

分类号 TP3 \_\_\_\_\_

密级 \_\_\_\_\_

UDC \_\_\_\_\_

编号 \_\_\_\_\_

# 中国科学院研究生院

## 博士学位论文

基于瀑布型分类器的人脸检测研究

闫胜业

指导教师 高文 教授

中国科学院计算技术研究所

申请学位级别 工学博士 学科专业名称 计算机应用技术

论文提交日期 2009年4月 论文答辩日期 2009年6月

培养单位 中国科学院计算技术研究所

学位授予单位 中国科学院研究生院

答辩委员会主席 卢汉清

## 声 明

我声明本论文是我本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，本论文中不包含其它人已经发表或撰写过的研究成果。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

作者签名：

日期：

## 论文授权使用授权书

本人授权中国科学院计算技术研究所可以保留并向国家有关部门或机构送交本论文的复印件和电子文档，允许本论文被查阅和借阅，可以将本论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编本论文。

（保密论文在解密后适用本授权书。）

作者签名：

导师签名：

日期：



## 摘 要

人脸检测肩负着在一幅输入图像中搜索并确定人脸是否存在以及如果存在，确定人脸位置和大小任务。本文针对当前主流的基于统计学习和瀑布型分类器检测框架的人脸检测方法，在收集到更大规模样本集的基础上，重新对当前主流的瀑布型分类器检测和训练技术进行了系统分析。为了构建面对更加复杂样本时速度更快、精度更准的检测器，对主流瀑布型分类器训练和使用方法进行一系列的改进。这些方法可以概括为三个方面：降低瀑布型分类器的训练负担并提高检测精度，提高瀑布型分类器的检测速度，优化瀑布型分类器的速度\精度的折中调节算法。

首先，为了解决存在大规模训练人脸样本集和非人脸样本集的情况下，瀑布型分类器训练所需要的时间代价过高的问题，提出了训练过程中对正例样本和反例样本交叉自举的矩阵式学习算法。简单说来，对于瀑布式分类器，矩阵式学习并不仅在相邻的子分类器间对巨大的反例样本集进行“自举”，在每个子分类器的学习过程中，对大规模正例样本集也进行自举以降低学习中直接使用的训练正例样本集大小。当然，矩阵式学习并不只是简单的对正例样本和反例样本进行自举，针对矩阵式学习中的交叉自举，进一步提出了累加式样本自举方法和特征继承技术。在自举过程中，相邻的子分类器的共享信息被用来进行更加快速的学习，这些共享信息包括两个方面，分类器中已经学到的特征和已经自举得到的样本。

其次，提出一种更加高效的非人脸窗口预排除方法，进一步提高了人脸检测的速度，这包括了一种新的特征——局部组合二值特征，以及相应的以特征为中心的瀑布型检测算法。在提出的非人脸窗口预排除方法中，特征计算方式更加简单，避免可能重复的特征计算。局部组合二值特征的基本单元是二值特征，二值特征是图像上相邻区域间灰度差异的二值信息，即像 Haar 特征那样的多个区域间对比的更“黑”更“白”关系。局部组合二值特征是组合二值特征的一个有效子集，而组合二值特征表示的是多个不同二值特征值共生的信息。在局部组合二值特征中，组合的方式被限制在类似局部二值模式组合像素点时的方式，形成局部组合二值特征。局部组合二值特征数目少，描述能力强，被用来进行预分类器的学习。预分类器采用了一种类似“注意机制”的以特征为中心的瀑布型分类器运行技术，以特征为中心的运行机制首先针对某个显著特征模板对图像每个位置计算出特征值，构成特征图像，然后在穷举窗口上应用学习得到的本特征模板在各位置的统计信息，并进一步判断窗口为人脸或者非人脸。无论是赖以计算的特征模板还是窗口内特征模板的统计信息都使用学习的方法得到，具体来说，最终采用的模板是多个候选得到的最优模板，统计信息是 Boosting 学习得到的瀑布型分类器。

最后，提出一种基于搜索的瀑布型分类器精度\速度的折中算法。基于一个已有的 Boosting 瀑布型分类器，提出的方法能够构建多个速度标准下的瀑布型分类器，使得人

脸检测算法在应用中能够根据不同速度标准获得更优的分类性能。构建算法对构成瀑布型分类器的强分类器阈值空间进行搜索，以获得计算代价更优的瀑布型分类器。为了降低搜索的计算代价，搜索开始于正例分类正确率为百分之百的瀑布型分类器，以后每一轮的运行在当前瀑布型分类器基础上，搜索进一步损失小部分正例分类正确率条件下所有可能的瀑布型分类器，并获得当前计算代价最小的瀑布型分类器。然后，在当前获得的瀑布型分类器基础上，迭代运行，直到获得的瀑布型分类器正例分类正确率达到用户指定的正例分类正确率。

本文提出的方法在包括正面人脸检测和常用的多姿态范围内的人脸检测问题进行了实验，与别的已经公开发表过的相关工作相比，在已公开的标准人脸测试集上的实验结果表明，本文算法在精度方面具有优势，同时，在一幅  $320 \times 240$  的图像上，多姿态人脸检测可以达到近实时的应用需求。

**关键词：**人脸检测；瀑布型分类器；二值特征；大规模样本集；自举

## **Research on Face Detection Based on Cascaded Classifier**

Yan Shengye (Computer Application)

Directed By Gao Wen

Face detection task includes automatically searching in an input image and deciding whether or not there are faces, if the answer is YES, giving the size and location of each faces further. Provided with an enormous face dataset collected, aiming at finding more statistical results, we revisit the most popular Boosting Cascade method proposed by Viola and Jones. Under new conditions, both detection accuracy and speed need be improved. Seres of methods are proposed to solve the problem. In detail, these solutions are described as follow.

Firstly, aiming at promoting the classification performance of the face detector, we introduce enormous face training set into the learning process of cascaded face detector. To solving the difficulty of very high time cost of training, we proposed a novel training method named Matrix-Structural Learning (MSL) as an extension to Viola and Jones' cascade learning method for object detection. Briefly speaking, unlike Viola and Jones' method that learn linearly by bootstrapping only negative samples, the proposed MSL method bootstraps both positive and negative samples in a matrix-like structure. Moreover, an accumulative way is further presented to improve the training efficiency of MSL by inheriting features learned previously and training sample set bootstrapped previously during training procedure.

Secondly, we describe a novel fast and accurate face detection method. The whole detection procedure is partitioned into two phases. Firstly, a fast pre-classifier with high detection rate is run to reject a large number of simple non-faces. Then a more complex classifier with high detection rate such as the one learned from MSL is run to reject more complex non-faces. The novel techniques include the training and the running of pre- classifier. In the pre-classifier, firstly, a novel type of feature called the binary feature is proposed. The binary features modified Haar features to keep only the ordinal relationship rather than the absolute difference between the accumulated intensities. Further, several neighboring binary Haar features are assembled to capture their co-occurrence similar to the coding of local binary pattern (LBP). The assembled binary feature is called a Locally Assembled Binary (LAB) feature which is used to train the pre-classifier. Secondly, an efficient detection method called feature-centric method is adopted to build an efficient pre-classifier. Different from the feature-centric method, we further cascade the classifier learned by feature-centric method to improve the computation efficiency.

Finally, based on the ready-made Boosting cascade, we study how to build series of faster cascades with different computational cost for practical use. The new cascade is allowed to lose some detection rate. We propose to search the latent cascaded classifiers for the one with minimal computation cost. To unload the computation burden, the search algorithm is run

sequentially forward. The search starts from the cascaded classifier with true positive rate 1. In each round of run, the best cascaded classifier is found out from all latent cascaded classifiers which are built with the same true positive rate a little smaller than the true positive rate of the selected cascaded classifier of the last round. The search algorithm is ended when target true positive rate is attained. Our contributions include three aspects. Firstly, a search-based method is proposed to build cascaded classifier. Secondly, we consider all possible cascaded classifier in the search-based method. Finally, different computational costs of different type of features in the cascaded classifier are considered in the evaluation of the computational costs of the cascaded classifiers.

To investigate the proposed method, experiments are committed on both frontal face detection and multi-view face detection. The experimental results on several public standard face test sets show that the classification accuracy of the proposed methods are comparable or better than the methods reported by others as to our knowledge. At the same time, the face detection system based on the proposed method has a nearly real-time detection speed on multi-view face detection.

**Keywords:** face detection, cascaded classifier, binary feature, enormous training set, bootstrap

# 目 录

摘 要.....	I
目 录.....	V
图目录.....	IX
表目录.....	XIII
<b>第一章 引言.....</b>	<b>1</b>
1.1 人脸检测的研究背景、内容及意义.....	1
1.2 人脸检测面临的困难 .....	3
1.3 人脸检测的研究现状 .....	5
1.4 本文研究问题的提出 .....	18
1.5 本文的主要贡献及章节安排 .....	19
<b>第二章 人脸检测数据集制备及检测性能评估 .....</b>	<b>23</b>
2.1 数据集收集.....	23
2.2 人脸样本的制作 .....	24
2.2.1 人脸样本对齐 .....	24
2.2.2 人脸样本生成.....	26
2.2.3 人脸样本的人工净化 .....	31
2.3 反例数据集.....	34
2.4 人脸检测算法的性能评估 .....	36
2.5 小结 .....	37
<b>第三章 面向大规模训练样本集的矩阵结构瀑布型分类器学习 .....</b>	<b>39</b>
3.1 Boosting 瀑布型分类器检测框架基础 .....	39
3.2 面向大规模训练样本的矩阵式学习.....	40
3.2.1 自举正例样本 .....	42
3.2.2 累加式样本自举方法 .....	45
3.3 讨论：另一种计算复杂度高的算法.....	45
3.4 特征继承 .....	47
3.5 实验 .....	48
3.5.1 实验设置 .....	48

3.5.2 正面人脸检测 .....	49
3.5.3 多姿态人脸检测 .....	52
3.6 总结 .....	53
<b>第四章 局部组合二值特征及快速预处理分类器 .....</b>	<b>61</b>
4.1 由粗到细的人脸检测系统框架 .....	61
4.2 局部组合二值特征 .....	63
4.2.1 二值 Haar 特征 .....	63
4.2.2 组合二值 Haar 特征 .....	64
4.2.3 局部组合二值特征 .....	65
4.3 使用局部组合二值特征进行人脸检测 .....	66
4.3.1 以特征为中心的检测方法 .....	66
4.3.2 以特征为中心的瀑布型预分类器 .....	69
4.3.3 以窗口为中心的深入处理分类器 .....	70
4.3.4 多姿态人脸检测上的应用 .....	71
4.4 实验配置和实验结果 .....	71
4.4.1 实验配置 .....	72
4.4.2 局部组合二值特征在正面人脸上的分辨能力调查 .....	72
4.4.3 正面人脸检测 .....	73
4.4.4 多姿态人脸检测器 .....	75
4.5 小结 .....	76
<b>第五章 基于搜索的瀑布型分类器构建方法 .....</b>	<b>83</b>
5.1 Boosting 瀑布型分类器中的特征 .....	84
5.2 精度指定条件下瀑布型分类器速度优化 .....	86
5.2.1 瀑布型分类器的计算复杂度 .....	86
5.2.2 不同检测精度计算复杂度优化 .....	87
5.2.3 扩展：基于每个弱分类器的瀑布型分类器调节算法 .....	90
5.2.4 反例样本集重采样 .....	91
5.3 实验 .....	91
5.3.1 输入瀑布型分类器 .....	91
5.3.2 正面人脸检测 .....	91
5.3.3 多姿态人脸检测 .....	92
5.4 小结 .....	93
<b>第六章 人脸检测原型及系统实现 .....</b>	<b>95</b>
6.1 动态链接库 JFD 模块实现 .....	95

6.2 多姿态人脸检测视频演示系统.....	96
6.3 小结 .....	97
<b>第七章 总结与展望.....</b>	<b>99</b>
7.1 总结 .....	99
7.2 展望 .....	101
<b>参考文献.....</b>	<b>103</b>
<b>致 谢.....</b>	<b>i</b>
<b>作者简介.....</b>	<b>iii</b>



## 图目录

图 1.1 人脸检测在相关研究学科中的地位 .....	2
图 1.2 人脸检测结果图示 .....	3
图 1.3 自然图片中的“疑似人脸”示例 .....	4
图 1.4 人脸检测中常用的金字塔式搜索 .....	5
图 1.5 基于“穷举窗口搜索”策略的人脸检测系统流程图 .....	6
图 1.6 Sung 等人建立的人脸和非人脸簇[SP98] .....	8
图 1.7 Rowley 等人建立的人脸检测框架[RBK98b] .....	9
图 1.8 Viola 等人提出的 Haar 特征 .....	10
图 1.9 积分图及其矩形内像素灰度值的快速计算示意图 .....	10
图 1.10 Barlett 等人提出的扩展 Haar 特征[BLFM03] .....	11
图 1.11 Lienhar 等提出的旋转 Haar 特征[LM02] .....	11
图 1.12 武等提出的扩展 Haar 特征[武 05] .....	12
图 1.13 Li 等提出的扩展 Haar 特征[Li02] .....	12
图 1.14 Huang 等提出的扩展 Haar 特征[HAL06] .....	12
图 1.15 弱分类器的类型 (a) 刀切式阈值型 (b) 均匀划分的分段信度函数 (c)非等区间 阈值型函数 .....	14
图 1.16 瀑布式分类器 .....	17
图 1.17 嵌入式瀑布型分类器 .....	17
图 2.1 本文使用的 15 种多姿态人脸类别 .....	24
图 2.2 不同研究者使用的正面人脸训练样本实例 .....	25
图 2.3 本文使用的正面样本剪切方案 .....	25
图 2.4 正面竖直和人脸样本示例 .....	27
图 2.5 正面左倾斜人脸样本示例 .....	27

