

天津大学

毕业设计（论文）说明书



学 院 计算机科学与技术

专 业 计算机科学与技术

年 级 2003 级

姓 名 陈颖

指导教师 何英华

2007 年 6 月 15 日

天津大学

毕业设计（论文）任务书

题目：基于特征的人脸检测系统的设计与实现

学生姓名陈颖

学院名称计算机科学与技术

专 业计算机科学与技术

学 号3003204004

指导教师何英华

职 称讲师

一、原始依据（包括设计或论文的工作基础、研究条件、应用环境、工作目的等。）

工作基础：人脸检测的研究目前已成为计算机视觉和模式识别领域中一个重要的研究课题。迄今为止，涌现出了各种各样的人脸检测算法，但在实用要求中仍面临着许多严峻的问题，还有很大的探索空间。本课题的提出是在基于目前人脸检测研究的基础上发展创新，利用 OpenCV 开源计算机视觉库中的一些通用算法，设计并实现一个更有效的基于特征的人脸检测系统。

研究条件：一台 CPU 为 Intel P4 2.8E，内存为 512M 的 PC 机。编程语言采用 C++，使用 VC++ 6.0 平台，利用 OpenCV 1.0 开源计算机视觉库，测试图像集由数码相机采集和互联网下载。

应用环境：灰度图像和彩色图像的人脸检测，所支持的文件格式具体包括：Windows 位图文件格式-BMP, DIB；JPEG 文件格式-JPEG, JPG, JPE；便携式网络图片-PNG；便携式图像格式-PBM, PGM, PPM；Sun rasters-SR, RAS；TIFF 文件-TIFF, TIF。

工作目的：设计并实现一个基于特征的人脸检测系统。该系统对于给定的可能包含人脸的输入图像，能够判断其中是否有人脸；若有，则将所有的人脸从背景中标记出来。该系统要能适应各种复杂环境，能对任意背景下的任意数目和大小的人脸进行检测，同时要具有快速的检测速度。

二、参考文献

- [1]黄福珍，苏剑波．人脸检测[M]．上海：上海交通大学出版社，2006．1—14．
- [2]张宏林，蔡锐．Visual C++数字图像模式识别技术及工程实践[M]．北京：人民邮电出版社，2003．363—370．
- [3]王金庭，杨敏．人脸检测技术研究[J]．计算机系统应用，2006，4：31—33．
- [4]周书仁，梁昔明，叶吉祥等．基于脸部信息和支持向量机的人脸检测[J]．计算机应用，2006，26（5）：1032—1034．
- [5]Intel．Open Source Computer Vision Library (OpenCV) [EB/OL]．<http://www.opencv.org.cn>，2007-6-1．
- [6]黎松，平西建，丁益洪．开放源代码的计算机视觉类库 OpenCV 的应用[J]．计算机应用与软件，2005，22（8）：134—136．
- [7]Rein-Lien Hsu, Mohamed Abdel-Mottaleb, Anil K. Jain. Face Detection in Color Images[J]. IEEE TRANS. PAMI, 2002, 24(5): 696—706．

三、设计（研究）内容和要求（包括设计或研究内容、主要指标与技术参数，并根据课题性质对学生提出具体要求。）

1. 研究内容：

- (1) 熟悉数字图像处理的基本内容以及人脸检测的基本知识。
- (2) 熟悉 OpenCV 中的基本函数，包括图像的输入、输出、显示、缩放等等。
- (3) 熟悉 OpenCV 中的人脸检测部分的函数。
- (4) 实用 VC++ 搭建人脸检测部分的框架。
- (5) 实现人脸特征提取部分的功能，包括肤色在色彩空间中的聚类特性以及肤色模型的建立；人脸眼睛，鼻子，嘴等面部细节特征等。
- (6) 实现人脸检测部分的功能。可以尝试两种途径，第一种根据检测出的人脸特征的色彩或者特征之间的几何关系直接检测出人脸。第二种首先根据提取出的训练样本的特征训练出检测人脸的分类器，然后使用分类器来进行测试样本的人脸检测。
- (7) 探讨光照变化环境下的人脸检测的特点，并设计算法尝试更好地解决这个问题

2. 主要指标与技术参数：

人脸检测的检测正确率，错误报警率，检测的开始到人脸被检测出的时间长短等。

3. 具体要求

学习并掌握 OpenCV 开源计算机视觉库中的一些通用算法；熟悉 VC++ 编程。

指导教师（签字）

年 月 日

审题小组组长（签字）

年 月 日

天津大学本科生毕业设计（论文）开题报告

课题名称	基于特征的人脸检测系统的设计与实现		
学院名称	计算机科学与技术	专业名称	计算机科学与技术
学生姓名	陈颖	指导教师	何英华
（内容包括：课题的来源及意义，国内外发展状况，本课题的研究目标、研究内容、研究方法、研究手段和进度安排，实验方案的可行性分析和已具备的实验条件以及主要参考文献等。） 人脸检测问题最初来源于人脸识别。20 世纪 90 年代以来，随着网络安全和电子商务等应用需要的剧增，人脸识别由于具有非侵犯性以及直接、友好和方便的特点，成为最有潜力的生物身份鉴别方式，出现了实际的人脸识别系统和商用产品。而人脸检测是自动人脸识别系统的一个关键环节，人脸检测算法的精度直接影响着整个识别系统的性能。这种应用背景使得人们对人脸检测的各个重要方面引起了极大的研究兴趣，人脸检测也逐渐独立出来成为一个专门的研究方向。 随着研究的深入和应用的扩大，今天，人脸检测已远远超过了人脸识别的范畴。人脸检测除了应用于人脸识别外，在视频会议，出入口控制，证件验证，基于内容的图像检索（CBIR）以及智能人机交互等领域都有着极其重要的作用，人脸检测已经逐渐发展成为计算机视觉和模式识别领域的一个重要的研究课题。 目前，国外对人脸检测问题的研究很多，美国，英国等许多国家有大量的研究组从事人脸检测的研究，这些研究受到军方，警方以及一些公司的高度重视和资助，比较著名的有美国麻省理工大学（MIT）的媒体实验室和人工智能实验室，卡内基梅隆大学（CMU）的机器人研究所以及 Illinois 大学的 Beckman 研究所等。MPEG7 标准组织已经建立了人脸识别草案小组，人脸检测算法也是一项征集的内容。随着人脸检测研究的深入，国际上发表的有关论文数量也大幅度增加，每年的国际会议上关于这方面的专题也屡屡可见。在国内，也有许多单位和学校从事人脸检测与识别相关的研究。如清华大学，上海交通大学，亚洲微软研究院，中科院计算技术研究所和中科院自动化研究所等都有人员在从事人脸检测相关的研究。自 1995 年以来，国外一些公司开发了多个人脸识别系统，其中大多数产品将人脸检测作为系统的一个重要组成部分，有代表性的有如下产品：德国 Cognitec Systems 公司的 Face VACS 人脸识别系统，美国 Identix 公司的 FaceIt 人脸识别系统，加拿大 Imagis 公司的 ID-2000 面部识别软件，美国 Viisage 公司的 FaceTools 人脸识别系统等。国内也有一些关于人脸检测方面的研究已经或正在转化为商品软件，如上海银晨智能识别科技有限公司开发的人脸检测音视频压缩卡，广州超越软件公司的 BioFace 系			

统等等。

虽然人脸检测领域的研究已经取得了一些可喜的成果,但在实用要求中仍面临着许多严峻的问题。人脸的非刚性,表情、姿态、发型和化妆的多样性,眼镜、胡须等附属物的存在,以及环境、光照的复杂性都给正确检测人脸带来了困难。要让计算机像人一样方便准确地检测出无约束环境下任意图像中的人脸,还需要进一步的深入研究,还有很大的探索空间。如果能够找到解决这些问题的方法,成功地构造出人脸检测系统,将为解决其他类似的复杂模式的检测问题提供重要的启示。

本课题的研究目标是利用 OpenCV 开源计算机视觉库,设计并实现一个适用于各种复杂环境的基于特征的人脸检测系统。该系统对于给定的可能包含人脸的输入图像,能够判断其中是否有人脸;若有,则将所有人脸从背景中标记出来。

本课题的研究内容如下:

1) 利用 OpenCV 中的基本函数,实现数字图像处理的基本功能,包括图像的输入、显示、缩放、旋转、保存等等。开发人脸检测系统 GUI 软件界面。

2) 为适应环境的复杂性,处理图像中存在的噪声,光照变化等,实现数字图像预处理的相关功能,包括图像的亮度/对比度调节、平滑、腐蚀、膨胀、直方图均衡化、Gamma 校正、光照补偿、色彩平衡等等。

3) 熟悉 Boosted Cascade 方法,利用 OpenCV 中的人脸检测部分的函数,实用 VC++ 搭建人脸检测部分的框架,实现人脸检测的基本功能。

4) 研究人脸检测算法,利用人脸特征,包括肤色特征及人脸眼睛等器官特征,改进人脸检测功能的实现,降低误检率,提高检测正确率。

5) 探讨光照变化环境下的人脸检测特点,并设计算法尝试更好地解决此问题。

本课题采用基于肤色特征的人脸检测算法和基于 Boosted Cascade 的人脸检测算法,利用 OpenCV 开源计算机视觉库中的一些通用算法等手段来设计和实现基于特征的人脸检测系统。

本课题进度安排如下:

2006 年 11 月—2006 年 12 月,完成开题报告,熟悉数字图像处理的基本内容。

2006 年 12 月—2007 年 1 月,熟悉人脸检测的基本知识。

2007 年 1 月—2007 年 2 月,熟悉 OpenCV 中的基本函数和人脸检测部分的函数。实用 VC++ 搭建人脸检测部分的框架。

2007 年 2 月—2007 年 3 月,实现基于肤色特征的人脸检测部分的功能。

2007 年 3 月—2007 年 4 月,实现基于 Boosted Cascade 的人脸检测功能,并对算法进行研究,尝试利用人脸特征,包括肤色特征及眼睛特征等设计改进方案。

2007 年 4 月—2007 年 5 月,探讨光照变化环境下的人脸检测的特点并设计算法解决这个问题。完成论文初稿。

2007 年 6 月,完善系统,完成论文终稿并进行答辩。

下面对实验方案的可行性作简要分析。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、表情等的变化情况都能保持不变，有相对稳定性，况且大多数和背景物体的颜色相区别，因此肤色特征在人脸检测中是最有效的特征之一。Boosted Cascade 算法检测正确率较高且具有极高的运行速度，是目前现有的人脸检测算法中效果最好的算法之一，本课题利用 OpenCV 开源计算机视觉库，使用 Boosted Cascade 算法进行人脸检测，这也是一种可行并且有效的做法。本课题还希望在算法上做一些改进以提高系统性能并适应一些如光照变化、表情和姿态变化等复杂情况。当然人脸拍摄的复杂环境及人脸模式的多样性等因素可能会给系统带来一定问题，不过，总体来说，本课题的方案是可行的。

本课题的实验条件如下：一台 CPU 为 Intel P4 2.8E，内存为 512M 的 PC 机。编程语言采用 C++，使用 VC++ 6.0 平台，利用 OpenCV 1.0 开源计算机视觉库，测试图像集由数码相机采集和互联网下载。

主要参考文献如下：

- [1]黄福珍，苏剑波．人脸检测[M]．上海：上海交通大学出版社，2006．1—14．
- [2]张宏林，蔡锐．Visual C++数字图像模式识别技术及工程实践[M]．北京：人民邮电出版社，2003．363—370．
- [3]王金庭，杨敏．人脸检测技术研究[J]．计算机系统应用，2006，4：31—33．
- [4]周书仁，梁昔明，叶吉祥等．基于脸部信息和支持向量机的人脸检测[J]．计算机应用，2006，26（5）：1032—1034．
- [5]Intel．Open Source Computer Vision Library (OpenCV) [EB/OL]．<http://www.opencv.org.cn>，2007-6-1．
- [6]黎松，平西建，丁益洪．开放源代码的计算机视觉类库 OpenCV 的应用[J]．计算机应用与软件，2005，22（8）：134—136．
- [7]Rein-Lien Hsu，Mohamed Abdel-Mottaleb，Anil K. Jain．Face Detection in Color Images[J]．IEEE TRANS．PAMI，2002，24(5)：696—706．

选题是否合适： 是 ☐ 否 ☐

课题能否实现： 能 ☐ 不能 ☐

指导教师（签字）

年 月 日

选题是否合适： 是 ☐ 否 ☐

课题能否实现： 能 ☐ 不能 ☐

审题小组组长（签字）

年 月 日

摘 要

目前, 人脸检测已成为计算机视觉和模式识别领域中最活跃的研究课题之一。论文对人脸检测进行了详细综述, 并着重介绍了在利用基于肤色特征的人脸检测算法和基于 Boosted Cascade 的人脸检测算法进行人脸检测方面的一些成果。

论文第一部分介绍了基于肤色特征的人脸检测算法。该算法首先构建肤色模型得到候选人脸, 然后对每个候选区域进行脸部细节验证, 最后输出满足验证的人脸。论文还通过光照补偿和色彩平衡等预处理功能解决光照变化问题。实验结果表明该算法能适应光照、表情变化和部分遮挡的情况, 检测正确率高且速度快。

论文第二部分介绍了基于 Boosted Cascade 的人脸检测算法。该算法采用积分图的图像表示方法和级联式的分类结构, 能快速准确地检测出人脸。论文还提出了通过肤色验证、眼睛验证来排除误检区域及正面与侧面检测结合的改进方案。改进后的算法在不影响检测速度的前提下, 能够有效提高检测正确率, 降低错误报警率。

关键词: 人脸检测; 肤色模型; Boosted Cascade 算法; 光照变化;

ABSTRACT

Face detection is currently one of the most active research topics in the domain of computer vision and pattern recognition. This paper detailedly summarizes the topic of face detection and emphatically introduces some achievements in face detection using the approach based on skin color information and Boosted Cascade arithmetic.

In the first part of this paper, an approach to face detection based on skin color features is introduced. First, one method that builds the skin color model is used to segment the candidate face areas in a color image. Then, the facial details of each candidate face area are used to examine. Finally, the faces that satisfy all requirements are marked in the original image. The paper also solves the problem of lighting variances by adding preprocessing functions of Lighting Compensation and Color Balance. The experimental results show that the proposed approach is effective to detect faces with variations in lighting, facial expression and partial occlusion.

In the second part of this paper, an approach to face detection based on Boosted Cascade arithmetic is introduced. This arithmetic adopts the image representation of integral image and cascade classificatory structure, so that the arithmetic can detect faces quickly and accurately. This paper also presents solutions of improving Boosted Cascade arithmetic, that is using the model of skin color or eyes feature to verify the result of detection and combining frontal face detection with profile face detection. The improved arithmetic can increase the True Positive Rate and decrease the False Alarm Rate without slowing down the speed.

Key words: face detection; skin color model; boosted cascade arithmetic; lighting variances;

目 录

第一章	前言.....	1
1.1	引言.....	1
1.2	课题背景和意义.....	1
1.3	国内外研究概况.....	2
1.4	课题研究内容.....	3
1.5	全文结构.....	4
第二章	人脸检测相关理论及 autofaceGFB 系统简介..	5
2.1	人脸检测相关理论及研究情况.....	5
2.1.1	人脸检测基本思想和方法简介	5
2.1.2	人脸检测算法的评价	6
2.2	autofaceGFB 人脸检测系统简介	7
2.2.1	系统简介和功能说明	7
2.2.2	系统的总体结构	8
2.3	autofaceGFB 人脸检测系统开发环境说明	9
第三章	基于肤色特征的人脸检测.....	10
3.1	概述.....	10
3.2	肤色模型理论.....	10
3.3	光照变化下的人脸检测策略.....	12
3.4	基于肤色特征的人脸检测的实现.....	15
3.4.1	算法伪代码	15

3.4.2	算法实现流程图.....	17
3.5	测试及评价.....	17
第四章	基于 Boosted Cascade 算法的人脸检测.....	20
4.1	概述.....	20
4.2	Boosted Cascade 算法原理	20
4.2.1	Haar 特征的选取及特征值的计算.....	21
4.2.2	Boosted Cascade 多层分类器级联结构.....	22
4.3	基于 Boosted Cascade 算法的人脸检测的实现.....	24
4.3.1	算法伪代码	24
4.3.2	算法实现流程图	25
4.3.3	测试及评价	25
4.4	基于肤色验证的 Boosted Cascade 人脸检测方案...	27
4.5	基于眼睛验证的 Boosted Cascade 人脸检测方案...	30
4.6	正面与侧面检测结合的人脸检测方案.....	32
第五章	总结与展望.....	35
5.1	全文总结.....	35
5.2	研究展望.....	35
参考文献		37
附录		38
外文资料		
中文译文		
致谢		

第一章 前言

1.1 引言

目前，人脸检测已成为计算机视觉和模式识别领域中最活跃的研究课题之一。所谓人脸检测，是指在输入图像中判断是否存在人脸，若存在，则将所有人脸从背景中标记出来。

人脸检测大致可以划分为两种类型^[1]：一类是在简单的人脸图像中进行人脸检测，这类图像包括各类证件上的大脸照，坐在摄像头前面录制的包含人头和肩膀的图像，以及用于进行人脸识别研究而专门拍摄的带有很大限制条件的人脸图像。这里将这一类人脸检测称为“简单人脸检测”。另一类则是在任意的图像中进行人脸检测，对图像没有任何限制条件，图像的背景可以是任意的，人脸的个数可以是任意的，人脸的大小和出现的位置也可以是任意的。这里将这一类人脸检测称为“通用人脸检测”。本课题就是尝试设计并实现一个属于“通用人脸检测”范畴的，基本能够适用于各种复杂情况，准确率较高的，检测耗时较少的基于特征的人脸检测系统。

1.2 课题背景和意义

人脸检测问题最初来源于人脸识别，一个完整的人脸自动识别系统应该包括人脸检测、特征提取以及匹配识别这三大基本过程^[2]。早期的人脸识别研究主要针对具有约束条件较强的人脸图像，往往假设人脸位置已知或很容易获得，即属于“简单人脸检测”范畴，因此人脸检测问题的重要性并没有凸显出来。进入 20 世纪 90 年代以后，随着社会的发展，诸如视觉监控、远程教育、智能人机交互技术及信息安全等各方面都迫切希望能够进行快速、有效的身份验证。生物特征因为自身具有的稳定性和差异性，已经成为身份验证的主要手段。与其他利用视网膜识别（无法确保对人体的安全性）及指纹识别（涉及个人隐私）等人体生物特征进行身份验证相比，人脸识别的应用具有非侵犯性以及直接、友好和方便的特点，成为最有潜力的生物身份鉴别方式，随即出现了实际的人脸识别系统和商用产品，而这种应用背景要求自动人脸识别系统能够对一般环境图像具有一定的适应能力，即适应“通用人脸检测”范畴的应用，由此所面临的一系列问题使得人们对人脸检测的各个重要方面引起了极大的研究兴趣，人脸检测也逐渐独立出来成为一个专门的研究方向。

随着多媒体和人工智能在安全、信息等领域的应用日益广泛，人脸检测在各方面的需求也日渐增多，今天，人脸检测的应用范畴已远远超过了人脸识别，它可以被应用在各种不同的领域，比如身份认证、楼宇安全控制、重要场所中的监控和信息安全领域（计算机的登录控制、应用程序安全使用、数据库安全访问和文件加密、实现局域网和广域网的安全控制、保护电子商务的安全性）等等^[1,2]。

