor 小波展开时也使用的这种方法。可以取一系列以等量倍数增长的波长和间隔的方向构成多方向、多角度的 Gabor 滤波器组<sup>[35]</sup>。人们在使用 Gabor 滤波器组对图像进行获取的时候，基本上都是根据滤波器进行输出所得到的绝对值或者是幅值进行操作的。

### 2.2.2 积分图

2001 年 Viola 等人在提出以上几点的同时也提出了积分图（Integral Image）概念，提出积分图的目的是为了更加迅速准确的获取矩形表征。倘若  $i(x,y)$  是一幅图像的像素灰度，那么其积分图像值是：

$$ii(x,y) = \sum_{x_1 \leq x, y_1 \leq y} i(x_1, y_1)$$

只需一次逐点扫描原图像就可计算出来这幅输入图像 I 的积分图像值是多少。

$$s(x,y)=s(x,y-1)+i(x,y)$$

$$ii(x,y)=ii(x-1,y)+s(x,y)$$

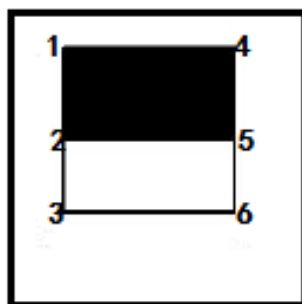


图 2.2 求矩形特征像素差示例

通过计算积分图上六个点就能得到两个矩形内像素和的差，同理通过计算积分图上的八个点就能得到三个矩形内像素和的差。如下图所示，可以通过扫描图像求出下图各

点像素的数值，由此可推计算  $(5+1-2-4)$  就可得下图中灰色区域的像素值的和了，则通过计算  $(2+6-3-5)$  可得下半部分区域像素值的和，综合以上区域的值，那么通过  $(5+1-2-4-2-6+3+5)$  的计算得出该矩形 Harr-like 特征值为两个公式的差，则如图 2.2 所示：

Harr-like 的主要特征主要通过三大类表现出来，两个矩形内像素灰度值分别求和之差就可以得出 a 与 b 的矩形 Harr-like 特征，而要算出 c 的矩形 Harr-like 特征则需要用两边矩形框的灰度值之和减去中间矩形的灰度值之和，同理用两条对角线的矩形框的灰度值之和的差值就可以得出 d 矩形 Harr-like 的特征。

## 2.3 AdaBoost 算法的简单介绍

P.Viola 先生是首位提出 AdaBoost 算法的人，方法大致上是：提取出目标的是特征，然后从目标图像上提取特征值，再结合目标的形状信息和运动信息进行 Adaboost 分类器的训练。通过训练最终证明这种检测器的准确性较高，只在某些特定环境会出现准确性下降的情况。

实际上 AdaBoost 算法简单的说就是一种把比较弱的分类方法结合在一起的分类方法。像图 2.3 所说的，如果要用小段线把图 a 中的红色、深蓝色球分开来，像图 b 和图 c 那样仅仅画一条线是不能正确分开的。

如果如图 b 那样先画出一条错误率最小的线段，深蓝色球就会被划分到红色区域，那么在下次划分区域时，我们首先要考虑的就是那部分深蓝色的球，（如 c 图 所示），因而最终会得到一个如图 d 所示的准确划分。

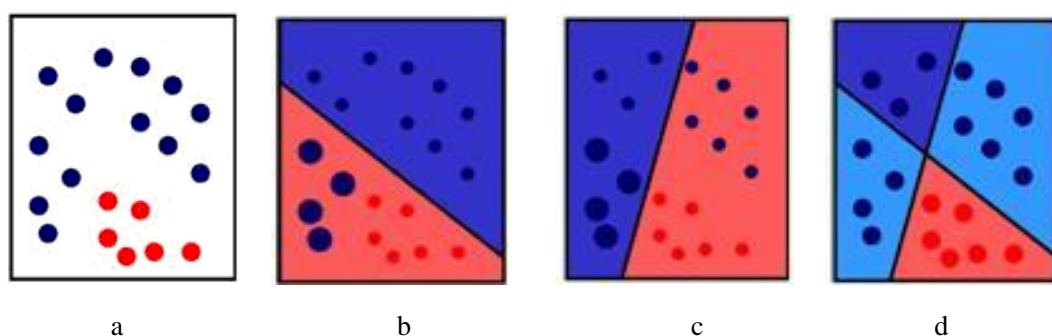


图 2.3 红球和蓝球的划分

Adaboost 算法是以积分图、级联检测器和 Adaboost 算法为根本的方法，该方法主要由：使用“积分图”快速计算特征值；使用 Adaboost 算法挑选出矩形特征也就是弱分类

器，再将弱分类器构造组合为一个强分类器；串联若干训练后的强分类器组成级联结构的层叠分类器这三大部分组成的。

具体 Adaboost 算法的检测系统框架如下图所示：

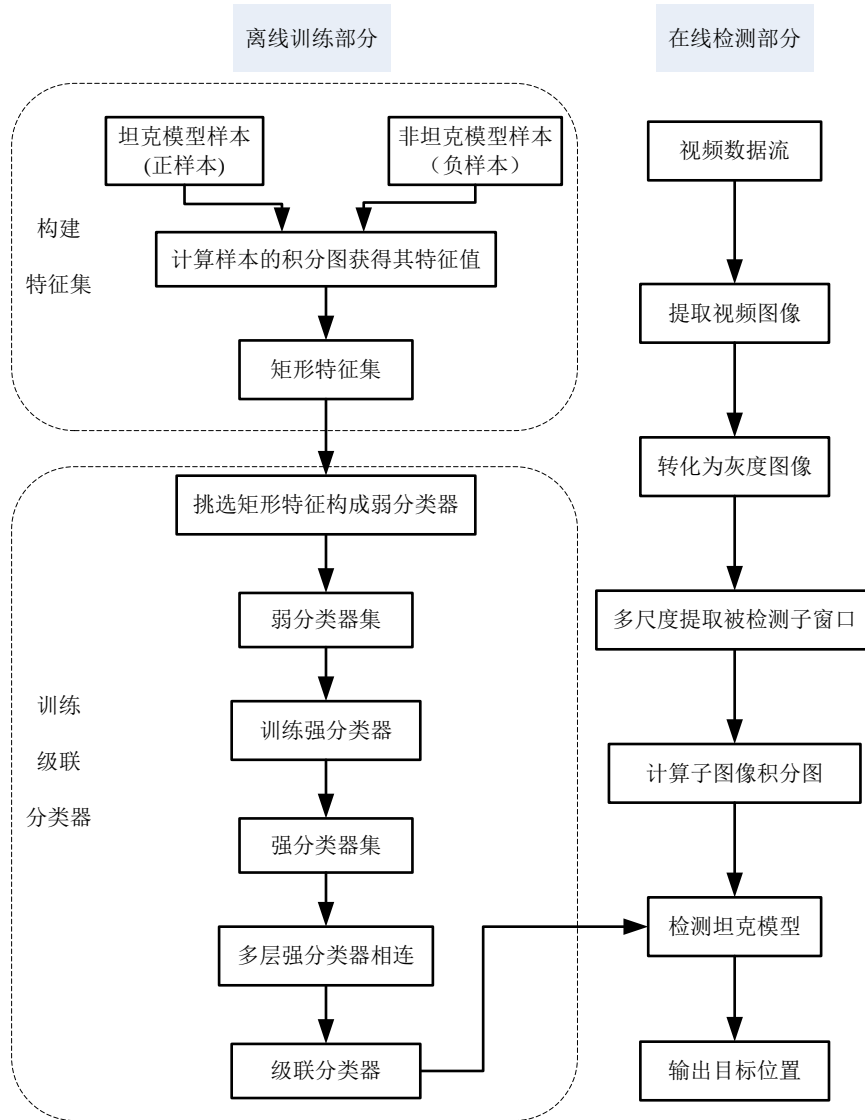


图 2.4 Adaboost 算法检测系统架构图

## 2.4 样本的构建

前面讲到了训练 Adaboost 分类器，对目标车辆模型多角度拍摄得到的正负样本是首先要做的。包含目标车辆的训练数据可以通过视频采集来获取，可以从序列图像中手

工切割出  $20 \times 20$  大小的灰度图像车辆模型的样本图像作为正样本。需要注意的是样本尺寸限制了检测窗口的最小尺寸，如图 4.3 和 4.4 所示。



图 4.3 正样本



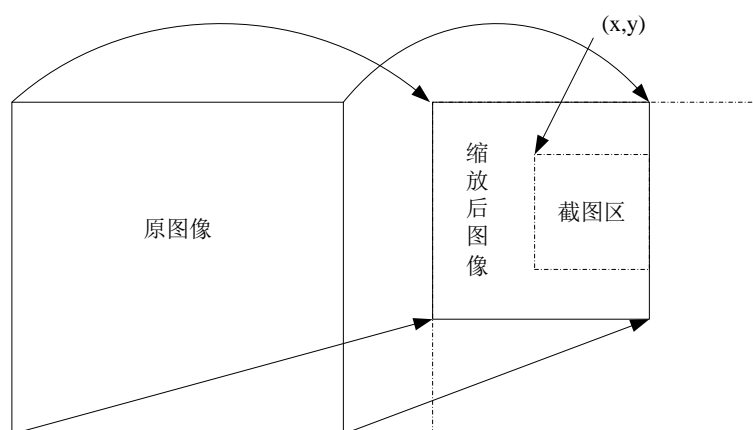
图 4.4 负样本

在训练过程中，负样本的选取决定了训练效果的好坏，当然也就是说负样本的数量比正样本越多越好。需要注意的是如果负样本数量有限就会影响到分类器的层级限制，最终导致样本训练进入死循环。

针对系统的应用环境，负样本选取的绝大多数是室外风景、室内环境、建筑物等背景图像，选数量一般为 6000 幅，而且图像大小一般大于正样本。负样本选取的内容越复杂，系统适应的背景也就越多元化。

从较大尺寸的图像中截图的方法也可以获得负样本，但是截图的负样本图像的裁剪必须要大于正样本。在实际操作的过程中，可以随便从负样本图像中拿出尺寸和正样本大小相同的一小部分区域当作负样本，具体操作流程如下：

- 1)首先确定好截图的区域，并假设其左上角的坐标为  $x$ 、 $y$ ;
- 2)确定一个缩放后仍然能够包含被选中的抠图区域为最小缩放比例;
- 3)按上面的最小缩放比例对初始负样本图像进行等比例缩放;
- 4)从缩放好的图像上抠出需要的负样本，如图 2.5 所示;



负样本截图过程示意图

识别错误样本是在样本训练时的一个再学习过程，再学习过程也是提高分类器识别性能的必经步骤，需要重点掌握。

## 2.5 分类器的训练

想要达到视频监控实时性的要求并迅速提高系统的计算速度就要从各个方面来降低计算成本。最直接的方法就是将一个复杂的分类器拆分为许多简化的分类器(弱分类器)，然后进行筛选，训练分类器从而组成一些较为复杂的强分类器，最后把强分类器层层相连，就会形成一个较为复杂的组成级联分类器也就是(Cascaded Strong Classifier)<sup>[54]</sup>。

### 2.5.1 弱分类器

每一个弱分类器都对应于一个 Haar 特征，每一个 Haar 特征参数又可以来进行定义每一个弱分类器。如何从大量的 Haar 特征中挑出最优的 Haar 特征并制成分类器用到人脸检测中，这是 Adaboost 算法训练需要解决的关键问题。

简单地说 Adaboost 的核心思想就是为分类器（弱分类器）<sup>[55]</sup>做不同的集训，然后把得到的不同分类器集合在一起形成一个更强的最终分类器也就是强分类器。下图是弱分类器分类函数的构造：

$$h_j(x) = \begin{cases} 1, & p_j f_j(x) < p_j \theta_j \\ 0, & \text{其他} \end{cases}$$

可以看出， $h_j$  表示弱分类器的值， $\theta_j$  则代表了阈值， $p_j = \pm 1$  是方向值，用来控制不等式的方向， $f_j(x)$  为该矩形特征在  $x$  上的值。

弱分类器根据统计方式的不同又分为单阈值分类器和双阈值分类器两大类。阈值  $\theta_j$  只能一次性存在单阈值分类器中，然而  $p_j$  不是+1,就是-1；Haar 特征中的“黑白”两个部分的灰度值的对比特点在单阈值分类器得到了充分体现，两个区域间的灰度值具有一定差异是作为两个区域的分类依据。而双阈值分类器则分别对应了两个方向的参数因而具有上下两个阈值分类。弱分类器的输出结果可以为真或者假，分别由 1 或者 0，代表。

$m$  代表训练的正样本个数， $n$  代表负样本的个数， $x_j$  代表样本，如果  $i=1,2, \dots, m+n$ ，则可以通过以下几个步骤实现最优阈值的选取：

1) 计算样本中特征值  $j$  的  $f_j(x_i)$  ( $i=1,2, \dots, m+n$ )，并对其进行排序，获得排序后的序列  $\{f_j(\tilde{x}_1), f_j(\tilde{x}_2), \dots, f_j(\tilde{x}_{m+n})\}$ ；

2) 观察特征值序列，可以分别计算出阈值所对应的带权分类误差数如：  
 $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_{m+n}$ ；

3) 从带权分类误差集  $\{\varepsilon_1, \varepsilon_2, \dots, \varepsilon_{m+n}\}$  中选取最小的带权分类误差，该分类误差所表现出来的是最优阈值，确定了  $p_j$  方向符后，弱分类器就可以从训练后而得到。

弱分类器也是阈值和方向值训练后确定结果，一般要选取一个矩形特征的最优特征值  $f$ ，充分利用这个最优特征值训练样本可使得弱分类器的分类误差降到最小。利用弱分类器确定了目标图像与非目标图像后还是不能有效的处理对象。进一步将这些弱分类



器合并成强分类器来使用才是王道。

### 2.5.2 强分类器

如图 2.6 所示，强分类器是如何形成的构成的。

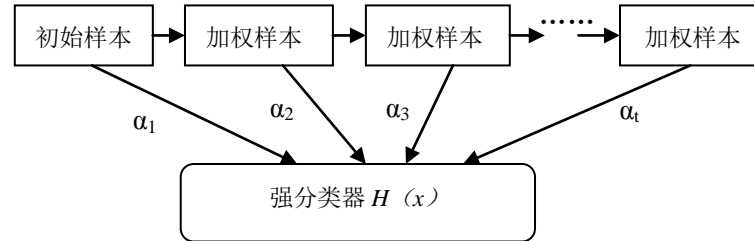


图 2.6 强分类器构成示意图

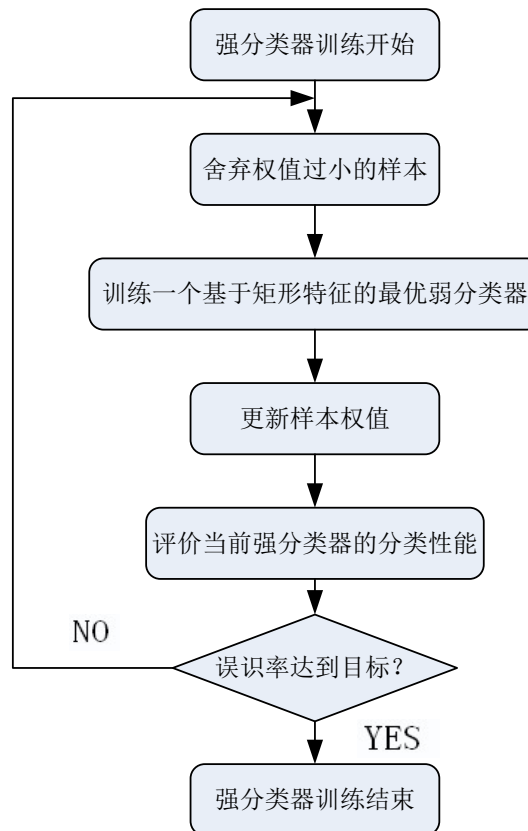


图 2.7 强分类器训练示意图

强分类器的训练是通过一轮又一轮的训练淘汰来完成的，每一轮都会均训练出一个最佳弱分类器淘汰差的，在训练开始之初，每个样本的地位及权重都是相等的，最低检测率  $d_{\min}$  和最大误检率  $f_{\max}$  是必须要提前知道的。初次训练结束是要舍弃某些样本分类