图 2.6 正面右倾斜人脸样本示例 .....	27
图 2.7 左半侧垂直人脸样本示例 .....	28
图 2.8 左半侧左倾斜人脸样本示例 .....	28
图 2.9 左半侧右倾斜人脸样本示例 .....	28
图 2.10 右半侧垂直人脸样本示例 .....	28
图 2.11 右半侧左倾斜人脸样本示例 .....	29
图 2.12 右半侧右倾斜人脸样本示例 .....	29
图 2.13 左全侧垂直人脸样本示例 .....	29
图 2.14 左全侧左倾斜人脸样本示例 .....	29
图 2.15 左全侧倾斜人脸样本示例 .....	30
图 2.16 右全侧垂直人脸样本示例 .....	30
图 2.17 右全侧左倾斜人脸样本示例 .....	30
图 2.18 右全侧右倾斜人脸样本示例 .....	30
图 2.19 正面人脸样本集中不期望出现的和期望出现的一些样本 .....	31
图 2.20 人工净化样本工具之打开文件 .....	32
图 2.21 人工净化样本工具之直方图均衡化显示按钮 .....	33
图 2.22 人工净化样本工具之直方图均衡化显示 .....	33
图 2.23 人工净化样本工具之基本信息显示 .....	34
图 2.24 去掉用于人脸图像后的用于收集非人脸样本的部分图片示例 .....	35
图 2.25 正面人脸正确检测窗口所在的区域（阴影显示） .....	37
图 3.1 Real AdaBoost 学习算法 .....	41
图 3.2 本章中用到的符号及涵义 .....	42
图 3.3 Sung 等人提出的自举算法 .....	43
图 3.4 子分类器学习中的正例样本自举 .....	44
图 3.5 矩阵式学习 .....	44

图 3.6 线性结构学习.....	46
图 3.7 使用特征继承技术后第 $i$ 个子分类器的特征构成.....	48
图 3.8 本文实验使用的五种 Haar 特征类型 .....	48
图 3.9 重新挑选特征数目对比 .....	51
图 3.10 两种方法对已有特征获得的分类器分类判别能力对比.....	51
图 3.11 矩阵式学习在 CMU+MIT 正面人脸测试集上的 ROC 曲线.....	52
图 3.15 CMU 侧面人脸测试集上的 ROC 曲线.....	53
图 3.12 矩阵式学习正面人脸检测结果图片 1 .....	54
图 3.13 矩阵式学习正面人脸检测结果图片 2 .....	55
图 3.14 矩阵式学习正面人脸检测结果图片 3 .....	56
图 3.16 矩阵式学习多姿态人脸检测结果图片 1 .....	57
图 3.17 矩阵式学习多姿态人脸检测结果图片 2 .....	58
图 3.18 矩阵式学习多姿态人脸检测结果图片 3 .....	59
图 4.1 两阶段瀑布型分类器 .....	62
图 4.2 一个特征被多个窗口共享示意图 .....	63
图 4.3 组合多个二值特征 .....	65
图 4.4 组合使用的二值 Haar 特征.....	65
图 4.5 局部组合二值特征 .....	66
图 4.6 以特征为中心的检测方法 .....	68
图 4.7 为以特征为中心的分类器挑选尺度的过程.....	69
图 4.8 多姿态人脸检测流程 .....	71
图 4.9 CMU+MIT 正面人脸测试集上的 ROC 曲线.....	73
图 4.10 CMU+MIT 正面人脸测试集上的性能曲线 .....	74
图 4.11 CMU 侧面人脸测试集上的 ROC 曲线.....	76
图 4.12 使用局部组合二值特征预检后的结果图片 1 .....	77

图 4.13 使用局部组合二值特征预检后的结果图片 2.....	78
图 4.14 使用局部组合二值特征预检后的结果图片 3.....	79
图 4.15 使用局部组合二值特征预检后的结果图片 4.....	80
图 4.16 使用局部组合二值特征预检后的结果图片 5.....	81
图 5.1 两种解决应用中不同速度需求的方案。(a)训练多个分类器，(b)训练一个高性能分类器，并调节得到多个速度下的分类器参数.....	83
图 5.2 Boosting 瀑布型分类器.....	84
图 5.3 部分样本的信度曲线.....	87
图 5.4 构建一个正例测试集上检测率为 1 的瀑布型分类器算法.....	88
图 5.5 构建损失指定个数正例样本情况下的瀑布型分类器算法.....	89
图 5.6 集成分类器到瀑布型分类器.....	90
图 5.7 CMU+MIT 正面人脸测试集上的 ROC 曲线.....	92
图 5.8 CMU 侧面人脸测试集上的 ROC 曲线.....	93
图 6.1 视频演示系统界面.....	96
图 6.2 视频演示系统之检测姿态范围选择.....	97

## 表目录

表 3.1 一些分类器最终自举得到的人脸训练集大小.....	49
表 3.2 累加自举方法和原始自举方法达到相同性能指标需要的自举迭代总次数对比 .....	49
表 3.3 第 15 个子分类器自举过程中, 不使用特征继承技术, 训练各子分类器需要新添加特征的数目和相应的训练样本数目 .....	49
表 3.4 第 15 个子分类器自举过程中, 使用特征继承后, 训练各子分类器需要新添加特征的数目和相应的训练样本数目 .....	50
表 4.1 3×3 大小局部组合二值特征的分类器对每个特征处设置层级, 检测率设定为 0.99984, 误检率的变化列表 .....	70
表 4.2 等分类错误率情况下不同尺度模板局部组合二值特征分类器的特征数目 .....	72
表 4.3 3×3 尺度下局部组合二值特征集成分类器在对每个特征处设置层级, 检测率设定为 0.99984, 误检率的变化情况 .....	73
表 4.4 后处理分类器(无预分类器)和及其加上预处理分类器之后(预分类器)的检测结果对比.....	74
表 4.5 正面人脸检测器在一个自选测试集上的测试结果.....	74



# 第一章 引言

## 1.1 人脸检测的研究背景、内容及意义

自人类诞生之日起，为了与客观世界做斗争，人类发明了各种各样的生产生活工具来提高自身改造客观世界的能力。人类使用棍棒、石头以及各种交通运输工具等等来延伸人的四肢，发明了重量、长度的测量工具来对客观世界进行感知，发明了语言文字来进行人类社会内部人与人之间的信息沟通，发明了算盘等计算工具来延伸人类大脑的计算能力，这些发明创造大大提高了社会活动的运作效率。

除了对延伸人类某个局部功能的生产生活工具的追求之外，人类从未放弃对更加智能机器的追求，智能的机器能够模仿人类所拥有的智能，自动进行分析决策并执行，更进一步的解放人类在劳动中的劳力者地位，另外，智能机器还可以在人类不宜的恶劣环境中代替人类进行客观世界的考察、改造等等。但是，在远古时代，这只是一个梦想。

近代科学的发展使得人类对热、光、力、震动等等抽象物质有了更深的认识，人类发明了各种传感设备来延伸人类感知器官的功能，机器感知的出现使得人类能够处理这些抽象的信号。这些抽象信号的感知使得机器有望(部分)代替人的触觉、视觉、听觉等功能。另一方面，在对计算能力的强烈需求环境下，人类发明了计算机。电子计算机的发明，使得人类在计算能力上大大提高，并且电子计算机也很快应用于各种社会事务的处理中，近年，计算机更是几乎深入走进了每个人的生活，我们有幸可以目睹计算机发展的飞快速度。

伴随着客观环境的逐步成熟，智能机器的需求，诞生了相关的一系列学科，而其中一个研究热点就是计算机视觉。作为人工智能的一个重要领域之一，计算机视觉是指用计算机实现人的视觉功能，包括对客观世界的三维场景的感知、识别和理解。计算机视觉研究的基础就是图像。图像就是机器对光信号的感知，图像感知设备包括照相机，摄像机等等。伴随着图像采集设备和互联网的发展，每个人可以拥有的图像数量大大提高。由于计算机视觉具有的潜在应用十分广泛，所涉及的学科知识极其繁多，研究的问题又极富挑战性，因此它一直是计算机学科中的一门热门学科，并吸收了许多从事心理学、神经科学、生理学、生物物理学、数学与计算机学科等各种学科研究人员的关注。

人脸检测是计算机视觉中图像识别的子领域之一，在计算机视觉中，人脸存在着特殊的地位，人脸是人与人之间交流沟通时视觉最关注的部位之一，从人脸上相互交流的人们可以获取表情、年龄、性别、种族等等信息。相应地存在人脸表情识别、年龄识别、性别识别等其它相关的人脸感知研究子领域。而在智能人机交互中，用户是人机交互界面的主体，计算机作为一种“智能体”参与了人类的通讯活动，在处理人机关系时，根据人为中心的思想，应当充分考虑人的因素，因此，智能人机接口要解决的首要问题就

是计算机如何感知人的存在，这是人机交互的前提，脸部是人类携带信息最丰富的部位，是人的重要特征。从认知学的角度讲，人类从视觉通道层次上感知和识别人首要信息载体就是人脸，使计算机具有感知到人是否存在的视觉功能，这项技术就是人脸检测。

人脸检测在人脸相关的感知任务中处于首当其冲的位置，是人脸识别、人脸合成与动画、人脸特征检测等视觉任务的必须环节。人脸检测是人脸识别的第一步，人脸识别在人脸检测获得的人脸子窗口基础上进行进一步的分类判断，并获得人脸的身份信息。早期的人脸识别直接从人脸图像窗口中进行人脸识别，是一种非自动人脸识别。还存在着半自动人脸识别，和全自动人脸识别。半自动人脸识别在人脸检测的基础上人工进行特征点的定位并对齐待识别的人脸窗口来辅助进一步的人脸识别。全自动的人脸识别自动的进行特征点定位并对齐人脸窗口进行人脸识别。人脸检测在相关研究学科中的地位可以参看图 1.1。

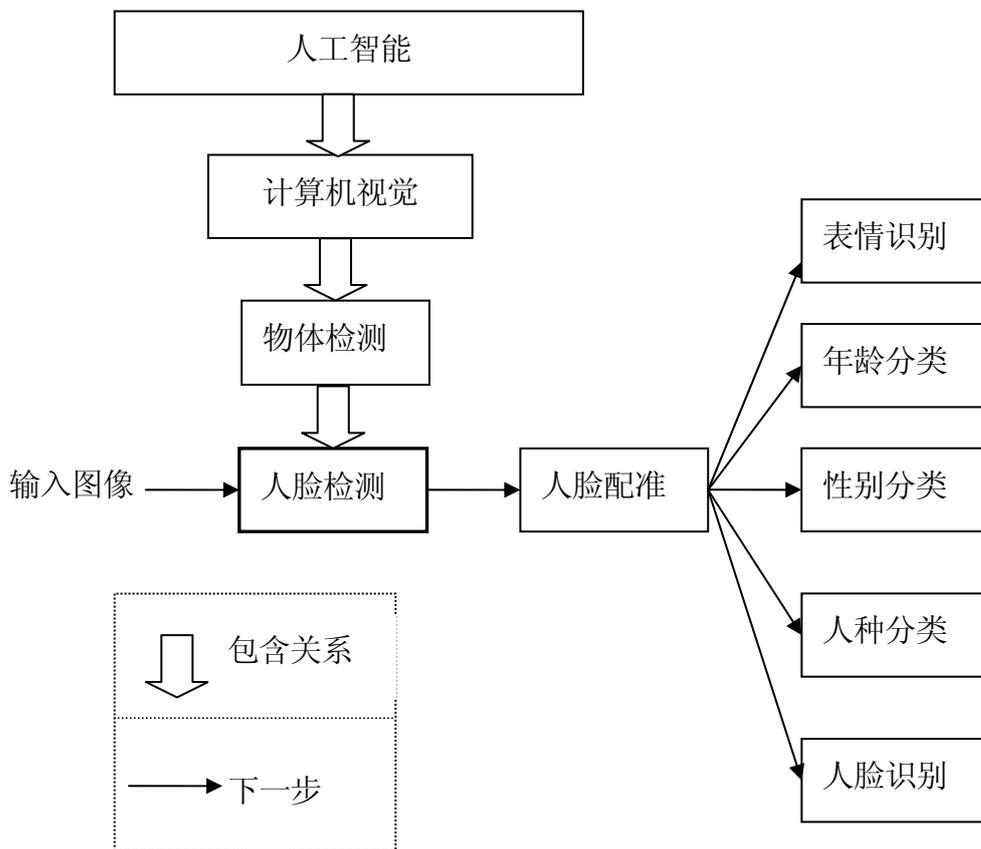


图 1.1 人脸检测在相关研究学科中的地位

当然，在其它的人脸感知任务中，比如人脸表情识别、年龄识别、性别识别、人种分类，人脸检测依然处于一个前提的位置，这些应用都在人脸检测结果的基础上，进一步进行。所以，从某种意义上说，人脸检测结果的准确性对这些视觉问题的最终解决具有很重要的作用。人脸检测是图像处理、统计学习、概率统计理论、特征提取及建模、信号处理等等学科的交叉应用，它们之间存在着千丝万缕的联系。

人脸检测任务的目标是对任意给定的一幅图像或者一组图像序列，使用机器自动判

定该图像或图像序列中是否存在人脸，如果存在人脸的话，找出人脸位置和大小的过程[YKA02]。图 1.1 给出了一些人脸检测的结果，很清楚地解释了人脸检测的任务。



图 1.2 人脸检测结果图示

人脸检测相关的具体应用领域包括可视电话、电视会议、可与人交流的智能计算机、多媒体智能娱乐和游戏、身份认证和安全保卫自动监控系统等。人脸检测横向推广就是物体检测，人脸只是纷繁世界中各种物体的一种，是物体检测的一个应用分支，它与物体检测存在着包含和被包含的关系。物体检测探索如何对客观世界中的物体进行有效的检测，是计算机视觉的一个重要分支。

相对于其它物体的检测问题，人脸模式显得相对简单。从人脸构造上说，人脸是一个近似的刚体。一方面，由于每个人遗传变异不同，人类拥有不同的头骨模型，不同的皮肤、五官等构成人脸的因素。另一方面，遗传所保持一致性，使得人脸又是一个拥有大致规则的物体。所以，在物体检测领域，人脸检测的研究首先引起了研究者们极大的关注。在类似的问题中，一个简单问题的解决往往能够给其它更为复杂问题的解决提供非常多的信息，来帮助其进一步解决。