其常见的应用归纳起来有以下几个方面：

- 1) 人脸识别。如前所述，人脸检测最初来源于人脸识别，是人脸自动识别系统中至关重要的环节，因此，人脸识别是人脸检测重要的应用，其应用前景广泛，如刑侦破案、证件验证、出入口控制以及视频监控等。
- 2) 视频会议。可视电话和视频会议是基于内容的编码传输方法的典型应用，而他们所感兴趣的传输内容是人脸。一个鲁棒的、实时的人脸检测系统所能提供的精确的人脸区域是这些应用切实可行的前提。
- 3) 图像与视频检索。基于内容的图像检索尤其是视频检索的研究方兴未艾，人脸作为一种重要的、稳定的、具有一定语义的检索特征，在新闻片、故事片等各种题材类型的视频中广泛存在，可成为图像检索的条件，因此成为这类系统中的重要组成部分，利用检测所得人脸，可以有效标注、索引及分类视频。
- 4) 智能人机交互。是在人机之间实现类似于人类之间的非精确的自然交互方式，它是集语音、手语、人脸、表情、唇读、头势、体势等多通道为一体的计算机智能接口系统。显然，在这种以人为中心的交互方式中，人脸检测是系统中十分重要的一环。

1.3 国内外研究概况

目前，国外对人脸检测问题的研究很多，美国，英国等许多国家有大量的研究组从事人脸检测的研究，这些研究受到军方，警方以及一些公司的高度重视和资助。对人脸检测问题的研究比较著名的有美国麻省理工大学（MIT）的媒体实验室和人工智能实验室，卡内基梅隆大学（CMU）的机器人研究所以及 Illinois 大学的 Beckman 研究所等^[1,2]。

MPEG7 标准组织已经建立了人脸识别草案小组，人脸检测算法也是一项征集的内容。随着人脸检测研究的深入，国际上发表的有关论文数量也大幅度增加，每年的国际会议上关于这方面的专题也屡屡可见。

在国内，也有许多单位和学校从事人脸检测与识别相关的研究。如清华大学，上海交通大学，亚洲微软研究院，中科院计算技术研究所和中科院自动化研究所等都有人员在从事人脸检测相关的研究。

自 1995 年以来，国外一些公司看准了人脸识别系统在商贸和保安方面的广阔应用前景，动用了大量人力和物力，独立研制或与高校合作，开发了多个人脸识别系统，其中大多数产品将人脸检测作为系统的一个重要组成部分，有代表性的有如下产品：德国 Cognitec Systems 公司的 Face VACS 人脸识别系统；美国 Identix 公司的 FaceIt 人脸识别系统；加拿大 Imagis 公司的 ID-2000 面部识别软件；美国 Viisage 公司的 FaceTools 人脸识别系统等。

国内也有一些关于人脸检测方面的研究已经或正在转化为商品软件，如上海

银晨智能识别科技有限公司开发的人脸检测音视频压缩卡, 广州超越软件公司的 BioFace 系统等等。

1.4 课题研究内容

目前, 理想条件下的人脸检测问题已经取得了很大的进展, 但在复杂环境下, 对任意光照条件、面部有遮挡、姿态和表情变化等情况下的人脸检测仍是一个比较困难的问题。由于人脸自身结构复杂, 某些局部特征具有随机性(如眼镜、胡须、发型等)而且还存在着不同表情的人脸, 以及光照变化等环境的复杂性, 成像时亮度、对比度等因素的影响使图像不清晰, 人与背景区别不大, 这些都给正确检测人脸带来了困难, 尤其是光照条件的变化十分影响人脸检测系统的性能。要让计算机像人一样快速准确地检测出任意环境下任意图像中的任意数目的人脸, 还需要进一步的深入研究, 还有很大的探索空间。如果能够找到解决这些问题的方法, 成功地构造出人脸检测系统, 将为解决其他类似的复杂模式的检测问题提供重要的启示。

本课题针对这种情况, 在国内外同类工作的研究基础上, 利用 OpenCV 开源计算机视觉库中的一些通用算法, 设计并实现一个属于“通用人脸检测”范畴的, 基本能够适用于各种复杂情况, 准确率较高的, 检测耗时较少的基于特征的人脸检测系统, 取名为 autofaceGFB。该系统对于给定的可能包含人脸的输入图像, 能够实现人脸检测的功能, 判断出其中是否有人脸, 若有, 则将所有的人脸从背景中标记出来。该系统还需要能够应对输入图像质量不佳及光照变化等情况, 具备一些图像预处理的功能, 以适应环境的复杂性, 处理图像中存在的噪声, 光照变化等。

本课题的研究内容主要可以归结为如下几点:

- 1) 利用 OpenCV 中的基本函数, 实现数字图像处理的基本功能, 包括图像的输入、显示、缩放、旋转、保存等等。开发人脸检测系统 GUI 软件界面。
- 2) 为适应环境的复杂性, 处理图像中存在的噪声, 光照变化等, 实现数字图像预处理的相关功能, 包括图像的亮度/对比度调节、平滑、腐蚀、膨胀、直方图均衡化、Gamma 校正、光照补偿、色彩平衡等等。
- 3) 熟悉肤色模型的创建和 Boosted Cascade 方法, 利用 OpenCV 中的人脸检测部分的函数, 实用 VC++ 搭建人脸检测部分的框架, 实现人脸检测的基本功能。
- 4) 研究人脸检测算法, 利用人脸特征, 包括肤色特征及人脸眼睛等器官特征, 改进人脸检测功能的实现, 降低误检率, 提高检测正确率。
- 5) 探讨光照变化环境下的人脸检测的特点, 并设计算法尝试更好地解决这个问题。

1.5 全文结构

第二章 人脸检测相关理论及 autofaceGFB 系统简介。本章主要介绍人脸检测相关理论及研究情况，将人脸检测的基本思想和方法做一介绍，给出通用的基于特征的人脸检测系统 autofaceGFB 的总体结构，并介绍一下系统的开发环境。

第三章 基于肤色特征的人脸检测。本章主要阐明人脸肤色模型的理论基础和建立过程，以及介绍光照变化环境下的人脸检测问题的解决方案，并给出了基于肤色特征的人脸检测的实现流程和实验结果分析。

第四章 基于 Boosted Cascade 算法的人脸检测。本章主要介绍 Boosted Cascade 算法及其改进方案的原理和实现，并对实验结果进行了分析和对比。

第五章 总结与展望。本章对全文内容进行总结，并对将来的工作进行展望。

第二章 人脸检测相关理论及 autofaceGFB 系统简介

2.1 人脸检测相关理论及研究情况

2.1.1 人脸检测的基本思想和方法简介

人脸检测问题可描述为：任意给定一幅图像，判断其中是否存在人脸，若存在，则将所有的人脸从背景中标记出来。人脸检测实际上是一个两类的人脸鉴别问题^[1,2]，即在一幅图像中只有“人脸”和“非人脸”之分。人脸检测的基本思想是基于知识或统计的方法对人脸建模，比较所有可能的待检测区域与人脸模型的匹配度，从而得到可能存在人脸的区域。其方法大致可分为以下几类：

1. 基于肤色模型的方法

在彩色图像中，颜色是人脸表面最为显著的特征之一，利用颜色检测人脸是很自然的想法。Yang 等在考察了不同种族，不同个体的肤色后认为人类的肤色能在颜色空间中聚成单独的一类，而影响肤色值变化的最主要因素是亮度变化。即人类的肤色在去除亮度的色度空间表现有聚类特性，这种聚类性可用来检测人脸。综合有关文献，利用肤色特征进行人脸检测主要有以下三种不同的形式：①将肤色判断作为人脸检测的核心方法。②将肤色判断作为人脸检测的预处理。肤色的判断具有明显的速度上的优势，大大提高人脸检测算法的整体性能。③将肤色判断作为人脸检测的后期验证。这对于提高人脸检测算法的正确率，降低误报率有很大的帮助。

2. 基于模板匹配的方法

五官的拓扑位置信息是最早被用来检测人脸的信息。基于拓扑结构的模板匹配法的中心思想是将人脸的五官位置比例关系做成模板，用此模板在所检图像中逐点匹配搜索，计算输入图像与模板的匹配值，然后确定一个阈值，用以判断该输入图像中是否包含人脸。匹配值大于所定阈值处认为存在人脸，并把它标记出来。基于模板匹配的方法实现简单，但计算复杂度较高，速度也较慢，而且不能适应光照、姿势、人脸大小的变化。

3. 弹性匹配法

这种方法主要基于小波特征。在图像的敏感位置，小波变换后生成的特征矢量的模较大。人脸的相似度可用拓扑图的“距离”来表示。用拓扑图分别代表已知和待识别人脸，根据匹配拓扑图算出它们的“距离”，作为人脸的相似度准则。由于小波特征分析是一种时频分析，所以此方法对于光线、尺寸和角度具有一定的不变性。但由于图像中大量与识别无关的信息没有过滤而使计算变得复杂是此方法的一大缺点。

4. 特征脸法

这种方法主要基于 KL 变换。从压缩能量的角度来看，KL 变换是最优的，

它不仅使得从 n 维空间降到 m 维空间前后的均方误差最小，而且变换后的低维空间有很好的脸表达能力。更重要的是 KL 变换去掉了向量的各个分量间的相关性，这样如果在三维特征空间中则可以把特征限制在一个立方体中，这对于分类来说就简单多了。但是特征脸识别和检测方法存在对偏差特别敏感的缺陷，研究表明，随着光线、角度及人脸尺寸等因素的导入，识别率急剧降低。

5. 人工神经网络法

人工神经网络（Artificial Neural Network, ANN）的方法是通过训练一个网络结构，把模式的统计特性隐含在神经网络的结构和参数之中。基于人工神经网络的方法对于复杂的，难以显示描述的模式，具有独特的优势。但是基于神经网络的人脸识别和检测方法计算量特别大，难于实现实时应用。

6. 支持向量机法

支持向量机（Support Vector Machines, SVM）是 Vapnik 等提出的基于结构风险最小化原理（Structural Risk Minimization Principle, SRM）的统计学习理论，用于分类与回归问题。SRM 使 VC（Vapnik Chervonenskis）维数的上限最小化，这使得 SVM 方法比基于经验风险最小化（Empirical Risk Minimization Principle, ERM）的人工神经网络方法具有更好的泛化能力。对于人工神经网络中一些难以逾越的问题，如：模型选择和过学习问题、非线性和维数灾难问题、局部极小点问题等都得到了很大程度上的解决。

7. 基于积分图像特征法

基于积分图像（Integral Image）特征的人脸检测方法是 Viola 等提出的一种算法，它综合使用积分图像描述方法、Adaboost 学习算法及训练方法、级联分类器。积分图像是指该像素以前、以上所有像素灰度和。级联分类器是多个分类器的组合，分类器的输入是待检测图像子窗口，输出集是用来表示该检测窗口是否人脸模式{T, F}。

8. 基于概率模型的方法

基于概率模型方法的一种思路是计算输入图像区域属于人脸模式的后验概率，据此对所有可能的图像窗口进行判别。Schneiderman 等将概率估计的方法用于检测正面旋转人脸和侧面人脸，同时使用多分辨率信息复用和由粗到精搜索的策略提高检测的速度。

2.1.2 人脸检测算法的评价

人脸检测系统的输出准确与否需要一个客观的标准进行评价。对各种人脸检测方法作性能上的对比，无疑将会促进人脸检测技术的发展。人脸检测算法的评价可通过如下两种概率得到^[2]：

1) 检测正确率（True Positive Rate, TPR）

检测正确率反映了人脸检测系统检测出真实人脸的能力，由下式定义：

$$\text{检测正确率} = \frac{\text{正确检测到的人脸数目}}{\text{测试集中包含的人脸数目}} \times 100\%$$

2) 错误报警率 (False Alarm Rate, FAR 或者 False Positive Rate)

错误报警率是指人脸检测系统将不是人脸的区域错误地认为是人脸,它反映了检测系统对于非人脸区域的拒识能力,与测试集中图像的尺寸、背景复杂程度等有关,由下式定义:

$$\text{错误报警率} = \frac{\text{错误报警数目}}{\text{处理的检测窗口数目}} \times 100\%$$

实际上,人脸检测器的正确检测能力与拒识能力是相互关联的两个方面,降低人脸检测器的敏感性阈值,可以使检测正确率提高,同时错误报警率也会增加,反之亦然。

2.2 autofaceGFB 人脸检测系统简介

2.2.1 系统简介和功能说明

本课题设计并实现的通用的基于特征的人脸检测系统取名为 autofaceGFB,其中 G 是指代 general-service (通用的),表明该系统属于“通用人脸检测范畴”,可检测任意背景下,所含人脸数目和大小任意的,无任何限制条件的图像;FB 是指代 feature-based (基于特征的),所谓基于特征,这里的“特征”的概念是很宽泛的,因为人脸的特征有很多,如人脸器官的特征,各器官之间的几何分布特征,图像像素的灰度特征,颜色特征,人脸轮廓特征,纹理特征,Haar 小波特征以及一些广义测度特征等等,都是系统可以利用起来进行人脸检测的特征。

autofaceGFB 人脸检测系统是一个用于对静态图像进行人脸检测的软件,也包括一些常用的图像处理功能,使用方便,功能较齐全。该系统为多文档应用软件,能够读写多种格式的静态图像,具体包括 Windows 位图文件 (BMP, DIB); JPEG 文件 (JPEG, JPG, JPE); 便携式网络图片 (PNG); 便携式图像格式 (PBM, PGM, PPM); Sun rasters (SR, RAS); TIFF 文件 (TIFF, TIF)。

autofaceGFB 人脸检测系统支持灰度图像和彩色图像的人脸检测,实现了基于肤色特征的人脸检测,基于 Boosted Cascade 算法的人脸检测,并且提出了对 Boosted Cascade 算法的几个改进方案,检测正确率较高,错误报警率较低,处理速度较快,对不同图片的适应性较强。同时为了查看人脸检测的结果,如检测耗时以及检测出来的人脸数目等数据,系统还设计实现了人脸检测输出参数信息对话框,将检测结果统计并反映到该对话框中,方便对系统进行测试时的数据统计。

autofaceGFB 人脸检测系统还加入了一些常用的图像处理功能,如图像放大

和缩小、旋转、镜像、查看图像灰度直方图、平滑、腐蚀、膨胀、直方图均衡化、亮度/对比度调节、Gamma 校正、光照补偿、色彩平衡、彩色图转换为灰度图等。另一方面，为了方便系统的操作，实现了对操作的撤销及重做功能，还将一些主要操作添加到右键菜单和工具栏中，方便用户使用。

2.2.2 系统的总体结构

由于输入图像的成像条件不同，光照变化、环境复杂程度也各异，为了增强对输入图像的适应性，autofaceGFB 人脸检测系统设计并实现了一些图像预处理功能，如图像放大和缩小、旋转、镜像、查看图像灰度直方图、平滑、腐蚀、膨胀、直方图均衡化、亮度/对比度调节、Gamma 校正、光照补偿、色彩平衡等功能。该系统设计并实现的第一个人脸检测方案是基于肤色特征的人脸检测，利用人脸颜色特征，使用肤色模型检测人脸，并结合人脸内部细节（如眼睛和嘴等器官）特征来进行验证。这一人脸检测算法具有很强的适应能力，在人脸姿态、表情变化以及面部部分遮挡等复杂情况下依然能够获得很高的检测正确率。2001 年美国的 Viola 和 Jones 提出了用易于计算的 Haar 小波特征来描述人脸模式，用 Boosted Cascade 算法来训练人脸检测分类器，取得了人脸检测领域的突破性进展，实现了实时检测视频中的人脸图像，而且准确率也非常高。autofaceGFB 人脸检测系统也使用了这种算法作为系统的另一个人脸检测方案。为了排除某些复杂环境下被 Boosted Cascade 算法误检的人脸区域，降低系统的错误报警率，还提出了三个改进方案，分别为：基于肤色验证的 Boosted Cascade 人脸检测方案、基于器官验证的 Boosted Cascade 人脸检测方案和正面与侧面检测相结合的人脸检测方案。autofaceGFB 人脸检测系统通过综合实现上述各方案，对光照、尺度、姿态、旋转、表情和部分遮挡等方面的变化具有较强的适应性和较高的检测率。

综上，本文设计的 autofaceGFB 人脸检测系统的总体结构用图 2-1 表示如下：

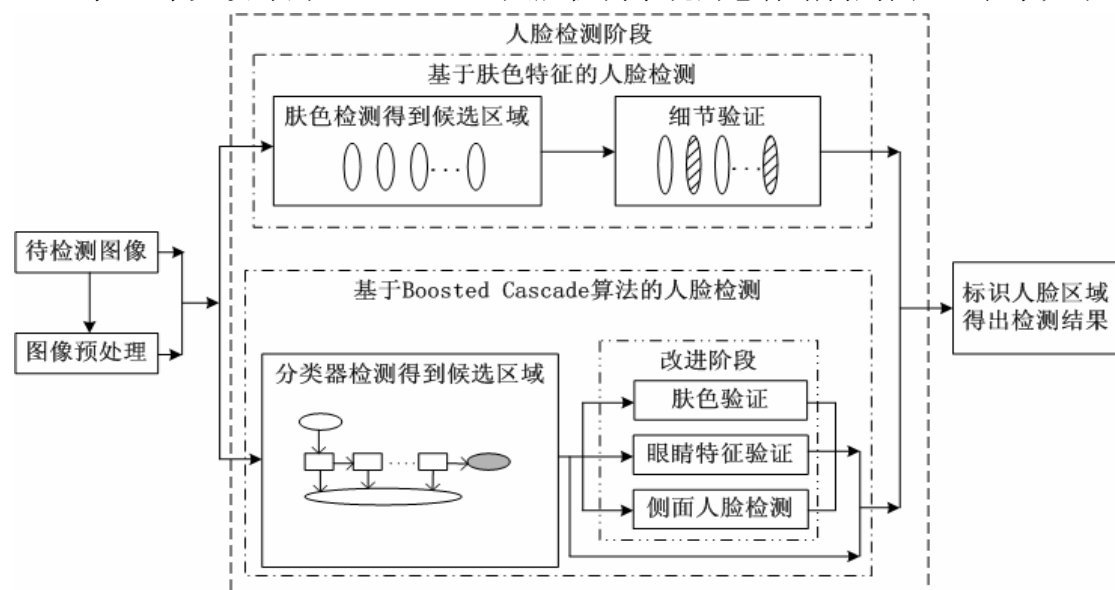


图 2-1 autofaceGFB 人脸检测系统的总体结构图

2.3 autofaceGFB 人脸检测系统开发环境说明

autofaceGFB 人脸检测系统使用 Microsoft Visual C++ 6.0 作为开发平台，并使用了 Intel 公司开发的 OpenCV 1.0（Open Source Computer Vision Library，开放源代码的计算机视觉类库）作为系统的开发工具。OpenCV 是由 Intel 公司位于俄罗斯的研究实验室所开发，它是一套可免费获得的由一些 C 函数和 C++ 类所组成的库，用来实现一些常用的图像处理及计算机视觉算法^[3]。由于 OpenCV 的源代码是完全开放的，对非商业应用和商业应用都是免费的，而且源代码的编写简洁高效，特别是其中大部分的函数都已经过汇编最优化，以使之能高效而充分地利用英特尔系列处理芯片的设计体系，对于 Pentium 系列的处理器而言，OpenCV 的代码执行效率是非常高的，所以近年来在国内外的图像处理相关领域中被广泛地使用，作为第二次开发的理想工具，成为一种流行的图像处理软件。OpenCV 集成了一些比较常见的用于图像处理及计算机视觉的算法，用 C/C++ 语言进行实现，在实际应用中，可以减少很多重复劳动，达到更高的效率。