正确的分类器的权重；如果出现分类错误的情况，就需要增加这些样本的权重。一层层的做下去，后面训练提取出的简单分类器就会自动加强对那些分类错误样本的训练强度。

训练程度的逐步加深，强分类器会逐渐减弱识别错误的能力，一旦达到了我们设定的预期标准，强分类器就会停止工作，我们因此得到新的强分类器，反之则相反。就是不断地通过训练得到新的弱分类器，用来降低强分类器的误识率，直到达到预期目标为止。图 2.7 是训练强分类器的流程图：

强分类器的训练步骤如下文：

a. 如  $(x_1, y_1), (x_2, y_2), \dots, (x_i, y_i), \dots, (x_n, y_n)$  就是训练样本提前给出的一个示例， $i$  表示  $x_i$  中的一个样本，当  $y_i=0$  时表示本训练样本为负样本也就是非目标，当  $y_i=1$  时则表示样本为正样本也就是目标。总共的训练样本数量用  $n$  来表示， $y_i \in \{0,1\}$ ；

b. 权重的初始化： 负样本的计算方法：  $D_t(i) = 1/M, i = 1, \dots, M$ ,

$D_t(i)$  是表示第  $t$  次循环中第  $i$  个样本的误差权重。

c. 当  $t=1, \dots, T$  的情况出现时，就需要循环执行下面的步骤：

1. 方程式表示权重的归一：

$$q_i(i) = \frac{D(i)}{\sum_{i=1}^n D(i)}$$

2.  $f$  代表特征，对每一个特征来说训练一个弱分类器  $h(x, f, p, \theta)$ ；对应所有特征的弱分类器的加权错误率的计算如下：

$$\varepsilon_t = \sum_i q_i |h(x_i, f, p, \theta) - y_i|$$

3. 从 2 中会确定出一个弱分类器再从这部分若分类器中，找出一个最小错误率  $\varepsilon_t$  的最佳弱分类器  $h_t$ ；

$$\varepsilon_t = \min_{f, p, \theta} \sum_i q_i |h(x_i, f, p, \theta) - y_i| = \sum_i q_i |h(x_i, f_t, p_t, \theta_t) - y_i|$$

$$h_t(x) = h(x, f_t, p_t, \theta_t)$$

4. 如下公式更新全部样本中的的权重:

$$D_{t+1}(i) = D_t(i)\beta_t^{1-e_i}$$

当出现  $\beta_t = \varepsilon_t / 1 - \varepsilon_t$ ,  $e_i = 0$  时表示  $x_i$  被正确地分类, 当出现  $e_i = 1$  时表示  $x_i$  被错误地分类;

5. 最终得出的强分类器公式为:

$$H(x) = \begin{cases} 1 & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{其它} \end{cases}$$

其中  $\alpha_t = \log 1/\beta_t = \log \frac{1-\varepsilon_t}{\varepsilon_t}$ 。

6 储存训练的最终结果。

### 2.5.3 级联分类器

如果用一个较强的分类器分类判别要检测图像中所有的子图像, 就需要训练非常多的弱分类器才能得到你所需要的强分类器, 而且数量将会非常庞大, 但是我们可以采用级联的方法得到, 即: 将含有较少弱分类器的强分类器进行级联。

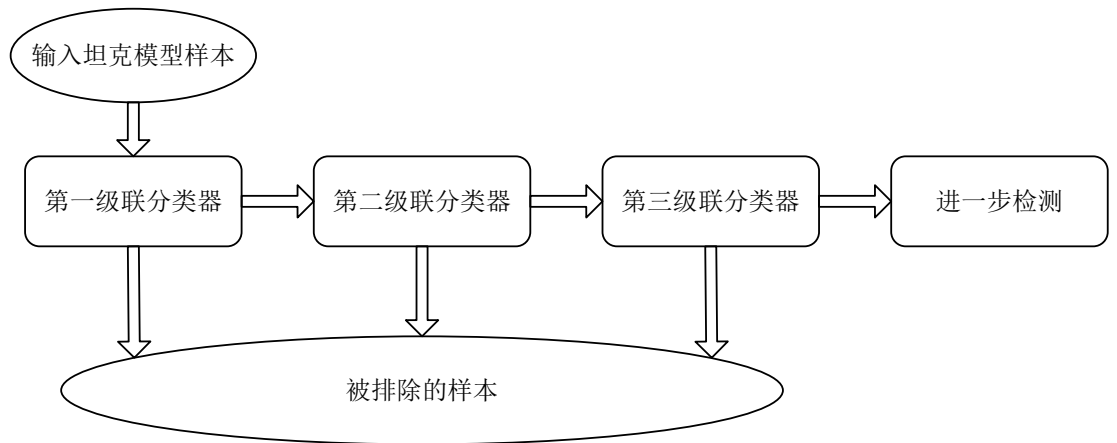


图 2.8 级联分类器的结构

如图 2.8 所示, 较多强分类器的组合就是级联分类器结构。从整体上看, 级联结构和退化了的决策树是完全一致的。通常不确定目标的窗口通过各级分类器就会被确认;

反之则会结束对该子窗口的检测，下一个子窗口就会自动开始检测。这种级联结构的分类器的好处是一方面提高检测率，另一方面降低了运算的时间。

每一个级联分类器都是由 Adaboost 方法训练得到的多个强分类器构成的，而且分类器级别越高复杂程度越高检测也越严格，如果调整参数会让每一级都能通过所有的目标样本，拒绝掉一大部分非目标样本。由一到两个特征组成的最初几级分类器比较简单，但是它们的检测速度却很高，不但在短时间内排除了大量非目标样本，而且在后期通过的候选窗口明显减少，这样使得在原有的时间内矩形特征数的计算增多。图 2.9 是训练过程示意图：

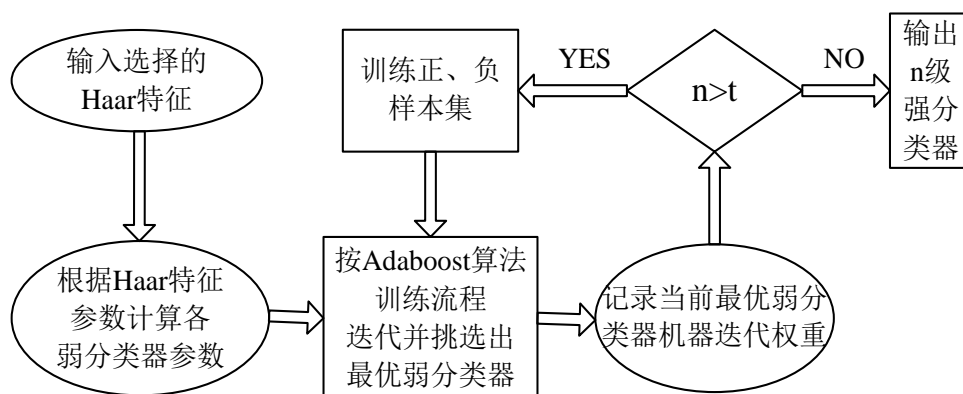


图 2.9 训练过程示意图

虽然级联分类器的设计不是不可以改变的，但保持检测率的同时降低误检率还是级联分类器设计必须应遵守的原则。级联分类器的个数和级数相关，一开始的几层分类器的分类能力只能达到大于一半的水平，到最后的能把所有非人脸样本拒绝。这说明分类器的分类能力是越来越高的。

看上图就可以得出训练过程的几个步骤：

1) 首先获取训练正负样本的个数，把  $P$  假设为所需的正样本个数， $N$  假设为所需的负样本个数；

2) 然后读取正负样本得到初始正样本集和负样本集并设 for  $i=1:n$ ;

① 开始训练一个强分类器如：  $H_i(x)$

② 保存上述强分类器的参数，即  $H_i(x)$  中参数、阈值以及负样本

③ 将训练得到的前  $i$  个强分类器  $H_1(x)$ ，  $H_2(x)$ ， .....，  $H_i(x)$  进行组合，验证并

淘汰正样本集并更新正样本数量  $P$  的值。

④同样的方法对负样本集进行验证及淘汰，判断出正确的负样本，

⑤及时补充负样本集的数量使之达到  $N$  个，使用组合前  $i$  层强分类器对候选负样本进行检测并且将之前误判的负样本加入到样本集中，需要注意的是正样本不需要更新和增加。事实上也是被误判的负样本才能加入到负样本集中参加之后训练。

4)如果出现  $i$  小于  $n$  的情况，就需要重复上述第一步训练。

5)将训练结果作好记录保存；

若  $f_i$  为每个强分类器的最大误检率，则  $F = \prod_{i=1}^n f_i$  代表  $n$  级级联分类器的最大误检

率，由此推出系统强分类器的层数公式为  $n = \log_{f_{\max}} F_{\max}$ 。

同样， $d_i$  为第  $i$  级强分类器的检测率， $n$  级的级联分类器的检测率则为  $D = \prod_{i=1}^n d_i$ 。

由上公式可以看出最后得到的级联分类器的误判率是最低的，因此通过级联足够多的具有相对较高误判率的强分类器就可以完成构造一个具有很低误判率的级联分类器。所以要做到误检率趋近于无限小分类器的级数就必须足够多。而在实际中训练一个具有很低的误判率的强分类器是相当困难的，级联分类器的这个优势在训练强分类器的时候可以充分被利用，进而保证了强分类器尽最大可能具有高的检测率和相应低的误判率。

为了使运动目标能够有效地检测出来，所以保持收敛是保证级联分类器检测结果的重要保证。也有可能会出现过训练的问题，所谓过训练就是指在训练过程中样本数量较少，导致训练的结果较差或出现一定的偏差。

## 2.6 本章小结

本章主要介绍了人脸检测技术的几种主要方法，重点阐述了 Adaboost 算法的 Haar 特征、积分图像、分类器以及级联分类器。

### 第三章 基于 FPGA+ARM 的高速图像数据采集处理平台

整个高速图像数据采集处理平台的架构如图 3.1 所示,摄像头选用 OmniVision 公司生产的 OV6620 数字摄像头,内置两个 8 位高速 AD 转换器,分辨率为 356x292。SRAM 选用 CYPRESS 公司的 CY62146EV30LL 静态 RAM 芯片,容量为 256K\*16。

系统采用 FPGA+ARM 构架,完成数据采集存储、图像预处理、数据传输和图像处理。首先通过  $I^2C$  总线对 OV6620 内部寄存器参数进行配置,数据输出模式设置为 GRB4:2:2,摄像头开始工作后,首先经过 74LVT16245 电平转换芯片将 5V 电压转换为 FPGA I/O 口的 3.3V,然后 FPGA 对接受到的图像信号进行灰度值预处理,采用两块 SRAM 乒乓操作实时存储处理后数据,最后通过 ARM 总线将图像发送给 ARM 进行实时处理,根据结果对门进行控制。

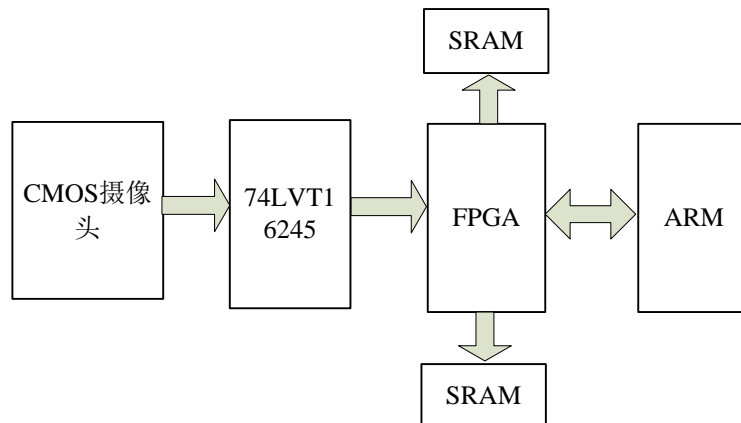


图 3.1 图像采集存储处理流程图

#### 3.1 CMOS图像传感器硬件电路设计及功能介绍

##### 3.1.1 OV6620图像传感器基本工作原理

OV6620/OV6120 的 CMOS 图像传感器方框图如下,OV6620 传感器包括一个 356 x 292 分辨率的图像阵列,一个模拟信号处理器,双 8 位模/数转换器,模拟视频复用器,

数字数据格式器和视频端口，SCCB 接口和寄存器，数字控制包括时钟模块、曝光和黑白平衡。

OV6620/OV6120 传感器是 1/4-inch CMOS 成像设备。该传感器含有约 101376 像素。其设计是基于逐行转换的场积累读出机制和同步像素读出机制的电子快门。传感器的色彩过滤器包括一个按行交替方式的原色 RG / GB 阵列。

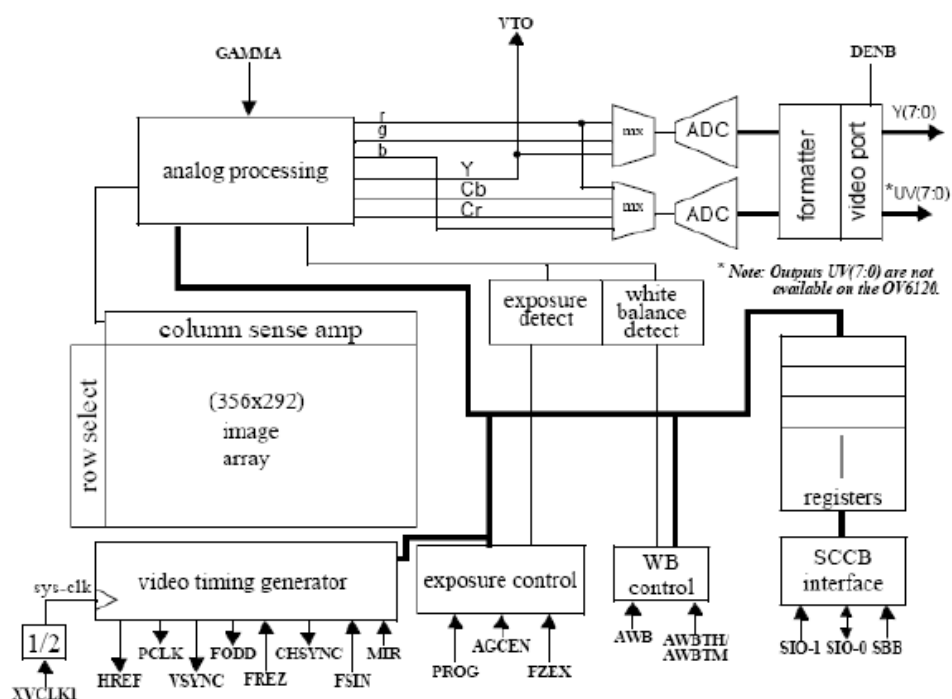


图 3.2 Ov6620 结构框图

OV6620 特性：

1. 101376像素（352x288）,1/4棱镜，CIF/QCIF格式
2. 逐行扫描读出
3. 数据格式：YCrCb 4:2:2, GRB 4:2:2 ,RGB Raw Data(RGB原始数据)
4. 8/16 bit视频数据：CCIR601、CCIR656、ZV端口
5. 宽动态范围，抗模糊，零拖尾效应
6. 电子曝光、增益、白平衡控制
7. 图像增强：亮度、对比度、伽玛、饱和度、锐度、窗口等
8. 内部/外部同步