所以，在过去的十几年中，人脸检测引起了研究者们非常广泛的关注，并成为计算机视觉领域的研究热点之一。

## 1.2 人脸检测面临的困难

人脸检测的输入是图像，图像是光感设备对三维景物的二维感知，外界光源发出的光线照射于人脸之上，经过人脸表面反射后，由光感设备记录每个相应二维位置上光感而获得了最终的图像。人类面部本身是一个附于头颅骨的生理器官，检测人脸所遇到的困难主要来自于人脸的个体差异、光照条件的不同、成像条件的不同等等。另外，由于三维的物体在空间中与成像设备之间的角度不同，造成了多个视角下或者是姿态下不同的人脸。总的说来，人脸检测遇到的困难可以归结为：

- 内部因素：由于遗传生长等条件的不同，不同人的脸会呈现出差异，这包含了每个部位，比如肤色、眼睛、鼻子、嘴等的基本形状都会呈现出差异。这些差异会直接给人脸检测带来困难；

- 光照及成像条件：光源的方向、强度等都会使得人脸在图像中呈现出差异。另外，光线传播过程中的遮挡，也会带来面部图像的差异，光照始终是视觉问题中的一个比较重要的问题。成像设备的不同，比如各种成像质量参差不齐的扫描仪、监控摄像头等会带来低质量的面部图像，另外，成像设备的焦距、光圈、快门速度和畸变等不同都会使得最终的人脸图像存在较大的差异；
- 多视角人脸：由于成像设备与人脸都可以移动，其二者位置关系变化带来视角和遮挡关系的变化，从而导致采集到的人脸图像呈现出很大差异，甚至呈现出完全不同的模式，从而产生了所谓的多姿态人脸检测问题。
- 速度\精度两难问题：计算机有限的计算访问速度和存储空间会给人脸检测带来很大的困难。在实际环境中，由于无法预知人脸在图像中出现的位置和大小，人脸检测通常需要对所有可能出现的人脸尺度和位置进行穷举搜索，这需要很大的计算量。以  $320 \times 240$  大小的图片为例，一般要穷举十万多个候选区域。另一方面，非人脸模式规模更为巨大，且其中存在着大量“疑似人脸”，这批“疑似人脸”也是人脸检测需要做出正确决策的候选，这就要求花费更多的计算代价、获得更多的细节信息以检测人脸。图 1.3 给出了一些“疑似人脸”示例。



图 1.3 自然图片中的“疑似人脸”示例

### 1.3 人脸检测的研究现状

人脸感知研究始于 20 世纪 60 年代，早期的研究大多集中在人脸识别问题上，直到 80 年代末，90 年代初，人们才把注意力转向人脸检测问题上来。最初人们的研究主要着重于静态图像中的人脸检测，如照片中自动定位人脸等。随着智能视频监控、电视会议等应用需求的增强，动态人脸检测的研究逐步成为人脸感知领域的研究热点之一。在 2001 年以后的几年里曾经达到一个研究高峰，最近几年随着人脸检测的精度和速度逐步提高，并逐步进入产业化阶段，研究热度又略有下降。但是比人脸检测更为困难的其它物体检测比如人体检测吸引了更多研究者的注意。

从图像中搜索出人脸的方式来讲，人脸检测中存在两种搜索人脸区域的方法。一种方法采用了类似于人类“注意”机制的方法，此类方法对人脸的搜索并不是从直接搜索图像子窗口开始的，而是从搜索显著（salient）特征开始，搜索过程中，当图像某个区域中没有出现显著特征，就认为此图像区域没有人脸出现，放弃进一步的人脸判别；而当图像某个区域中出现了显著特征，则认为出现显著特征的位置可能存在人脸，再在这些拥有显著特征的区域内进一步对显著特征构建人脸概率模型，并进一步判断人脸窗口的存在。显然，为了稳定的搜索出人脸区域，显著特征要能够稳定地搜索获得，否则将漏检某些人脸，这要求显著特征必须能够对几何变换、光照变化等保持不变性质。目前看来，寻找这种“不变特征”具有很大的挑战。人脸检测中基于肤色的方法都可以划归为此类方法[YA98, 陈 03, HAJ02]，另外，比如 Miao 等人提出了基于边缘特征的变形模板匹配的方法[MWSC99]。但是很遗憾，现在看来寻找稳定的不变特征依然任重道远，比如，肤色，就会因为光照的变化而变得不稳定。尽管如此，从物体检测的角度来说，如果要对场景中的更多数量的物体进行检测，此方法依然是一个很有前途的方法，因为方法中各个物体共享使用的不变特征，可以大大提高物体检测的效率，在待检测物体种类比较多时优势尤其明显。从目前对人类识别客观世界场景中各种类别物体的识别过程研究来看，人类自身很可能就采用了此种物体识别方法。

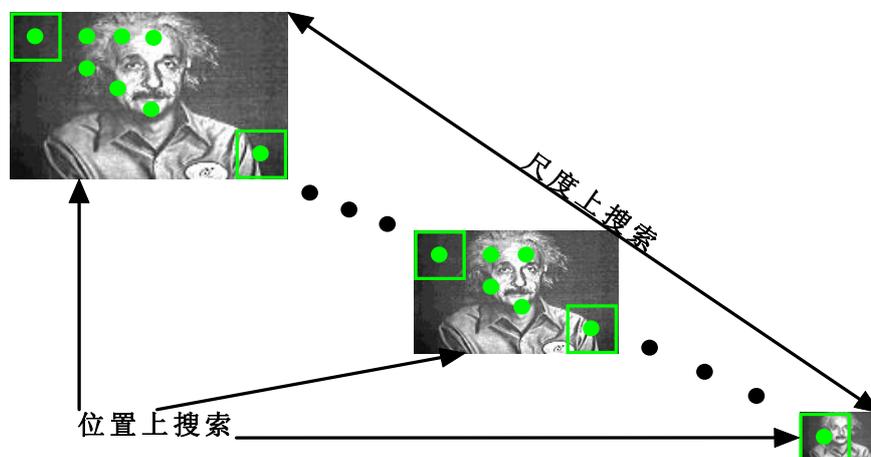


图 1.4 人脸检测中常用的金字塔式搜索

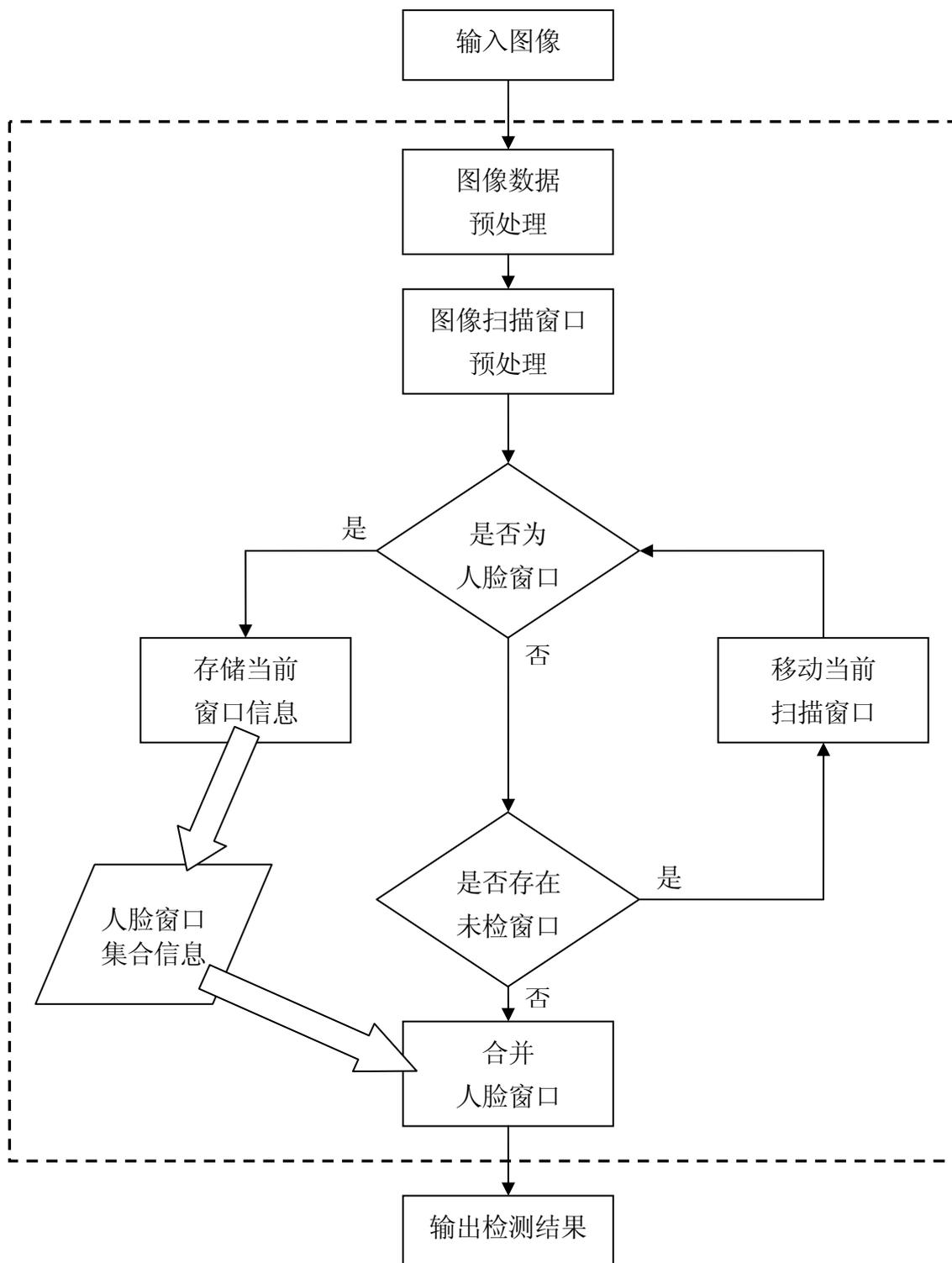


图 1.5 基于“穷举窗口搜索”策略的人脸检测系统流程图

虽然人类识别过程采用不变显著特征的方法获得了成功，但是在自动物体识别领域始终没有发现稳定的显著特征，能够进行高性能、低漏检的物体检测。另一种搜索图像中人脸的方法获得了相当的成就，叫做“穷举窗口搜索”方法。与采用显著特征搜索来进行候选人脸窗口判断不同，此类方法首先将图像中的每个位置穷举出相同大小的候选

人脸窗口，在每个窗口上应用预先建立的用以区分人脸和非人脸的分类器进行人脸区域的判断。为了应付不同尺度的人脸，穷举窗口搜索的过程不仅应用于原始输入图像上，还应用在各种缩小或者放大后的原始图像上。穷举窗口搜索的过程参见图 1.4，图中矩形窗口即为穷举过程中的候选人脸窗口，各个尺度下的图像从左上到右下逐步缩小排列，并在每个尺度图像上应用穷举窗口过程。另外，基于此类方法的人脸检测系统的流程图见图 1.5 所示。目前基于穷举窗口的方法在人脸检测中占据了更为重要的地位，被广泛应用，本文提出的方法就建立在这种框架之下。

从模式分类方法的角度来讲，为了对每个候选区域做出正确的模式判断，人脸检测方法可以粗略的划分为基于规则的方法和基于学习的方法。基于规则的方法在文献 [YKA02, Hje01, LJ04, 梁 02] 中有比较详细的介绍，本小节将简单回顾一些相关的方法。

基于规则的方法主要出现在人脸检测研究的早期。通过人为的归纳出一些由人类总结出来的规则并使用计算机进行描述，以用来区分人脸和非人脸模式。举例来说，可能的规则有“人脸可以分割得到是个具有大致一致亮度的单元”，“人的头部是一个近似的椭圆”，“人的双眼和嘴部构成等边三角形”，“如果将正面人脸向双眼方向的求每个位置的灰度和，那么从脸下部到脸上部，这个灰度和的变化会呈现出一定的规律，在嘴巴、鼻子、眼睛部位会出现山谷”等等 [YH94, Kan73, KP97]。基于规则的方法具有简单、容易理解的特点。Kim 等提取边缘特征，利用建立的椭圆模板进行椭圆匹配来检测人脸 [KKSP00]。Wang 等对椭圆模板进行了改进 [WT00]，采用一个椭圆环模型来代替椭圆，并加入了边缘的方向特征，实现了简单背景的人脸检测。Fröba 采用了基于 EOM (Edge Orientation Matching) 的方法 [FE03]，利用边缘强度信息建立一个 EOM 模板，在检测时进行模板匹配。Miao 等人提出了基于变形模板匹配的方法 [MWSC99]，由于特征计算简单，速度也很快，他们还将该方法用来进行多姿态的人脸检测，取得了较好的效果。此外，类似的方法还有 [刘 97] 等。

所有这些基于规则的方法都需要根据人的先验知识来人为设计人脸的模板但是由于人脸的多变性，如何设计鲁棒、普适的人脸规则非常具有挑战性。目前看来，人类能够描述的关于人脸的特征和规律相对于千变万化的非人脸来说是极其有限的，从某种意义上说，依靠人类来总结人脸模式所呈现的所有规律并转换为计算机的表达方式是不可能的。所以，基于规则的方法缺乏通用和稳定的区分性，往往不能胜任在实际环境中复杂场景下的人脸检测任务：由于无法完全描述的规则造成人脸模式匹配上造成了漏检，或者由于并不完全正确的规则造成非人脸判断为人脸的误检。

与基于规则的方法不同，基于学习的方法在进行模式分类的时候，采用的是关于人脸模式的某种概率模型。蕴含在这些概率模型中的关于人脸模式的判定规则并非显式地由研究人员描述得到，而是来源于大量的人脸和非人脸样本在表观模式上体现出来的统计规律。对于这些统计得到的规律，有很多是无法描述、难以理解的。正是由于这一特点，基于学习的方法的性能取决于统计学习算法的学习能力以及训练数据对于实际概率分布的表达能力，而不受我们所能描述的知识所限制。也因此，基于学习的方法往往能