第三章 基于肤色特征的人脸检测

3.1 概述

对彩色人脸图像而言，肤色是人脸的重要信息，不依赖于面部的细节特征，对于人脸的旋转、面部表情、面部被部分遮挡（如墨镜、围巾）等等复杂变化情况都能保持不变，有相对稳定性，况且大多数情况下会和背景物体的颜色相区别，因此肤色特征在人脸检测中是最有效的特征之一。同时，基于肤色的人脸检测速度很快，适于实时使用。本课题开发的 autofaceGFB 人脸检测系统就利用了人脸的肤色特征来实现人脸检测功能。另一方面，肤色检测的错误报警率较高，如手，四肢等也会被认为是人脸区域。这里采取的应对措施是，对用肤色模型检测出来的候选区域进行细节验证，排除人体四肢及颜色近似肤色的区域等非人脸区域，来降低错误报警率。具体的细节验证方法采用 Sobel 方差验证，因为人脸区域包含五官，用 Sobel 方差处理后五官与皮肤交界处因像素灰度差较大而明显的显现出来，但人体四肢因不具备这样的条件就可以被排除掉。

3.2 肤色模型理论

所谓肤色模型，也和其他的数学建模一样，即用一种代数的（解析的）或查找表等形式来表达哪些像素的色彩属于肤色，或者表征出来某一像素的色彩与肤色的相似程度。这种模型主要通过色彩格式的变换，使得肤色在某一色彩或变形的色彩空间中呈良好的聚类特性，然后将这一聚类在色彩空间中的分布区域用一种简便的代数解析式加以表达^[4]。

肤色信息的使用可以有效简化人脸定位，但是它容易受到复杂背景和亮度变化的影响^[5]。为了克服光照对肤色检测的影响，采用能将色彩中的亮度分量分离出来的 YCbCr 色彩空间，这样在建模时可以不考虑亮度分量，而仅考虑色彩分量，希望借此来克服光照的影响。YCbCr 色彩格式的计算过程和空间坐标表示形式比较简单，而且实验结果表明在 YCbCr 色彩空间中肤色的聚类特性比较好。本模块就采用了 Anil K. Jain 等提出的基于变形 YCbCr 色彩空间的肤色模型^[6,7]，它利用了非线性分段色彩变换（Nonlinear Color Transformation）这个关键技术，得到的肤色模型有效地降低了亮度对肤色聚类区域的影响。

YCbCr 色彩格式是直接由 RGB 色彩格式通过线性变换得到的，其亮度分量 Y 并不是完全独立于色度信息而存在的，所以肤色的聚类区域也是随 Y 的不同而呈非线性变化的趋势。Anil K. Jain 等人从 Heinrich-Hertz-Institute (HHI) 图像库中的 137 幅图像中手工选取了 853571 个肤色像素点，并将其绘制在 YCbCr 空间及其二维投影子空间 Cb-Cr 空间中，得到的结果如图 3-1 所示^[6]。

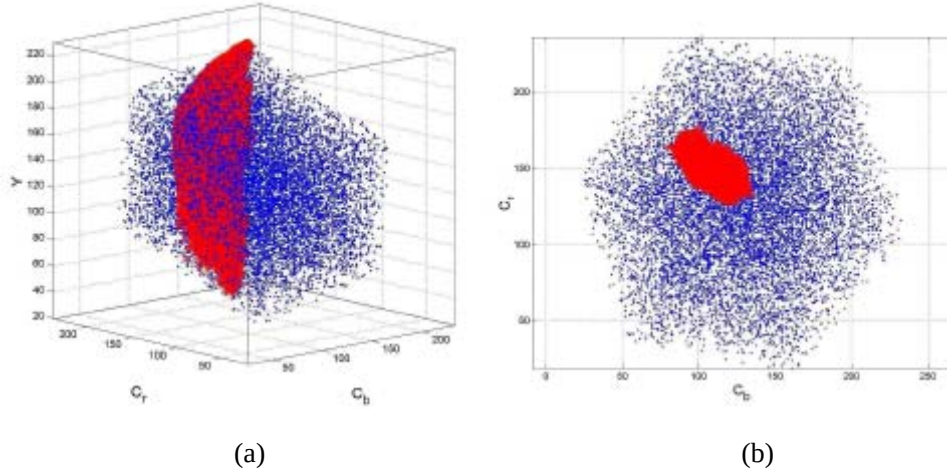


图 3-1 (a)为肤色点于 YCbCr 空间, (b)为肤色点在 Cb-Cr 空间投影

从图中可以看到, 在 YCbCr 色彩空间中, 肤色聚类是呈两头尖的纺锤形状, 即在 Y 值较大和较小的部分, 肤色聚类区域是随之缩减的。如果在 Y 值不同的地方, 我们取 Cb-Cr 子平面的投影, 那么得到的结果是不同的。从中可以得出结论, 简单地排除 Y 分类, 按照传统的做法在二维的 Cb-Cr 子平面中寻求肤色的聚类区域是比较粗糙的, 不完善的, 还必须考虑 Y 值不同造成的影响, 从而对 YCbCr 色彩格式进行非线性分段色彩变换。图 3-2(a)和(b)反映了我们的肤色聚类的边界上, Cb 和 Cr 两分量随 Y 变化的情况。用这 4 个边界来限制肤色聚类区域可以很好地适应亮度过明或过暗的区域, 从而使肤色模型的鲁棒性大大提高。

经过了非线性分段色彩变换得到的色彩空间我们用 YCb'Cr'来表示。YCbCr 坐标空间到 YCb'Cr'坐标空间的变换过程推导如下^[6]:

(1)图 3-2(a)和(b)中, 肤色区域的中轴线分别用 $\overline{C_b}(Y)$ 和 $\overline{C_r}(Y)$ 来表示, 可以得到 $\overline{C_b}(Y)$ 和 $\overline{C_r}(Y)$ 的表达式为:

$$\begin{aligned} \overline{C_b}(Y) &= \begin{cases} 108 + \frac{(K_i - Y) \cdot (118 - 108)}{K_i - Y_{\min}}, & \text{if } (Y < K_i) \\ 108 + \frac{(Y - K_h) \cdot (118 - 108)}{Y_{\max} - K_h}, & \text{if } (K_h < Y) \end{cases} \\ \overline{C_r}(Y) &= \begin{cases} 154 - \frac{(K_i - Y) \cdot (154 - 144)}{K_i - Y_{\min}}, & \text{if } (Y < K_i) \\ 154 - \frac{(Y - K_h) \cdot (154 - 132)}{Y_{\max} - K_h}, & \text{if } (K_h < Y) \end{cases} \end{aligned} \quad (3-1)$$

其中 K_i 和 K_h 为非线性分段色彩变换的分段阈值, 分别为: $K_i=125$, $K_h=188$ 。 $Y_{\min}$ 和 $Y_{\max}$ 为根据实验数据得到的肤色聚类区域中 Y 分量的最小值和最大值, 分别为: $Y_{\min}=16$, $Y_{\max}=235$ 。

(2)同样，图 3-2(a)和(b)中，肤色区域的宽度分别用 $W_{C_b}(Y)$ 和 $W_{C_r}(Y)$ 来表示，表达式为：

$$W_{C_i}(Y) = \begin{cases} WLC_i + \frac{(Y - Y_{\min}) \cdot (WC_i - WLC_i)}{K_i - Y_{\min}}, & \text{if } (Y < K_i) \\ WHC_i + \frac{(Y_{\max} - Y) \cdot (WC_i - WHC_i)}{Y_{\max} - K_h}, & \text{if } (K_h < Y) \end{cases} \quad (3-2)$$

其中 i 代表 b 或 r, $W_{C_b} = 46.97, WLC_b = 23, WHC_b = 14, W_{C_r} = 38.76, WLC_r = 20, WHC_r = 10$

(3)最后，根据上面的结果得到如下的非线性分段色彩变换公式：

$$C'_i(Y) = \begin{cases} (C_i(Y) - \overline{C_i}(Y)) \cdot \frac{W_{C_i}}{W_{C_i}(Y)} + \overline{C_i}(K_h), & \text{if } (Y < K_i) \text{ or } (K_h < Y) \\ C_i(Y), & \text{if } (Y \in [K_i, K_h]) \end{cases} \quad (3-3)$$

经过这样的非线性分段色彩变换，我们的肤色聚类在 $YC_b'Cr'$ 空间中的分布情况如图 3-2(c)所示。再将其投影到 $Cb'-Cr'$ 二维子空间，可得到实用的肤色聚类模型，投影图如图 3-2(d)所示。

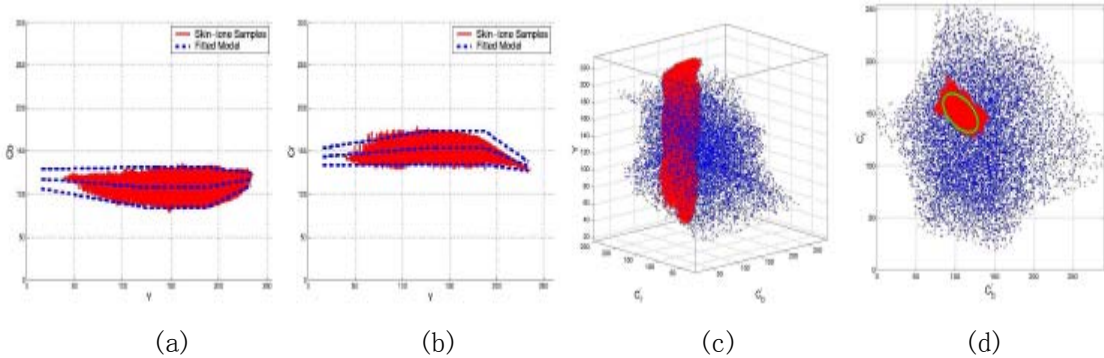


图 3-2 (a)为 C_b 随 Y 的变化，(b)为 C_r 随 Y 的变化，(c)为变换后的肤色聚类分布，(d)为投影后的肤色聚类模型

按照传统的方法，可用一个椭圆来近似这一肤色区域，如图 3-2(d)中绿色的椭圆。我们得到它的解析表达式为：

$$\frac{(x - ecx)^2}{a^2} + \frac{(y - ecy)^2}{b^2} = 1, \quad \text{其中, } \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} C'_b - cx \\ C'_r - cy \end{bmatrix}$$

解析式中的常量分别为： $cx=114.38$, $cy=160.02$, $\theta=2.53$ (弧度), $ecx=1.60$, $ecy=2.41$, $a=28.39$, $b=16.03$ 。这里的常量是我经过实验处理，稍作修改的，与 Anil K. Jain^[6]等提出的肤色模型中给出的常数数值上有点差异。

3.3 光照变化下的人脸检测策略

对基于肤色特征的人脸检测来说，图像中不同亮度和颜色的光照是影响人脸

检测的重要因素。为了能够有效地克服由于光照的亮度变化及有色光的照射给检测造成的困难，autofaceGFB 人脸检测系统除了使用前面提到的非线性分段色彩变换技术外，还使用了 Anil K. Jain 等提出的光照补偿技术^[6]，以及 Shumate 等提出的用于消除有色光干扰的色彩平衡技术^[8]，通过使用这些技术，来使基于肤色模型的人脸检测适应光照变化的环境。另外，出于显示器的物理特性和人的感知特性，有时需要对图像进行 Gamma 校正^[9]。由于上一小节已经对非线性分段色彩变换技术作了详细介绍，这里主要介绍下 Gamma 校正、光照补偿和色彩平衡技术。

(1) Gamma 校正

Gamma 校正可以有效弱化光照对图像质量的影响^[9]。其公式如下：
$$\text{new_pixel_value} = \text{old_pixel_value}^{1/2.22}$$
。如图 3-3 所示，(a)为原图像，(b)为经过 Gamma 校正后的图像，图像细节更加清晰，光照条件得到改善，图像质量有所提高。(c)和(d)分别为 Gamma 校正前后得到的肤色区域，可见，Gamma 校正后图像原本较暗的左半边脸也可以被检测出来。

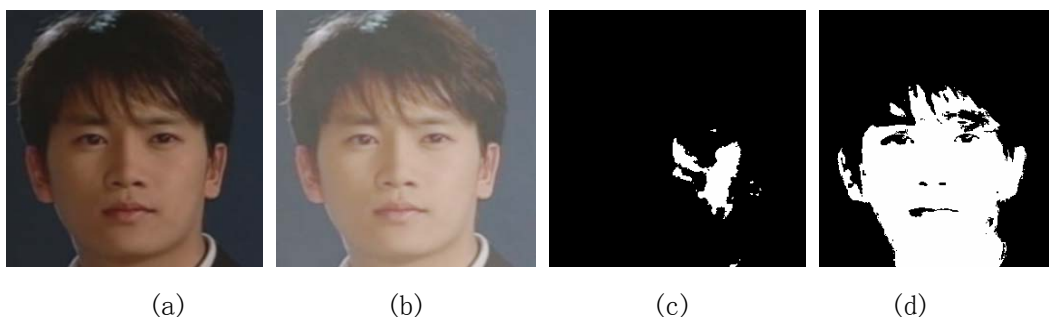


图 3-3 Gamma 校正的作用

(2) 光照补偿

光线补偿的想法的提出主要是考虑到肤色等色彩信息经常受到光源颜色、图像采集设备的色彩偏差等因素的影响，而在整体上偏离本质色彩而向某一方向移动，即通常所说的色彩偏冷、偏暖，照片偏黄、偏蓝等等。这种现象在艺术照片中更为常见。所以 Anil K. Jain 等提出，为了抵消这种整个图像中存在着的色彩偏差，将整个图像中所有像素的亮度从高到低进行排列，取前 5%的像素，如果这些像素的数目足够多（如大于 100），我们就将它们的亮度作为“参考白”，即将它们色彩的 R、G、B 分量值都调整为最大的 255。整幅图像的其他像素点的色彩值也都按这一调整尺度进行变换^[6]。

可从两个角度考察这一做法的合理性^[6]：一方面，绝大部分的图像中都包含有纯白色，特别是包含人脸在内的图像中，眼球外围等处通常就是纯白色，所以将具有最大亮度像素的色彩值调整为纯白色是合理的；另一方面，存在色彩偏差的图像在原来是白色的区域有着很明显、直观的体现，所以按照这些区域的调整

方法对整个图像进行调整也是比较合理和有效的。

图 3-4 的例子^[7]表现了光线补偿的作用，其中(a)是原图像，显然发生了偏黄现象，(b)是经过光线补偿后得到的图像，而(c)和(d)分别是对(a)和(b)进行肤色分割得到的结果，从中可以很明显地看出光线补偿的作用。

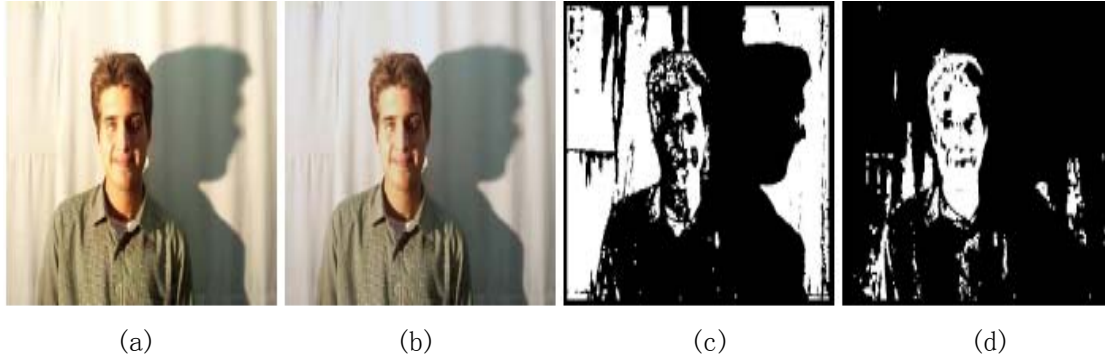


图 3-4 光线补偿的作用

(3) 色彩平衡

在利用肤色模型检测肤色像素之前，需要消除可能的有色光干扰。本系统采用色彩平衡的方法来消除有色光的干扰，在这之前应先判断是否有有色光干扰，否则色彩平衡的结果可能会使图像中人脸的颜色发生较大改变，反而影响肤色模型的检测效果。本系统的色彩平衡引用了 Shumate 等的方法^[8]，其中心思想是“灰度世界假设（Gray World Suppose）”，即认为对一幅存在大量色彩变化的图像，RGB 三分量的均值趋近于同一值。在现实世界中，一般来说图像中的物体和背景的颜色变化是随机的，相互独立，因此这个假设是合理的。

对于计算机中的 RGB 图像，具体的处理方法描述如下：

- a) 计算图像中的 R, G, B 分量各自的平均值 $averb$, $averg$, $averr$ ，令它们的平均值为趋近值，即 $GrayWorldValue = (averb + averg + averr) / 3$ 。
- b) 对图像中的每一个像素，调整三分量值分别为：

$$b = (b * GrayWorldValue) / averb;$$

$$g = (g * GrayWorldValue) / averg;$$

$$r = (r * GrayWorldValue) / averr;$$
- c) 归一化所有像素的各个分量的值，让它们都在 [0,255] 范围内，令 $factor$ 为调整后像素分量的最大值，若 $factor$ 大于 255，则调整各分量为：

$$b = (b * 255) / factor;$$

$$g = (g * 255) / factor;$$

$$r = (r * 255) / factor;$$

这里给出一个例子说明色彩平衡的效果。图 3-5 中(a)是原图像，显然蓝色的有色光造成了皮肤的色彩偏移，皮肤模型(c)只能得到零星的皮肤区域；(b)是经

过色彩平衡后得到的图像，(d)是(b)进行肤色分割得到的结果，从中可以很明显地看出色彩平衡的作用。



图 3-5 色彩平衡的作用

3.4 基于肤色特征的人脸检测的实现

在基于肤色特征的人脸检测算法的具体实现中，对肤色进行建模得到肤色区域后，为了去掉人体四肢及其他近似肤色的非人脸区域，需要对候选肤色区域进行验证。Rein-Lien Hsu 使用嘴唇和眼睛的 Cr 和 Cb 规范值来辨别人脸，但由于很多人戴眼镜，并且有些女士涂不同颜色的唇彩而使得该方法的准确性不高^[10]。故本模块没有借鉴这种方法来判断候选区域是否包含眼睛、嘴等器官，而是使用 Sobel 差分方法对候选区域进行验证。因为人脸内包含眼睛，眉毛，嘴等器官，用 Sobel 方差处理后，五官与皮肤交界处因像素灰度差较大而明显的显现出来，但人体四肢等非人脸的肤色区域就不具备这样的条件，利用这一点可以被排除掉。这里对原输入图像候选区域的灰度图进行 y-方向的图像差分处理，然后对差分结果图进行二值化处理。再在二值化处理后的差分结果图中寻找轮廓，将像素数目过于小的部分忽略掉。因为每一个人脸器官特征与人脸区域有一个最大比例，因此当一个人脸候选区域上存在大于该区域 20%的“器官”时，将其剔除。另外，如果一个区域上面部特征的数量过多或过少，也将其视为非人脸区域而剔除。