9. 帧曝光/行曝光选项
10. 5V工作电压，低电源消耗
  - a) <80 mW 工作功率
  - b) <10uA 节电模式
11. 伽玛校正(0.45/0.55/1.00)
12. SCCB可编程 (400kb/s)
  - 色彩饱和度、亮度、对比度、白平衡、曝光时间、增益

### 3.1.2 OV6620 驱动时序

OV6620 的同步信号时序如下：垂直同步信号 VSYNC 为两个正脉冲之间扫描一帧的定时，即完整的一帧图像在两个正脉冲之间；水平同步信号 HREF 扫描该帧图像中各行像素的定时，即高电平时为扫描一行像素的有效时间；像素同步信号 PCLK 为读取有效像素值提供同步信号，高电平时输出有效图像数据，若当前图像窗口大小为  $320 \times 240$ ，则在 VSYNC 两个正脉冲之间有 240 个 HREF 的正脉冲，即 240 行；在每个 HREF 正脉冲期间有 320 个 PCLK 正脉冲，即每行 320 个像素。这就是 VSYNC、HREF、PCLK 三个同步信号之间的关系。OV6620 同步信号时序如图 3.3 所示。

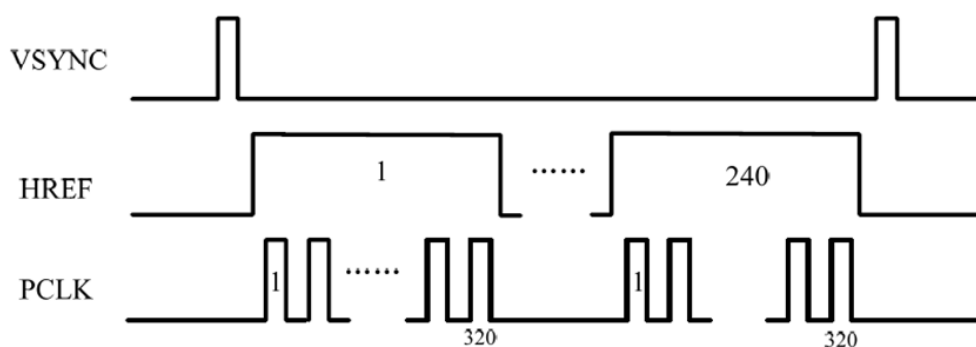


图 3.3 OV6620 同步时序图

### 3.1.3 SCCB 总线协议的时序分析

SCCB(OmniVision serial camera control bus)，即串行摄像机控制总线。OmniVision 公司已经定义和采纳的 SCCB 总线是一种三线结构的串行总线,用于完 成对绝大部分 OmniVision 系列图像传感芯片功能的控制。在简化的引脚封装中，SCCB 总线可以工



作在改进的两线工作方式下。

两线 SCCB 接口有两条通讯连接线, 即 SIO\_D(数据线)和 SIO\_C(时钟信号线)。两线 SCCB 连接框图如图 3.4 所示。

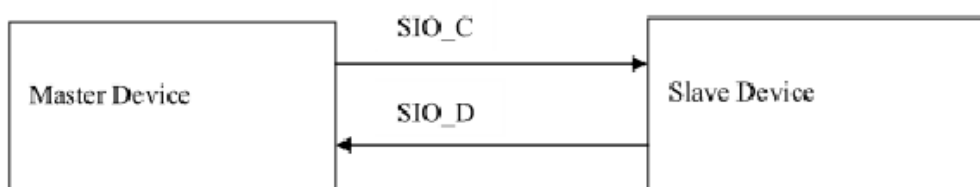


图 3.4 SCCB 连接框图

### 3.1.3.1 SCCB 总线协议的时序分析

在 SCCB 操作中,主设备必须做以下操作:

- (1)产生开始/停止信号
- (2)在 SCL 上施加串行时钟

(3)将 7 位从设备地址,读写位和 8 位子地址串行放到 SDA 上。读的一方必须在确认位时间里拉低 SDA, 返回一个确认位作为对从设备所写数据的确认。在写周期 OV6620 返回确认位, 在读周期里, 主设备返回确认位, 除非读的数据是最后一个字节。如果读的数据是最后一个字节, 主设备并不返回确认位, 通知从设备读周期可以终止。每一字节内, 最高位总是先传输。对于 OV6620, SCCB 总线通信仅需两个管脚:SCL 和 SDA。SDA 设置成开漏双向端口, SCL 为高时, SDA 上从高到低的转换表示开始。SCL 为高时, SDA 上从低到高的转换表示结束。只有主设备可以产生开始/结束信号。除以上两种情况外, 协议要求 SDA 在时钟 SCL 位高电平器件保持稳定。

只有当 SCL 为低时每一位才允许改变状态。图 3.5 和 3.6 所示分别为 SCCB 总线位传输的时序图和 SCCB 总线数据传输的时序图。

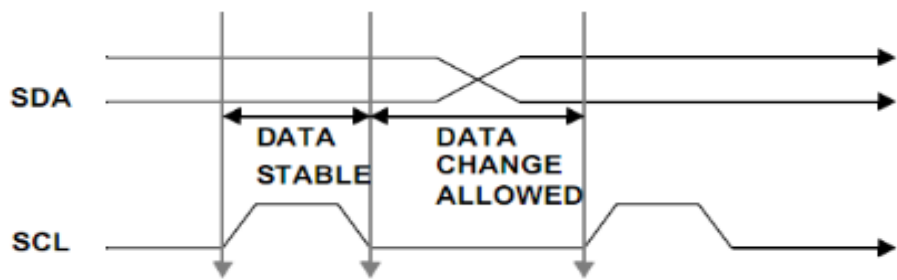


图 3.5 SCCB 总线位传输的时序图

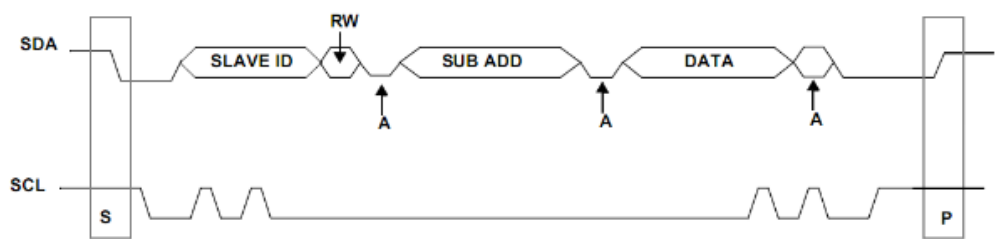


图 3.6 SCCB 总线数据传输的时序图

3.1.3.2 SCCB 总线传输实现方法

OV6620 支持单个从设备，当只有唯一的从设备时，OV6620ID 须置为 0XC0 (写) 和 0XC1(读)。在写周期中，SCCB 总线上的第一个字节是选择单个片内寄存器的子地址,第三个字节是读写该寄存器的数据。写一个未定义的子地址将被忽略。在读周期中，SCCB 总线上的第一个字节是读写该寄存器的数据，子地址用前面的。读一个未定义的子地址，返回数据不定。图3.7所示为主设备向从设备读写数据的结构图。

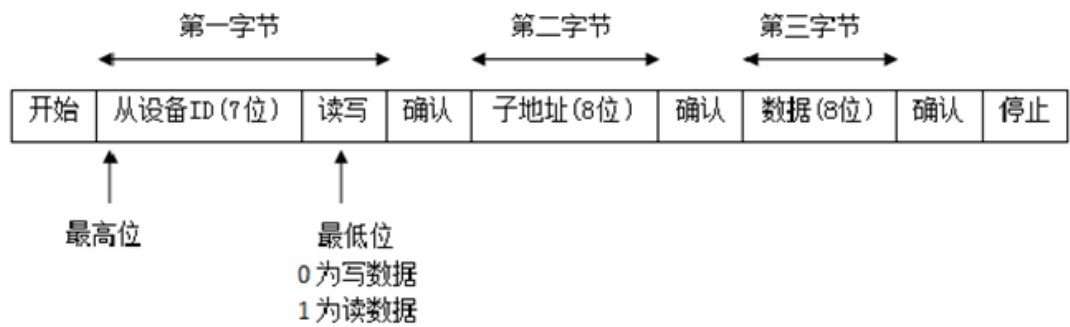


图 3.7 主设备向从设备读写数据的结构图

系统需要对摄像头进行配置，下表为系统常用到的 SCCB 寄存器表：

表 1 SCCB 寄存器表

地址	寄存器	缺省	R/W	说明
00	Gain	00	RW	GC[7:6]--未使用，读取时返回 x GC[5:0]—存储当前 AGC 增益值本寄存器自动更新。如果 AGC 许，内部控制电路将最佳增益存储在该寄存器中。 如果 AGC 止，寄存器中存 00
03	Sat	80	RW	Sat[7:0]—色饱和度调制，FF 最大，00 最小
05	Ctr	48	RW	Ctr[7:0]—对比度调整，FF 最大，00 最小
06	Brt	80	RW	Brt[7:0]—亮度调整，FF 最大，00 最大
12	COMA	24	RW	COMB[7]—SRST,1 启动软复位。所有寄存器设置到缺省值，芯片复位到确定状态并继续运行，复位后该位自动清除。 COMB[6]—MIRR 选择镜像图像 COMA[5]—1 使能 AGC COMA[4]—1 选择 8 位输出格式为 YUYVYUYV COMA[3]—1 选择视频输出格式为 RGB,0 为 YcrCb(YUV) COMA[2]—1 使能 AWB,0 禁止 AWB COMA[1]—1 使能颜色测试模块 COMA[0]—保留字位
13	COMB	01	RW	COMB[7: 6]—保留字位 COMB[5]—1 配置输入为 8bit 模式，0 为 16bit 模式 COMB[4]—1 使能字数输出为 CCIR656 格式 COMB[3]—1 水平同步,0 综合同步 COMB[2]—1 使 Y 和 UV 输出为三态 COMB[1]—1 触发单场传输 COMB[0]—1 使能自动调整模式

### 3.2 FPGA 控制模块电路设计及功能介绍

本采集与预处理模块的核心芯片选用 xilinx 公司的 Spartan-3E 系列 FPGA 芯片，资源位 50 万逻辑门，包含 4656 个 Slices，360K 块 RAM，20 个硬件乘法器，具有极高的性价比，完全满足设计容量要求。

#### 3.2.1 电源模块

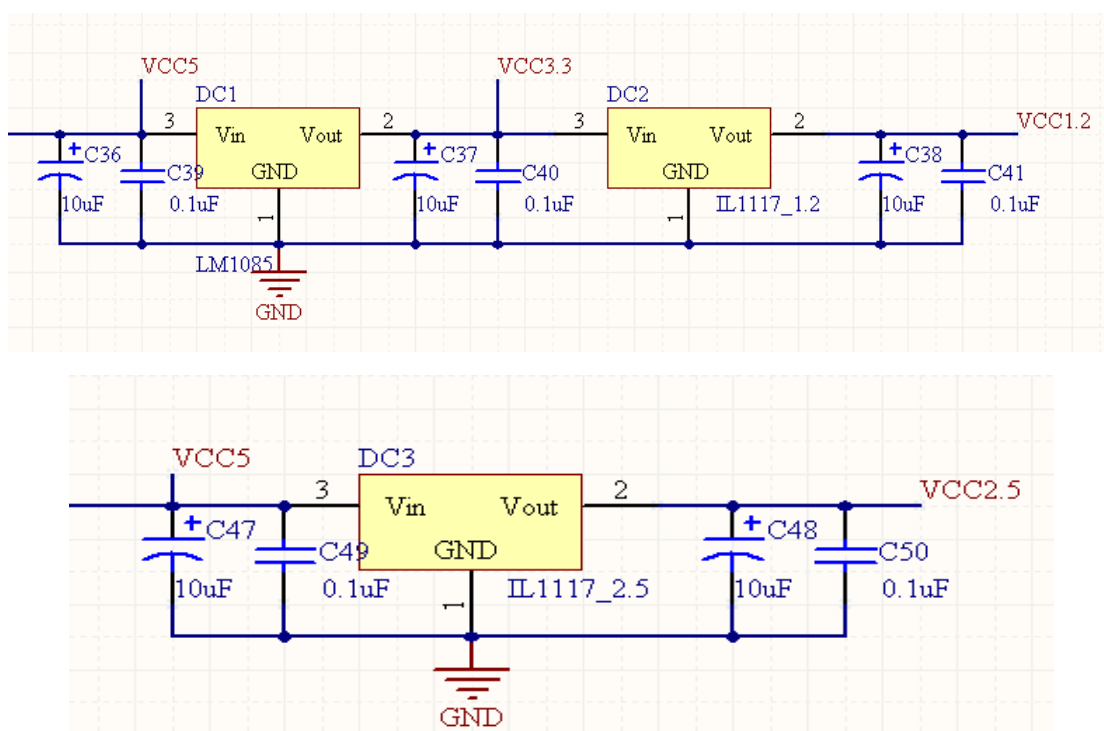


图 3.8 电源模块原理图

FPGA 正常工作需要 3.3V、2.5V 和 1.2V 三个电压，由于电源板为 5V 供电，选用 National Semiconductor 公司的 LM1085 稳压芯片完成 5V 转 3.3V 电压，IL1117-1.2 完成 3.3V 转 1.2V 电压，IL1117-2.5 完成 5V 转 2.5V 电压，提供给 I/O 口供电、内核电压、晶振、外围配置电路等。