够处理更加一般的复杂环境下的人脸检测问题。

近年来统计学习理论的快速发展为基于学习的方法提供了良好基础，基于数据进行统计学习的方法在人脸检测领域取得了非常好的效果，因此得到了广泛重视。

Osuna 等人提出了采用支持向量机 (Support Vector Machine, 简称 SVM) 来检测图像中出现的人脸的方法，其中 SVM 采用了二次规划 (Quadratic Programming) 的方法，它可以在处理大批量的人脸正例和反例时，只选取那些位于分界面附近的样本来完成学习过程[OG97]。

Sung 等人提出的基于样本学习的方法见文献[SP98, Sun96]。他们采用 k-均值聚类方法在人脸特征空间中建立六个“人脸”簇(Clusters)和包围人脸簇的六个“非人脸”簇，并且使用高斯模型来描述这些簇。Sung 等人建立的人脸和非人脸簇如图 1.6 所示。这些人脸样本和非人脸样本的大小是  $19 \times 19$ 。值得强调的是为了解决人脸检测中非人脸样本的选取问题，Sung 等人使用了“自举(Bootstrap)”的方法：首先使用全部的人脸样本和少量的非人脸样本训练一个初始分类器，然后该分类器在一组不含人脸的图像中进行检测，并将所有误检为人脸的子窗口加入非人脸样本库，并和人脸样本一起重新训练一个新的分类器。以上过程不断迭代，直到收集了足够的非人脸样本。这种非人脸样本的收集方法后来被许多研究者使用。

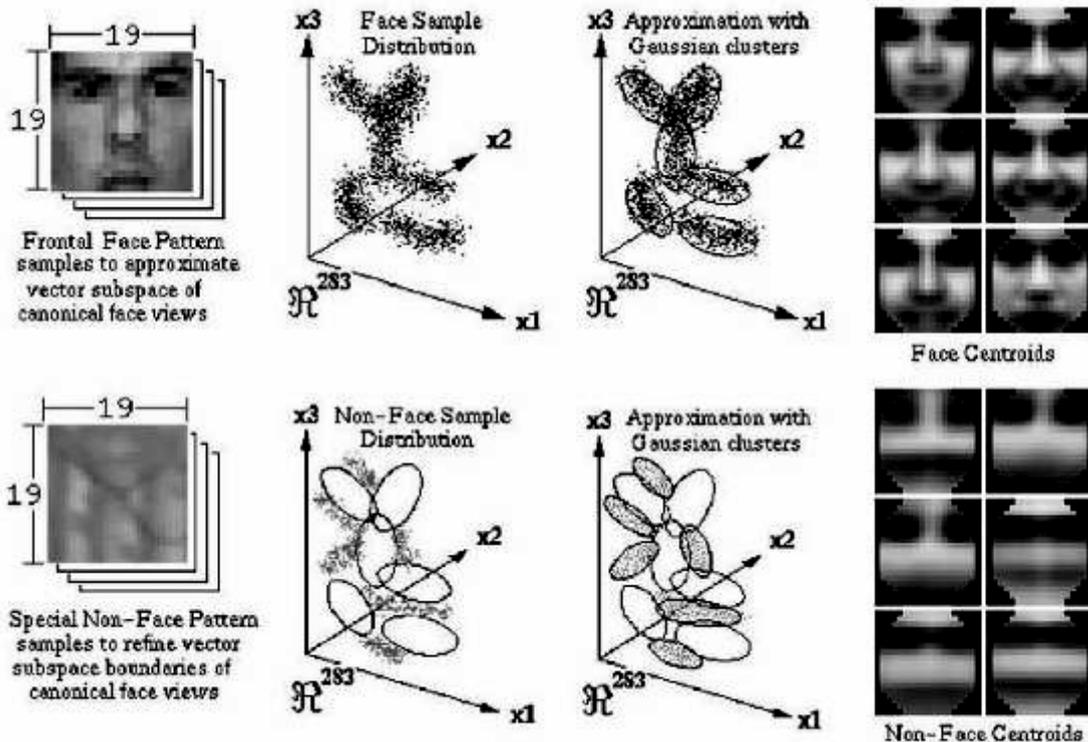


图 1.6 Sung 等人建立的人脸和非人脸簇[SP98]

Rowley 等人采用了两个多层的神经网络来从人脸和非人脸样本中学习它们各自的模型（如灰度值，像素的空间联系）[RBK96, RBK98a, RBK98b, Row99]。其正面人脸检测器由两部分组成：多个神经网络组成的前端检测器和综合前端检测器的决策模型。

为了检测有平面内旋转的人脸，他们又训练了一个新的神经网络对每个扫描子窗口中可能存在的人脸姿态进行判决，并根据判决得到的人脸旋转角度对该窗口进行反旋转，再将该窗口送入正面的人脸检测器进行检测[RBK98b]。他们的神经网络是由一批大小为 $20 \times 20$  人脸样本和非人脸样本训练得到的。神经网络的输出范围为 $[-1, 1]$ ，在对输入窗口进行判决时，如果神经网络的输出接近-1，则为非人脸，接近 1，则为人脸。图 1.7 给出了 Rowley 等使用的人脸检测框架。在文献[RBK98a]中，Rowley 等人构建了一个公共的测试数据库 MIT+CMU 人脸测试库，该测试库在人脸检测领域被广泛使用，作为不同算法的性能测试数据集。同样采用神经网络进行人脸检测的方法还有[FBVC02]。

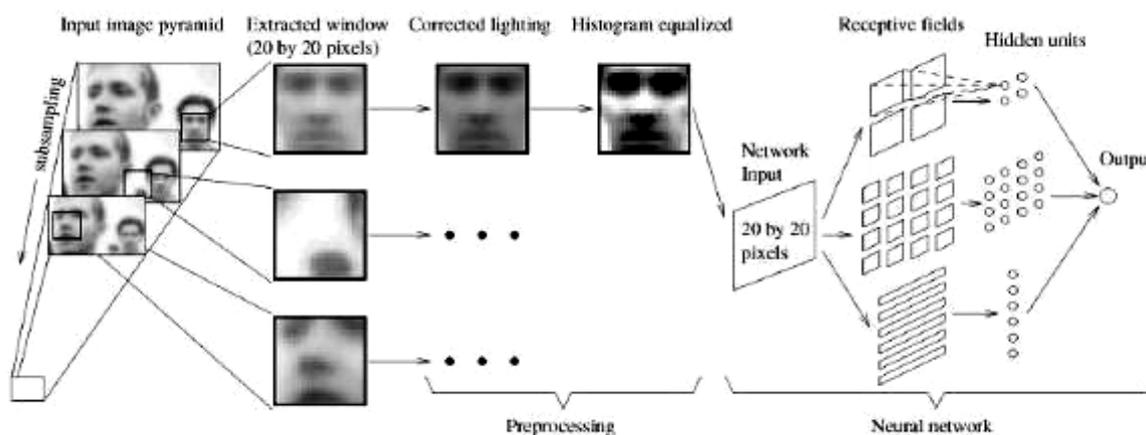


图 1.7 Rowley 等人建立的人脸检测框架[RBK98b]

Schneiderman 等提出了一种基于后验概率估计的方法来对三维物体进行检测[SK98, SK00, Sch00]。该方法利用贝叶斯原理将后验概率转化为一个计算类条件概率密度的问题，将难以估计的先验概率用一个比率参数代替，作为调节检测器敏感度的参量。他们先对样本进行小波变换，并对提取出的特征进行直方图统计，求得人脸和非人脸模式的概率分布，然后用得到的概率来检测正面和侧面人脸[Sch04]。

Yang 等人提出了一种基于 SNoW 的学习机制，用于检测具有不同特征、表情、姿态和光照条件下的人脸[YRA00]。Liu 提出了一种基于 Bayes 特征判别的人脸检测方法并采用 PCA(Principal Components Analysis)进行降维以减小计算量[Liu03]。同样，Heisele 也使用了类似的方法以提高检测速度[HSPP03]。

上面这些方法可以参考综述性文献[YKA02, Hje01, LJ04, 梁 02]。在这之后，一个重要的进展是 Viola 等人提出的采用 AdaBoost 算法进行人脸检测的方法，该算法利用 Haar 特征实现了实时的人脸检测[VJ01a, VJ01b, VJ04]。他们提出的算法在保证人脸检测精度与以前最好的方法具有可比性的基础上，检测速度大幅度提高，实现了第一个准实时的人脸检测系统，计算效率远远超过了过去所有的方法。在  $384 \times 288$  大小的图片上，Pentium III700MHz 的计算机上检测时间大约为 67 毫秒。由于速度上极大的优越性，这个工作在人脸检测领域引起了极大的关注，成为人脸检测领域一个里程碑式的工作，以后几乎所有的工作都受到此方法巨大的影响。

Viola 等人提出的方法主要包括三个方面：

- 1、采用 Haar 特征及其快速计算方法以及基于 Haar 特征的刀切式弱分类器构建方法。
- 2、通过 AdaBoost 学习算法将上述弱分类器集成为强分类器。
- 3、瀑布型多阶段分类器结构。

### Haar 特征、快速计算方法及其刀切式弱分类器构建

Viola 等人提出的 Haar 特征如图 1.8 所示。Haar 特征表示的是阴影矩形与无阴影矩形之内所有像素的灰度和的差值。矩形内所有像素的灰度和可以通过一种叫做积分图的辅助图像通过少量的加减法获得。积分图如图 1.9(a)所示，它每个像素的值为其水平左边，上边所有像素的和（图中红色像素为当前像素，阴影区域的像素和为其像素值）。通过对如图 1.9(b)所示的 a、b、c、d 像素点处的积分图像素值进行如下操作，即可获得由 a、b、c、d 构成的矩形内的像素和。操作如下： $a$  处积分图值 $+d$  处积分图值 $-b$  处积分图值 $-c$  处积分图值。这样一次矩形内像素和的求值只需要 3 次加减法。

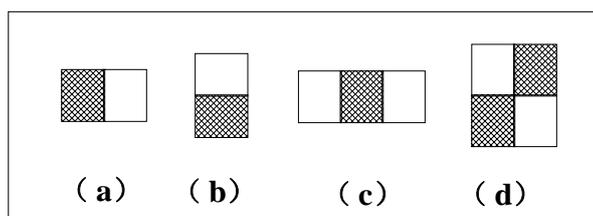


图 1.8 Viola 等人提出的 Haar 特征

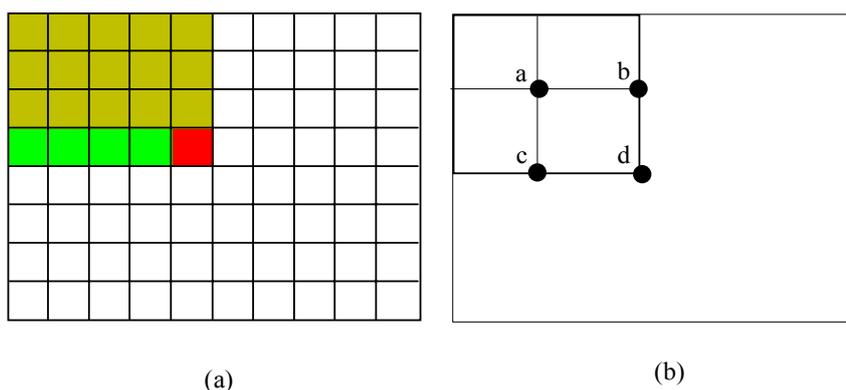


图 1.9 积分图及其矩形内像素灰度和的快速计算示意图

Viola 等人的方法之所以在速度上获得成功，首先是因为采用了以积分图作为辅助进行的 Haar 特征快速计算方法。在一些早期与人脸检测相关的研究工作中，特征提取是直接基于扫描窗口这个高维向量的。如 Rowley 等人[RBK96, RBK98a, RBK98b]就直接将整个扫描窗口作为人工神经网络的输入。虽然 PCA 和 LDA 这类方法能将输入模式从高维空间降到具有很强区分能力的低维空间中，但由于降维过程涉及到计算稠密高维向量的内积，它们都无法避免大量复杂的浮点乘法运算。因此这些方法并不太适用于需要对大

量扫描窗口进行模式判决的人脸检测问题，尤其是那些具有实时应用需求的检测系统。

继 Viola 等人提出的方法之后，人脸检测研究领域涌现出了不少针对特征提取改进的新方法。针对传统 Haar 型矩形特征的平面上的布局及计算策略，很多方法进行了特定的扩展。Barlett[BLFM03]等人根据正面直立人脸的对称性，提出了如图 1.10 所示的对称形 Haar 特征。Lienhart 等人[LM02]提出了旋转 45 度的矩形构成的 Haar 特征，并进一步提出了旋转 45 度图像积分图的概念，借助旋转 45 度的积分图图像，旋转 45 度的矩形仍然可以实现类似竖直矩形的快速计算方式(图 1.11)。武等人[WAHL04, 武 05]通过对矩形进行更多样的组合，得到了如图 1.12 所示的扩展 Haar 特征集合。Li 等人[Li02, LZSZ02, LZZBZ02, LZ04]则进一步放宽了矩形相互间的位置关系，允许不同矩形之间相对平移，从而形成了如图 1.13 所示的矩形特征集合。Huang 等人[HAL06]考虑到计算效率和分类精度的结合提出了粒度特征，从一个开放的简单 Haar 特征集合开始，将特征矩形进行删除、增加、替换等操作(见图 1.14 所示)，从而提出了一种扩张的、基于粒度(简单来说，每个矩形都为正方形)空间内的扩展特征。增加了这些操作以后，特征数目急剧增加，Huang 进一步提出使用启发式搜索算法进行特征集的构建。另外，Masada 等人提出了一种使用遗传算法生成新特征的特征构建算法[MCWW08]。