3.4.1 算法伪代码

基于肤色特征的人脸检测算法的伪代码如下：

1. 输入待检测图像。
2. 如有需要进行图像预处理，如 Gamma 校正、光照补偿和色彩平衡等，以改善光照条件，利于后面的肤色检测。
3. 进行色彩空间转换，由 RGB 色彩格式通过线性变换得到 YCbCr 色彩格式。转换公式如下：

$$Y = (257 * r + 504 * g + 98 * b) / 1000 + 16;$$

$$Cr = (439 * r - 368 * g - 71 * b) / 1000 + 128;$$

$$Cb=(-148*r-291*g+439*b)/1000+128;$$

4. 对 YCbCr 空间进行非线性转换，用来消除色度对亮度的依赖关系。
5. 皮肤颜色建模。这里用到的主要是 YCbCr 空间中的色度信息，建模的形式是椭圆公式匹配两个色度分量的距离。它的解析表达式为：

$$\frac{(x - ecx)^2}{a^2} + \frac{(y - ecy)^2}{b^2} = 1$$

6. 膨胀处理，连接脸部区域中不连续的块。
7. 腐蚀处理，使非人脸区域变小，恢复到膨胀前的大小。
8. 中值滤波去除孤立噪声点，以免影响到后面的操作。
9. 再次膨胀和腐蚀，主要是为了得到人脸区域的像素点，方便后面定位人脸区域。
10. 大致定位人脸区域，在二值肤色图像中寻找轮廓，用下面的循环遍历每个轮廓。

for i=1,...,轮廓数目 n

1) 计算第 i 个轮廓的面积;

2) if (轮廓面积>=600)

认为是人脸候选区域，计算轮廓的最外面矩形边界;

3) 使用 Sobel 差分方法对候选区域进行验证。

a) 对原输入图像候选区域的灰度图进行 y-方向的图像差分处理;

b) 对差分结果图进行二值化处理

if (灰度值>=70)

像素置为白色;

else

像素置为黑色;

c) 在二值化处理后的差分结果图中寻找轮廓

for j=1,...,轮廓数目 m

if (像素数目<33)

此部分面积过于小，将其忽略掉;

else if (像素数目>该候选区域面积的 20%)

该特征面积过大，将其剔除;

4) if (Sobel 验证得到的符合条件的面部特征数<2)

面部特征的数量过少，将其视为非人脸区域而剔除;

else if (Sobel 验证得到的符合条件的面部特征数>15)

面部特征的数量过多，也将其视为非人脸区域而剔除;

else

满足验证条件，用红色矩形框在原输入图像中标识出来。

3.4.2 算法实现流程图

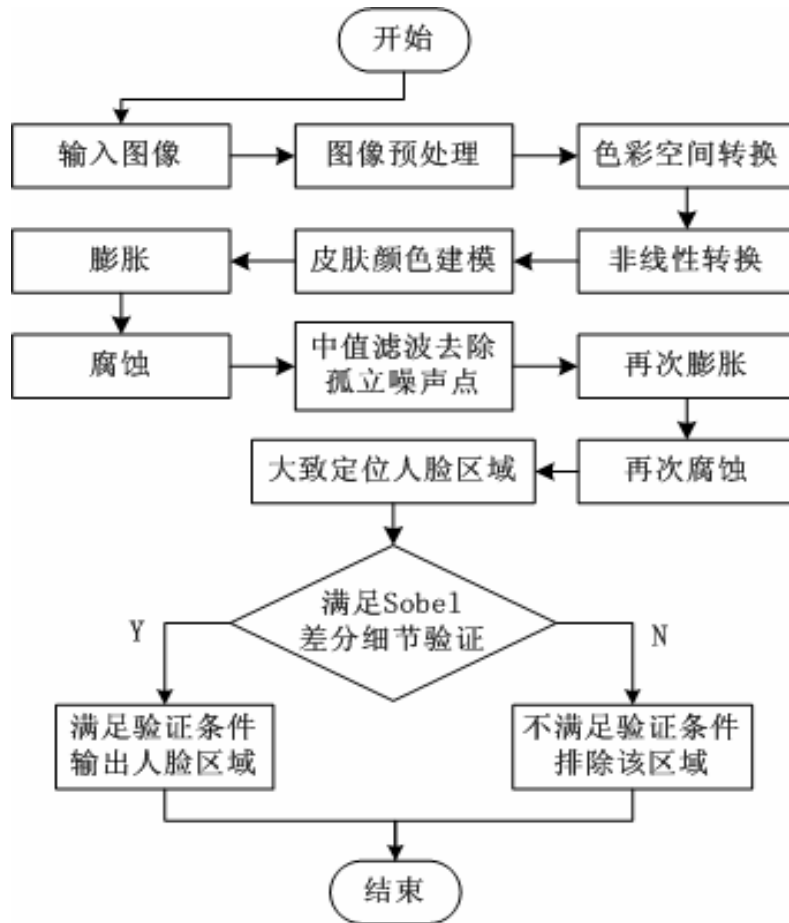


图 3-6 基于肤色特征的人脸检测算法流程图

3.5 测试及评价

测试的人脸图像由数码相机采集和互联网下载，具有不同的大小，不同的复杂背景，光照有亮有暗，部分有有色光照射，人脸尺寸有大有小，既有单人脸也有多人脸，既有正面人脸也有侧面人脸，有白种、黄种和黑种人，除了人脸，图像中还出现人体四肢等皮肤区域。测试图像共有 100 幅，其中包括 618 个人脸，正确检测出 557 个人脸，检测正确率达到 90.13%，共出现 142 个错误报警，经过 Sobel 验证后，错误报警数减少到 33 个。每幅图像的检测速度随图像大小而变化，图像越小，检测耗时越少。如果图像的大小在 500×500 像素左右，则检测时间小于 0.8 秒，如果图像的大小在 1000×1000 像素左右，则检测时间在 1 秒至 1.5 秒左右。可以看出，基于肤色特征的算法检测人脸速度很快，能够达到实时的要求。图 3-7 给出了一些检测结果。从图中可以看出，该算法可以正确检测不同光照条件下，人脸图像大小各不相同，肤色各异的人脸图像，且能够检测任意姿态下，任意表情的人脸，检测效果非常好。在测试中发现，在亮度极低和

极高的极端情况下，会发生漏检，因为此时的肤色区域已经不在肤色模型限制范围内了。而当四肢出现交叠时，特别是多人的合照中两人的四肢交叠在一起时，容易发生误检。因为此时也比较容易满足 Sobel 验证，很难排除这些区域。



图 3-7 检测结果示例

为了说明程序中 Sobel 验证的作用，下面给出了几幅进行 Sobel 验证前后的对比图。图 3-8(a)中基于肤色特征的人脸检测算法检测后有一个被识别为人脸的手区域，而在应用 Sobel 验证后重新进行检测得到的检测结果中，如图 3-8(b)所示，手被排除掉了。同样，图 3-8(c)中有三个四肢区域被误检，应用 Sobel 验证后，图 3-8(d)所示，误检区域被排除。



图 3-8 Sobel 验证前后对比图

第四章 基于 Boosted Cascade 算法的人脸检测

4.1 概述

本课题开发的 autofaceGFB 人脸检测系统还采用了 Boosted Cascade 算法作为人脸检测的另一个基本方法，该算法检测正确率较高且具有极高的运行速度，是目前现有的人脸检测算法中效果最好的算法之一，且对于光照变化环境下的人脸，该算法也具有较好的适应性。但该算法具有误检率高的缺点，且检测结果直接依赖于所使用的分类器，当使用正面人脸分类器时，对侧面人脸检测效果不佳。本模块针对该方法检测人脸的不足之处，分别有针对性的提出了改进方案。其一，为了降低误检率，分别进行肤色验证和眼睛器官验证，能有效去掉一些误检区域，从而最大限度的降低系统的错误报警率，而又尽量不影响正确率。其二，由于单纯使用正面人脸分类器的 Boosted Cascade 方法对侧面人脸的检测不是很有效，而同时用正例和反例样本训练出来的分类器过于复杂且检测效果并不好，故在正面人脸检测的基础上，另外加载用侧面人脸样本训练过的分类器去补检被漏检的一些侧面人脸，采用正面与侧面人脸检测相结合的方法，大大提高了检测正确率，而且检测速度也是可以接受的。基于 Boosted Cascade 算法的人脸检测及其改进方案的总体结构用图 4-1 表示如下：

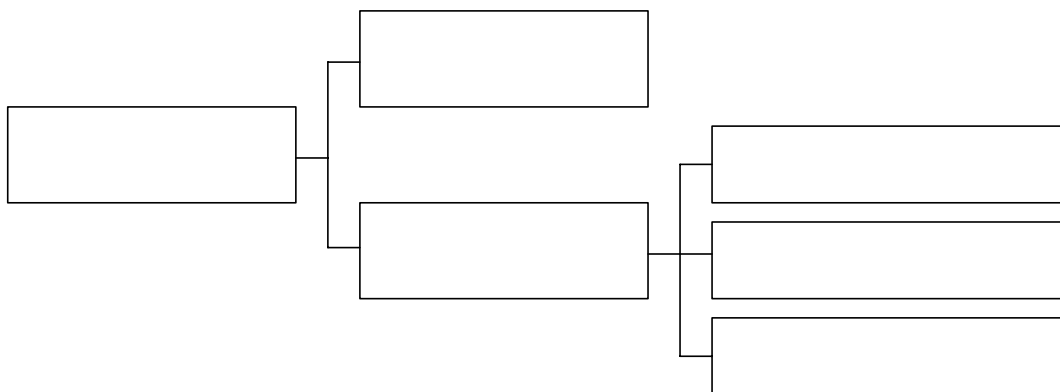


图 4-1 基于 Boosted Cascade 算法的人脸检测的总体结构

4.2 Boosted Cascade 算法原理

Boosted Cascade 算法最初由 Paul Viola 提出，并由 Rainer Lienhart 对这一算法进行了改善^[11]。该算法的主要思想^[11,12]是利用样本（成百上千幅样本图片）的 Haar 特征进行分类器训练，得到一个级联的 boosted 分类器，能将候选子窗口的人脸和非人脸区分开来。分类器中的“级联”是指最终的分类器是由多级分类器级联组成，其中每一级分类器都是一个“强分类器”，是若干个被称之为“弱分类器”的集成，而每一个弱分类器对应一个“弱特征”（Haar 特征），该特征可以在一定程度上区分人脸和非人脸。分类器训练完以后，就可以应用于输入图像

中感兴趣区域的检测。检测到人脸区域分类器输出为 1，否则输出为 0。该算法中包括两个关键的知识点：①Haar 特征的选取和计算。这些特征必须足够简单且容易计算，才能达到运算速度的要求。②多层分类器级联结构的建立。这涉及到如何从诸多的 Haar 特征中挑选出最有效的特征组合来训练分类器，使得它足以区分人脸和非人脸。下面对这两点分别作详细说明。

4.2.1 Haar 特征的选取及特征值的计算

目前的算法主要利用下面的 Haar 特征^[11]，如图 4-2 所示。

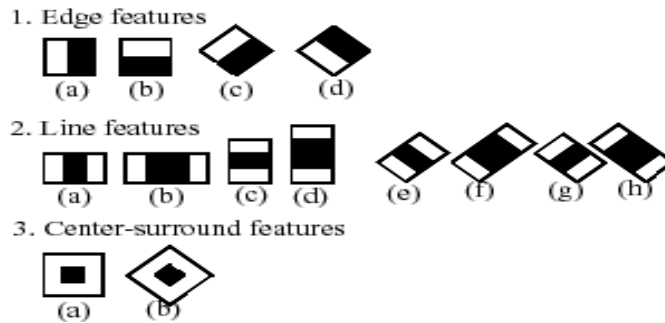


图 4-2 Haar 特征：1. 边界特征（Edge features）；2. 线特征（Line features）；3. 中心特征（Center-surround features）

每个特定分类器所使用的 Haar 特征可以用形状、感兴趣区域中的位置以及比例系数来定义。Haar 特征值是指图像上两个或者多个形状大小相同的矩形内部所有像素灰度值之和的差值。例如在第二行特征（2c）的情况下，响应计算为覆盖全部特征整个矩形框（包括两个白色矩形框和一个黑色矩形框）像素的和减去黑色矩形框内像素和的三倍。每个矩形框内的像素和都可以通过积分图很快计算出来。对于一个输入图像，像素处的积分图像值定义为^[12]：

$$ii(x, y) = \sum_{x_1 \leq x, y_1 \leq y} i(x_1, y_1) \quad (4-1)$$

如图 4-3 所示，即图中用黑色填充部分的所有像素灰度的和。有了积分图，上述矩形特征就可以通过很少的计算量得到。任意一个矩形内的像素和可以由积分图上对应的四点得到，如图 4-4 所示。两个矩形内像素和的差可以由积分图上的八点得到。由于上述矩形特征中的矩形都是相邻的，所以两个矩形的特征只需要计算积分图上对应的六点，三个矩形的特征需要计算八点，四个矩形的特征需要计算九点。

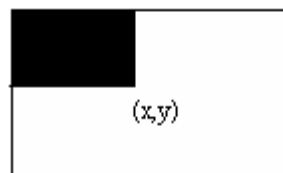


图 4-3 点 (x, y) 处的积分图像值

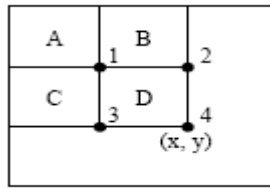


图 4-4 图中矩形 D 内的像素和可由积分图上的四点得到。积分图上 1 是矩形 A 内的像素值的和，2 是矩形 A+B 内的像素值的和，3 是矩形 A+C 内的像素值的和，4 是矩形 A+B+C+D 内的像素值的和，所以 D 内像素和为 $4+1-2-3$

为了得到一个输入图像 I 的积分图像，只需要逐点扫描原图像一次，就可以计算出来。记 $s(x, y) = \sum_{y_1 \leq y} i(x, y_1)$ 是原图像 (x, y) 点所在列纵坐标不超过该点的所有像素灰度的和，则有下面的递推公式^[12]：

$$\begin{aligned} s(x, y) &= s(x, y-1) + i(x, y), \\ ii(x, y) &= ii(x-1, y) + s(x, y). \end{aligned} \quad (4-2)$$

进行多尺度检测时，采用缩放特征模板的方法，即使在任意尺度搜索，也可以使用这一个积分图像。也就是说，整个检测过程，只需要扫描原图一遍就可以了。可见，Haar 特征确实是简单且容易计算。

Boosted Cascade 算法将大量的 Haar 矩形特征作为人脸的特征集，丰富的矩形特征充分体现了人脸模式的特点，并且每个矩形特征对应一个弱分类器，使得算法有丰富的弱分类器可供选择。

图 4-5 显示了由 Boosted Cascade 算法选择的两个 Haar 特征的例子^[12]。其中第 1 行为 Haar 特征弱分类器示意，第 2 行第 1 列为一幅被测图，后两列分别为将弱分类器放在典型的目標区域的情况。从实验结果来看，检测器显然分别利用了眼部区域亮度低于脸颊和双眼亮度低于眉心的特点确定了人脸的候选区域。



图 4-5 由 Boosted Cascade 方法学习出的两个 Haar 特征的例子

4.2.2 Boosted Cascade 多层分类器级联结构

分类器进行学习时，使用两个训练集对分类器进行训练，即正例样本集和反例样本集，其中正例样本是指人脸样本，反例样本指其它任意图片，所有的样本图片都被归一化为同样的尺寸大小（例如， 20×20 ）。而在一个 20×20 像素大小的小窗口内，就有数十万个不同大小及其在小窗口所在位置不同的 Haar 矩形特

征，这些特征并不是都对人脸检测有效，因此从特征选择的观点来看，需要从这么多的特征中挑选出最有效的特征组合，并用挑选出的这些特征构成一个分类器。Boosted Cascade 算法可以实现上述功能。

每一层强分类器都是若干 Haar 弱特征的带权值的线性组合，为了达到预定的性能，即预定的检测正确率和错误报警率，找到有效的弱特征是十分重要的。每个弱分类器对应一个弱特征，被设计为从能对正例和反例样本进行正确分类的所有弱分类器的集合中选择错误率最小的一个。一个子窗口 x 的弱分类器 $h_j(x)$ 包括如下部分^[13]：该子窗口的矩形特征值 $f_j(x)$ ，阈值 θ_j 以及指示不等式方向的符号因子 p_j ，如下式所示：

$$h_j(x) = \begin{cases} 1 & , \quad p_j f_j(x) < p_j \theta_j \\ 0 & , \quad \text{其他} \end{cases} \quad (4-3)$$

分类器学习的算法基本框架如下^[13]：

1. 给定一组包含人脸和非人脸的训练图像。
2. 对每一幅训练图像初始化权值为 ω_1 。
3. For $t=1, \dots, T$
 - 1) 归一化权值 ω_t ；
 - 2) 对每一个特征 j ，训练一个分类器 h_j ，且该分类器只对应一个特征；
 - 3) 选取使错误率达到最小的分类器 h_t ；
 - 4) 更新权值 ω_{t+1} 。
4. 得到最终的强分类器 $h(x)$ 。

若干个弱分类器线性组合成一个强分类器，再由很多强分类器级联组成最终的分器，多层分类器级联结构如图 4-6 所示。

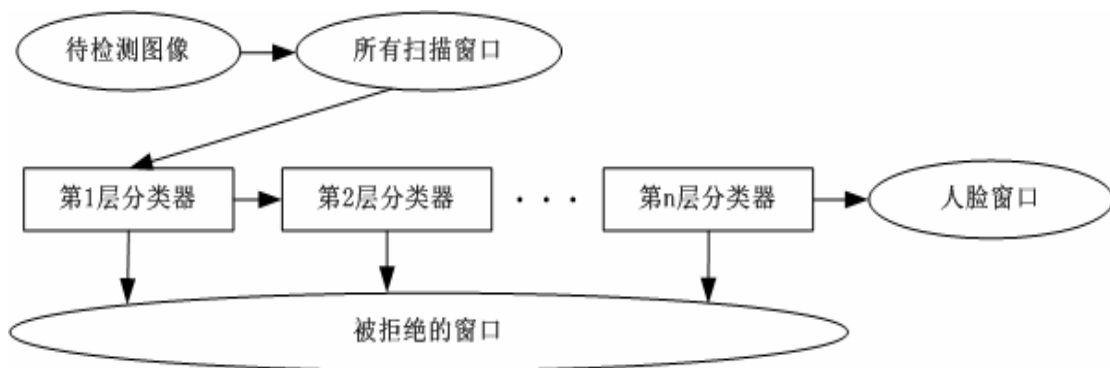


图 4-6 多层分类器级联结构

它是一种由粗到细的结构，其中每一层都可以选取一个 boosting 算法（权重投票），并利用基础分类器的自我训练得到，这里的基础分类器（弱分类器）是至少有两个叶结点的决策树分类器。这样，每一层作为一个强分类器，都经过阈值调整，能让全部正例样本通过，而拒绝很大一部分非人脸样本。用于训练后一

层分类器的反例样本集是从当前分类器对非人脸图像集进行检测后判断为正例的但事实上没有人脸的那些图像中获得的，显然，分类器的层数越高，对于该分类器的反例就越接近人脸，这时就对该层分类器加入更多的弱特征来排除反例。

整体的检测正确率和错误报警率可以通过确定多层分类器的各个阶段的检测正确率和错误报警率来达到。而每一层的检测正确率和错误报警率在该层分类器的学习过程中达到，如果该层分类器不能够排除足够的反例，则增加弱特征的个数并重新进行分类器的学习过程直到达到要求为止。