#### 3.2.2 时钟模块

设计选用 50MHZ 有源晶振，供电电压为 3.3V，满足系统处理速度方面的要求，电路图如图 3.9 所示：

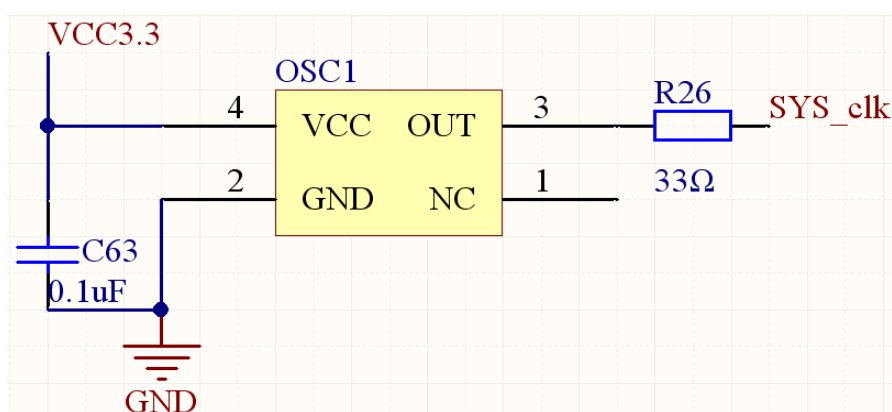


图 3.9 时钟模块原理图

### 3.2.3 程序下载电路

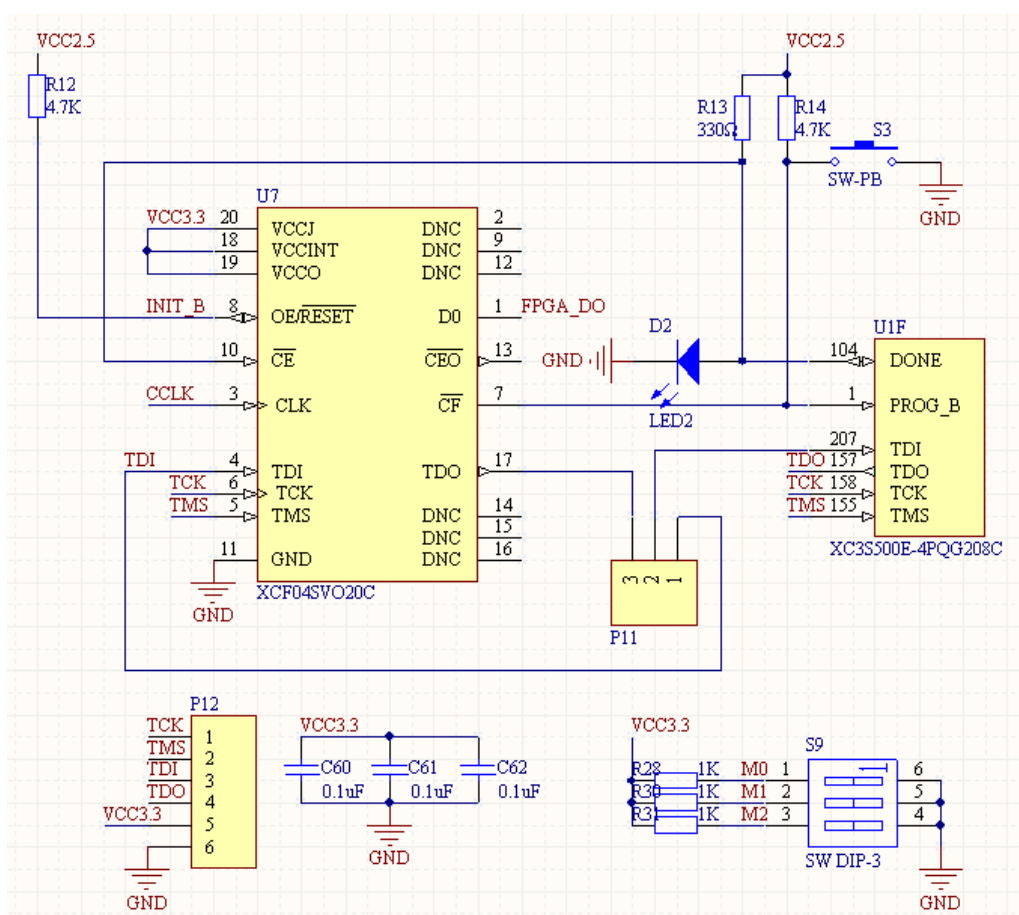


图 3.10 FPGA 配置模块电路图

Xilinx 公司 FPGA 提供多种配置模式<sup>[37]</sup>：主串模式、从串模式、边界扫描模式、SPI 模式，模式配置通过 M0/M1/M2 3 个管脚电平的高低选择。本系统选择主串模式，待配

### 3.2.4 OV6620电路

图 3.11 OV6620 电路图

工作流程如图 3.12 所示:

(2) FIFO 深度设置为 1024, 位宽 16 位, 当 FIFO 出现半满信号时, 读取 512 个数据, 然后等待下一次半满信号, 以此循环。

像素点的灰度值。在硬件实现过程中,采用流水线操作,3个数据相加后,采用 xilinx 自带 IP 核设计除法器模块进行除以 3 操作,将结果按照 RGB565 格式还原成 16 位的数据。

(4) 经过灰度值处理后,数据在 FPGA 控制下写入 SRAM1 中,然后等待第二组 512 个数据。

(5) 系统采用两片 SRAM 以乒乓方式工作,可提高后续图像处理系统的实时性。当 SRAM1 中存储满一帧数据后,ARM 产生片选信号,控制总线切换模块将 ARM 总线连接到 SRAM1 上,并将 SRAM2 切换到 FPGA 上。开启 SRAM1 的写使能和 SRAM2 的读使能,将下一帧的数据存储到 SRAM2 中,同时 ARM 开始对 SRAM1 中的一帧数据进行处理。当 SRAM2 存满第二帧数据后,再通过 ARM 片选将 ARM 总线连接到 SRAM2 上,将 SRAM1 切换到 FPGA 上,这时开启 SRAM1 的读使能和 SRAM2 的写使能。如此循环,使两个存储单元始终都在读写数据,保证了数据采集的连续性。

CMOS 摄像头的帧频率为 25HZ,每帧又分为奇数场合偶数场,每 40ms 输出一帧数据,ARM 在下一帧数据到来前,已完成前一帧数据的处理,使得系统可以实时对摄像头数据进行处理。

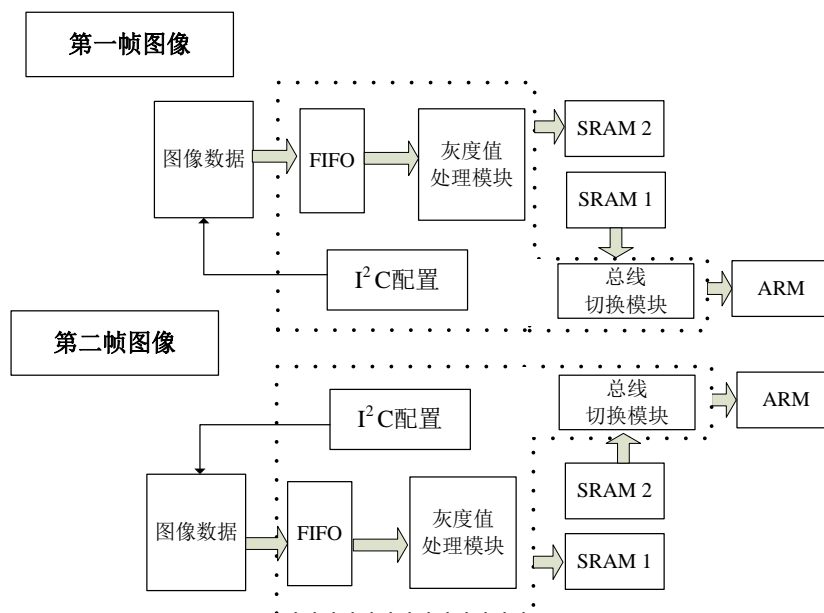


图 3.12 FPGA 硬件流程

### 3.2.5.1 FIFO缓存探测器数据仿真

为了将 OV6620 输出的图像信号自动地存入 FIFO, 采用像素同步信号 PCLK 与行同步信号 HREF 之间做与操作, 输出为 fifo 提供读时能的方法。

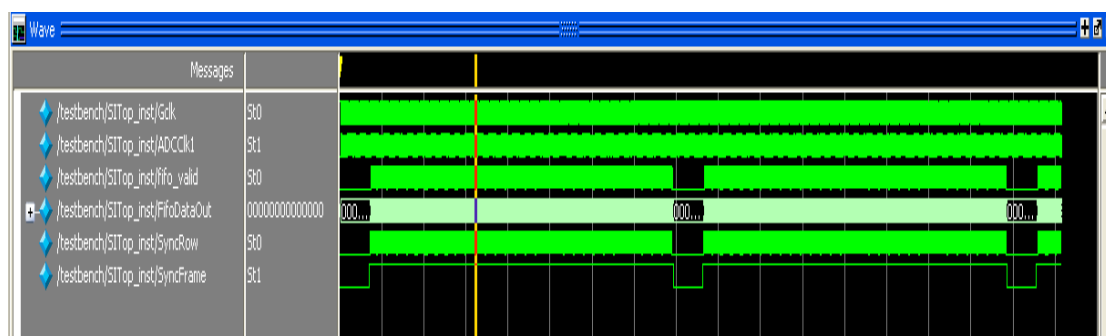


图3.13 FIFO读使能信号

图3.13为读取数据模块仿真结果: 当 OV6620 输出有效像素时, HREF 和PCLK皆为高, 则 FIFO 的 WE 为高, fifo开始读取有效数据, 当HREF 和PCLK有一个不为1时, 数据则无效, 不进行读取。

### 3.3.5.2 SRAM读写模块仿真

SRAM选用CYPRESS公司的CY62146EV30LL静态RAM芯片, 容量为256K\*16。通过控制SRAM的高8位低8位选通、片选和读写使能信号, 完成SRAM对数据的缓存和乒乓读写操作, 如图3.14所示。

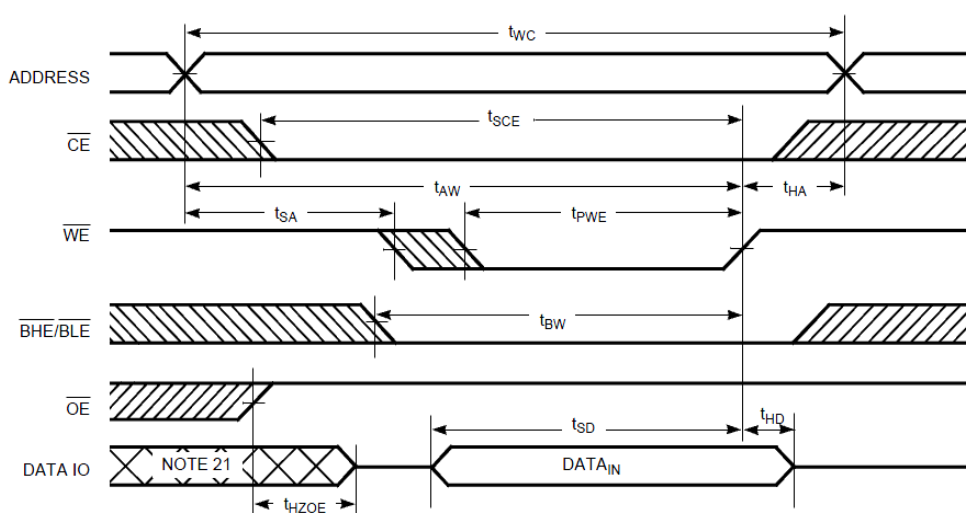


图3.14(a) SRAM写操作时序图



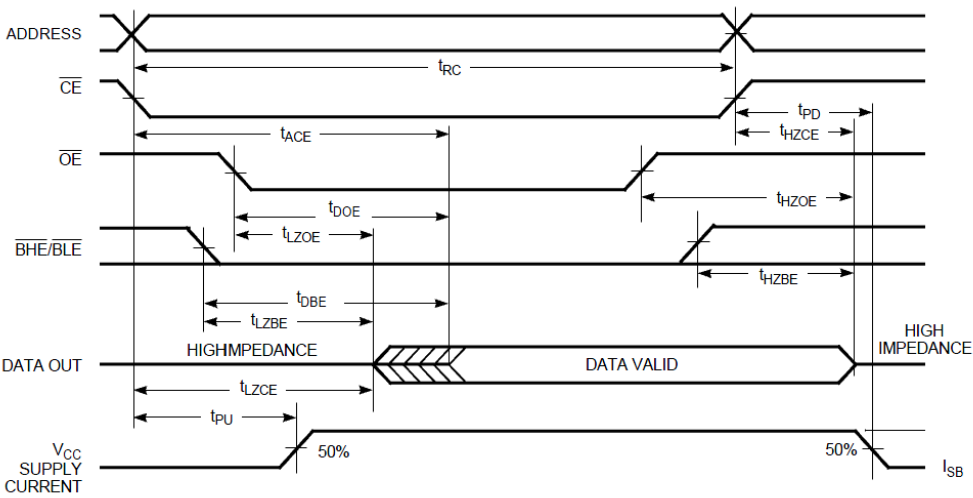


图3.14(b) SRAM读操作时序图

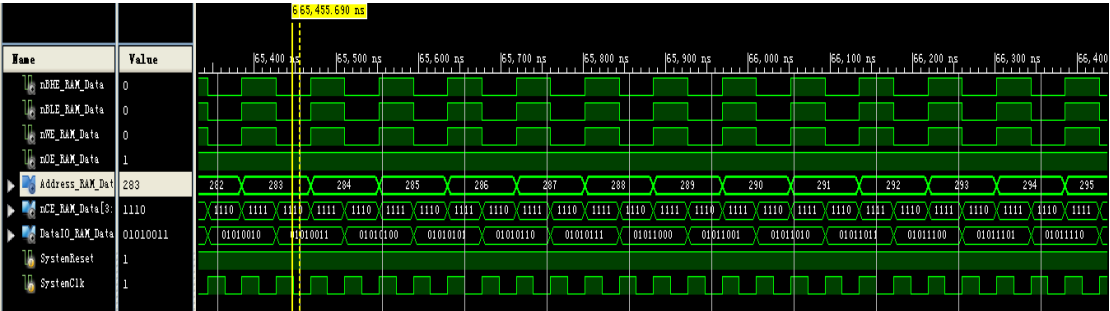


图3.14(c) SRAM写操作仿真图

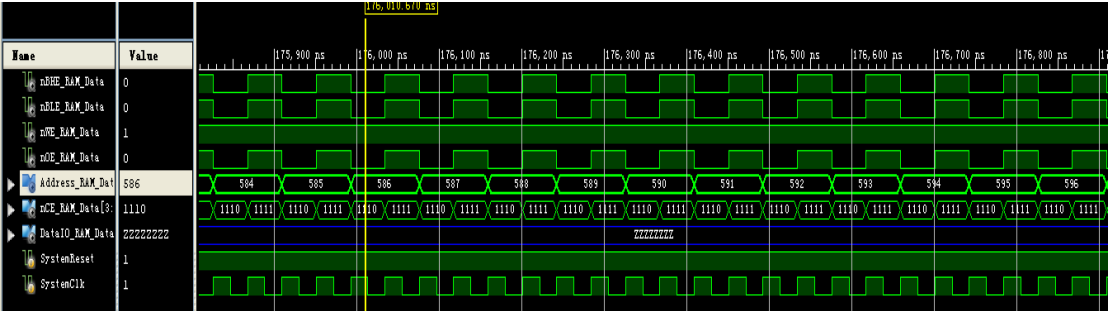


图3.14(d) SRAM读操作仿真图

3.3.5.3 灰度值预处理模块仿真

由于灰度值处理中需要用到除以3操作，Verilog中虽然提供了除法运算符“/”，但是只有当除数为2的幂次方的时候才是可以综合的，其余情况都不可以综合，而且在进行整数除法时，结果值要略去小数部分。本系统选用ISE自带的除法器IP核完成求平均

值运算。

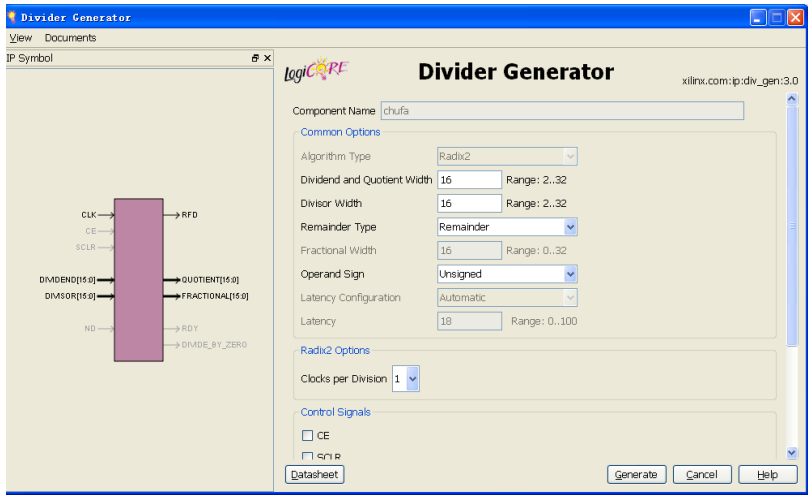


图3.15 Cordic算法设置页面

DIVIDEND: 被除数的输入接口。

DIVISOR: 除数的输入接口。

QUOTIENT: 商输出端口，商的位宽等于被除数的位宽。

REMAINDER: 余数输出端口，如COE为小数coe，则输出为除法的小数备份，位宽由XCO的小数位宽参数而设定；如CORE不为小数coe,则输出为整数除法的整数余数，位宽等于除数的位宽。