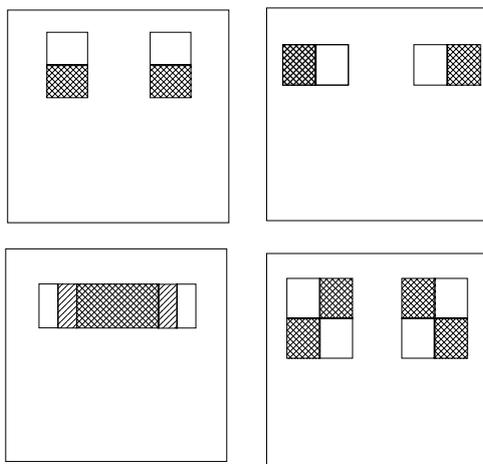


图 1.10 Barlett 等人提出的扩展 Haar 特征[BLFM03]

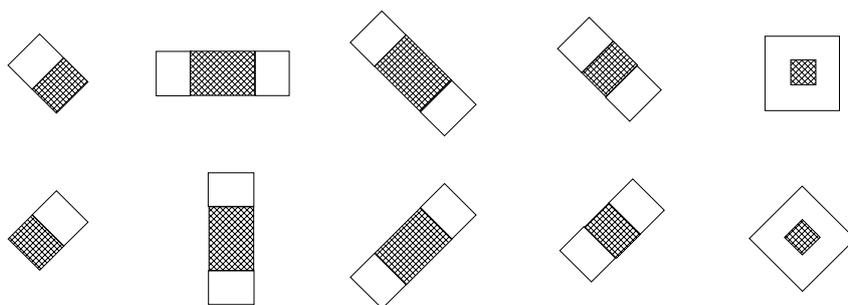


图 1.11 Lienhar 等提出的旋转 Haar 特征[LM02]

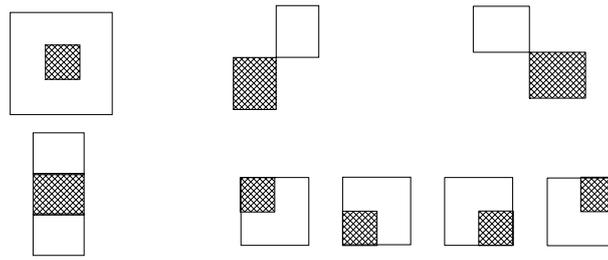


图 1.12 武等提出的扩展 Haar 特征[武 05]

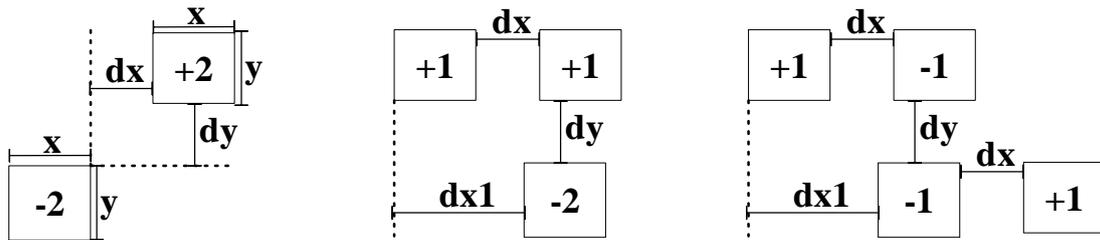


图 1.13 Li 等提出的扩展 Haar 特征[Li02]

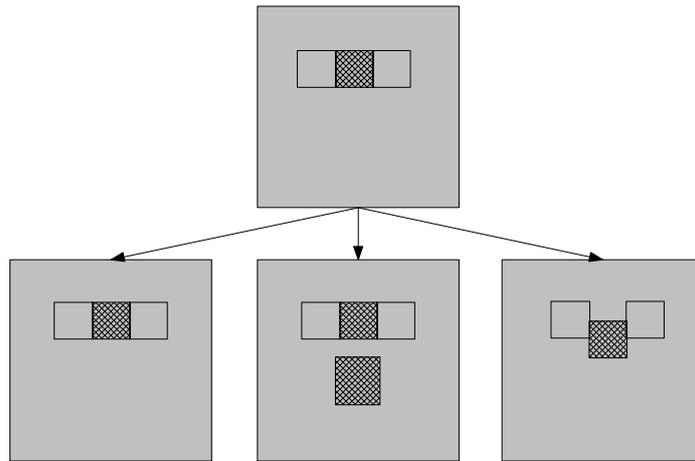


图 1.14 Huang 等提出的扩展 Haar 特征[HAL06]

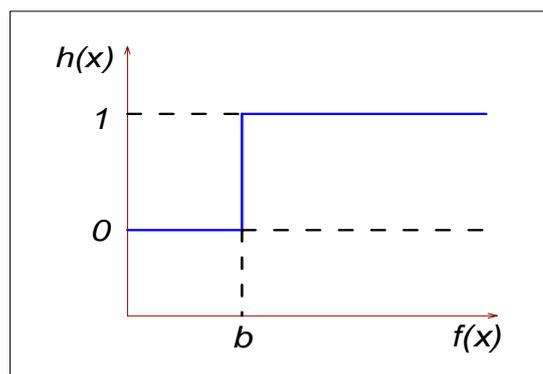
另外, Baluja 等人[BSR04]只取图像中任意两个像素的灰度差异并作为特征,并将这些特征使用 AdaBoost 进行分类器学习,应用于深度旋转 5 个姿态人脸的分类。Abramson 等人[AS05]定义两个正负像素集合,两个集合内的像素分别有序对应,比较分别在两个集合中的每个像素上进行,特征值只提取两个像素灰度差异的符号信息,是二值特征。当两个集合中所有像素对(pair)之间的符号信息与要求的符号信息一致时,特征值取值为 1,否则取值为-1。特征的二值化回避了检测前期的预处理过程(比如 Viola 等人中使用的均值方差规范化),因而能够提高检测的效率。Hadid 等人[HPA04]使用局部二值模式 [OPH96] (Local Binary Pattern) 形成的直方图作为特征进行分类器的学习并进行人脸检测和识别。另外 Zhang 等人[ZZ04, ZGCZ06]在人脸检测中也使用了此特征,并进一步将其应用于文字和汽车的检测问题中。

以上是对 Haar 特征一些比较直观的改进, 另外, Liu 等人[LS03]将 Haar 特征推广为一个更为广泛的概念——线性投影向量。Haar 特征实际上是将二维图像变换成一维向量上的一个形状直观、容易理解的线性变换, 在这个线性变换中, 投影向量取值为 0, -1 或者 1, 举例来说, 对于如图 1.8 所示的 Haar 特征, 矩形阴影区域的像素在投影向量中所对应的元素值取值为 1, 非阴影矩形区域内的像素在投影向量中所对应的元素值取值为-1, 其它所有位置的像素在投影向量中所对应的元素值取值 0。Liu 等人将 Haar 特征中这种矩形框布局的限制完全抛弃, 使得特征可以是任何的投影向量。特征搜索过程中, 为了降低的计算复杂度, 对 Haar、高斯滤波特征空间均匀下采样, 提取期望较好的特征作为候选, 并线性组合这些特征作为候选新特征, 使用 Kullback-Leibler 散度作为评价标准对线性组合新特征的系数进行学习, 从而获得具有更强特征描述能力的所谓 KL 特征。而 Mita 等人[MKH05]将若干个基于阈值判决的 Haar 特征组合到一起得到联合 Haar (Joint Haar Feature) 特征, 联合的方法是对多个基于阈值判决的 Haar 特征构成的弱分类器采用二值编码的方式进行编码, 并将此编码值作为新的联合特征的值, 从而得到了区分能力更强的特征。

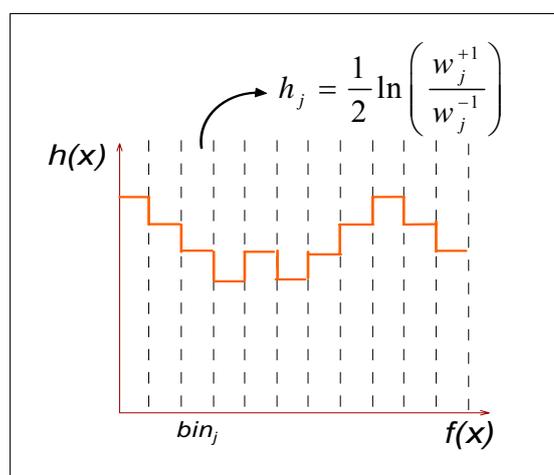
在基于 AdaBoost 的人脸检测体系结构中, 弱分类器处于底层, 由特征提取和函数映射两部分组成。其中, 特征提取负责从扫描窗口中获取最基本的可区分性信息, 而函数映射则需根据特征提取的结果, 采用简单快捷的方式对扫描窗口进行模式判别。

Viola 等人采用的函数映射模式如图 1.15(a)所示, 采用在特定 Haar 特征  $f(x)$  的值空间寻找一个使得错误率最小的阈值  $b$ , 并将特征值空间划分为以  $b$  为界的两个取值分别为 1 和 0 的两个子空间, 其中取值 1, 表示此弱分类器决策为人脸, 而取值为 0, 则表示决策为非人脸。

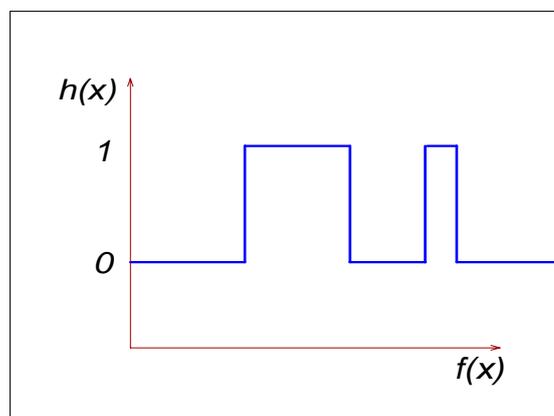
针对特征映射函数, 为了提高弱分类器的判决精度, 加快 AdaBoost 的收敛速度, Liu 等人使用了直方图函数取代阈值判决[LS03], 武等人将 Real AdaBoost[SS99]学习算法引入人脸检测问题中, 并提出了基于等区间划分查找表的弱分类器[W AHL04], 此类弱分类器将每个特征的连续特征值空间等划分为若干个离散的特征值区间, 在每个区间内给出一个候选窗口的特征值如果出现在区间时其为人脸窗口的信度, 并不真正做出人脸和非人脸的二值决策, 信度使用 Real AdaBoost 算法学习获得, 具体来说, 是训练阶段提取的特征值落入本区间的正例样本数目和反例样本数目之间比值的对数函数, 此映射函数如图 1.15(b)所示。而 Xiao 等人[XZST07]则仍然使用 AdaBoost 算法, 提出了基于贝叶斯的刀切式弱分类器, 弱分类器仍然只输出 1 和 0, 但是特征值空间被划分为若干个取值为 1 或者 0 的子空间, 在每个子空间内根据样本的分布确定取值为 0 或者 1, 如图 1.15(c)所示。他们还提出了子空间的搜索方法, 以降低弱分类器的分类错误率。这种方法描述特征值空间内的样本分布能力大大加强, 所以提高了弱分类器的判别能力, 但是同时也增加了学习一个弱分类器所需要的时间。Le 等人[LS06]讨论了如何借助熵分析的方法来优化弱分类器中对于特征空间的划分。



(a)



(b)



(c)

图 1.15 弱分类器的类型 (a) 刀切式阈值型 (b) 均匀划分的分段信度函数 (c) 非等区间阈值型函数

另外, 为了提高检测器的训练速度, Pham 等人对特征值在样本空间的分布使用高斯函数建模[PC07b], 刀切式阈值型弱分类器基于正例和反例样本分布的高斯函数均值和方差而构建, 每个 Haar 特征的高斯分布均值和方差可以通过在整幅样本图像上预先计算的辅助矩阵进行简单的运算获得。关于辅助矩阵, 比较容易理解的是均值求取辅助矩阵, 它就是积分图均值图像, 对所有样本积分图加和求取均值图像后, 通过在均值图像上应用 Haar 特征计算相关操作, 就可以获得此 Haar 特征形成高斯函数的均值。关于方差,

也可以采用与均值计算存在着类似但更复杂的过程计算获得。所以，一旦预先计算好辅助矩阵，特征选择过程中的候选特征弱分类器的构建将降低为少数的计算操作，从而提高整个训练过程的速度。王等人则提出了在训练时对特征值进行排序以避免重复操作降低训练算法的计算复杂度[王 04]。

## 基于 AdaBoost 的强分类器学习及其改进

每个强分类器都是瀑布型分类器的一个子分类器，使用学习算法学习获得，具体可以参考如图 1.16 中相关的关于瀑布型分类器内容的描述。Viola 提出了使用 AdaBoost 对一个拥有相当数量的 Haar 特征集进行学习的方法，具体来说，AdaBoost 方法挑选部分特征并将其构建为一个线性加和模型，形成一个预测的信度。AdaBoost 是 Boosting 算法的一种，详细的关于 Boosting 算法来龙去脉的相关介绍可以参考 3.1 节。

继 Viola 之后，存在很多针对人脸检测特定问题对 Boosting 算法进行改进的算法，下面我们将进行简单的介绍。

人脸检测中以及物体检测中存在一种不对称现象，不对称现象可以描述如下：在人脸模式和非人脸模式构成的空间中，人脸模式的子窗口在图像中所占的比例非常小，而非人脸模式的子窗口所占的比例非常大；为了避免漏检，学习的过程中就不能对人脸样本和非人脸样本同等对待。具体来说，不仅要最小化学习过程中的错误拒绝率(FRR: False Reject Rate)和错误接受率(FAR: False Accept Rate)，而且更要关注于缩小 FRR 的上界。对于瀑布型分类器的每个子分类器来说，每个子分类器的训练目标都要使得错误拒绝率接近于零，而错误接受率则可以高很多，甚至可以高达 50%。Boosting 初始设计的时候并没有考虑到这种不对称现象，为了更好的满足人脸中的不对称要求，各种解决不对称的算法被提出。Viola [Vio01]引入了不对称 AdaBoost 算法(Asymmetric AdaBoost)，该方法还用于快速的图像检索和人脸检测。Hou 等人则提出了另外一种不对称 AdaBoost 算法[HLT06]，并进一步提出了一种快速瀑布型分类器学习算法[WBM08]。另外，Pham 等提出一种基于样本权重修改的不对称学习算法，根据指定的不对称程度，初始时赋予正例样本更大的权重，以达到更好的学习目标，他们的实验证明了这种不对称算法在保证正确率相同的情况下，可以有效的降低弱分类器的数目[PC07a, PHC08]。