在图像检测中，被检窗口依次通过每一级分类器，这样在前面几层的检测中大部分的候选区域就被排除了，全部通过每一级分类器检测的区域即为目标区域。由于靠前的层拒绝了大部分的非人脸样本，而且，前面的层使用的矩形特征数很少，所以计算起来非常快。越往后，通过的候选窗口越少，尽管由于矩形特征增多，单个窗口的计算时间加长，但由于实际检测时，在输入图像中对应着人脸的窗口非常少，所以真正引起所有层都要计算的窗口数非常少，与以往的人脸检测算法相比，Boosted Cascade 算法具有很高的运算速度。为了检测整幅图像，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。为了搜索不同大小的目标物体，分类器被设计为可以进行尺寸改变，这样比改变待检图像的尺寸大小更为有效。所以，为了在图像中检测未知大小的目标物体，扫描程序通常需要用不同比例大小的搜索窗口对图片进行几次扫描。

4.3 基于 Boosted Cascade 算法的人脸检测的实现

使用 Boosted Cascade 算法进行人脸检测，所使用的分类器的性能直接影响着检测结果的好坏。而在 OpenCV1.0 版本中提供了具体的基于 Haar 特征进行分类器训练的程序 HaarTraining。该程序源码由 OpenCV 自带，且可执行程序在 OpenCV 安装目录的 bin 目录下，具体训练时只需要根据不同情况输入所需要的命令行参数即可。OpenCV 也提供了训练好的人脸检测分类器文件，在安装目录的 data 目录下。在本系统的具体实现中，为了减少重复劳动，集中精力于算法的研究和改进上，提高系统的开发效率，故而采用了 OpenCV1.0 版本中提供的训练好的人脸检测分类器 haarcascade_frontalface_alt_tree.xml，这是通过大量正例和反例样本的训练得到的，检测效果非常好。

4.3.1 算法伪代码

使用 Boosted Cascade 算法进行人脸检测的伪代码如下：

1. 输入待检测图像。
2. 加载分类器文件。
3. 创建动态内存，用于存储检测到的一序列候选人脸矩形框。
4. 创建一单通道图像，将待检测图像转换为灰度图并存放在此单通道图像

中，而原待检测图像不受影响。

5. 缩小灰度图尺寸。
6. 将缩小后的灰度图进行直方图均衡化处理。
7. 清空第 3 步创建的内存区域。
8. 检测图像中的人脸区域，将找到的这些人脸区域作为一序列的矩形框返回。
9. 输出检测到的人脸区域。

for $i=0, \dots, \text{faces} \rightarrow \text{total}$

- 1) 找到第 i 个检测到的目标矩形中心，将其乘以缩放的比例系数还原到初始待检图像中的位置，并作为圆心；
- 2) 以目标矩形长宽和的四分之一并乘以缩放因子作为半径；
- 3) 在初始待检图像中画出红色的圆。

4.3.2 算法实现流程图

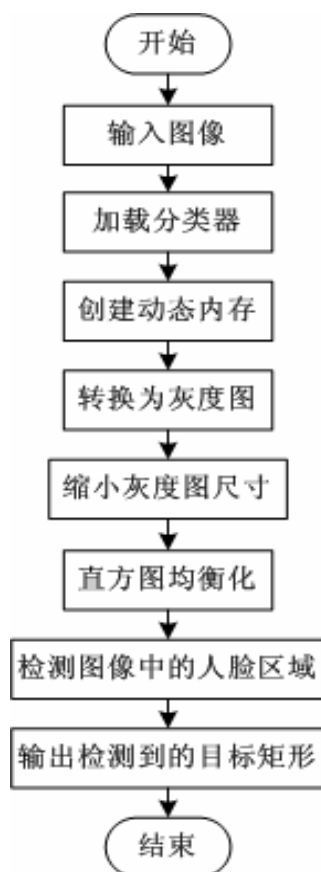


图 4-7 Boosted Cascade 算法流程图

4.3.3 测试及评价

测试的人脸图像由数码相机采集和互联网下载，包括 100 幅不同大小的人脸图像，具有不同的表情变化（微笑、庄重、怪脸），肤色变化（白色、黄色、黑

色人种），眼睛的变化（睁眼、闭眼），离镜头远或近，光照的变化，脸的姿态变化（微扬、微低、偏转），戴眼镜与否，头发发型的变化等，取得了良好的测试结果。这 100 幅人脸图像中共包括 618 个人脸，正确检测出 564 个人脸，检测正确率达到 91.26%，共出现 16 个错误报警。每幅图像的检测速度随图像大小而变化，图像越小，检测耗时越少。如果图像的大小在 500×500 像素左右，则检测时间小于 0.5 秒，如果图像的大小在 1000×1000 像素左右，则检测时间在 1 秒至 2 秒左右。可以看出，Boosted Cascade 算法检测人脸速度很快，基本达到实时的要求。图 4-8 给出了一些正确检测结果。从图中可以看出，该算法可以正确检测不同光照条件下，人脸图像大小各不相同，肤色各异的人脸图像，尤其对于正面人脸，检测效果相当之好。对于人脸偏转程度较小的侧面人脸，能够正确检测，但若人脸偏转程度较大，检测结果不甚理想。另一方面，系统可以检测到有一定表情的人脸，但若表情比较特殊时，不能正确检测。而对于人脸部分被遮挡的情况（如戴墨镜等），也不能正确检测到人脸。图 4-9 给出了一些错误检测结果（包括漏检和误检）。



图 4-8 正确检测结果示例



图 4-9 误检和漏检结果示例

4.4 基于肤色验证的 Boosted Cascade 人脸检测方案

在单纯使用 Boosted Cascade 方法的具体检测过程中，有一些非人脸区域被误识别为人脸，而这其中有些情况是可以利用人脸肤色信息很快排除的区域。在正确的人脸区域中，肤色应该是占主导地位的像素色彩值，所以利用肤色这一线索可以排除掉检测结果中在灰度图像中很像人脸，而对应到彩色图像中根本不是肤色的区域。基于上述原因，我设计了这个对 Boosted Cascade 算法的改进方案，即首先用 Boosted Cascade 算法检测出人脸的候选区域，然后通过人脸肤色模型进行验证，如果候选区域中符合肤色模型验证的像素点个数达到某一比例值，就认为该候选区域为人脸，否则将其看作非人脸区域。

这里仍然使用亮度与色度分开的色彩空间 YCbCr 来建立肤色模型，但为了缩短检测时间，使系统具有更高的效率，没有使用第三章介绍的肤色模型，而是采用另一种由 Douglas Chai^[14]提出的比较简单且好用的肤色模型，该模型虽然比较简单粗糙，但作为后期验证来使用，它的精度已经足够了。因为人脸中不同区域的肤色差异主要在亮度上，故该模型首先对 RGB 图像的亮度归一化，以缩小肤色之间的差异。具体的肤色分布区间为： $Cr \in [123, 175]$, $Cb \in [93, 133]$ ，这是由

Douglas Chai^[14]通过实验选取不同光照、环境、分辨率下的去除眼睛、嘴、眉毛和鼻子等器官的人脸，得到的肤色分布区间，如图 4-10 所示。

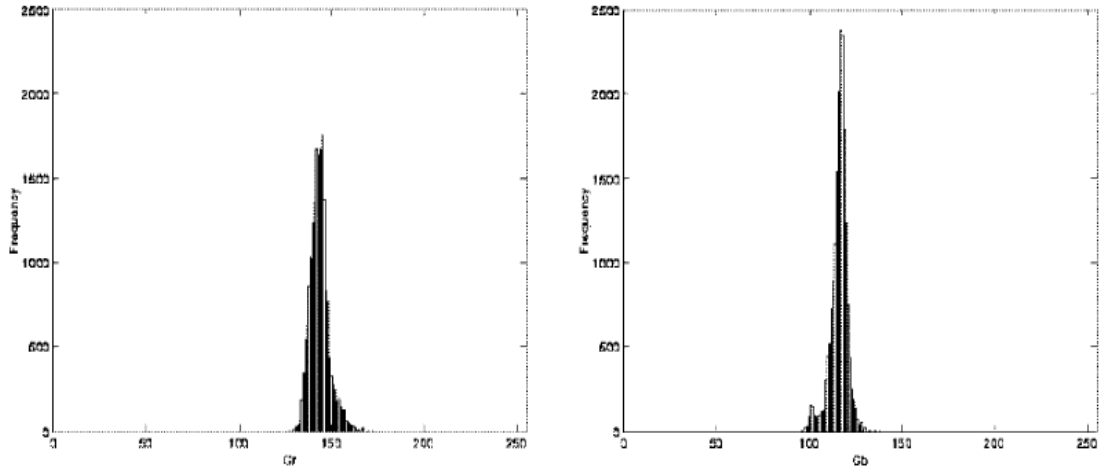


图 4-10 肤色分布区间

一. 算法伪代码

使用基于肤色验证的 Boosted Cascade 算法进行人脸检测的伪代码如下：

1. 输入待检测图像。
2. 加载分类器文件，利用 Boosted Cascade 算法检测出人脸的候选区域。
3. 对候选人脸进行肤色验证。

for i = 0,...,face->height

for j = 0,...,face-> width

- 1)对 RGB 图像亮度归一化，采用如下简单的公式^[15]：

$$\begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} r \\ g \\ b \end{bmatrix} \cdot \frac{255}{r + g + b} \quad (4-4)$$

- 2)进行色彩空间转换，RGB 空间转换到 YCbCr 空间，公式为^[16]：

$$\begin{aligned} Cr &= 128.0 + 112 * r / 255 - 93.786 * g / 255 - 18.214 * b / 255, \\ Cb &= 128.0 - 37.797 * r / 255 - 74.203 * g / 255 + 122 * b / 255 \end{aligned} \quad (4-5)$$

- 3)if (Cr ∈ [123,175] & & Cb ∈ [93,133])

肤色计数加 1;

4. 肤色计数值除以候选区域像素计数值，得到肤色像素所占比例值
5. if (肤色像素所占比例 >= 70%)

认为是人脸区域，找出圆心和半径，用红色的圆标识出来。

else

认为是非人脸区域，排除掉。

二. 算法实现流程图

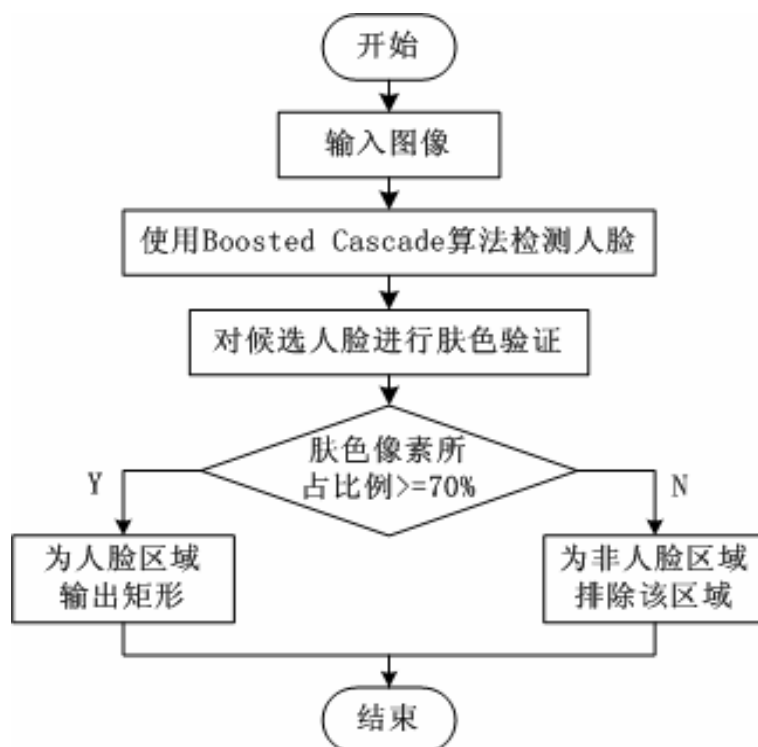


图 4-11 基于肤色验证的 Boosted Cascade 算法流程图

三. 测试结果对比及评价

在图 4-12(a)中 Boosted Cascade 算法检测后有一个被识别为人脸的足球，现在应用肤色验证的方法重新进行检测，保留肤色匹配度大于等于 70%的区域，得到的检测结果中，如图 4-12(b)所示，足球被排除出人脸区域了。



图 4-12 进行肤色验证前后对比图 1

在图 4-13(a)中 Boosted Cascade 算法检测后有一个被误识别为人脸的小区域（在图的下部），现在应用肤色验证的方法重新进行检测，得到的检测结果中，

如图 4-13(b)所示，该误检区域被排除了。



图 4-13 进行肤色验证前后对比图 2

通过实验结果可以看出，所使用的 Boosted Cascade 算法能够在人脸检测中快速准确地检测出人脸，并且通过使用人脸肤色验证，能够有效地提高检测的正确率，减小出现检测错误的几率，具有明显的检测效果。而且肤色验证算法简单，运行速度很快，不影响 Boosted Cascade 算法的实时性。

4.5 基于眼睛验证的 Boosted Cascade 人脸检测方案

在 Boosted Cascade 算法出现误报，而误报区域又恰好近似肤色的情况下，或者整幅图像是在复杂的光照环境下拍摄的，出现严重的肤色偏离，无法使用肤色模型，这时用 4.4 节肤色验证方法就不奏效了，那么如何解决这种情况呢？因为眼睛器官的存在也是人们识别人脸的一个重要信息，而且在面部特征诸如眉毛，眼睛，鼻子，嘴当中，双眼是最为稳定的特征。所以我设计一个基于眼睛器官验证的 Boosted Cascade 人脸检测方案，尝试利用眼睛的信息排除掉那些被 Boosted Cascade 算法误认为人脸的区域，降低系统的误检率。当然利用器官的验证在人脸区域比较大的图像中可能会收到更好的效果。该改进方案首先用 Boosted Cascade 算法检测出人脸的候选区域，然后进行器官验证，加载眼睛分类器，再次利用 Boosted Cascade 算法进行眼睛检测，若能检测到一个眼睛区域，证明该候选区域为人脸，否则为非人脸，排除掉这种候选区域。

一. 算法伪代码

使用基于眼睛验证的 Boosted Cascade 算法进行人脸检测的伪代码如下：

1. 输入待检测图像。

2. 加载分类器文件，利用 Boosted Cascade 算法检测出人脸的候选区域。
3. 对候选人脸进行眼睛器官验证。
 - 1)加载眼睛分类器；
 - 2)利用 Boosted Cascade 算法检测出眼睛区域；
 - 3)if（检测到的眼睛数目==1）

认为该候选区域包含一双眼睛，用绿色矩形标识出眼睛区域；

else

认为该候选区域不包含眼睛。
4. if（候选区域包含一双眼睛）

认为是人脸区域，找出圆心和半径，用红色的圆标识出来。

else

认为是非人脸区域，排除掉。

二. 算法实现流程图

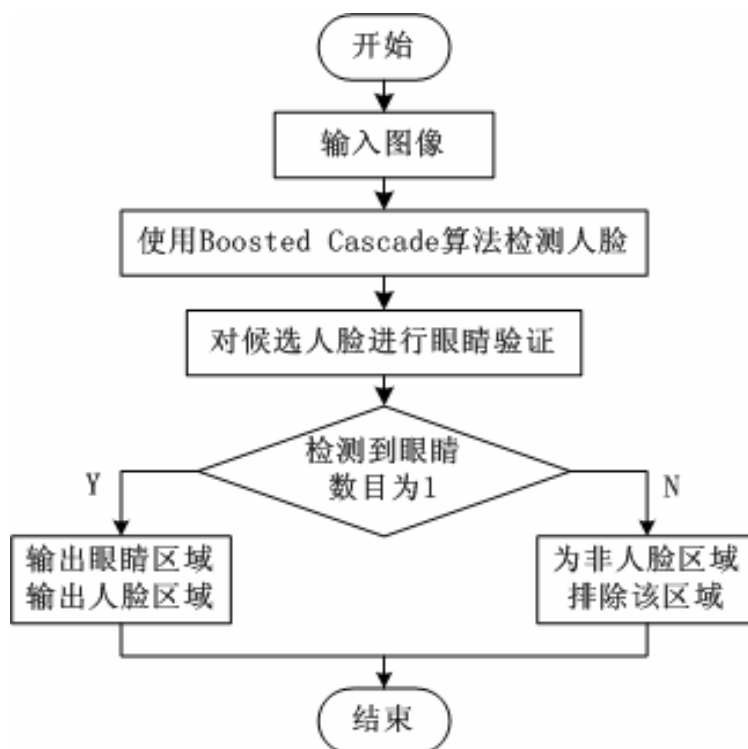


图 4-14 基于眼睛验证的 Boosted Cascade 算法流程图

三. 测试结果对比及评价

在图 4-15(a)中 Boosted Cascade 算法检测后有两个被误识别为人脸的非人脸区域，现在应用眼睛验证的方法重新进行检测，保留具备眼睛器官的部分，得到的检测结果中，如图 4-15(b)所示，两个误检的区域被排除掉了。

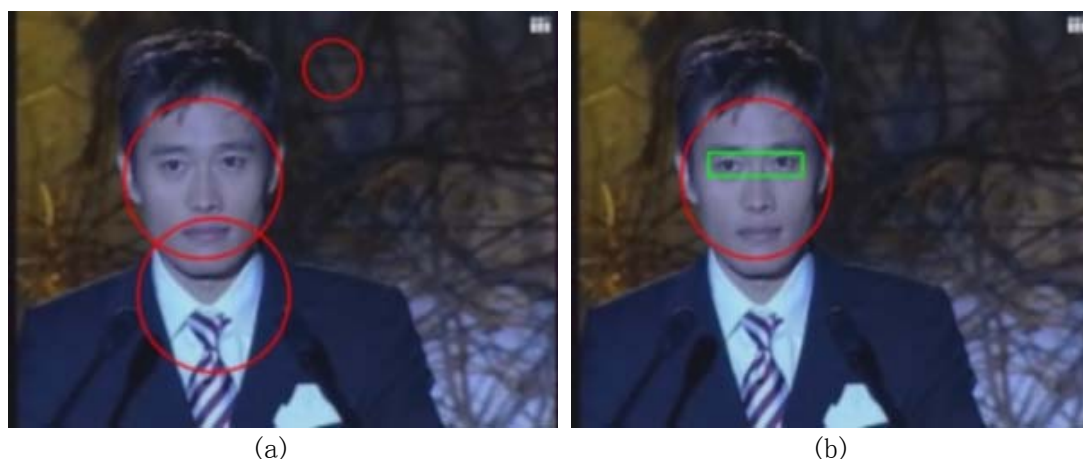


图 4-15 进行眼睛验证前后对比图

通过实验结果可以看出，通过使用眼睛器官验证，能够有效地提高检测的正确率，减小出现误检的几率，具有明显的检测效果。而且眼睛验证算法运行速度很快，不影响 Boosted Cascade 算法的实时性。但该改进方案仅适用于人脸区域比较大，眼睛清晰可见的图像，对那些小人脸图像或图像质量不佳、眼睛区域模糊不清的图像不适宜用这种改进方案。