图3.16 除法器模块的RTL图

数据在数据准备信号 RFD 上升沿信号后第 18 个时钟的上降沿开始输出，下图为采用 Modelsim 对 sin/cos 模块的仿真验证：

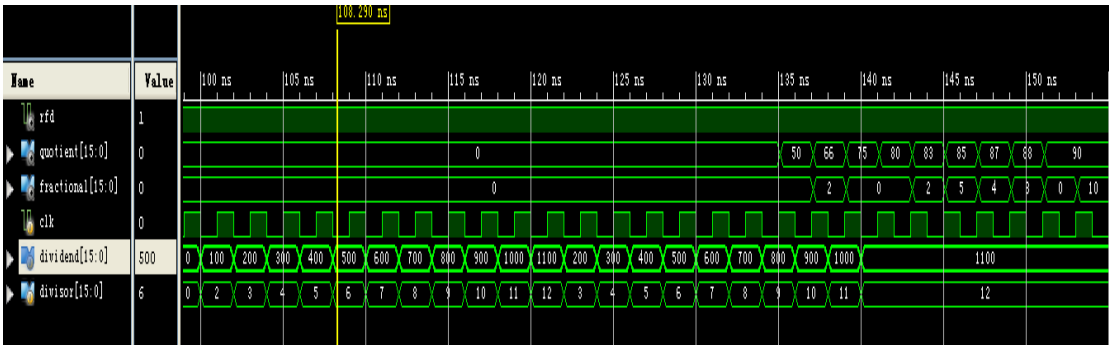
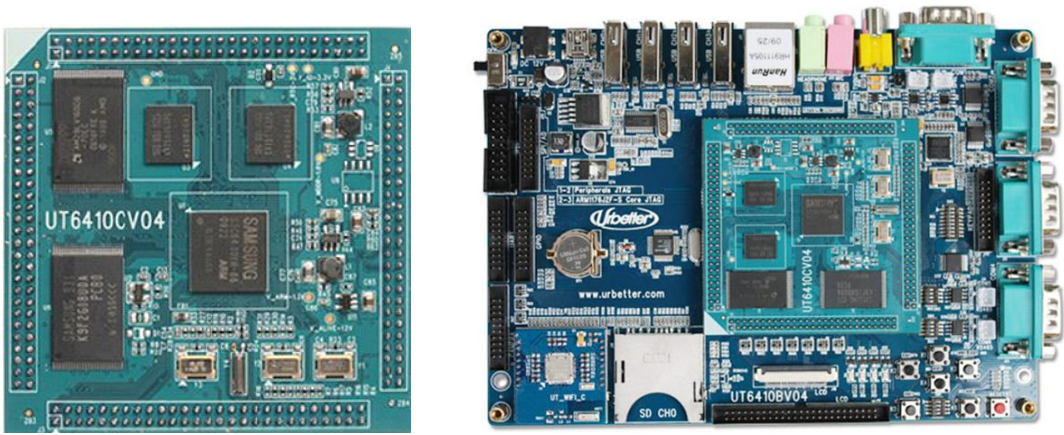


图3.17 除法器仿真结果

3.3 ARM 嵌入式系统的硬件结构

ARM11 系列微处理器具有低功耗、高数据吞吐量、高性能等特点，在软件开发方面也提供了极大地便捷，它的新型存储器提高了操作系统的性能，还提供了新指令来加速实时性能和中断的响应，并且可以适应多处理器的配置，比如 ARM+DSP、ARM+FPGA 的组合，以及从非 ARM 处理器上移植软件，使得 ARM 与其他处理器共享数据更加容易。本设计选用友坚公司的 UT6410BV04 是一款功能全面、性价比高的 ARM11 处理器开发板，开发板采用核心板+底板的设计方式，支持多种不同的液晶屏显示以及支持多个主流嵌入式操作系统。



(a) 核心板图; (b) 地板图;

图 3.18 核心板+底板实物图

ARM 嵌入式系统主要由 S3C6410 处理器、SDRAM、Flash、TFT 液晶屏、调试串口、USB 接口、总线接口等几部分组成，如图 3.19 所示。

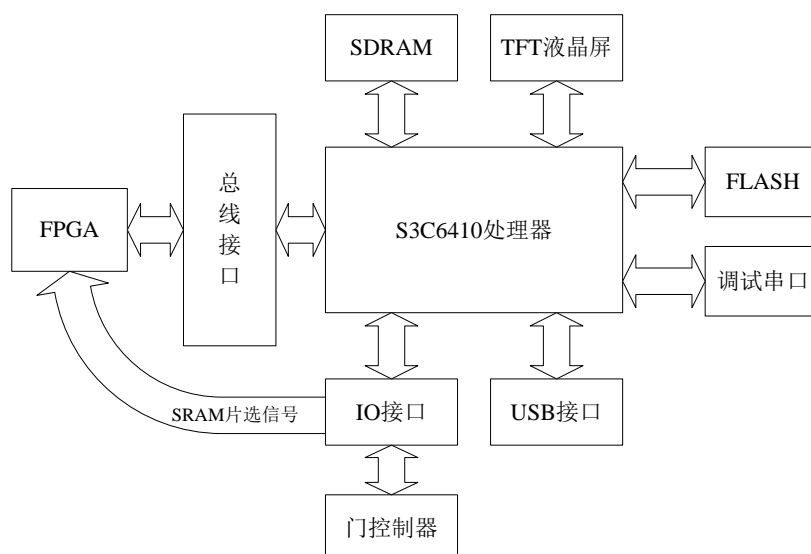


图3.19 ARM嵌入式系统的硬件结构

ARM 嵌入式系统通过总线的方式读取两片 SRAM 中的图像数据，这两片 SRAM 地址的切换通过 ARM 系统的 IO 口控制，将得到的一帧图像数据进行处理，根据处理结果控制门的开关，处理结束后再导入下一帧的图像，如此循环工作。为了实现图像的实时处理，必须在下一帧图像到来之前处理完毕。

### 3.4 嵌入式系统软件平台的建立

#### 3.4.1 嵌入式系统的介绍

随着嵌入式系统开发技术的发展，操作系统平台的功能比以前更为强大。作为一种全新的操作系统，嵌入式系统是一个名副其实的软件和硬件相结合的综合体，针对嵌入式环境所要求的资源和内存进行了优化，该系统可被应用于各种类型的设备上。它的出现不仅可以满足当今硬件供应商不断发展的各种需求，而且目前在嵌入式领域被广泛使用<sup>[55-57]</sup>。

本系统选用 Linux 系统嵌入式作为软件开发平台，具体原因如下：

Linux 操作系统在很多高级应用中占有很大市场。操作界面直观、丰富，可用鼠标双击调用应用程序，可用鼠标右键调出可选项菜单，全新的硬件管理、软件的兼容性出色，美观的使用者接口，可以保障不同的数据运行安全的需要。这些突出特性都将更能满足使用者的应用需求。

在种类繁多的嵌入式操作系统中，Linux 操作系统几乎可以不用考虑计算机硬件的配置要求。不但具有很高的可靠性和效率，还包含了高效性和灵活性的特征。可以帮助用户更好地使用计算机，同时充分发挥计算机和外设的功能。不仅保证系统运行得到令人满意的效果，而且具有多任务、多用户的能力。

Linux 被广泛应用于嵌入式系统中，多用户操作系统可以免费获得，它符合 POSIX 标准，可以支持多个使用者同时使用该系统，共享系统的磁盘、外设、处理器等系统资源。每个用户也可以同时启动多个任务，允许使用窗口、图标和菜单对系统进行操作，这一方式使得工作效率更高。

Linux 系统是一个通用的性能较高的操作系统，具有多任务特点，提供了全功能，交互界面专为电脑初级用户而设计，让初学者也能快速理解和使用。在同一时间，用户可以依赖该系统同时执行多个程序，多任务为电脑形成一种稳定、安全的软件模块集合提供了可能。提供多任务处理的方式可以启动多个应用程序，在此系统下，微处理器运行程序的时间大大缩短，方便了用户操作。

嵌入式 Linux 实现了模块化内核，将程序和操作系统嵌入，这使得操作系统能力和易用性等各方面功能更强，用户可以根据不同的任务，通过可配置内核裁剪系统内核，还可以选择特定内核模块减少体积，这种做法保证了操作系统系统资源占用率较低，而且性能比较稳定，能够给用户极大的拓展空间。

可移植的操作系统，能够在一个特定环境里运行，Linux 成功的做到了这一点。操作系统在移植的过程中，电脑能够按其自身的方式运行，适应了控制方式及通信协议的变化，不需要另外增加特殊的通信接口，大大改善了平台的可移植性，并大幅度降低了通常软件开发的成本和时间。

Linux 良好的用户界面可以支持庞大的软件系统，传统用户界面和系统调用，可在不影响性能的情况下，运行占用更多内存资源的应用软件。为用户程序提供低级、高效率的服务。Linux 向用户提供了两种界面：除了基于文本的外，还包括调用给用户编程时使用的界面。通过这个界面可以按照自己的要求随时调用命令，从而为用户扩充系统功能提供了更高级的手段。

### 3.4.2 交叉编译开发环境的建立

交叉编译器在嵌入式系统的软件开发中启动关键性作用，交叉编译是指在 PC 上使用交叉编译器将软件代码编译成可在其他嵌入式平台上运行的软件程序的过程，例如，在 PC 上编译的程序，可以在 ARM 平台上运行。这样做的主要优点是本机执行的程序代码不需要在本机编译，而是在其他平台上进行编译的，由于在嵌入式平台中没有足够的资源进行编译，所以使用交叉编译器可以大大减少了嵌入式软件的编译难度。

ARM 嵌入式系统的开发过程一般采用该方法。

交叉编译环境的建立，一般需要在 PC 中安装 Linux 操作系统，本系统是在 PC 中安装虚拟机，在虚拟机中安装 linux 系统，并在 linux 中安装交叉编译器，交叉编译器有很多种，本文使用 arm-none-linux-gnueabi-gcc-4.3.3 交叉编译器，使用该编译器编译过的程序就可以移植到 ARM 嵌入式设备中了。如图 3.20 所示。

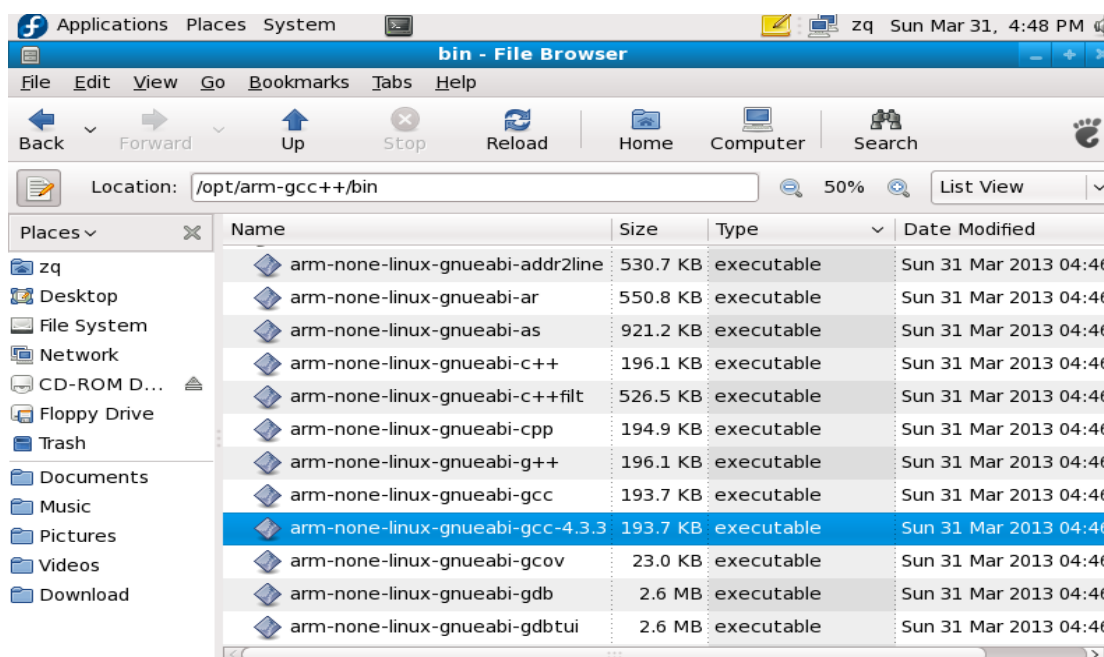


图 3.20 arm-none-linux-gnueabi-gcc-4.3.3 交叉编译器

### 3.4.3 引导程序的移植

#### 3.4.3.1 Boot loader 引导程序

在特定的硬件平台上，Bootloader 一般在操作系统执行前载入，它的存在，可以帮

助用户最快的加载操作系统，通过内存、中断控制器的初始化，调用系统所需的模块，建立适当的系统软硬件环境，对于嵌入式系统，**Bootloader** 能够获得更快的执行速度，可以启动相关的键盘、鼠标、显卡、多媒体的驱动程序，确保操作系统内核正常运行。

**Bootloader** 是一段很有用的引导小程序，我们可以通过它从目标机上的某个固态存储设备上，将操作系统加载到 **RAM** 中运行，具体作用包括：初始化 **LED**、设置参数区域，启动参数和内核页表等信息，建立内存空间的映射表，并构造参数结构和标记，通过这段小程序，启用指令、数据高速缓存，确保硬件设备的正常工作。

**Linux** 系统是通过 **Bootloader** 引导启动的，可以指导用户特定的 **BootLoader** 设计与实现。在初始化系统之后，执行 **POST**（加电自检）来标识存在的设备并报告有何问题，设置 **CPU** 的速度和时钟频率，建立内存空间的映射图，括正确地设置系统的内存控制器的功能，通过此步骤，电脑开始启动操作系统。

**Bootloader** 程序还可以执行很多用户输入的命令，这是因为它不但依赖于 **CPU** 的体系结构，而且依赖于嵌入式系统板级设备的配置。**BIOS** 在完成硬件检测和资源分配后，**CPU** 通常都从地址 **0x00000000** 处开始执行，将操作系统的内核装载到存储器中的合适位置上去运行。因此，**Bootloader** 软件确保了主机和目标机之间通过串口来输入输出，读取用户控制命令等功能的实现。

作为引导操作系统内核运行的一段代码，不论是在本地加载模式还是远程下载模式下，**Bootloader** 的启动过程都是多阶段的，系统在上电或复位时在最短的时间内完成启动过程，在启动过程中，会根据该启动参数进行相应的初始化工作，最终完成操作系统的加载，这个小程序，使系统应用更加广泛和便捷，大大提高了处理器调试的效率。

#### 3.4.3.2 U-BOOT 的介绍

**u-boot** 的优点众多，不仅具备了开放源码，而且支持多种嵌入式操作系统内核，该命令支持多个处理器系列，可以为用户提供了交互功能，不仅具有较高的可靠性，稳定性也很强，可以实现几十个常用的命令，对系统的软硬件资源进行合理的配置与管理。高度灵活的功能设置，让用户告别枯燥的操作方式，满足操作系统不同引导要求，使 **U-Boot** 调试操作中的一切都变得轻松简单。