在多姿态人脸检测上，Li 等人提出了金字塔检测器结构[Li02, LZSZ02, LZZBZ02, LZ04]，通过利用 FloatBoost 将学到的弱分类器构造成一个强分类器，该系统是一个从粗到精、从简单到复杂的金字塔型的人脸检测系统。该系统是第一个近实时的多姿态人脸检测系统。Ma 等人提出了含有视角分类单元的树形结构 [MQ03]，而 Jones 等人则提出了基于决策树进行视角估计的检测器结构 [JP03]。Lin 等人提出了多类别 Boost 算法 (Multi-Class Boosting)[LL05]，而 Zhang 等人提出了基于输出编码错误修正(Error Correcting Output Codes)的多类别 Boosting 算法 [ZGC06]。Huang 等人提出了向量 Boosting 的方法 (Vector Boosting) [HWAL04]，并将其用于多姿态的人脸检测，即任意

角度的深度旋转(ROP: Rotation-Off-Plane)和平面旋转(RIP: Rotation-In-Plane)的人脸检测[HWAL04, HALL05, HALL07]。在他们的多姿态人脸检测算法中,他们采用了一系列的技术,包括:提出了基于宽度优先搜索(WFS: Width-First-Search)的树结构的人脸检测器、用 Vector Boosting 算法来学习强分类器输出的矢量结果、采用了基于域分割(Domain-partition-based)弱分类器的学习、探究不同粒度空间(Granular space)中稀疏的特征以及如何在特征空间中采用启发式搜索等。

为了提高检测器的训练速度, Wu 等人还提出基于直接特征选择(Direct Feature Selection)的强分类器训练方法[WRM03],直接特征选择认为 AdaBoost 算法成功的关键在于最终学到的集成分类器获得的特征集合来源于一个非常丰富的候选特征集,而对于每次更新权重重新调整弱分类器依赖性不大,所以直接特征选择抛弃了样本权重更新和重新每次弱分类器训练重新更新每个特征弱分类器的过程,在原始的样本集上基于每个样本权重为 1 预先计算所有的弱分类器,并直接应用这些弱分类器进行特征选择构建集成分类器,从而节省了每次弱分类器训练对每个特征重新训练弱分类器的过程,提高了训练时间。Fan 等人采用类似 Wu 的方法进行快速训练[FL06]。

此外,还有 Sahbi 提出的综合 SVM 和分级的方法进行人脸检测的系统[Sah03],艾等人采用 AdaBoost 和 SVM 进行分类器的融合以提高分类器的性能[艾 03];综合 AdaBoost 和其它强分类器的文献还有[FE03],此外,其它针对 AdaBoost 算法进行改进的文献还有[Lyu05]等。

文献[LKP03]对 Viola 等人提出的方法采用不同的 Boosting 算法进行了对比实验。

### 瀑布型分类器学习及其改进

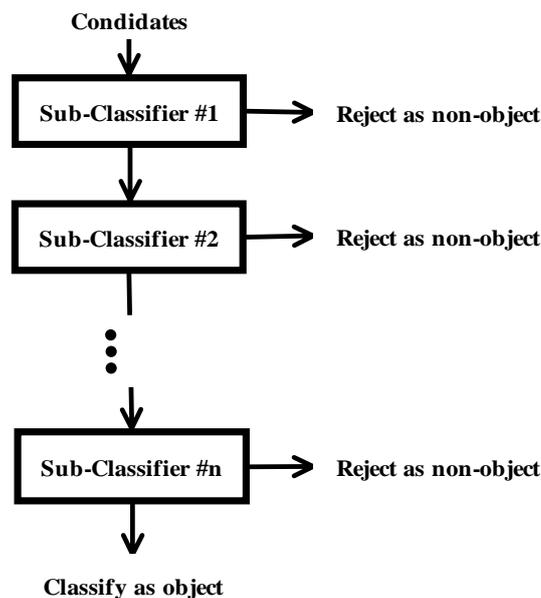


图 1.16 瀑布式分类器

瀑布型分类器的检测结构如图 1.16 所示。瀑布型分类器由多个子分类器构成，这些子分类器从前到后顺序排列，只有通过前面所有的子分类器，才能进入下一个子分类器。每个子分类器都进行一次二值判决，决定输出窗口是人脸还是非人脸。本质上，瀑布型分类器对非人脸空间中的样本进行序贯决策，简单的样本在靠前位置被简单快速的子分类器排除，而复杂样本则在靠后位置被相对计算复杂度高但更强的子分类器排除。这符合非人脸样本的分布规律，客观世界中的非人脸样本集中，简单非人脸样本占了大部分，而复杂的非人脸样本只占据了一小部分，所以，瀑布型分类器能够很好的提高分类器的运行效率。

继 Viola 之后，为了增强各个强分类器之间的联系，充分利用已经获得的置信度信息，Xiao 等人提出了 Boosting Chain 的方法[XZZ03, XLZ04]，而 Sochman 等人则通过 WaldBoost 方法将这种思想进一步渗透到每个弱分类器上[Luo05]。Huang 等人提出一种嵌入式瀑布型分类器(nested cascade)来解决此问题[BMR06]。嵌入式瀑布型分类器的检测结构如图 1.17 所示，如图中所示，每个 Nested Classifier 将前一层子分类器的置信度重新构建一个弱分类器，从而建立了各层子分类器之间的联系。

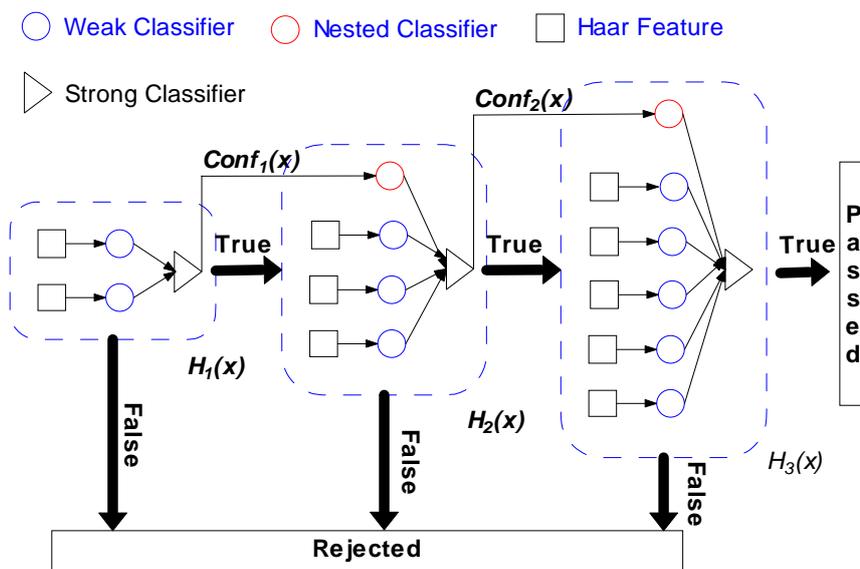


图 1.17 嵌入式瀑布型分类器

在检测器结构的层次上，为了能够平衡每层强分类器和整体强分类器的关系，使得系统性能更优化，Sun 等人[SRB04]、Luo[Luo05]、Brubaker 等人[BMR06, BWS08] 和 Dundar 等人[DB07]分别提出了用于估计瀑布型检测器整体性能的算法，并将其应用于每个子分类器的训练。

Bourdev 等[BB05]和 Chen 等[CY05]分别提出了两种方法训练瀑布型分类器，他们方法的主要特点是优化用于人脸检测的瀑布型分类器的计算复杂度。

此外，Hayashi 等人采用 AdaBoost 的方法来检测图像中分辨率非常小的人脸（例如  $6 \times 6$  像素大小），并将该算法用于视频图像中的人脸检测[HHD06]。文献[LLK03]提出了

一种不同于瀑布型分类器那样的二叉截枝树，而是没有截枝的决策树方法，以提高人脸检测对复杂人脸的鲁棒性。文献[CWYS05, CWYS07, WCSC06, CCG04]尝试对训练集进行扩展和下采样，扩张是为了获取更多可供使用的样本，下采样是为了优化样本的分布，增强其代表性。

不针对 Viola 等人的方法进行进一步改进，另外一些进展还包括 Garcia 和 Delakis 提出的采用卷积神经网络来进行快速的人脸检测[GD04]。文献[BH03, HSP07, BH02, HSMP03]研究了基于部件的人脸检测。文献[KS03] 调查了头肩图像和人脸图像共同使用时人脸检测的情况，结果表明此策略能够消除更多误检，从而提升性能的结果。文献[AAPO5]研究了使用对应不同角度的多个分类器进行投票来去除训练集中包含的“野点”。文献[Bha86]使用人脸轮廓的边缘信息进行人脸的进一步改善。

#### 1.4 本文研究问题的提出

如何构建速度上更快、精度上更准的人脸检测器始终是研究者们最终需要解决的问题，为了实现此目标，综合以上关于人脸检测相关文献的描述，可以发现，人脸检测方法经历了从早期的基于先验知识的方法到当前主流的基于统计学习的方法的发展过程。人脸检测方法对样本的依赖性也经历了从弱到强的一个过程，在基于先验知识的方法中，人脸检测方法只能使用少数（几百上千）的典型范例样本对先验知识进行进一步的验证，在统计方法中，研究者们收集更多的样本（上千上万）进行分类器的自动学习。在当前基于统计学习的各种人脸方法中，人脸检测器的学习过程中使用的样本大都不超过数十万个。本文工作首先从样本的收集开始，通过依赖近年来数码产品发展带来的大规模数字图像的出现，从而进行了更大规模人脸图像的收集，此举的目的是期望以大样本集来带动学习得到的分类器能够获得更准确的人脸检测器。实际上，针对一个人脸检测器的学习，本文工作收集制作了数十万的样本。

大规模训练样本集的引入放大了当前最流行的基于 Viola 等人提出瀑布型分类器的训练和检测框架所存在问题，主要有三点：

其一，瀑布型分类器的训练负担过大问题。在基于 Viola 等人提出瀑布型分类器的训练框架下，即使在样本集不是很大的情况下，训练起来就需要花费较多的时间，精确地说 Viola 等人的正面人脸检测器花了 2 个星期进行训练(正面人脸样本五千多个)。现在，在已经拥有数十万或者更多正例样本的条件下，也就是说，训练样本规模增加了数十倍的情况下，怎样改进瀑布型分类器的训练方法，以提高训练效率，同时还要保证大样本集带来的好处——提高分类器的检测性能？

其二，在能够检测更为复杂人脸的检测器中，检测器的计算复杂度增加的问题。大规模更多变化样本集的引入导致了当前主流方法——Viola 等人提出瀑布型分类器的训练框架——学习获得的人脸检测器在高性能的情况下计算复杂度大幅度增加。如何在保证在人脸检测器拥有高分类性能的同时，缓解人脸检测过程中的瀑布型分类器计算复杂

度过大的问题？

其三，无论如何，目前的计算机硬件条件和人脸检测技术还不能做到在很多在实际的应用场合中检测系统既获得期望的分类性能，同时又获得期望的检测速度。那么如何对瀑布型分类器进行更优的调节，使得在保证系统分类性能的情况下，获得更优的检测速度；或者说，在保证检测速度的情况下，获得跟有的检测分类性能？

## 1.5 本文的主要贡献及章节安排

基于 Viola 等人提出瀑布型分类器的训练和检测框架，针对上节提出的几个问题，本文在瀑布型分类器学习的各个层次提出相应的解决方法。论文的主要贡献概括如下：

1. **提出了一种基于大规模样本集的瀑布型分类器学习算法。**针对瀑布型分类器学习中同时存在大规模正例样本集和反例样本集的情况，提出一种叫做矩阵式学习的训练算法。简单说来，在瀑布式分类器的训练过程中，矩阵式学习不仅在相邻的子分类器间对大规模反例样本集进行“自举”，而且，在每个子分类器自身的学习过程中，对大规模正例样本集也通过自举方式收集小部分具有代表性的正例样本进行最终强分类器的学习，矩阵式学习过程中的各个子分类器形成一个类似矩阵的结构，所以被称作矩阵式学习。更进一步，矩阵式学习不只简单地对大规模正例样本集和反例样本集进行自举，在自举过程中，相邻的子分类器的共享信息被用来进行更加快速的学习。这些共享信息包括两个方面，分类器训练过程中已经获得的特征和样本。
2. **提出了局部组合二值特征的概念以及使用局部组合二值特征构建快速预处理分类器的方法。**这包括了一种新的特征——局部组合二值特征，以及相应的以特征为中心的瀑布型检测算法。在提出的非人脸窗口预排除方法中，特征计算方式更加简单，避免可能重复的特征计算。局部组合二值特征的基本单元是二值特征，二值特征是图像上相邻区域间灰度差异的二值信息，即像 Haar 特征那样的多个区域间对比的更“黑”更“白”关系。局部组合二值特征是组合二值特征的一个有效子集，而组合二值特征表示的是多个不同二值特征值共生的信息。在局部组合二值特征中，组合的方式被限制在类似局部二值模式组合像素点时的方式，形成局部组合二值特征。局部组合二值特征数目少，描述能力强，被用来进行预分类器的学习。预分类器采用了一种类似“注意机制”的以特征为中心的瀑布型分类器运行技术，以特征为中心的运行机制首先针对某个显著特征模板对图像每个位置计算出特征值，构成特征图像，然后在穷举得到的窗口上应用学习得到的本特征模板在各个位置的统计信息，并进一步判断窗口为人脸或者非人脸。无论是赖以计算的特征模板还是窗口内特征模板的统计信息都使用学习的方法得到，具体来说，模板由多个候选挑选得到的最优模板，统计信息是 Boosting 学习得到的瀑布型分类器。