4.6 正面与侧面检测结合的人脸检测方案

前面 4.3 节的 Boosted Cascade 算法能够检测出大部分的正面人脸和部分人脸偏转程度较小的侧面人脸，但对于一些偏转程度较大的侧面人脸就无法检测出。这是因为所使用的分类器的训练样本是基于正面人脸和小偏转侧面人脸的，偏转程度大的侧面人脸会被分类器识别为非人脸而排除掉。另一方面，使用大量偏转大的侧面人脸作为训练样本训练出的分类器又只适用于检测大偏转侧面人脸，不能兼顾检测正面人脸。而若把大量的正面和侧面人脸拼凑到一起作为正例样本去训练分类器，会使分类器过于庞大和复杂，且检测效果也不好。针对这种情况，我提出了一种将正面人脸检测与侧面人脸检测相结合的检测方案，也是基于 Boosted Cascade 方法，但这一方案将使用两个分类器做人脸检测，一个用来检测正面人脸，另一个用来检测侧面人脸，配合使用，互相补充，在很大程度上提高了系统的人脸检测的正确率，使系统的适用面更广，检测效果更佳。

这一方案首先加载正面人脸检测分类器，进行正面人脸检测。其次，为了避免重复检测，将检测出来的正面人脸区域置为全黑色，这样进行下一步侧面人脸检测时可避开这些区域（因为这些区域为全黑色，不符合任何一个 Haar 特征，所以进行侧面人脸检测时，侧面人脸级联结构分类器的第一级分类器就会淘汰掉这些区域，耗时很短）。最后，加载侧面人脸分类器，对整个待检测图像重新扫描一遍，进行侧面人脸检测。这样，正面和侧面的人脸就都可检测出来，大大提高了检测正确率，同时不会耗费太多时间，检测速度也是可以接受的。

一. 算法伪代码

使用基于眼睛验证的 Boosted Cascade 算法进行人脸检测的伪代码如下：

1. 输入待检测图像。
2. 加载正面人脸检测分类器文件，利用 Boosted Cascade 算法检测出正面人脸的候选区域。
3. 输出检测到的正面人脸区域。找出圆心和半径，用红色的圆标识出来。
4. 将检测出的正面人脸区域置为全黑。
5. 加载侧面人脸检测分类器文件，再次利用 Boosted Cascade 算法检测出侧面人脸区域。
6. 输出检测到的侧面人脸区域。找出圆心和半径，用绿色的圆标识出来。

二. 算法实现流程图

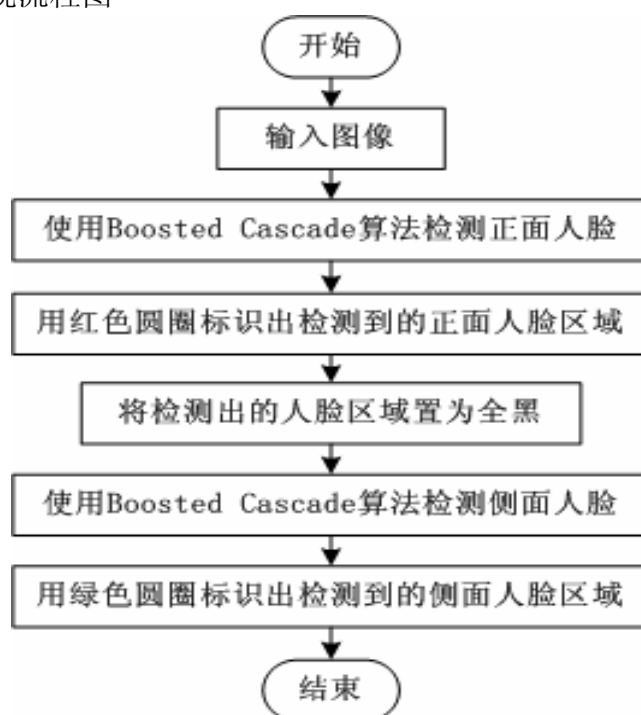


图 4-16 正面与侧面检测结合的 Boosted Cascade 算法流程图

三. 测试结果对比及评价

在图 4-17(a)中使用正面人脸分类器的 Boosted Cascade 算法检测后有一个漏检人脸，即最右侧的侧面人脸没有检测出来。在图 4-17(b)中使用侧面人脸分类器的 Boosted Cascade 算法检测后也有一个漏检人脸，即中间的正面人脸没有检测出来。现在应用正面与侧面检测结合的改进方案重新进行检测，得到的检测结果中，如图 4-17(c)所示，三个人脸区域都成功检测到了。

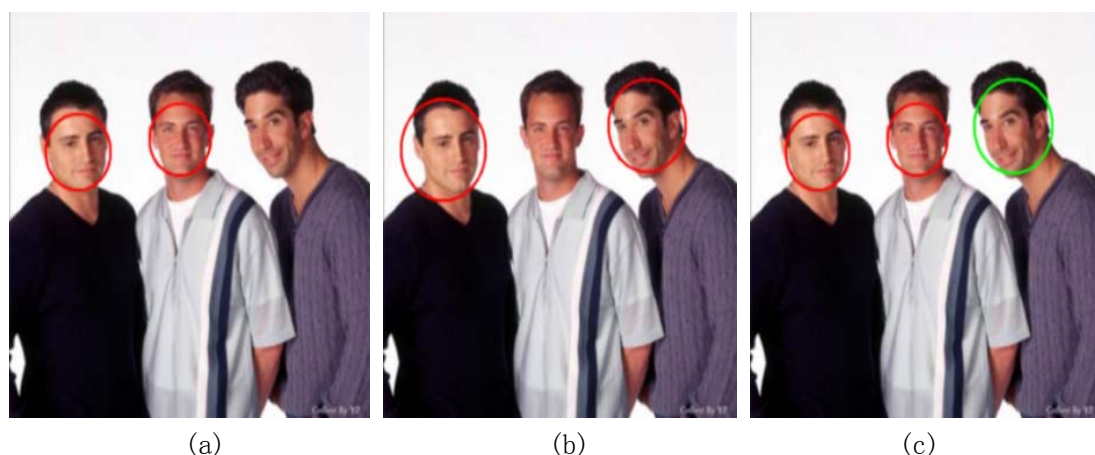


图 4-17 正面与侧面检测结合前后对比图 1

图 4-18(a)是测试图像原图，图 4-18(b)是使用正面人脸分类器的 Boosted Cascade 算法的检测结果，可看到里面有一个漏检人脸，即最右侧的侧面人脸没有检测出来。而在图 4-18(c)中使用侧面人脸分类器的 Boosted Cascade 算法检测后有三个漏检人脸，即左侧三个正面人脸没有检测出来。现在应用正面与侧面检测结合的改进方案重新进行检测，得到的检测结果中，如图 4-18(d)所示，五个人脸区域都成功检测到了。

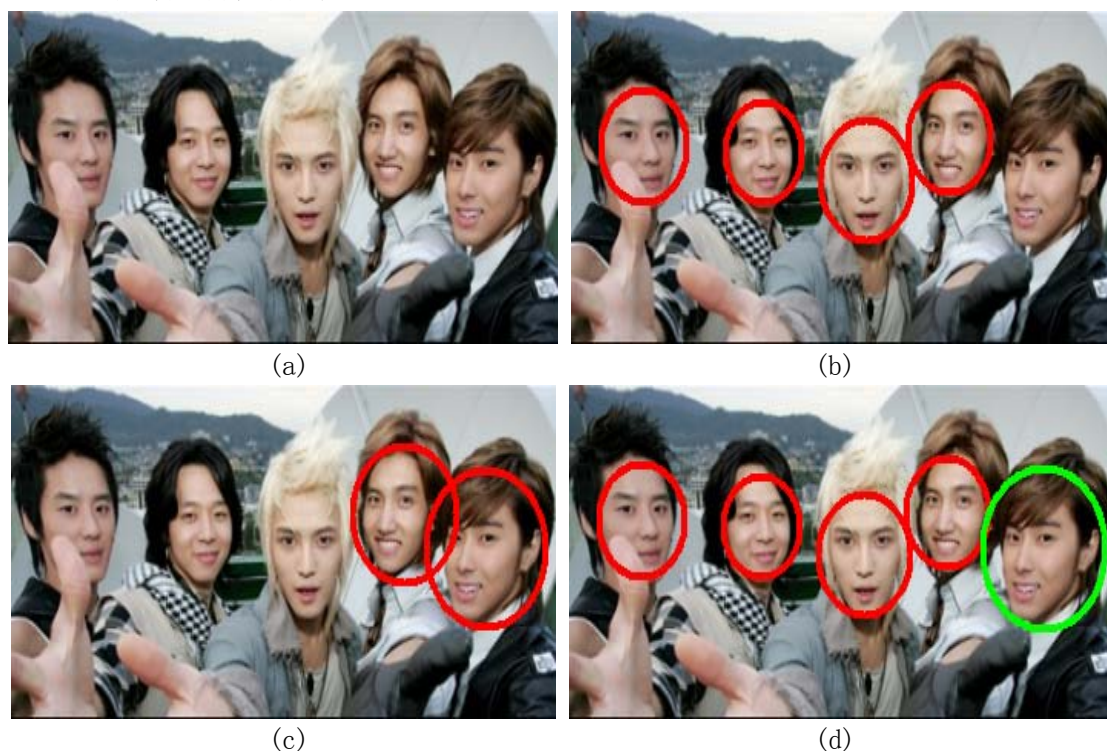


图 4-18 正面与侧面检测结合前后对比图 2

通过实验结果可以看出，通过使用正面与侧面检测结合的改进方案，能够有效地提高检测的正确率，具有很好的检测效果。而且不会耗费太多时间，检测速度也是可以接受的。

第五章 总结与展望

5.1 全文总结

本文对人脸检测这个热门的研究课题进行了详细综述,并着重介绍了我在利用基于肤色特征的人脸检测算法和基于 Boosted Cascade 的人脸检测算法进行人脸检测方面的一些成果。本文成功设计并实现了一个通用的基于特征的人脸检测系统 autofaceGFB,集图像预处理和人脸检测功能于一身,使用方便,功能较完善,对复杂情况适应能力较强。

本文首先介绍了基于肤色特征的人脸检测算法。该算法首先构建肤色模型,利用肤色特征进行人脸检测得到候选人脸区域,然后对每个候选区域运用 Sobel 方差进行脸部特征细节验证,最后输出满足验证的人脸。该方法利用了肤色这一人脸的重要信息,对复杂环境的适应能力强,能够适用于表情变化丰富、人脸姿态任意及面部部分被遮挡的情况,检测正确率较高,而且基于肤色特征的检测算法检测速度很理想,适于实时使用。本文还通过使用 Gamma 校正、光照补偿和色彩平衡等预处理手段,来适应光照变化的复杂情况。

本文还介绍了基于 Boosted Cascade 的人脸检测算法。Boosted 即指级联分类器的每一层都可以从中选取一个 Boosting 算法(权重投票),并利用基础分类器的自我训练得到。就 Boosting 本身而言,其目标是为了提高任何给定的学习算法的分类准确率。Boosting 方法的机制可以保证最终的分类器在训练样本集合上有任意高的准确率。这种方法的人脸检测器是目前从精度和适用范围而言比较成功的一类方法。Boosted Cascade 算法采用积分图的图像表示方法和级联式的分类结构,从而使实时快速的人脸检测成为可能。本文采用这种方法来设计和实现人脸检测系统,得到的系统具有检测正确率高,快速实时的优点。本文还针对该方法处理效果不好的几种情况,分别有针对性的提出了改进方案。为了降低错误报警率,进行了肤色验证和眼睛器官验证,能有效去掉一些误检区域,从而最大限度的降低了错误报警率,而又尽量不影响正确率。为了使系统能有效检测侧面人脸,提高系统的检测正确率,还设计了将正面与侧面人脸检测相结合的改进方案,大大提高了系统的适用范围。从测试结果也可以看出本文提出的改进方案是有效的。

综上,本文设计并实现的 autofaceGFB 人脸检测系统综合利用了基于肤色特征的人脸检测算法和基于 Boosted Cascade 的人脸检测算法,不但能满足实时人脸检测所需要的精度,而且在运行速度上有较好的表现,适应范围十分广泛。

5.2 研究展望

虽然本文设计实现的 autofaceGFB 人脸检测系统已经具有较好的性能,利用基于肤色特征的人脸检测算法和基于 Boosted Cascade 的人脸检测算法进行人脸

检测取得了较好的成果，但在思想和算法上还有改进和继续深入研究之处，使人脸检测功能的实现更为完善。比如目前对基于肤色特征的人脸检测算法的错误报警率的控制还不是很令人满意，有待提高。这一方面可以继续尝试修改人脸面部细节的验证规则，提高对面部特征的验证的准确率，而这也直接关系到系统的检测速度和检测正确率。最后，人脸检测的复杂性使得仅仅单独使用一种方法很难取得很好的检测效果，将各种方法有效综合运用将是以后人脸检测研究的必然趋势。

参考文献

- [1]王金庭, 杨敏. 人脸检测技术研究[J]. 计算机系统应用, 2006, 4: 31—33.
- [2]黄福珍, 苏剑波. 人脸检测[M]. 上海: 上海交通大学出版社, 2006. 1—14.
- [3]黎松, 平西建, 丁益洪. 开放源代码的计算机视觉类库 OpenCV 的应用[J]. 计算机应用与软件, 2005, 22 (8): 134—136.
- [4]刘洁, 张汗灵. 一种新的基于肤色模型的人脸检测算法[J]. 计算机工程与应用, 2006, 11: 70—71.
- [5]Christophe Garcia, Georgios Tziritas. Face detection using quantized skin color regions merging and wavelet packet analysis[J]. IEEE TRANS. MULTIMEDIA, 1999, 1(3): 264—277.
- [6]Rein-Lien Hsu, Mohamed Abdel-Mottaleb, Anil K. Jain. Face Detection in Color Images[J]. IEEE TRANS. PAMI, 2002, 24(5): 696—706.
- [7]张宏林, 蔡锐. Visual C++数字图像模式识别技术及工程实践[M]. 北京: 人民邮电出版社, 2003. 363—370.
- [8]游亚平, 李明, 袁保宗. 可变光照下的人脸检测[J]. 信号处理, 2004, 20 (2): 101—107.
- [9]Charles Poynton, A technical introduction to digital video [EB/OL]. <http://www.poynton.com/notes/TIDV/index.html>, 2003-9-25.
- [10]赵明华, 游志胜, 穆万军等. 基于肤色分割、数学形态学和几何方法的人脸检测[J]. 四川大学学报 (自然科学版), 2005, 42 (3): 70—71.
- [11]Intel. Open Source Computer Vision Library (OpenCV) [EB/OL]. <http://www.opencv.org.cn>, 2007-6-1.
- [12]Viola P, Jones M. Rapid Object Detection Using a Boosted Cascade of Simple Features[A]. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition [C]. Hawaii, USA: 2001, 511—518.
- [13]杜宇. 基于 Boosted Cascade 算法的人脸检测和跟踪系统[J]. 电子科技, 2006, 7: 67—70.
- [14]Douglas Chai, King N. Ngan. Face Segmentation Using Skin-Color Map in Videophone Applications[J]. IEEE TRANSACTIONS CSVT, 1999, 9(4): 551—564.
- [15]何良华, 邹采荣, 赵力. 一种基于肤色和模板梯度的人脸检测方法[J]. 应用科学学报, 2004, 22 (4): 487—492.
- [16]周书仁, 梁昔明, 叶吉祥等. 基于脸部信息和支持向量机的人脸检测[J]. 计算机应用, 2006, 26 (5): 1032—1034.

附 录

一、 基于肤色特征的人脸检测算法实现代码

struct FaceDetectReturn m_FaceDetectReturn; //人脸检测输出参数信息

struct FaceDetectReturn ImageProcessor::SkinDetect(IplImage* img)

```
{
    IplImage* skinimg=cvCreateImage(cvSize(img->width,img->height),8,3);
    cvCopy(img,skinimg); //创建一个待检测图像的副本
    int i,j;
    double t=(double)cvGetTickCount();//计时
    m_FaceDetectReturn.faceskincount=0;
    m_FaceDetectReturn.faceorgancount=0;
    m_FaceDetectReturn.facecount=0;
    for(i=0;i<img->height;i++)
    {
        for(j=0;j<img->width;j++)
        {
            float b=((uchar*)(img->imageData+img->widthStep*i))[j*img->nChannels];
            float g=((uchar*)(img->imageData+img->widthStep*i))[j*img->nChannels+1];
            float r=((uchar*)(img->imageData+img->widthStep*i))[j*img->nChannels+2];
            //计算得到 y, cr, cb 的数值
            float Y=(257*r+504*g+98*b)/1000+16;
            float Cr=(439*r-368*g-71*b)/1000+128;
            float Cb=(-148*r-291*g+439*b)/1000+128;
            //如果 y 数值在两个临界值之间, 保持不变
            if(Y<Kl || Y>Kh)
            {
                //调用非线性转换函数调整 Cr, Cb 的数值
                Cr = (Cr-_Cr(Y))*(Wcr/_WCr(Y))+_Cr(Kh);
                Cb = (Cb-_Cb(Y))*(Wcb/_WCb(Y))+_Cb(Kh);
            }
            //Cb 的系数常量
            const float cx=114.38;
            //cr 的系数常量
            const float cy=160.02;
            //角度常量
            const float theta=2.53;
            //x 轴线和 y 轴线的两个常量
            const float ecx=1.60;
            const float ecy=2.41;
            //长轴
            const float a0=28.39;
            //短轴
            const float b0=16.03;
            //相似度常量
```

```

const float judge=0.5;
//计算得到 x 轴数值
float x=cos(theta)*(Cb-cx)+sin(theta)*(Cr-cy);
//y 轴数值
float y=-sin(theta)*(Cb-cx)+cos(theta)*(Cr-cy);
//计算离心率
float temp=pow(x-ecx,2)/pow(a0,2)+pow(y-ecy,2)/pow(b0,2);
//如果满足要求即为肤色，否则为非肤色
if(temp<=1.0)
{
    //肤色
    ((uchar*)(skining->imageData+skining->widthStep*i))[j*skining->nChannels]=255;
    ((uchar*)(skining->imageData+skining->widthStep*i))[j*skining->nChannels+1]=255;
    ((uchar*)(skining->imageData+skining->widthStep*i))[j*skining->nChannels+2]=255;
}
else
{
    //非肤色
    ((uchar*)(skining->imageData+skining->widthStep*i))[j*skining->nChannels]=0;
    ((uchar*)(skining->imageData+skining->widthStep*i))[j*skining->nChannels+1]=0;
    ((uchar*)(skining->imageData+skining->widthStep*i))[j*skining->nChannels+2]=0;
}
}

int mouth;
int eye;
int Sobel;
float areacontour;
float arearect,coef;
IplConvKernel* element=cvCreateStructuringElementEx(3,3,1,1,CV_SHAPE_ELLIPSE,
NULL);

cvDilate(skining,skining,element,1);//膨胀
cvErode(skining,skining,element,1);//腐蚀
cvSmooth(skining,skining,CV_MEDIAN,3,0,0);//中值滤波去除孤立噪声点
cvDilate(skining,skining,element,5);//膨胀
cvErode(skining,skining,element,5);//腐蚀
cvReleaseStructuringElement(&element);
CvMemStorage* storage=cvCreateMemStorage(0);
CvSeq* contour=0;
IplImage* gray=cvCreateImage(cvSize(skining->width,skining->height),8,1);
cvCvtColor(skining,gray,CV_BGR2GRAY); //转换成灰度图
//找轮廓
cvFindContours(gray,storage,&contour,sizeof(CvContour),CV_RETR_TREE,
CV_CHAIN_APPROX_SIMPLE,cvPoint(0,0));
contour=cvApproxPoly(contour,sizeof(CvContour),storage, CV_POLY_APPROX_DP,3,1);
for( ; contour!=0;contour=contour->h_next)