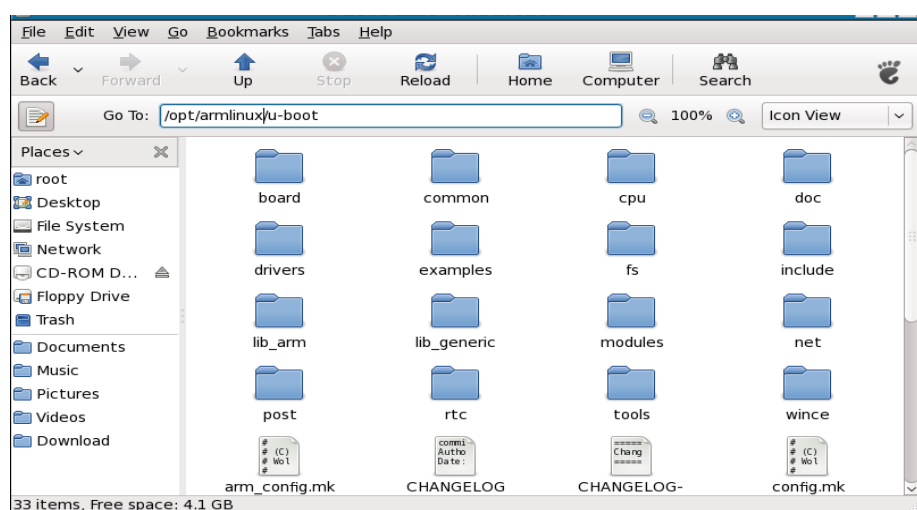


图 3.21 u-boot 目录

除此之外，u-boot 还具备丰富的设备驱动源码，操作系统只有通过 EEPROM、以太网以及其他串口；调用定点设备：鼠标、键盘、USB 等，保证用户可以使用一些较高层次的接口。u-boot 目录下子目录众多，有 18 个之多，这些目录分别存放管理不通的源程序。能够满足强大的网络技术支持，不仅包括丰富的开发调试文档，还包含了通用的函数，除此之外，大量的驱动和应用程序、工具或者文档也收录其中。具体如图 3.21 所示。

### 3.4.3.3 u-boot 的启动过程

u-boot 的启动分为两个阶段。第一个阶段的代码存放在 start.s 中，用汇编来实现，步骤如下。

(1) 定义入口。U-boot 启动需要一个启动的入口点，，通常这个入口放在 ROM 或者 FLASH 的 0x0 地址。

(2) 首先必须为加载 Stage2 准备好一段可用的 RAM 空间，将 ROM 中的程序复制到 RAM 中。需要明白两点：1 stage2 的可执行映像 在固态存储器设备的存放起始地址和终止地址。2RAM 空间的起始地址；

(3) 初始化堆栈。

(4) 转到 RAM 中执行。

第二个阶段的代码放在 board.c 中，用 C 语言实现。用 C 来实现目的是实现更复杂的功能和取得更好的代码可读性和可移植性。步骤如下：



- (1) 调用一系列的初始化函数。
- (2) 初始化 Flash 设备。
- (3) 初始化系统内存分配函数。
- (4) 如果目标系统拥有 NAND 设备，则初始化 NAND 设备。
- (5) 如果目标系统有显示设备，则初始化该类设备。
- (6) 初始化相关网络设备，填写 IP、MAC 地址等。
- (7) 加载系统内核和根文件系统

#### 3.4.3.4 u-boot 的移植

在 U-boot 中已经对很多主流的处理器进行了支持，本系统所使用的 ARM 嵌入式处理器 SC6410 包含在内，所以只需要将相应的配置文件导入 u-boot 的 config 文件，并针对开发板进行一些修改，然后使用交叉编译器进行编译，生成的 u-boot .bin 文件就可以烧写到 ARM 开发板中了。

### 3.5 linux 操作系统的移植

#### 3.5.1. linux 内核的移植

本系统移植的是 linux2.6.30.1，源代码可以从 linux 官网中下载，具体的移植步骤如下：

##### (1) 配置 linux 内核

在终端窗口中使用 `make menuconfig` 指令进入配置界面，配置菜单结构简单，操作方便。本系统是采用 ARM 开发板自带的配置文件，在其的基础上进行了一些修改。修改配置如下：

##### <1> ARM 系统的类型

本系统所使用的 ARM 处理器是三星的 S3C6410，所以配置如图 3.22 所示。

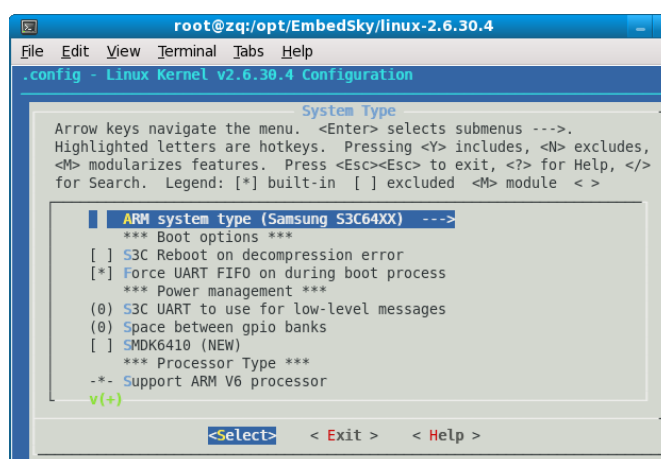


图 3.22 ARM 处理器类型

## <2> 支持 MTD 分区

本系统所使用的 nandflash 的分区方式是 MTD(Memory Technology Devices),所以需  
要将 MTD 设为 support, 如图 3.23 所示。

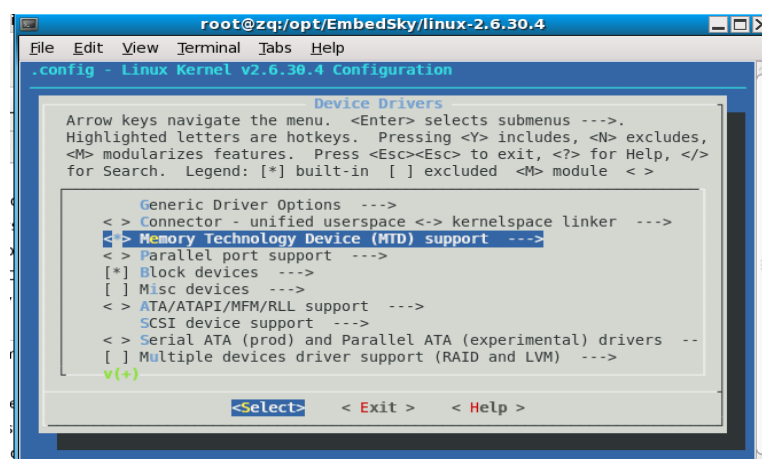


图 3.23 ARM 处理器类型

## <3> 配置支持的文件系统格式

Linux 支持的文件系统格式很多, 本系统所使用的根文件系统是 YAFFS2 所以需要  
将它设为 support。如下图所示:

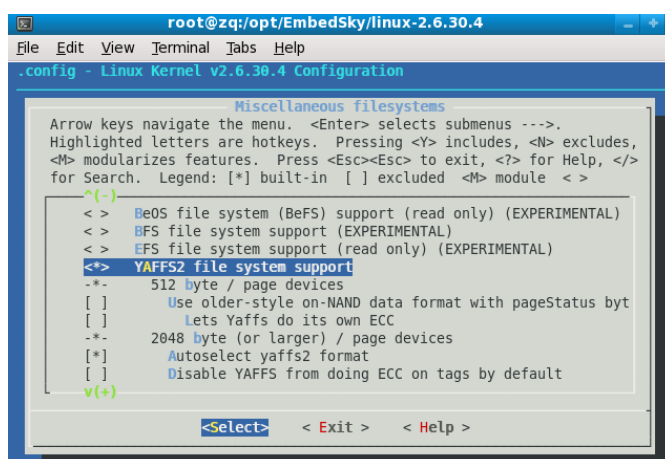


图 3.24 Linux 配置示意图

#### <4> 支持符合 EABI 标准的交叉编译器

处理器在对浮点进行运算时，如果处理器支持硬浮点运算，编译器会预设硬浮点运算 FPA(Float Point Architecture)，而没有 FPA 的处理器，会使用 FPE(Float Point Emulation 即软浮点)，这样处理速度会很慢。EABI(Embedded Application Binary Interface)的出现使得该问题得到了改善，它使用 Vector Float Point(矢量浮点)大幅提高浮点运算的能力。本系统所使用的的交叉编译 arm-linux-gcc 是符合 EABI 标准交叉编译器。

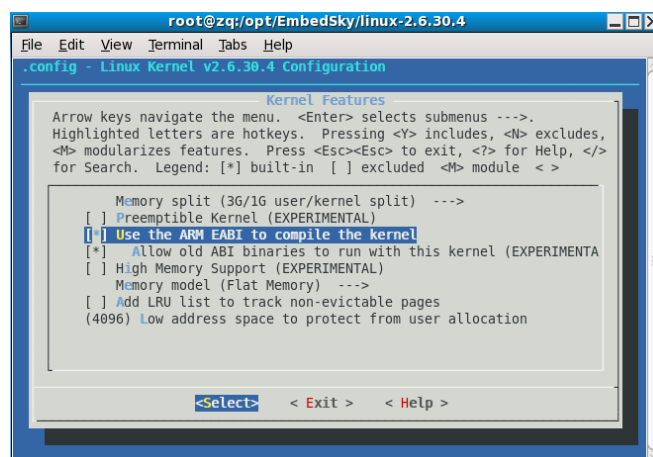


图 3.25 EABI 标准的交叉编译器

(2) 进入内核目录，修改 makefile，并对内核进行默认配置进行修改

ARCH ?=arm

CROSS\_COMPILE ?=arm-linux-

(3) 修改平台输入时钟

本系统所使用的时钟是 12M，所以需要将 linux 内核的时钟修改成 12M 找到内核源

码 arch/arm/mach-s3c2440/mach-s3c6410.c 文件，修改时钟为 12M，如图 3.26 所示。

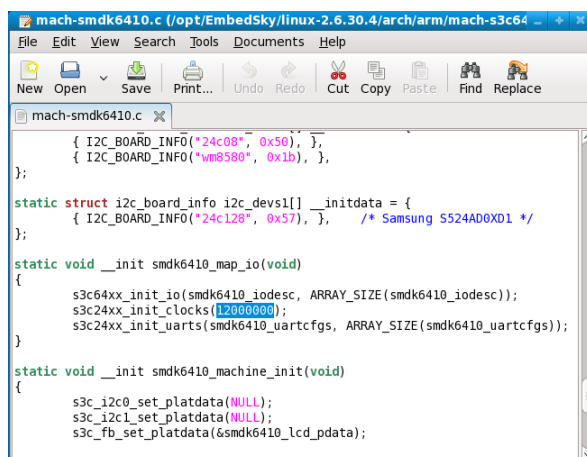


图 3.26 平台输入时钟修改示意图

#### (4) 修改 Nand flash 分区信息

修改 Nand flash 分区信息，需要修改文件 common-smdk.c，修改如下

```
static struct mtd_partition smdk_default_nand_part[] = { //nandflash
[0] = {
.name1 = "u-boot",
.offset1 = 0x00000000,
.size1 = 0x00050000,
},
[1] = {
.name1 = "kernel_linux",
.offset1 = 0x002F0000,
.size1 = 0x00200000,
},
[2] = {
.name1 = "linux_root_file",
.offset1 = 0x004F0000,
.size1 = 0x05000000,
}
};
```

可以看出，分区是按照系统上电加载的先后顺序分配的，第一分区存放 u-boot，第二分区存放 linux 系统内核，第三分区存放根文件系统。

### (5) 编译内核

与 u-boot 的编译类似，使用交叉编译器进行编译,编译指令：`make zImage`，生产 `zImage.bin` 文件，将该文件烧写到 nand Flash 相应的分区去，linux 内核的移植就算完成。

## 3.5.2 根文件系统的移植

对于根文件系统来说，在移植的时候，需要进行挂装，这样做可以确保 Linux 启动时挂载根文件系统。根文件系统顶层目录众多，按照构建根文件系统的规则，确保根文件系统的各个目录形成完整的文件系统。

第一个必须挂载的是：若系统不能从指定设备上，则系统会出错而退出启动。成功之后可以自动或手动挂载其他的文件系统。因此，一个系统中可以同时存在不同的

在 Linux 中，Linux 根文件系统是按照科学的文件系统构建方法和规范构建成的，根文件系统被挂载到根目录下“/”上后，通常情况下，使用 `mount` 命令形成一定的目录结构，在根目录下就有了文件目录，例如：`bin` 目录、`sbin` 目录等。具体如下所示：

首先，根文件系统中包括了许多的应用程序 `bin` 目录，根据存储设备的硬件特性、系统需求等有不同的应用场合，该目录下存放的这些命令可以被用户随时调用。系统配置实用程序是我们经常使用的系统程序，它可以控制启动项目，提高开机速度等。用户可以在 `bin` 目录下快速将其调出，例如：只要在“运行”对话框中输入并执行“`msconfig`”命令即可调出“系统配置实用程序”。

其次，`sbin` 目录存放的是基本的系统命令，通过启用这些命令，帮助用户完成启动系统任务，查看当前端口状态。通过这个命令的详细信息，我们就可以完全监控计算机上的连接，从而达到控制计算机的目的。在电脑出现问题的时候，管理员可以利用该文件夹的修复系统命令完成相应的功能。

再次，在 Linux 系统下，设备文件存放在 `dev` 目录里面。通过执行这个命令，可以启动具体硬件，确定设备的驱动程序和界面，该目录下的是设备文件，能够将硬件设备的特性及管理细节显示出来，是 Linux 中特有的文件类型，以文件的方式访问各种设备，通过系统调用 `open()` 打开设备文件，建立起与目标设备的连接。

除此之外，etc 目录下的文件具可选性，而 lib 目录能够运行根文件系统中的可执行程序，home 目录存放用户相关的配置文件。usr 目录存放的是共享、只读的程序和数据，该目录下的内容可以在多个主机间共享，也可以在嵌入式中随意精减。var 目录中存放可变的数据，比如 spool 目录（mail, news），log 文件，临时文件。proc 目录是临时生成的虚拟的文件系统，里面的文件可以操作控制系统。和它相同，mnt 也是一个空目录，用来临时挂载光盘或者硬盘。tmp 目录我们都很熟悉，这是系统的临时文件存放之处，一些程序在运行的过程中，需要生成临时文件，此时就会存储在 tmp 目录下。

对于 Unix 操作系统来说，根文件系统基本结构并不复杂，它通常包含三个组成部分，例如：JFFS、yaffs、NFS 等几种。虽然包括的种类并不多，但是，这些文件给 Linux 带来了许多强大和灵活的功能，具备了嵌入式操作系统的重要特征，可以确保系统库、内核模块和应用程序等的正常运行，在文件系统层，按照文件系统的操作规则对该文件进行相应处理。

嵌入式系统文件系统区别于其他一些传统根文件系统，它具备了复杂性的特点。为了对各类文件系统进行统一管理，用户可以通过选择合适的文件系统类型，把它放到实际的存储设备的合适位置，确保可执行映像依次存放在系统存储设备上。