3. 提出一种基于搜索的构建瀑布型分类器的算法。基于一个已有的 Boosting 瀑布型分类器，我们提出一种构建多个更加快速的瀑布型分类器方法。新的瀑布型分类器允许损失一定的正例分类正确率，损失的大小由用户指定。提出了一种新的基于搜索的算法，对可能存在的瀑布型分类器进行搜索，并获得计算代价更小的瀑布型分类器。为了降低搜索算法的计算负担，搜索算法递进迭代运行。搜索开始于正例分类正确率为百分之百的瀑布型分类器。每一轮的运行在当前瀑布型分类器基础上，搜索进一步损失小部分正例分类正确率条件下所有可能的瀑布型分类器，并获得当前计算代价最小的瀑布型分类器。然后，在当前获得的瀑布型分类器基础上，迭代运行，直到获得的瀑布型分类器正例分类正确率达到用户指定的正例分类正确率，算法终止，输出瀑布型分类器。贡献主要有三点：1、提出了一种基于搜索的快速瀑布型分类器构建算法；2、搜索算法中将所有可能的瀑布型分类器考虑在内；3、搜索算法中瀑布型分类器的计算代价评估考虑了其中包含的不同特征的不同计算代价。

借助以上提出的针对瀑布型分类器的新方法，本文提出的人脸检测算法在速度和精度上都获得了显著的进步。借助贡献 1 提出的新的训练算法，大大缓解了反例样本集和正例样本集数量巨大的情况下，当前人脸检测算法计算代价过高的问题。另一方面大规模多变化的反例样本集和正例样本集保证了学到分类器的分类性能。这为进一步提高人脸检测性能提供了一个很好的解决方式，收集更多变化的样本，并应用矩阵式学习，而且，矩阵式学习是一个为单机运行（非多机器上分布式学习）设计，实现简单，方便使用。

借助于贡献 2 提出的算法，本文构建了一个有效的快速预处理分类器作为传统 Haar 特征瀑布型分类器之外分类器的一部分。二值特征企图采用更为简单的计算来描述人脸的结构，所有的计算都基于局部区域间的黑白二值特性，进一步提出的预分类器检测方法基于这些二值特性直接对输入图像进行特性计算，并最大化的利用每个像素点上获得的信息进行粗分类，预分类器的加入能够大大降低检测器的计算复杂度。

不仅如此，针对基于 Boosting 学到的瀑布型分类器，本文进一步提出了对分类器进行计算复杂度调整的算法，这使得人脸检测在拥有不同速度\精度需求的不同应用场景使用起来更加方便、灵活、高效。

本文的章节结构安排如下：

第二章中介绍了本文所收集到的各种人脸样本集以及样本制作的方法，另外还介绍了人脸检测性能的评估方法。

第三章中在回顾传统瀑布型分类器方法的基础上，介绍了解决大规模样本集训练的矩阵式学习算法，并进行了详细的讨论。

第四章介绍了局部组合二值特征的构造动机、独特之处，并据此提出了以特征为中心的预分类器构建算法，并与传统检测方法进行了对比和讨论。

第五章介绍了给定一个 Boosting 算法得到的整体分类器，对整体分类器进行分割并

构建不同速度瀑布型分类器的方法。

第六章介绍了本文算法所实现的动态链接库模块情况和基于本文算法所实现的视频演示系统情况。

第七章对本文工作进行了总结和展望。



## 第二章 人脸检测数据集制备及检测性能评估

在第一章中已经讨论过，因为人类对自身所认识的模式并没有一个明确的答案，单单利用人类知识对人脸模式的认识和总结并转换为电脑程序进行人脸检测的方法到目前为止还没有证明其能够完成高性能人脸检测任务的需求，所以，研究者们很自然的选择了使用统计学习算法由计算机自动学习能够区分人脸模式与非人脸模式的规则。本文研究方法的主要内容就是基于统计学习获得人脸检测器的人脸检测方法。

用计算机进行统计学习获取分类知识作为一个智能的过程，人类的认识也只处于初级阶段，当前多数统计学习算法，在大量噪声的干扰下会出现各种各样的问题，使得最终学到的分类器不能达到预期的结果。分类器要获得更好的学习效果，要求提供给学习算法的输入样本集尽可能地缩小人为因素带来的“非正常样本”干扰，最大化的降低样本类内散度，降低分类器学习的盲目性。另一方面，为了提高分类器的对各种样本的适应能力，输入样本集又要包含尽可能多的合理范围内的变化。

所以，对于基于统计学习的人脸检测方法来说，样本集的收集成为了一个至关重要的工作。训练样本集直接影响最后分类器所能达到的分类性能，从某种意义上来说，基于一个不具有充分代表性的人脸样本集，学习算法几乎不可能得到一个性能优良的分类器。

对于人脸检测器的训练样本集来说，训练集中要能够包括更多的拥有各种变化的人脸，但是，又不能脱离了“人脸样本”本身的范畴。本章将介绍本文工作中所有使用的人脸样本集和非人脸样本集的相关情况。另外，本章还介绍了对于基于统计学习的方法而言，人脸检测器检测结果如何依据训练样本进行合适的正确性评估和总体性能评估方。

### 2.1 数据集收集

因为本文工作不仅涉及正面竖直姿态下的人脸检测，还涉及到多姿态人脸检测，所以我们不仅收集了正面竖直人脸图像数据，还收集了多姿态人脸图像数据。在以后的描述中，正面竖直人脸将作为一个典型问题进行研究，简单起见，将简单称其为正面人脸。本文工作中涉及的多姿态人脸姿态覆盖了日常生活中的常见姿态，具体来说，包括了平面外左右旋转 $-90^\circ$ 到正 $90^\circ$ ，平面内旋转 $-45^\circ$ 到 $45^\circ$ 的所有人脸。对于平面外上下旋转，并没有做单独的考虑，收集到的人脸图片中平面外上下旋转范围被控制在 $-40^\circ$ 到 $40^\circ$ 。

对于多姿态人脸，根据其姿态，将整体覆盖范围划分为更小的姿态范围。首先，将平面外左右旋转范围从 $-90$ 度到 $90$ 的姿态范围进一步划分为5个小范围：左全侧面人脸 $[-90^\circ, -50^\circ]$ 、右全侧面人脸 $[50^\circ, 90^\circ]$ 、左半侧面人脸 $[-50^\circ, -25^\circ]$ 、右半侧面人脸 $[25^\circ, 50^\circ]$ 和正面人脸 $[-25^\circ, 25^\circ]$ 。进一步，将平面内旋转 $[-45^\circ, 45^\circ]$ 划分为3个子范围：

左侧倾斜人脸 $[-45^\circ, -15^\circ]$ 、竖直人脸 $[-15^\circ, 15^\circ]$ 和右侧倾斜人脸 $[15^\circ, 45^\circ]$ 。多姿态人脸共包括了 15 种姿态，图 2.1 给出了这 15 种姿态的示意图。

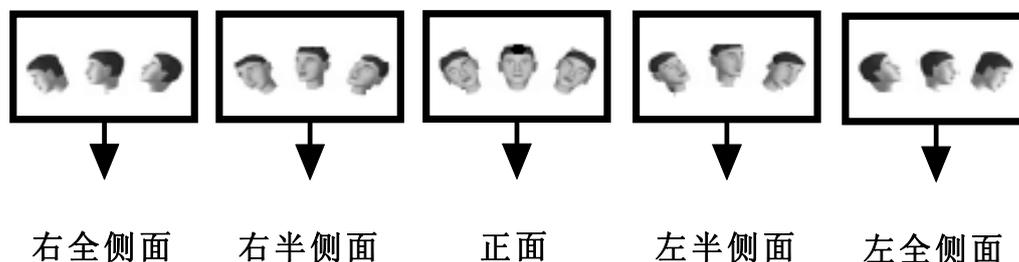


图 2.1 本文使用的 15 种多姿态人脸类别

本质上，在 15 种人脸姿态中，只需要 3 种姿态即可（如：左全侧面、左半侧面、正面），其它姿态可以通过这三种姿态进行平面内旋转和镜像操作等变换得到。

本文工作使用的正面人脸样本集中的原始人脸图像包含的变化比较广泛，从图像采集设备来说，包括数码相机、监控摄像头、摄像机、扫描仪等等；从人脸本身的变化来说，包括了不同人种、不同表情、不同性别、不同年龄阶段的人脸；最后，也是最重要的，收集过程中尽量收集更多不同光照情况下的人脸。对于遮挡情况下的人脸，本文工作并没有进行单独考虑，在本文以后所有的讨论中，如果没有特殊提示，提到的人脸都是没有遮挡，或者只有小部分遮挡情况下的人脸。

正面人脸样本数据集中的正面人脸图像来源包括互联网、FERET 数据集[PMRR97]，BioID 数据集[JKF01]、电视或 VCD 视频截屏、数码相机实拍等等，收集的图片中包含了将近 30000 张正面人脸图像。

大部分的半侧面和全侧面人脸图像从 700 个每幅图像只包含一个人脸的视频片段中获得，这些视频片段拍摄于普通光照的室内场景，每段视频包含一个人深度旋转从左全侧面到右全侧面的旋转变换。人脸在视频序列图像中所占的大小大约为  $100 \times 100$  像素左右。另外，为了进一步增加侧面人脸图像集合中人脸的变化，还从互联网上收集了一部分图像，图像集中共包含半侧面和全侧面人脸共 14087 个。半侧面人脸约含有 9000 个，全侧面人脸约含有 5000 个。

## 2.2 人脸样本的制作

收集到的图片中的人脸图像不能形成最终的人脸训练样本集，还需要标注特征点，根据特征点进行人脸对齐、尺度归一，来进一步降低人脸样本集的复杂度，以更好的提供给统计学习算法学习。

### 2.2.1 人脸样本对齐

人脸对齐是样本制作的第一步，样本制作包括了对齐、剪切和缩放三个过程。人脸

样本的对齐需要利用人脸上的关键特征点，这些特征点包括人眼、鼻子、嘴角、嘴中心等等，这些特征点通过人工标注获得。对齐人脸至少需要两个特征点，具体采用什么特征点进行人脸样本的对齐，以及在这些特征点的基础上如何剪切，通常由实现者自身的经验确定。无论如何，每个实现者在指定这些标注以及剪切方案时，目标都是尽量缩小样本的类内散度，扩大与非人脸集合之间的类间散度。图 2.2 给出了一些不同研究者在实验的过程中使用的正面人脸样本实例。研究者们也在探讨哪种样本对齐和剪切方式更加合理，但是并没有定论。

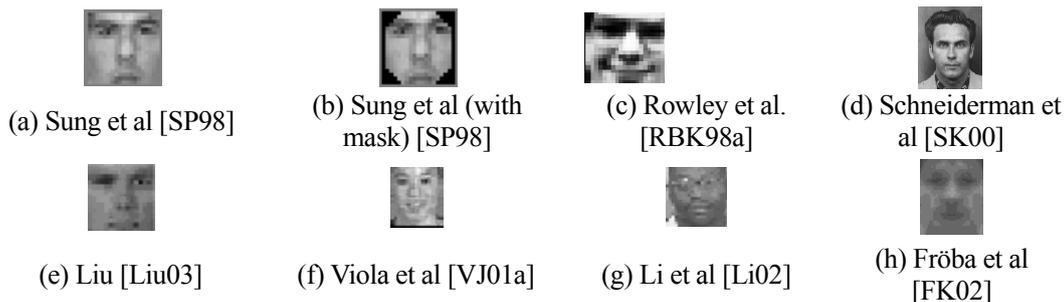


图 2.2 不同研究者使用的正面人脸训练样本实例

对人脸对齐来说，各个姿态人脸情况有所不同，下面，我们以正面人脸作为一个典型说起。目前看来，对于正面人脸来说，基于双眼为基准点的对齐方式经常被使用，归结起来，原因有两个。一方面，双眼特征明显，易标注，不像鼻尖之类的特征点，精确定位比较困难；另一方面，实验结果表明，以双眼为基准点获得的人脸样本能够获得性能优良的分器，检测结果与其它标注方法获得的相应相当或者更好。本文对正面人脸样本使用了人脸的双眼位置作为基准点进行对齐。

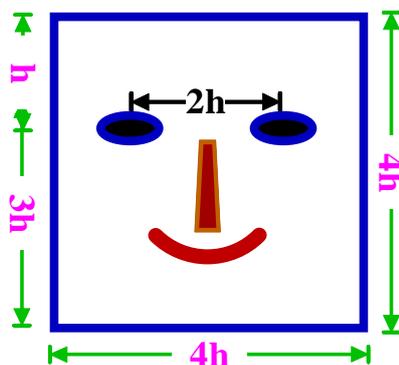


图 2.3 本文使用的正面样本剪切方案

有了双眼位置之后，对齐人脸的下一步工作就是确定将来训练人脸图像包含的内容，比如人脸样本中的信息包含整个头部各器官，还是只要几个核心器官——眉毛双眼鼻子嘴巴。无论最终做出何种选择，做出选择的依据都是最大化人脸训练样本集和非人脸样本集之间的可区分性。本文所做工作最终采用两个基准点来确定待面部器官内容，以人工标注了双眼位置的正面人脸为例，根据双眼两点之间的距离和中心来确定眼睛中心上边、下边、左边、右边需要保留的人脸范围比例。图 2.3 中给出了本文中使用的正面人脸样本所包含的范围，图中双眼之间的距离表示为  $2h$ ，对于一个双眼连线处于水平方