```

```

{
    areacontour=fabs(cvContourArea(contour,CV_WHOLE_SEQ));
    if(areacontour>=600) //当该区域面积大于等于 600 时看作候选区域
    {
        CvRect rect;
        rect=cvBoundingRect(contour,0);
        m_FaceDetectReturn.facecount++;
        CvPoint pt1,pt2;
        pt1.x=rect.x;
        pt1.y=rect.y;
        pt2.x=rect.x+rect.width;
        pt2.y=rect.y+rect.height;
        Sobel=Sobel_detect(rect,img); //进行 Sobel 验证
        if((Sobel>=2)&&(Sobel<=15))
        { //输出满足 Sobel 验证的区域
            cvRectangle(img,pt1,pt2,CV_RGB(255,0,0),3,8,0);
            m_FaceDetectReturn.faceorgancount++;
        }
    }
}

t=(double)cvGetTickCount()-t;
t=t/((double)cvGetTickFrequency()*1000);
m_FaceDetectReturn.dtime=t;
cvReleaseImage(&skinimg);
cvReleaseImage(&gray);
return m_FaceDetectReturn;
}

//进行 Sobel 验证处理
int Sobel_detect(CvRect rect,IplImage* img)
{
    cvSetImageROI(img,rect); //设置待检测图像的感兴趣区域
    //创建一个和矩形 rect 一样大小的 IplImage 类型的图像 smallface
    IplImage* smallface=cvCreateImage(cvSize(rect.width,rect.height),8,3);
    cvCopy(img,smallface); //将待检测图像的感兴趣区域复制到刚创建的图像中
    IplImage* gray=cvCreateImage(cvSize(rect.width,rect.height),8,1);
    cvCvtColor(smallface,gray,CV_BGR2GRAY); //转换成灰度图像
    cvSobel(gray,gray,0,1,3); //进行 y-方向 Sobel 处理
    cvThreshold(gray,gray,70,255,CV_THRESH_BINARY); //进行二值化处理
    CvMemStorage* storage=cvCreateMemStorage(0);
    CvSeq* contour=0;
    int result=0;
    float area=((rect.width)*(rect.height))*0.2;
    cvFindContours(gray,storage,&contour,sizeof(CvContour),CV_RETR_TREE,

```

```

        CV_CHAIN_APPROX_SIMPLE,cvPoint(0,0)); //找轮廓
for( ; contour!=0;contour=contour->h_next)
{
    //轮廓面积小于 33 的区域排除掉
    if(fabs(cvContourArea(contour,CV_WHOLE_SEQ))>=33)
    {
        if(fabs(cvContourArea(contour,CV_WHOLE_SEQ))<=area)
            result++;//轮廓面积小于等于候选区域 20%看作候选特征
        else
            {//轮廓面积大于候选区域 20%的区域排除掉
                result=0;
                break;
            }
    }
}
cvResetImageROI(img); //释放待检图像 img 的感兴趣区域
return result;
}

```

二、 Boosted Haar 分类器结构

本文使用了基于 Boosted Cascade 的人脸检测算法。该算法的主要思想是通过学习建立多层级联结构的分类器，最终形成一个有效的分类器，将候选子窗口的人脸和非人脸区分开来。分类器的输入是 Haar 特征。了解 Boosted Haar 分类器结构才能很好的理解基于 Boosted Cascade 的人脸检测算法，所以鉴于其重要性，现将 OpenCV 1.0 版本中 Boosted Haar 分类器结构简单介绍如下：

```

#define CV_HAAR_FEATURE_MAX 3
/* 一个 haar 特征由 2-3 个具有相应权重的矩形组成 */
typedef struct CvHaarFeature
{
    int tilted; /* 0 means up-right feature, 1 means 45--rotated feature */
    struct
    {
        CvRect r;
        float weight;
    } rect[CV_HAAR_FEATURE_MAX];
}CvHaarFeature;
typedef struct CvHaarClassifier
{
    int count; /* number of nodes in the decision tree */
    CvHaarFeature* haar_feature;
    float* threshold;
    int* left;
    int* right;
    float* alpha;
}

```

```

}CvHaarClassifier;
typedef struct CvHaarStageClassifier
{
    int    count; /* number of classifiers in the battery */
    float threshold; /* threshold for the boosted classifier */
    CvHaarClassifier* classifier; /* array of classifiers */
    /* these fields are used for organizing trees of stage classifiers,
       rather than just stright cascades */
    int next;
    int child;
    int parent;
}CvHaarStageClassifier;
typedef struct CvHidHaarClassifierCascade CvHidHaarClassifierCascade;
typedef struct CvHaarClassifierCascade
{
    int    flags;
    int    count;
    CvSize orig_window_size;
    CvSize real_window_size;
    double scale;
    CvHaarStageClassifier* stage_classifier;
    CvHidHaarClassifierCascade* hid_cascade;
}CvHaarClassifierCascade;

```

所有的结构都代表一个级联 boosted Haar 分类器。级联有下面的等级结构：

Cascade:

```

Stage1:
    Classifier11:
        Feature11
    Classifier12:
        Feature12
    ...
Stage2:
    Classifier21:
        Feature21
    ...
...

```

整个等级可以手工构建，也可以利用函数 `cvLoadHaarClassifierCascade` 从已有的磁盘文件或嵌入式基中导入。现在的目标检测分类器通常存储在 XML 或 YAML 文件中，而不是通过路径导入。从文件中导入分类器，使用函数 `cvLoad`。本文就是通过 `cvLoad` 函数加载分类器文件的。

三、 Boosted Cascade 算法实现代码

// A global variable

```

struct FaceDetectReturn m_FaceDetectReturn; //人脸检测输出参数信息
static CvMemStorage* storage = 0;
static CvHaarClassifierCascade* cascade = 0;
struct FaceDetectReturn ImageProcessor::FaceDetect(IplImage* img)
{
    //加载分类器
    cascade=(CvHaarClassifierCascade*)cvLoad(frontalface_cascade_name, 0,0,0);
    //创建动态内存
    storage=cvCreateMemStorage(0);
    //调用人脸检测处理函数
    face_detect(img);
    return m_FaceDetectReturn;
}
//人脸检测处理函数
void face_detect(IplImage* img)
{
    double scale=1.3; //图像缩放比例系数
    IplImage* gray=cvCreateImage(cvSize(img->width,img->height),8,1);
    IplImage* small_img=cvCreateImage(cvSize(cvRound(img->width/scale),
                                              cvRound (img->height/scale)),8,1);

    int i;
    cvCvtColor(img,gray,CV_BGR2GRAY); //转换为灰度图
    cvResize(gray,small_img,CV_INTER_LINEAR); //缩小尺寸
    cvEqualizeHist(small_img,small_img); //直方图均衡化处理
    cvClearMemStorage(storage); //清空用来存放目标矩形的内存区域
    if(cascade)
    {
        double t=(double)cvGetTickCount(); //计时
        //检测图像中的人脸区域
        CvSeq* faces=cvHaarDetectObjects(small_img,cascade,storage,1.1,2,0,cvSize(30,30));
        for(i=0;i<(faces?faces->total:0);i++)
        { //输出检测到的目标矩形
            CvRect* r=(CvRect*)cvGetSeqElem(faces,i);
            CvPoint center;
            int radius;
            center.x=cvRound((r->x+r->width*0.5)*scale);
            center.y=cvRound((r->y+r->height*0.5)*scale);
            radius=cvRound((r->width+r->height)*0.25*scale);
            cvCircle(img,center,radius,CV_RGB(255,0,0),3,8,0);
        }
        t=(double)cvGetTickCount()-t;
        t=t/((double)cvGetTickFrequency()*1000);
        m_FaceDetectReturn.faceskincount=0;
        m_FaceDetectReturn.faceorgancount=0;
    }
}

```

```

        m_FaceDetectReturn.facecount=faces->total; //检测到的人脸数目
        m_FaceDetectReturn.dtime=t; //检测人脸所用时间
    }
    cvReleaseImage(&gray);
    cvReleaseImage(&small_img);
}