具体包括以下几种：

#### (1) JFFS2

JFFS 嵌入式系统文件功能强大，不仅有初级版本 v1，还有 JFFS2 系统升级版，该系统基于 JFFS 开发，在闪存上使用 JFFS2 具有十分的必要性，因此，被嵌入式系统开发人员广泛使用，二者比较，jffsx 不适合用于 NAND 闪存，而 JFFS2 也可以用在 Linux，uClinux 中。JFFS 文件系统支持文件数据的压缩存储，设计中主要的思想沿袭了传统的嵌入式系统文件，符合关键的数据结构和垃圾收集机制。通过建立文件结构，可以选择日志结构的存储方式，在用于 NOR 型闪存方面作用明显，能够将文件系统的数据和原数据以节点的形式存储在闪存上，并找出所有的日志节点，释放在扫描过程中使用的临时信息，为用户节约了大量的时间。

#### (2) YAFFS

与 JFFS 相比，YAFFS 能更好地支持 NAND FLASH，是一种专门为 FLASH 设计的嵌入式文件系统。从性能上来说，它规避了挂载过程需要对闪存从头到尾的扫描的弊端，

无用功能的减少进一步提高了 PC 的反应速度，大大减少了内存空间的占用率。用户通过 YAFFS 文件系统，可以像操作硬盘上的文件一样管理 FLASH 中的数据，在系统断电后数据仍然存储在 FLASH 芯片中。

但是，YAFFS 存在挂载时间过长和损耗均衡性不足两方面的缺点，需要进一步改进和优化。随着嵌入式技术的发展，新出现的 YAFFS2 比 YAFFS 优势更明显，它是专为嵌入式系统使用 NAND 型闪存而设计的一种日志型文件系统，大幅提升读写速度。作为跨平台的文件系统，除了 Linux 和 eCos，还支持 WinCE，pSOS 和 ThreadX 等。这种改进在很大程度上提高了垃圾回收速度。

### (3) NFS

NFS 是一种网络文件系统，在嵌入式 Linux 系统的开发调试阶段，NFS 有属于自己的协议与使用的 port number，可以通过网络共享文件的技术挂载到嵌入式设备。用户在 NFS 的根文件系统中，可以管理 port 的对应与服务相关性的工作，可以很方便地进行档案的分享，修改根文件系统的内容。开发者需要在 Linux 服务器上进行所有的软件开发时，可以利用 Linux 的根文件系统获取各类信息，以及 devfs(设备文件系统)和 sysfs，用于维护设备文件。

为了使制作的内核支持更强，在 NFS 根文件系统下。用户一旦想做修改，可以在开发板从/mnt/nfs 目录中得到根文件系统的一级目录，。也可以进入文件目录查看根文件系统是否齐全。在这个系统中，具备了 linux 启动运行所需要的一切程序。同时，这些程序也可以在不同的系统间使用,有了它，用户就可以轻松地在 Linux 系统之间，实现资源共享和互访功能。

Linux 的根文件系统一般是使用 busybox 这个工具来生产各个文件目录的，只需要对 busybox 进行配置（类似于 linux 内核的配置）就能生成所需要的根文件系统。

将生产的根文件系统烧写到嵌入式系统的 nandflash 中，系统就可以正常工作了。

## 3.6 本章小结

从系统要实现的功能出发，采用 xilinx 公司 SP3E 芯片完成 CMOS 摄像头驱动以及数据的预处理，在 ARM11 平台上搭建了 LINUX 嵌入式操作系统，完成系统所需的软硬件平台。

## 第四章基于 OpenCV 的实时人脸检测算法的实现

### 4.1 OpenCV 简介

OpenCV 于 1999 年建立, 由 Intel 资助建立的开源项目, 现在由 Willow Garage 支持、维护。有数十万的爱好者为 OpenCV 编写程序, 为图像处理技术的广泛应用做出了重大的贡献。OpenCV 的优势:

跨平台性: 支持多种嵌入式操作系统内核, 如 Linux、windows、VxWorks,; 并且 opencv 支持多个处理器系列, 如 PowerPC、ARM、x86;

开源: OpenCV 是一个开源的项目, 代码完全公开且使用是完全免费的。

功能强大: OpenCV 提供矩阵、数组、序列等基本结构, 同时提供众多的数学计算函数, 包括差分运算、傅立叶分析、积分运算等, 以及各种封装好的图像处理函数。

实时性: 基于 OpenCV 设计好的代码可以编译生成二进制可执行文件, 大大提高了运算速度, 有效提高了实时性运算性能。

交互性: OpenCV 可以和包括 C、C++、matlab 等语言进行混编。

因此, OpenCV 可以很简单、方便的将图像处理技术应用于各个领域。

### 4.2 人脸分类器的训练

OpenCV 图像算法库中不光包括了很多图像处理和计算机视觉方面的通用算法库, 还有很多实用的运用程序, 这其中就包括了经典的 AdaBoost 级联分类器的训练程序, 通过该训练程序可以得到用于目标检测的级联分类器。

#### 4.2.1 样本的创建

样本的创建是对正样本而言的。OpenCV 中的应用程序 (createsamples.exe) 会将已标记好的正样本图像区域中计算出样本特征向量, 并将所有这些向量统一的打包成一个文件。



Createsamples.exe 所使用到的命令行参数如下：

- info 正样本矩形参数的文本路径。
- vec 训练好的正样本的文件名和路径。
- num 正样本数量总数。
- w 样本的宽度和高度。
- h 样本的高度

#### 4. 2. 2 分类器的训练

正负样本的工作都准备好以后，就可以使用 haartraining.exe 程序训练级联 Boost 分类器了， haartraining.exe 程序经常使用到的命令行参数如下：

- data 训练好的分类器所存放的路径。
- vec 由 createsamples.exe 程序创建的文件所存放的路径。
- bg 背景描述文件所存放的路径。
- npos-nneg 指每一个分类器阶段的正/负样本的数量。正样本一般为 2500，负样本一般为 5000。
- nstages 是指训练分类器的层数。
- nsplits 决定用于级联分类器的弱分类器。1 表示 stump 分类, 2 或者更多表示 CART 分类器。
- mem 训练所用的内存。决定训练的速度。
- sym-nonsym 指定训练的目标是否左右垂直对称。
- minhitrate 每层分类器所需要的最小的命中率。
- maxfalsealarm 每层分类器的最大错误报警率。
- mode 训练 haar 特征集的种类。all 表示使用 45 度矩形特征和垂直矩形。Basic 表示使用垂直矩形特征，
- w 训练样本宽度，以像素为单位。
- h 训练样本高度，以像素为单位。

在训练之前会有一个预处理步骤，就是要将负样本从背景图像中提取出来。这个负样本是随机在背景图像中任意选取一个矩形区域，并将其缩放到训练样本的尺寸而得到

的，并且需要通过前面所有层次的分类器。所以，当训练的层数越高时，寻找负样本的时间就花得越多。在训练过程中会出现一个数据栏，显示训练过程中的状况。其中“N”表示进行迭代的轮次；“%SMP”表示采用的样本的百分比（默认状况一般是 100%）；“F”表示是否存在翻转，“+”表示有（选用了-sym 参数才有效），“-”表示没有；“ST.THR”表示当前的判断阈值；“HR”表示 hitrate；“FA”表示 false alarm；“EXP.ERR”表示构成的强分类器目前的错误率。可以看到，HR 的值一般是略高于-minhitrate 参数，FA 会随着迭代次数的增加不断地下降，最终会降到低于规定的-maxfalsealarm 参数值而停止该层次分类器的训练。最后使用的特征个数也会和迭代的次数相关（图中使用的是 stumps 分类器，所以使用特征个数和迭代次数相同）。但是随着训练的进行，层次的深入，将正负样本区分的难度将越来越大，因此需要的特征也会越来越多，迭代的次数也越来越多。一般而言，训练样本数量越多，训练层次越多，训练的时间越长，如果想训练一个比较好的分类器一般要训练 2-3 天。训练一个 20 层的分类器，到了第 20 层的时候，预处理找寻负样本的时间通常会在几个小时以上，训练的时间也通常会用几个小时，第 20 层使用的特征会在一百至几百个不等，最后选取的特征会在几百到几千个不等，这比起一开始一二十万个的特征向量减少了很多。

训练过程中会在-data 文件夹中生成每一层分类器的参数，用记事本文件保存。最后训练完成后，会自动生成含有所有层次分类器参数的 xml 文件。

### 4.3 OpenCV 在 ARM 平台上的移植

所谓移植就是通过一定的转换修改工作在现有版本的基础上，生成软件单元的一个可在新的环境下运行的版本。软件单元在这里主要指应用程序、系统程序或一个程序的组件；一个软件系统是软件单元的集合；环境在这里具体指在安装过程中与所移植的软件产生相互作用的一切基础设施，通常包括处理器和操作系统，也可能包括 I/O 设备、各种软件库、网络或更大的人文和物理系统。目前软件移植研究的焦点主要集中于应用系统的移植，从软件工程的角度来看软件移植属于软件生命周期中的维护阶段，它是一种为适应变化了的环境而修改软件的活动<sup>[36]</sup>。

Opencv 视觉图像库在使用之前需要针对运行环境进行编译，生成相应环境下的静态链接库和动态链接库。下面是 ARM 嵌入式 Linux 环境下的编译过程。

(1) OpenCV 2.0 之后的版本，必须使用 CMake 创建 Makefile。增加环境变量路径：  
/usr /cmake/bin。

(2) 解压 OpenCV 2.1.1 到/usr/ Opencv-2.1.1 目录下，创建/usr/arm-OpenCV 目录，  
作为 CMake 编译的工作目录,安装交叉编译器 arm-linux-gcc 4.4.1，安装目前为  
/usr/arm/4.4.1，并添加环境变量：/usr/arm/4.4.1/bin。

(3) 在 windows 环境下，运行 cmake-gui 软件，源代码目录为:/usr/ /Opencv-2.1.1，  
编译目录为:/usr/arm-OpenCV,然后进行配置，配置过程保持 generator 为 Unix Makefiles，  
并选择 Specify options for cross-compiling 选项，操作系统栏填写 arm-linux，C 编译器  
栏填写/usr/arm-linux-gcc 的路径，C++ 编译器栏填写 arm-linux-g++的路径，程序库栏的  
Target Root 填写/usr/arm/4.4.1。

(4) 修改默认安装目录，默认的安装目录为/usr/local，对于交叉编译的库来说并不  
合适，所以我把 CMAKE\_INSTALL\_PREFIX 变量改为/usr/ /arm/lib/OpenCV/，修改  
CMakeCache.txt，在 CMAKE\_EXE\_LINKER\_FLAGS 后面加加上-lpthread -lrt，点击  
Generate 生成 Makefile。

(5)运行 make install,OpenCV 生成的库和头文件就安装到目录/usr/ /arm/lib/OpenCV/  
下，然后将生产的头文件和库文件放到根文件系统中，一起烧写到 nandflash 中。

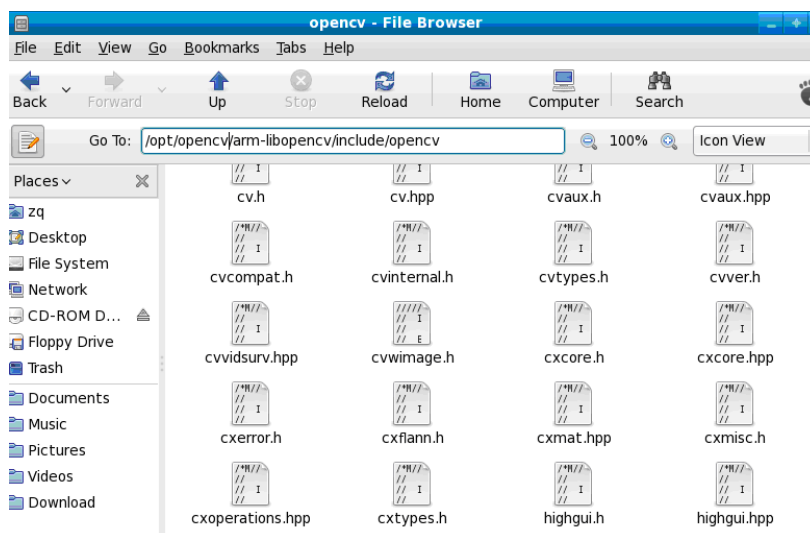


图 4.1 OpenCV 库文件

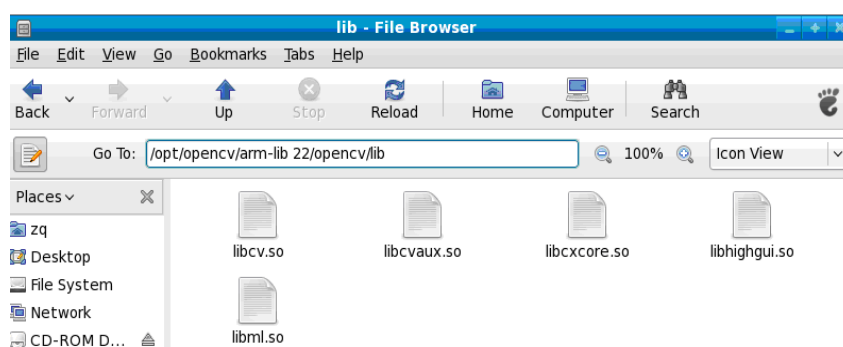


图 4.2 OpenCV 头文件

## 4.4 基于 OpenCV 的人脸检测

OpenCV 图像库成功移植到 ARM 嵌入式系统后，就可以在应用程序中调用图像库中的函数进行图像的处理了。

本系统使用 `cvHaarDetectObjects` 函数提取图像中的人脸的矩形区域，并且将这些区域作为一序列的矩形框返回。函数以不同比例大小的扫描窗口对图像进行几次搜索。每次都要对图像中的这些重叠区域利用 `cvRunHaarClassifierCascade` 进行检测。有时候也会利用某些继承技术以减少分析的候选区域，例如利用 Canny 边缘检测裁减的方法。函数在处理和收集到候选的方框（全部通过级联分类器各层的区域）之后，接着对这些区域进行组合并且返回一系列足够大的平均矩形。为了能对视频图像进行更快的实时检测，最小检测窗口设为  $70 \times 70$ ，程序如下：

```
Face_cascade=(CvHaarClassifierCascade*)cvLoad( cascade_name, 0, 0, 0 );  
img_storage = cvCreateMemStorage(0);  
CvSeq* faces =cvHaarDetectObjects(img, face_cascade,storage,1.2,2,0,cvSize(70, 70) );
```

## 4.5 实验结果及结论

系统主要由 FPGA 视频图像采集系统、ARM 嵌入式系统以及 TFT 显示屏组成。其中视频图像采集板卡主要用于采集数字图像并进行简单的图像预处理，ARM 嵌入式系统主要负责将视频图像采集系统传输过来的图像信息进行处理，并通过人脸检测算法进行人脸识别，最终的检测结果将在 TFT 屏上显示。

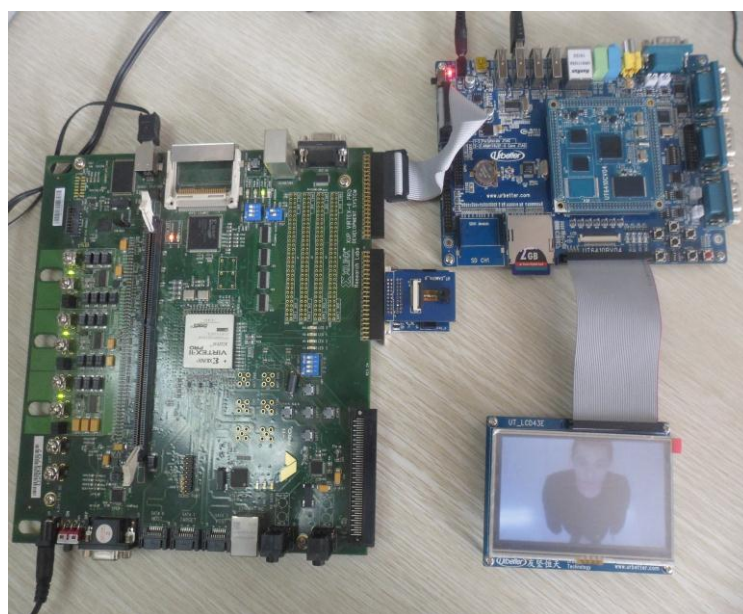


图 4.3 实验平台搭建

当系统采集到人脸信息时，系统将会在液晶显示器中用人脸大小的方框标记出人脸所在的位置，如图 5.2 所示。系统的人脸检测速度大约为 2 帧 / 秒，人脸在屏幕中以正常速度移动时，系统可实检测、定位人脸的信息。

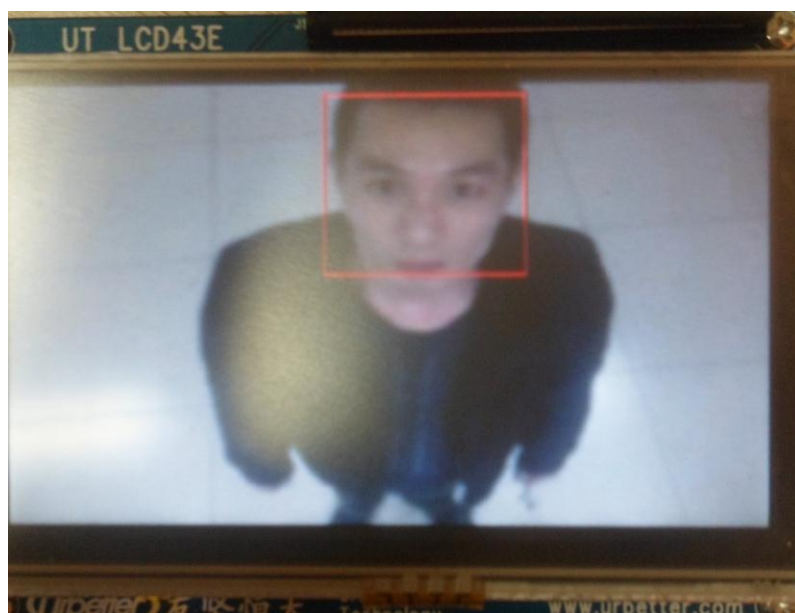


图 4.4 实验现场效果

实验结果表明，本文设计人脸检测系统检测精度高，实时性好，相比于传统的 PC 机人脸检测系统，具有体积小，成本低、便于安装携带等优点，而且在数据的处理的速度上满足应用要求，具有实用价值。

## 4.6 本章小结

本章主要内容是介绍了 OpenCV 的优点，使用 OpenCV 制作人脸分类器，并使用 OpenCV 库函数设计了人脸检测的应用程序，最后在搭建的硬件平台上进行人脸检测试验，实验结果表明：检测结果正确，实时性良好，系统运行稳定。

## 第五章 总结与展望

针对人脸识别实时性方面的需求,设计了一种基于 FPGA+ARM 的人脸检测系统,实现了图像采集、处理、检测及结果显示等功能,是嵌入式图像处理领域的一种新的尝试。

### 5.1 本论文的主要研究工作

本文主要完成如下工作:

- (1) 通过查阅大量国内外文献资料,研究了人脸识别的发展历程和常用的算法,针对人脸检测在许多实时性较强的场合应用,选择 FPGA+ARM 作为实现开发平台。
- (2) 比较了几种常见的人脸检测算法,重点对基于 Adaboost 算法的人脸检测的原理做了详细的分析和研究。
- (3) 根据系统功能要求,选用 xilinx 公司 SP3E 芯片完成 CMOS 的驱动控制和图像预处理,完成 FPGA 系统的硬件设计和软件实现。采用 2 块 SRAM 交替进行数据读写的方式,保证了数据采集的连续性和实时性。
- (4) 在 ARM11 平台上,建立 LINUX 系统的开发环境,并将 OPEN CV 移植到系统中,完成了基于 OPenCV 的人脸检测程序设计。实际搭建了图像采集的硬件平台并移植了设计好的软件程序,通过实验验证了系统的正确性和稳定性。

### 5.2 工作的不足及下一步的工作:

- 1) 受测试环境的影响较大,在复杂环境下,鲁棒性较差。
- 2) 当图像数据量较大时 ARM 处理器在数据处理方面还是不够快。可以考虑采用 FPGA+ARM+DSP 的构架,将 ARM 作为主处理器,FPGA 和 DSP 作为协处理器,进一步的提高系统的处理效率。

## 参考文献

1. 贾云得. 机器视觉[M]. 科学出版社, 2000.
2. 吕孟军, 郭琪, 张家峰. 航空发动机状态智能识别[J]. 微计算机信息, 2005, 2: 60-62.
3. 张翠平, 苏光大. 人脸识别技术综述[J]. 中国图象图形学报, 2000, 5(11): 885-894.
4. Zhao W, Krishnaswamy A, Chellappa R, et al. Discriminant analysis of principal components for face recognition[M]//Face Recognition. Springer Berlin Heidelberg, 1998: 73-85.
5. Wiskott L, Fellous J M, Kuiger N, et al. Face recognition by elastic bunch graph matching[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1997, 19(7): 775-779.
6. 田启川, 张润生. 生物特征识别综述[J]. 计算机应用研究, 2009, 26(12): 4401-4406.
7. 艾海舟, 乐秀宇. 面向视觉监视实时跟踪的动态背景更新方法[J]. 计算机工程与应用, 2001, 37(19): 104-106.
8. 束为, 荣钢, 边肇祺, 等. 利用掌纹进行身份自动鉴别方法的研究[J]. 清华大学学报(自然科学版), 1999, 39(1): 95-97.
9. John D.Woodward Jr.Nicholas M.Orlans 等著, 阵菊明,邓启威等译.生物认证[M].清华大学出版社.2004.
10. Boulgouris N V, Hatzinakos D, Plataniotis K N. Gait recognition: a challenging signal processing technology for biometric identification[J]. signal processing magazine, IEEE, 2005, 22(6): 78-90.
11. Jain A, Hong L, Pankanti S. Biometric identification[J]. Communications of the ACM, 2000, 43(2): 90-98.
12. Rein L H.Face detection and modeling for recognition[D].Michigan State University, 2002.



13. Chen L F, Liao H Y M, Ko M T, et al. A new LDA-based face recognition system which can solve the small sample size problem[J]. Pattern recognition, 2000, 33(10): 1713-1726.
14. 晨风. 嵌入式实时多任务软件开发基础[M], 北京: 清华大学出版社, 2004.
15. Bruce Powel Douglass. 嵌入式与实时系统开发—使用 UML、对象技术、框架与模式 [M], 北京: 机械工业出版社, 2005.
16. 杨学友,李博文,曲兴华. 基于 FPGA 高速智能化视觉传感器的研究[J]电子测量与仪器学报, 2003, (01) .
17. 邓海峰,余慧英,袁可风. 一种嵌入式图像处理平台的设计与实现[J]. 计算机应用, 2005, (S1) .
18. 姚聪,方勇,汪敏,潘志浩. 基于 ARM 的嵌入式图像采集处理系统及其无线传输[J]. 电子技术, 2003, (11) .
19. 梁路宏, 艾海舟. 基于多模板匹配的单人脸检测[J]. 中国图象图形学报: A 辑, 1999, 4(10): 825-830.
20. 刘青山, 卢汉清, 马颂德. 综述人脸识别中的子空间方法[J]. 自动化学报, 2003, 29(6): 900-911.
21. Coughlan J, Yuille A, English C, et al. Efficient deformable template detection and localization without user initialization[J]. Computer Vision and Image Understanding, 2000, 78(3): 303-319.
22. Yuille A L. Deformable templates for face recognition[J]. Journal of Cognitive Neuroscience, 1991, 3(1): 59-70.
23. Carey S, Diamond R, Woods B. Development of face recognition: A maturational component?[J]. Developmental Psychology, 1980, 16(4): 257.
24. Wiskott L, Fellous J M, Kuiger N, et al. Face recognition by elastic bunch graph matching[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1997, 19(7): 775-779.
25. Pentland A, Choudhury T. Face recognition for smart environments[J]. Computer, 2000, 33(2): 50-55.

26. Heisele B, Ho P, Poggio T. Face recognition with support vector machines: Global versus component-based approach[C]//Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on. IEEE, 2001, 2: 688-694.
27. Li S Z. Handbook of face recognition[M]. Springer-Verlag London Limited, 2011.
28. Yu H, Yang J. A direct LDA algorithm for high-dimensional data-with application to face recognition[J]. Pattern recognition, 2001, 34(10): 2067.
29. Sim T, Baker S, Bsat M. The CMU pose, illumination, and expression database[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2003, 25(12): 1615-1618.
30. 梁路宏, 艾海舟, 徐光祐, 等. 人脸检测研究综述[J]. 计算机学报, 2002, 25(5): 449-458.
31. 梁路宏, 艾海舟, 肖习攀, 等. 基于模板匹配与支持向量机的人脸检测[J]. 计算机学报, 2002, 25(1): 22-29.
32. 张燕昆, 杜平, 刘重庆. 基于主元分析与支持向量机的人脸识别方法[J]. 上海交通大学学报, 2002, 36(6): 884-886.
33. 温洲, 邵晓巍, 龚德仁. 基于 BP 神经网络的人脸识别后续分类算法[J]. 计算机应用, 2011, 31(2): 133-136.
34. 刘党辉, 沈兰荪, Lam K M. 人脸检测研究进展[J]. 计算机工程与应用, 2003, 39(28): 5-9.
35. 山世光, 高文, 唱铁钲, 等. 人脸识别中的“误配准灾难”问题研究[J]. 计算机学报, 2005, 28(5): 782-791.
36. 李子青. 人脸识别技术应用和 market 分析[J]. 中国安防, 2007, 8: 42-46.
37. Zhao W, Chellappa R, Phillips P J, et al. Face recognition: A literature survey[J]. Acm Computing Surveys (CSUR), 2003, 35(4): 399-458.
38. Phillips P J, Grother P, Micheals R. Evaluation methods in face recognition[M]//Handbook of Face Recognition. Springer London, 2011: 551-574.
39. Huang G B, Mattar M, Berg T, et al. Labeled faces in the wild: A database for studying face recognition in unconstrained environments[C]//Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition. 2008.

40. Gross R, Brajovic V. An image preprocessing algorithm for illumination invariant face recognition[C]//Audio-and Video-Based Biometric Person Authentication. Springer Berlin Heidelberg, 2003: 10-18.
41. 王荣本,游峰,崔高健,郭烈. 基于计算机视觉高速智能车辆的道路识别[J]计算机工程与应用, 2004, (26) .
42. 赵方鹏,杨建华,赵忠,刘文涛. 基于嵌入式 Linux 的网络视频监控系统[J]. 测控技术, 2007,(05).
43. 陈育香. 基于 ARM 的无线传感器网络多帧图像采集与处理[D]南京理工大学, 2011
44. 谢平. 嵌入式短距离无线音视频监控系统设计[J]. 计算机测量与控制, 2010,(08) .
45. 张远, 张首军, 汪俊. 基于 SOPC 的嵌入式网络摄像头的设计[J]. 西北大学学报(自然科学版), 2009,(01).
46. 龚向东,刘春平,黄虹宾,曾振兴. 一种基于 Nios II 软核的嵌入式图像采集处理系统设计[J]. 电子测量技术, 2010, (02) .
47. Ahonen T, Hadid A, Pietikainen M. Face description with local binary patterns: Application to face recognition[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2006, 28(12): 2037-2041.
48. 姜军, 张桂林. 一种基于知识的快速人脸检测方法[J]. 中国图象图形学报, 2002.
49. Yang G, Huang T S. Human face detection in a complex background[J]. Pattern recognition, 1994, 27(1): 53-63.
50. 王蕴红, 范伟, 谭铁牛. 融合全局与局部特征的子空间人脸识别算法[J]. 计算机学报, 2005, 28(10): 1657-1663.
51. Rowley H A, Baluja S, Kanade T. Neural network-based face detection[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1998, 20(1): 23-38.
52. 谢赛琴, 沈福明, 邱雪娜. 基于支持向量机的人脸识别方法[J]. 计算机工程, 2009, 35(16): 186-188.
53. 沈杰, 王正群, 邹军, 等. 基于连续隐马尔可夫模型的人脸识别方法[J]. 计算机工程与设计, 2008, 29(3): 707-709.
54. Viola P, Jones M J. Robust real-time face detection[J]. International journal of computer

- vision, 2004, 57(2): 137-154.
55. 涂刚, 阳富民, 胡贯荣. 嵌入式操作系统综述[J]. 计算机应用研究, 2000, 17(11): 4-6.
56. Bhatti S, Carlson J, Dai H, et al. MANTIS OS: An embedded multithreaded operating system for wireless micro sensor platforms[J]. Mobile Networks and Applications, 2005, 10(4): 563-579.
57. 林建民. 嵌入式操作系统技术发展趋势[J]. 计算机工程, 2001, 27(10).
58. Bovet D P, Cesati M. Understanding the Linux kernel[M]. O'Reilly Media, 2008.
59. 王学龙. 嵌入式 Linux 系统设计与应用[M]. 清华大学出版社, 2001.
60. 孙纪坤, 张小全. 嵌入式 Linux 系统开发技术详解: 基于 ARM[M]. 人民邮电出版社, 2006.
61. Roy S, Wolff C, Ingham P W. The u-boot mutation identifies a Hedgehog-regulated myogenic switch for fiber-type diversification in the zebrafish embryo[J]. Genes & Development, 2001, 15(12): 1563-1576.
62. Bradski G, Kaehler A. Learning OpenCV: Computer vision with the OpenCV library[M]. O'Reilly Media, Incorporated, 2008.
63. 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.
64. 曹志刚. 基于 LINUX+ ARM 的视频系统的应用和开发[D]. 北方工业大学, 2010.
65. 胡静波. 基于 OpenCV 的嵌入式视频监控系统关键技术的研究[D]. 云南大学, 2010.

## 攻读硕士学位期间发表的论文

### 一、发表的论文

- 1) 张挺飞, 樊永生, 孙凌云. 基于 FPGA+ ARM 的智能图像门控系统[J]. 电视技术, 2012, 36(17): 156-158.
- 2) 孙凌云, 樊永生, 张挺飞. 基于 MCA 和 Context 模型结合的磁共振图像去噪[J]. 电视技术, 2012, 36(23): 12-14.

## 致 谢

本文的研究工作是在导师樊永生教授的指导和帮助下完成的。在攻读硕士学位的两年多时间里，敬爱的樊老师以渊博的知识，严谨的治学态度，活跃的学术思想，一丝不苟的工作作风，极强的敬业精神深深地影响着我，使我在学习，思想，为人处事等方面取得了很大的进步，在此谨向樊永生导师表示深深的谢意和敬意。

同时要感谢实验室的孙凌云同学，在课题研究过程中给予的帮助和支持，感谢实验室的其他曾给予我帮助和支持的同学。

衷心感谢我的父母和亲人，感谢他们对我的关心、理解与支持。是他们的鼓舞使我走进研究生生活，也正是他们一如既往的关爱与呵护，使我顺利地完成了学业。

向所有曾帮助过我的老师、同学致以我最真挚的谢意！

谨向在百忙之中抽出时间审阅论文和参加答辩的各位老师表示我最由衷的谢意！