```

其中函数 `cvHaarDetectObjects` 是此 Boosted Cascade 算法中最重要的核心函数，鉴于其重要性，这里对这个函数的具体使用详细介绍一下。该函数检测图像中的人脸区域，将找到的这些人脸区域作为一序列的矩形框返回，存放在 `CvSeq*` 类型变量 `faces` 中。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 裁减方法。函数在处理和收集到候选的方框之后，接着对这些区域进行组合并且返回一系列各个足够大的组合中的平均矩形。下面是 `cvHaarDetectObjects` 函数原型：

```

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` 是 Haar 分类器级联的内部标识形式；`storage` 是用来存储检测到的一序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。例如 1.1 指将搜索窗口依次扩大 10%。`min_neighbors` 是构成检测目标的相邻矩形的最小个数；`flags` 是操作方式，当前唯一可以定义的操作方式是 `CV_HAAR_DO_CANNY_PRUNING`，如果被设定，函数利用 Canny 边缘检测器来排除一些边缘很少或者很多的图像区域，因为这样的区域一般不含被检目标，人脸检测中通过设定阈值使用了这种方法，并因此提高了检测速度；`min_size` 是检测窗口的最小尺寸，缺省的情况下被设为分类器训练时采用的样本尺寸（人脸检测中缺省大小是 20×20 ）。

外文资料

Author: Bernd Heisele, Thomas Serre, Massimiliano Pontil, Tomaso Poggio.

Title: Component-based Face Detection.

In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2001, 1: I-657—I-662.

中文译文

基于组件的人脸检测

作者: Bernd Heisele^{†‡}, Thomas Serre[†], Massimiliano Pontil[§], Tomaso Poggio[†]

[†]Center for Biological and Computational Learning, M.I.T., Cambridge, MA, USA

[‡]Honda R&D Americas, Inc., Boston, MA, USA

[§]Department of Information Engineering, University of Siena, Siena, Italy

{heisele, serre, tp} @ai.mit.edu pontil@dii.unisi.it

摘要

我们提出一个基于组件的可训练的系统,该系统可检测正面的和接近正面的静态灰度人脸图像。系统由一个两层结构的支持向量机(SVM)分类器组成。在第一层,各个组件分类器独立地对人脸的各个组件进行检测。在第二层,图像中已经在上一层被检测过的各个组件再统一由一个独立的分类器去检查,查看它们的几何结构是否与一个人脸几何模型相匹配。在本文中,我们还提出这样一个方法,它通过使用三维头模型来自动地学习人脸的各个组件。这个方法有一个优点,即不需要人工的交互来选择和提取各个组件。实验表明,相对于一个类似的在整体人脸模式上进行训练的系统来说,我们提出的这个基于组件的系统在处理人脸旋转的情况方面性能更为健壮。

1. 介绍

在过去的十年,计算机视觉研究领域中人脸检测得到深入地研究主要有以下两个原因。第一,人脸检测有诸多有意义的应用:它是人脸识别系统,监控系统及基于影像的计算机接口系统的一个组成部分。第二,人脸构成一类视觉上相似的物体,它简化了物体检测的通常困难的工作。

下面,我们简要介绍一下静态灰度图像的人脸检测技术。因为没有颜色和运动的信息可被获得,人脸检测可归结成一个纯的模式识别工作。文献[15]中提出来这样一种方法,通过结合聚类技术和神经网络技术来检测灰度图像中的人脸。此方法聚类一组训练图像来产生人脸和非人脸原型,使用一个多层的感知器来将输入模式和原型之间的距离分类。在文献[8]中提出,由一个多项式 SVM 分类器去检测正面的人脸。文献[10]提出了一个能够处理图像平面中旋转问题的系统。它包含两个神经网络,一个用来估计人脸的方位,另一个用来检测反向旋转的人脸。识别的步骤可经过独立训练的同一结构的网络之间的仲裁而得到改良[11]。文献[12]引入一种自然的贝叶斯方法,这种方法测定人脸图像中小矩形的亮度模式出现的经验概率。在文献[13]中系统通过添加一个单独的,在侧面人脸图像上

训练过的分类器而得到扩展，能去处理正面的和侧面的人脸图像。文献[6]提出另一个可能用来检测局部人脸的方法，它使用局部特征提取器来检测眼睛，嘴角和鼻尖，使用条件搜索来使这些特征的几何结构与一个模型的结构相匹配。文献[9]发表了一种相关的使用统计模型的方法。通过把多尺度多方位的过滤器应用于输入图像来提取局部特征。过滤器在训练集上的反应结果被作为高斯分布来建模。组件检测也适用于人脸识别。文献[18]中，在有弹性的网格节点处计算局部特征。在文献[1, 2]中，眼睛，鼻子和嘴的各自独立的模板得到匹配。最后，文献[7]提出一种使用 SVM 的基于组件的人脸检测方法。

基于部分或基于组件的物体检测背后有三个基本的思想。第一，某些类的物体可以被该物体的一些特有的部分和它们之间的几何关系很好地描述。第二，某些物体部分的模式在姿势变化的情况下可能比那个物体整体的模式变化量要少。第三，基于组件的方法在应对部分被遮挡的情况下可能比一个全局的方法性能更健壮。基于组件的方法的两个主要的问题是如何去选择物体的有差别的部分和如何去建模它们的几何结构。上面提到的方法要么人工地定义一组组件并对它们的几何结构进行建模，要么统一地将图像分割成各个组件并且假定它们之间统计上独立。

我们为学习来自三维头模型的有关组件提出一种技术。该技术开始于一组通过最小化预期的 SVM 错误率的界限而逐渐生长的小种子区域。这种方法有一个优点即不需要人工的交互来从训练集中选择和提取组件。一旦组件被测定，我们就训练一个包含两级 SVM 分类器的系统。在第一级上，各组件分类器独立地检测面部的各个组件。在第二级上，图像中的各个已经被单独检测过的组件再统一由一个独立的分类器检查，看看它们的几何结构是否与一个人脸的几何模型相匹配。

论文的大纲如下：第 2 节给出关于 SVM 学习的一个简短的描述。第 3 节我们描述基于组件的人脸检测系统。第 4 节我们提出一个方法用来自动从虚拟人脸图像中提取各个组件。第 5 节包含各项实验结果，以及整体的方法和基于组件的方法之间的分析对比。第 6 节总结这篇论文。

2. 支持向量机的学习

在这一节中我们简述 SVM(支持向量机)的基本理论[16]。SVM 应用于模式识别中来解决两类分类问题，这是通过测定独立的分类超平面¹来实现的，这种超平面能够使得离它最近的训练集中的点与它的距离达到最大值。这些点被称为支持矢量。如果输入空间中的数据不是线性可分的，可以引入非线性变换 $\Phi(\cdot)$ ，将输入空间中的数据点 $x \in \mathbf{IR}^n$ 映射到被称为特征空间的高维(可能无限大)空间 $\mathbf{IR}^p$ 。特征空间中的数据能被上面所描述的最优分类超平面所分隔开。映射 $\Phi(\cdot)$ 在 SVM

¹ SVM 理论也包括非可分离数据的情形，见[16]。

分类器中实现是通过一个在 IR^N 中定义内积的核函数 $K(\cdot, \cdot)$ ，亦即 $K(x, t) = \Phi(x) \cdot \Phi(t)$ 。SVM 的决定函数形式如下：

$$f(x) = \sum_{i=1}^{\ell} \alpha_i y_i K(x_i, x), \quad (1)$$

这里 ℓ 表示训练集中数据点的个数， $y_i \in \{-1, 1\}$ 是数据点 x_i 的类别标号。等式(1)中的系数 α_i 是二次规划问题的解[16]。

令 M 为支持矢量到超平面的距离的两倍。这个量被称为分类间隔并给出等式如下：

$$M = \frac{1}{\sqrt{\sum_{i=1}^l \alpha_i}}. \quad (2)$$

分类间隔是数据可分离性的指示器。实际上，SVM 的预期的错误率， EP_{err} ，满足如下的界限[16]：

$$EP_{err} \leq \frac{1}{l} E \left[\frac{D^2}{M^2} \right], \quad (3)$$

其中 D 为特征空间中包含数据点的最小区域的直径。稍后在论文中我们将尝试最小化这个量去自动提取各个组件。

3. 基于组件的人脸检测

3.1 动机

我们在前面的介绍中简要地提到了整体方法对于物体姿势的改变是高度敏感的。图 1 为线性分类的简单情形举例说明了这一问题。在多幅人脸图像上训练线性分类器的结果可以用一个人脸模板来表示，用图显示在图 1. a) 中。可见，即使很小的旋转，模板也明显脱离了那些被旋转过的人脸，这很清晰地显示在图 1. b) 和 c) 中。而基于组件的方法可通过独立地检测人脸的各个部分来尽力避免这个问题。在图 2 中，眼睛，鼻子，嘴分别用独立的模板来表示。对于一些小的旋转情况，相比较于在整体人脸模式中的改变量，在各个组件中的改变量是很小的。轻微地移动各个组件就足以去达成一个与被旋转过的人脸合理的匹配。



图 1 与单一模板的匹配。a) 中显示一个正面人脸的图形模板。人脸在图像平面上 b) 和在深度方向上 c) 轻微地旋转导致了人脸和模板之间相当多的不符之处。



图 2 与一组独立组件模板的匹配。a) 中显示一个正面人脸的各个组件图形模板。移动各组件模板能补偿人脸在图像平面上 b) 和在深度方向上 c) 的轻微地旋转。

3.2 系统的概述

图 3 给出了我们的两级的基于组件分类器的概述。在第一级上，各个组件分类器独立地检测人脸的各个组件。在展示的例子中，这些组件分别为眼睛，鼻子和嘴。我们使用线性SVM分类器，其中的每一个分类器都在一组提取出的面部组件上和一组随机选择的非人脸模式上进行训练。这些组件都是来自三维头模型的虚拟的 58×58 像素大小的人脸图像中自动提取出来的。在第二级上，几何结构分类器通过线性组合各组件分类器输出的结果来完成最终的人脸检测。给出一个 58×58 像素大小的窗口，在围绕着预期各组件位置处的小矩形搜索区域²内，各组件分类器连续的最大的输出结果被用来作为几何结构分类器的输入。搜索区域是根据训练图像中组件位置的普通的偏差以及标准的偏差而估算好的。我们也向几何分类器提供各被检测组件的精确位置，这个位置是相对于那个 58×58 像素大小窗口的左上角的。大致上，每个组件分类器有三个值要传递到几何分类器。系统是依下列各项进行计算的：我们用 x 表示输入图像，用 $\{x^t\}_{t=1}^T$ 表示被提取出的各个组件。给出一个组件分类器的决定函数如下：

$$f^t(x^t) = \sum_{i=1}^l \alpha_i^t K^t(x_i^t, x^t)$$

这里 K^t 表示被第 t 个分类器使用的核函数。几何结构分类器 $F(x)$ 是各组件分类器的输出和各被检测组件的图像位置 (h^t, v^t) 的一个线性组合：

$$F(x) = \sum_{t=1}^T c^t \cdot (f^t(x^t), h^t, v^t)^T,$$

其中系数矢量 c^t 是从各例子中学习得到：

$$\{(f^1(x_i^1), h_i^1, v_i^1, \dots, f^T(x_i^T), h_i^T, v_i^T, y_i)\}_{i=1}^\ell$$

这里的标号 y_i 对于人脸例子来说值为 1，而对于非人脸例子来说值为 -1， ℓ 表示例子的个数。

²输入图像的多尺度影响输出，解释了组件大小的改变。在我们的测试中，我们设置尺度范围为 $[0.75, 1.2]$ 。

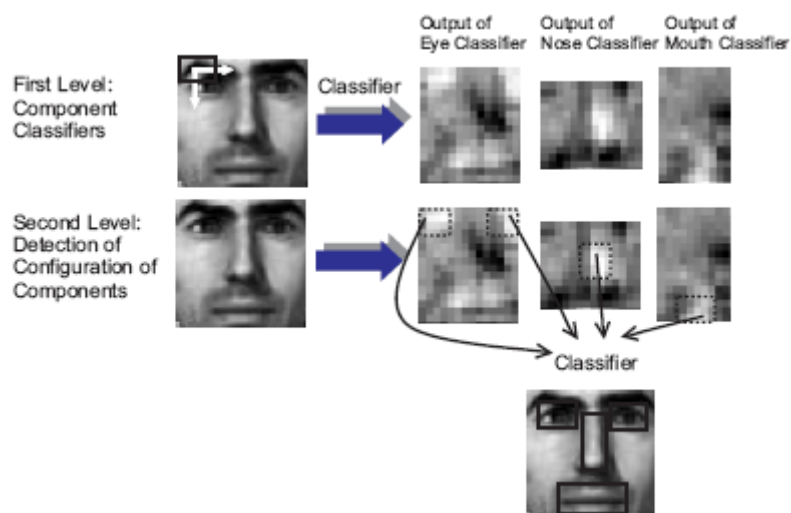


图3 使用四个组件的基于组件分类器的系统概述。在第一级，各组件大小的窗口（实线框）在人脸图像上移动，并且被各组件分类器进行分类。在第二级，在预期的搜索区域（虚线框）内的各组件分类器的最大输出，以及各组件的位置被送入几何结构分类器中。

3.3 训练数据

提取人脸模式通常是需要人工完成的，乏味的，耗时的的工作。而采用这种基于组件的方法，我们将不得不从训练集中的所有图像上人工地提取每一个独立的组件。这一过程只在组件数量少的情况下可行。因为这个原因我们使用三维头模型[17]来产生训练数据。通过绘制三维头模型我们能够自动地产生大量的处在任意姿势下和任意光照下的人脸。除了三维信息，我们也掌握图4显示的一组参考点之间的三维一致性关系。这些一致性关系使得我们可以自动地提取定位在参考点周围的面部组件。最初我们有7个获取自三维扫描仪的头模型，其他的头模型是通过所有原始头模型间的三维变种而产生的。变种的产生是通过让这些原始头模型在 -30° 到 30° 之间进行深度方向上的旋转，加之使人脸被环境光和一指向人脸中心的单束光照射，图5举出了一些例子。这些光线的位置在方位角度上从 -30° 到 30° 间变化，在高度上从 30° 到 60° 间变化。大体上，我们一共产生了2457幅 58×58 像素大小的人脸图像。

反例样本训练集最初有10,209幅 58×58 像素大小的非人脸模式，它们是随机地从502幅非人脸图像中提取出来的。然后我们用自举的方法来扩大训练数据，得到一些相似于人脸的非人脸模式。通过这样做，我们训练了一个独立的，线性的SVM分类器，并且将它应用到先前使用的502幅非人脸图像集上。那些错误的肯定(FPs)被添加到非人脸训练数据去建立最终的13,654大小的非人脸训练集。

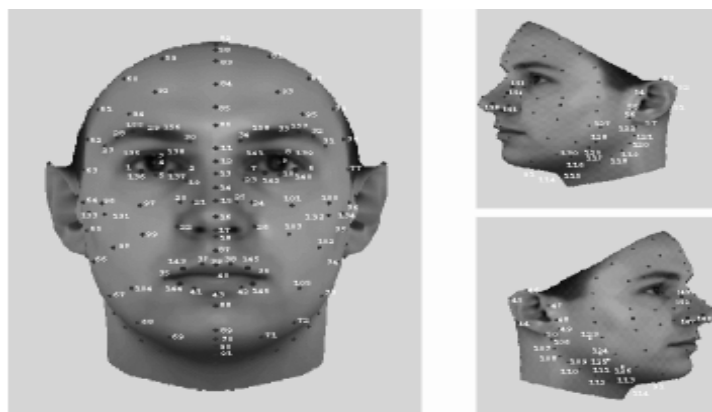


图 4 头部模型上的参考点，它们被用来三维变种和面部组件的自动提取。



图 5 虚拟人脸的例子。

4. 学习各组件

基于组件的方法的一个主要的问题是如何去选择那组有差别的物体的部分。对于人脸这一类别，一个明显的组件的选择可能是眼睛，鼻子和嘴。然而，对于其他类的物体，人工地定义一组直觉上有意义的组件可能是困难的。根据各组件的可识别的强度和它们对抗姿势和光照改变的健壮程度来自动地选择组件可能比人工地选择它们更合理。

解决自动检测组件问题的一个途径是在大小和位置随机的组件上训练大量的分类器。各组件能够根据各分类器的训练结果而被排列和挑选，如预期的错误率的界限。然而，这种方法在训练阶段中是计算广泛的。

相对于使用一大组任意组件的方案，另一个替换方案是明确地产生出有辨识力的组件。根据这一思想，我们开发出一个方法能够从一组虚拟的人脸图像上自动地测定各矩形的组件。算法开始于一个小矩形的组件，它定位在脸部预先挑选的点（例如左眼的中心）周围。注意，我们可以在所有的人脸图像上定位相同的面部点，因为我们掌握三维头模型间点对点的一致关系。组件从所有的虚拟人脸图像中被提取出来去建立一个正例训练集。我们也生成一个非人脸模式的训练

集，它有与组件同样的矩形形状。在组件数据上训练了 SVM 之后，我们根据估计的预期错误率的上界来估计 SVM 的性能。根据等式(3)我们计算：

$$\rho = \frac{D^2}{M^2}, \quad (4)$$

这里 D 表示包含支持矢量的特征空间 IR^p 中的最小区域³的直径， M 是等式(2)给出的分类间隔。在测定 ρ 之后，我们来对组件进行扩大，这通过向（上，下，左，右）四个方向之一将矩形拓展一个像素来实现。然后，我们再次产生训练数据，训练SVM并测定 ρ 。我们向所有四个方向反复这样做并且最终保留使 ρ 减少最多的那个拓展。这个过程持续进行直到向所有四个方向的拓展导致一个 ρ 的增量。在我们的实验中，我们开始于 14 个 5×5 大小的种子区域，它们大多分布在眼睛，鼻子和嘴附近。图 6 展示了组件生长之后的结果；表 1 给出了各组件的大小。

Components	Width	Height
Eyebrows	19	15
Eyes	17	17
Between eyes	18	16
Nose	15	20
Nostrils	22	12
Cheeks	21	20
Mouth	31	15
Lip	13	16
Corners of the mouth	18	11

表 1 被学习的各个组件的大小

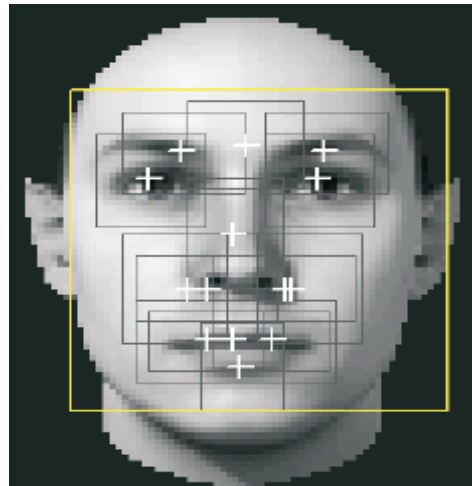


图 6 十四个学习的组件。十字形标志标明各组件的中心。

³在我们的实验中，我们用特征空间的维数 p 代替等式(4)中的 D^2 。这是因为我们的数据点处于一个长度为 1 的 p 维立方体中，所以包含数据的最小区域的半径等于 $\sqrt{p}/2$ 。这一近似值主要是因为计算的原因，因为为了计算 D 我们需要解决一个优化问题[8]。

5. 实验

在我们的实验中，我们将基于组件的系统与一个在整体人脸模式上受过训练的分类器进行了比较。这里，基于组件的系统包括 14 个用作组件检测的线性 SVM 分类器以及一个独立的作为几何结构分类器的线性 SVM。而整体人脸分类器是一个单独的在整体人脸模式的灰度值上训练过的线性 SVM。这两者的分类器的训练数据包括 2,457 幅虚拟的灰度人脸图像和 13,655 幅 58×58 像素大小的非人脸灰度图像。

正例测试包括 1,834 幅人脸图像，它们是经过在 -30° 到 30° 之间进行深度方向上的旋转得到的。这些人脸是从 CMU PIE 数据库[14]中人工提取的。相对的，反例测试集包括 24,464 幅有难度的非人脸模式，它们是一个快速人脸检测器[5]从网络图像⁴中收集来的。在此基础上，我们相对于非人脸测试图像的数量来计算错误的肯定(FP)比率。图 7 展示了 SVM 整体人脸分类器（线性核和多项式核）与一个基于组件分类器之间的对比情况，其中这个基于组件分类器是由 14 个线性 SVM 组件分类器及 1 个线性 SVM 几何结构分类器组成的。这时，作为基准，我们还添加了一条 ROC 曲线，表示在 19×19 像素大小的真实人脸图像上训练过的第二角度多项式内核的 SVM。文献[3]中详细地描述和评估了这个人脸检测器，它是在包括正面和近似正面人脸图像的 CMU 测试集[10]中运行最好的人脸检测系统之一。可以这样说，基于组件的系统优于所有的整体人脸系统。图 8 显示了基于组件系统产生的一些检测结果。

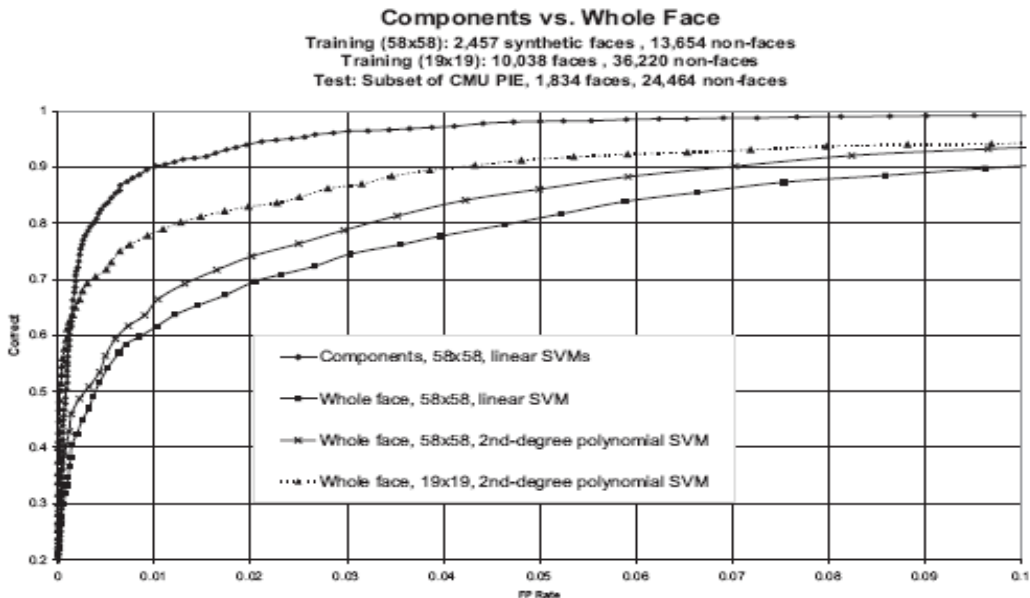


图 7 分别表示整体人脸分类器和 14 个组件分类器的 ROC 曲线。

⁴测试数据库和实验的一个详细的描述[4]可以在 MIT/CBCL 网页上找到。

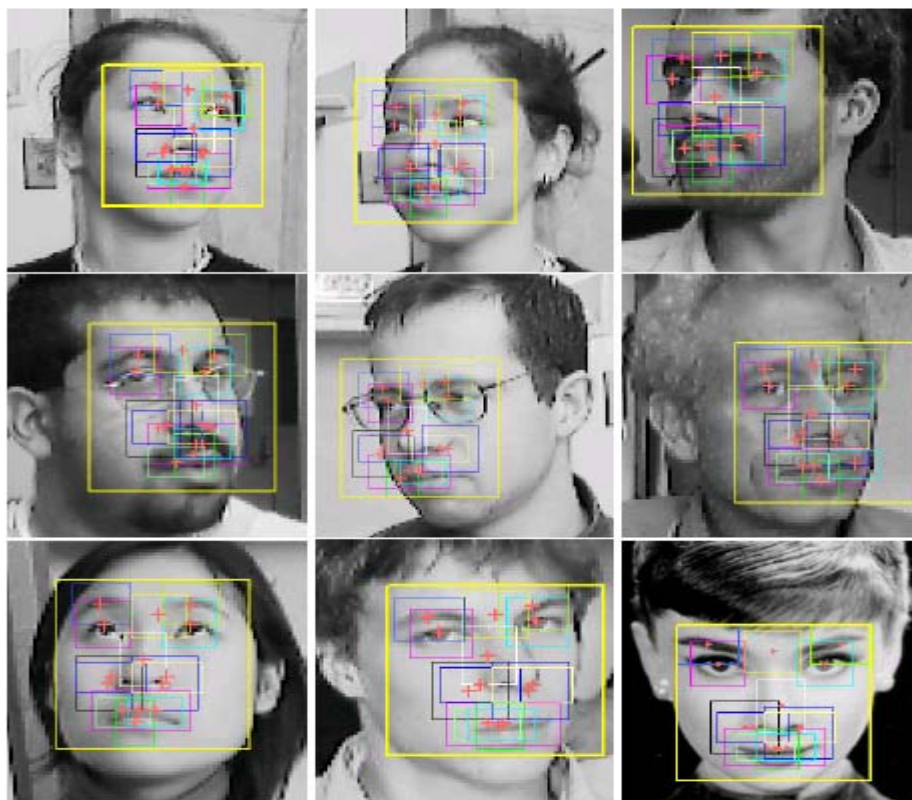


图 8 被包含 14 个组件的系统检测的人脸。

6. 总结

我们提出一个使用 SVM 分类器的基于组件的人脸检测系统。系统用一个两级分类器实现检测功能。在第一级，各个组件分类器独立地检测人脸的各个部分。在第二级，几何结构分类器组合各组件分类器的结果并执行最后的检测步骤。在真实人脸图像上的实验表明，相比于整体人脸检测系统，我们提出的系统在分类性能上有了显著的提高。我们还提出了包括来源于 SVM 理论的各种策略的一个区域生长的方法，能够从一组三维头模型中学习相关的各个组件。其中，三维头模型的使用使得我们能够自动地提取各组件并任意地改变光照和视角。而且，基于组件的分类系统和组件学习的技术可以应用于计算机视觉中的其他物体检测工作。

致谢

作者十分感谢提供了三维模型的 V. Blanz 和 T. Vetter。这项研究是由 DARPA 通过编号为 N00014-00-1-0907 的合同以及国家科学基金会通过编号为 IIS-9800032 的合同部分赞助的。其他的支持是由 the DFG, Eastman Kodak, Compaq 和 Honda 等提供的。

参考文献：

- [1]D. J. Beymer. Face recognition under varying pose. AI Memo 1461, Center for Biological and Computational Learning, M.I.T., Cambridge, MA, 1993.
- [2]R. Brunelli and T. Poggio. Face recognition: Features versus templates. IEEE Transactions on Pattern Analysis and Machine Intelligence, 15(10):1042–1052, 1993.
- [3]B. Heisele, T. Poggio, and M. Pontil. Face detection in still gray images. AI Memo 1687, Center for Biological and Computational Learning, MIT, Cambridge, MA, 2000.
- [4]B. Heisele, T. Serre, S. Mukherjee, and T. Poggio. Feature reduction and hierarchy of classifiers for fast object detection in video images. In Proc. IEEE Conference on Computer Vision and Pattern Recognition, Hawaii, 2001.
- [5]B. Heisele, T. Serre, and T. Poggio. Component-based face detection. AI Memo, Center for Biological and Computational Learning, MIT, Cambridge, MA, 2001.
- [6]T. K. Leung, M. C. Burl, and P. Perona. Finding faces in cluttered scenes using random labeled graph matching. In Proc. International Conference on Computer Vision, pages 637–644, Cambridge, MA, 1995.
- [7]A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(4):349–361, 2001.
- [8]E. Osuna. Support Vector Machines: Training and Applications. PhD thesis, MIT, EE/CS Dept., Cambridge, MA, 1998.
- [9]T. D. Rikert, M. J. Jones, and P. Viola. A cluster-based statistical model for object detection. In Proc. IEEE Conference on Computer Vision and Pattern Recognition, volume 2, pages 1046–1053, Fort Collins, 1999.
- [10]H. A. Rowley, S. Baluja, and T. Kanade. Rotation invariant neural network-based face detection. Computer Science Technical Report CMU-CS-97-201, CMU, Pittsburgh, 1997.
- [11]H. A. Rowley, S. Baluja, and T. Kanade. Neural network based face detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(1):23–38, 1998.
- [12]H. Schneiderman and T. Kanade. Probabilistic modeling of local appearance and spatial relationships for object recognition. In Proc. IEEE Conference on Computer Vision and Pattern Recognition, pages 45–51, Santa Barbara, 1998.
- [13]H. Schneiderman and T. Kanade. A statistical method for 3d object detection applied to faces and cars. In Proc. IEEE Conference on Computer Vision and Pattern Recognition, pages 746–751, 2000.
- [14]T. Sim, S. Baker, and M. Bsat. The cmu Pose, Illumination, and Expression (PIE) database of human faces. Computer Science Technical Report 01-02, CMU, Pittsburgh, 2001.

- [15]K.-K. Sung. Learning and Example Selection for Object and Pattern Recognition. PhD thesis, MIT, AI Lab, Cambridge, MA, 1996.
- [16]V. Vapnik. Statistical learning theory. JohnWiley and Sons, New York, 1998.
- [17]T. Vetter. Synthesis of novel views from a single face. International Journal of Computer Vision, 28(2):103–116, 1998.
- [18]L. Wiskott, J. M. Fellous, N. Kruger, and C. von der Malsburg. A statistical method for 3d object detection applied to faces and cars. In Proc. IEEE International Conference on Image Processing, pages 129–132, 1997.6

致 谢

本论文是在我的导师何英华老师的精心指导和亲切关怀下完成的。值此论文完成之际，我对何老师在百忙之中对我的辛勤培养和教导表示衷心的感谢！在做毕业设计期间，何老师一直关心课题的进展，并提出了很多意见和建议，使我对课题的认识不断深入，为我打开了通向科研的一扇大门。

在论文的完成过程中，同在一个课题小组的同学对我的工作提出了许多宝贵的建议，他们在论文和编程方面给了我很多的指点和帮助，在此深表谢意。还要感谢与我同宿舍的同学，多次充当“模特”给我拍摄，使我获得充足的测试样本，能顺利地进行人脸检测系统的测试。

同时感谢我的父母、家人，感谢他们对我的学业、生活给予的关心和支持！

最后感谢天津大学四年来对我的教育和栽培，今后无论本人从事什么工作，必定会尽心尽力，做出成绩，为母校争光！