向的人脸来说，双眼中心上边部分高度为  $h$ ，双眼中心下边部分高度为  $3h$ ，双眼中心左边部分和右边部分宽度都为  $2h$ 。在此范围下，所有以此标准对齐后的正面人脸图像中，眉毛、双眼、鼻子和嘴巴几乎都被包含其中。另外，对齐后的人脸图像中还包含一部分除这些核心器官之外的部分区域，这些区域范围限制的原则是尽量不包含大量背景信息和头发信息，这些信息变化缺少规律，不利于分类器学习。

对于半侧面人脸样本，本文选择的基准点仍然是双眼位置。并进一步根据双眼之间的距离和中心确定了人脸的范围。

对于全侧面人脸，则以可见的一只眼睛作为一个基准点，以嘴巴中心为另一个基准点，并进一步根据这两个基准点之间的距离和中心确定人脸的范围。

## 2.2.2 人脸样本生成

在对齐、确定人脸样本包含的范围之后，每幅原始人脸图像再经过缩放，以使所有人脸样本在一个较窄的尺度范围内，就可以获得最终的人脸样本，本文选择的人脸样本模板是  $24 \times 24$  像素大小。

另外，对于每个原始人脸样本，有必要为其生成一些可控制的虚拟人脸样本，以增加样本的变化。虚拟样本的生成的方法之一是采用局部范围内的平面内旋转变换，此方法可以应用于所有姿态下的人脸。根据 2.1 节对人脸检测中使用的姿态的定义可知，每个姿态分类器需要覆盖平面内旋转角度  $30^\circ$  的范围，而原始采集的人脸只存在于一个角度上，这使得人脸样本在平面内旋转空间的变化并不能被覆盖。平面内旋转虚拟样本的生成可以通过旋转原始图像并重新进行剪切缩放获得。对于每个姿态所控制的连续的平面内旋转角度范围，生成一个虚拟样本需要确定每个虚拟样本的具体平面内旋转角度，也就是对连续旋转角度进行离散化，并生成每个角度下的虚拟样本，本文采用了均匀离散化的方法，对于每个姿态平面内旋转角度控制范围，生成了 5 个均匀的离散角度下的虚拟样本，比如对于竖直情况下，虚拟样本的 5 个旋转角度分别为  $-12^\circ$ ， $-6^\circ$ ， $0^\circ$ ， $6^\circ$  和  $12^\circ$ ，而对于左倾斜情况下，虚拟样本的 5 个旋转角度分别为  $-42^\circ$ ， $-36^\circ$ ， $-30^\circ$ ， $-24^\circ$  和  $-18^\circ$ ，最后，对于右倾斜的情况，虚拟样本的 5 个旋转角度分别为  $18^\circ$ ， $24^\circ$ ， $30^\circ$ ， $36^\circ$  和  $42^\circ$ 。对于正面人脸来说，考虑到正面样本在水平方向的对称性，虚拟样本的另一种生成方法是水平镜像，此虚拟样本生成方法也在本文的样本制作过程中被采用。

最终，通过如下一节将要讨论的人工净化样本操作（见下节）之后，未生成虚拟样本前的正面人脸集合包括人脸图像 23608 张。对每个姿态的通过对原始正面人脸图像进行镜像和平面内 5 个角度（ $-12^\circ$ ， $-6^\circ$ ， $0^\circ$ ， $6^\circ$  和  $12^\circ$ ）下虚拟样本的生成，最终的正面人脸训练集大小为 236080 个。

对于其它姿态，通过人工净化之后，并将存在相互镜像关系的两个姿态下（左半侧和右半侧，左全侧和右全侧）进行镜像并归一到一个姿态下后，最终收集到 24000 个全侧面人脸图像和 60000 张半侧面人脸图像。通过平面内 5 个角度虚拟样本的生成，得到

总的全侧面和半侧面人脸训练集大小分别为 120000 张和 300000 张。

如前面所述，其它姿态下的人脸图像可以通过水平镜像和平面内旋转 $\pm 30^\circ$  此三种姿态下的人脸图像获得，并制作出相应的样本集。

各姿态下的一些人脸样本实例见图 2.4 到图 2.18。



图 2.4 正面竖直和人脸样本示例

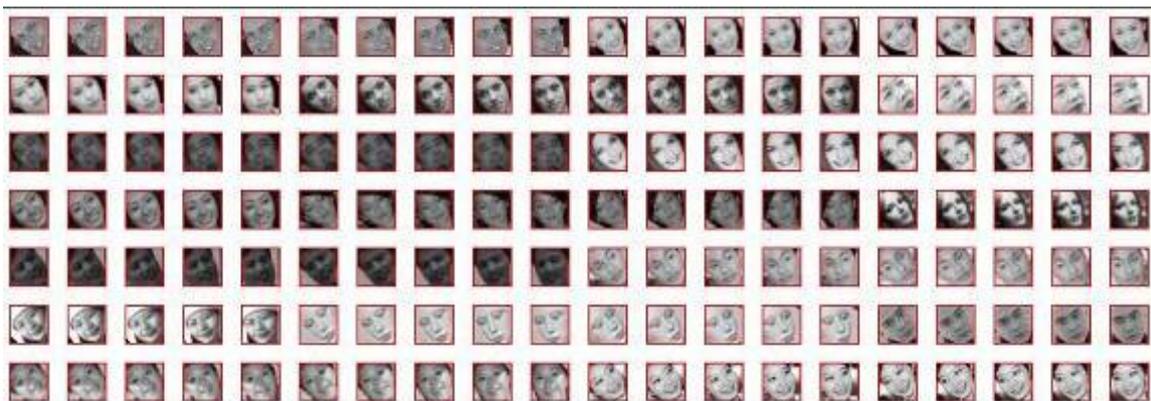


图 2.5 正面左倾斜人脸样本示例

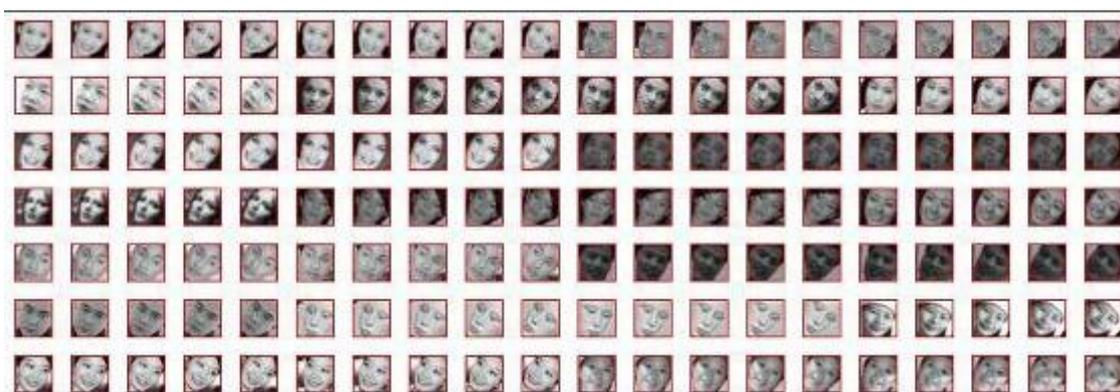


图 2.6 正面右倾斜人脸样本示例



图 2.7 左半侧竖直人脸样本示例

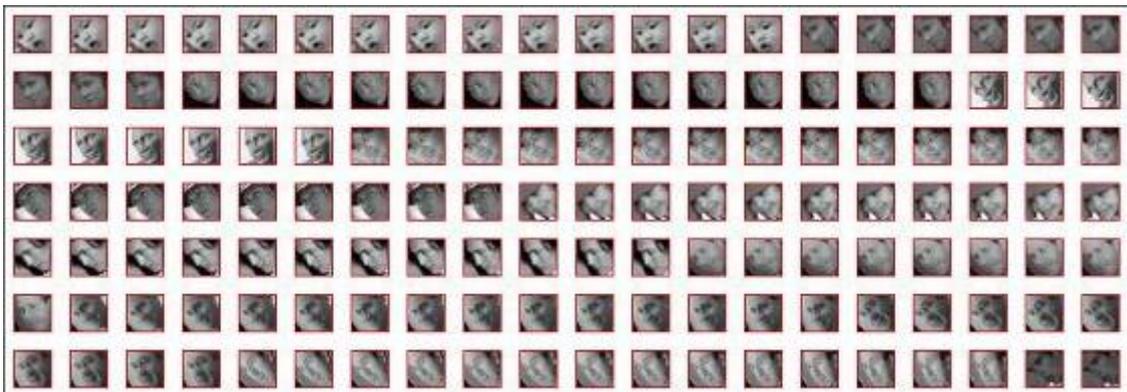


图 2.8 左半侧左倾斜人脸样本示例



图 2.9 左半侧右倾斜人脸样本示例

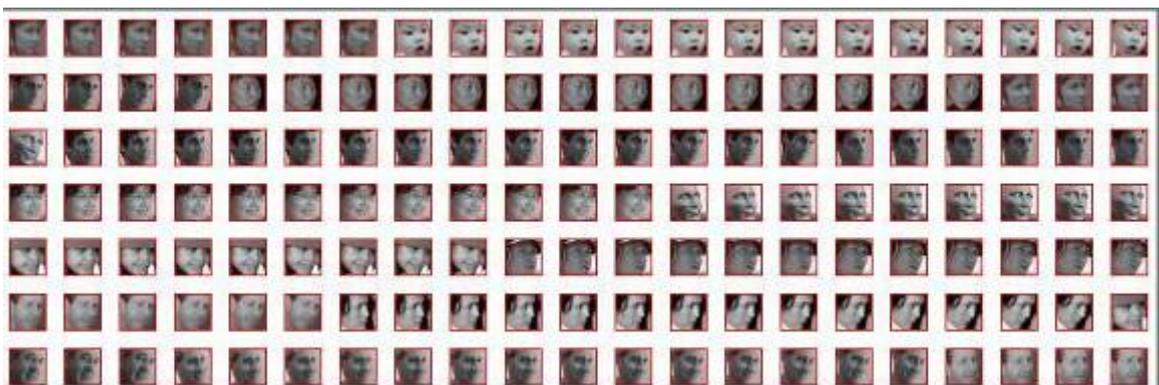


图 2.10 右半侧竖直人脸样本示例



图 2.11 右半侧左倾斜人脸样本示例

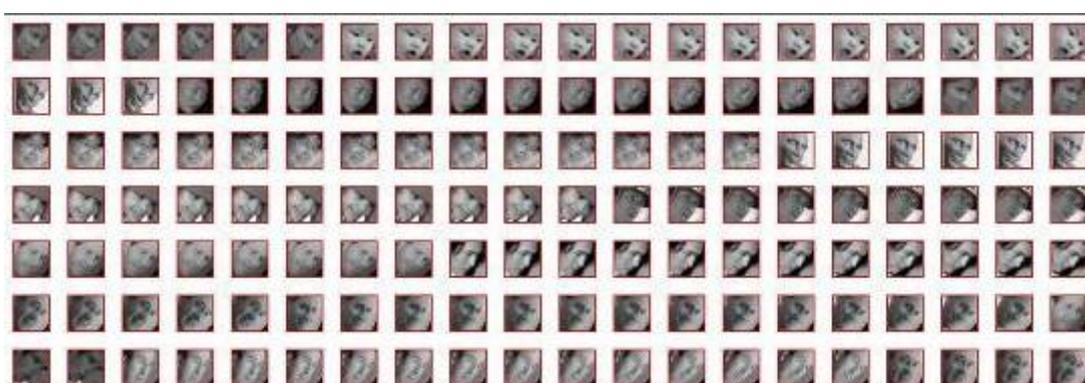


图 2.12 右半侧右倾斜人脸样本示例



图 2.13 左全侧竖直人脸样本示例



图 2.14 左全侧左倾斜人脸样本示例



图 2.15 左全侧倾斜人脸样本示例



图 2.16 右全侧竖直人脸样本示例

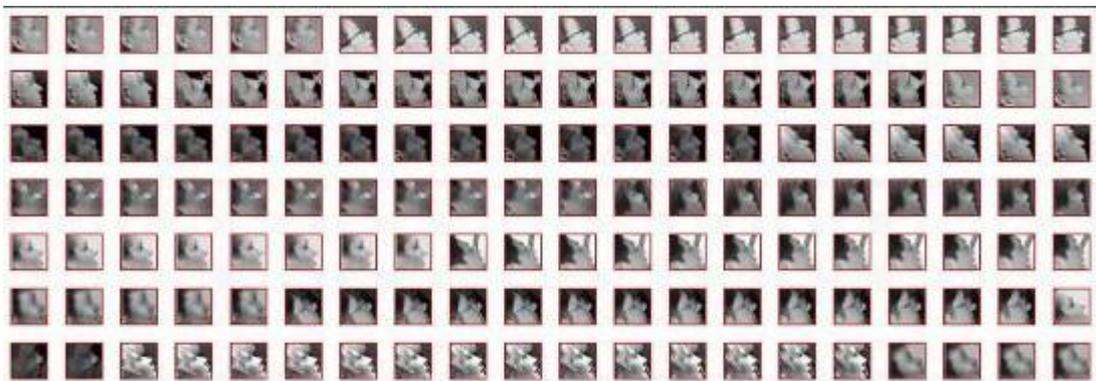


图 2.17 右全侧左倾斜人脸样本示例



图 2.18 右全侧右倾斜人脸样本示例

### 2.2.3 人脸样本的人工净化

在标注的过程会出现标注错误，使得特征点不准确，这样在错误的特征点基础上生成的样本并不是期望的人脸样本，一些出现标准点错误导致的不希望出现在最终人脸样本集中人脸如图 2.19(a)所示，为了对比，2.19(b)还给出了理想的人脸样本。

不希望出现的人脸样本会增加统计学习算法学习分类器的难度，干扰检测器的学习，影响分类性能。虽然也有其它方法来回避噪声样本，但都不能保证完全回避噪声样本的干扰。然而，通过人工去除“劣质样本”，这是可以避免的，我们对收集的原始人脸手工地进行了进一步的净化，人为对每个样本进行检查，并去掉每个不希望出现的样本。



(a)不期望的正面人脸样本



(b)期望的正面人脸样本

图 2.19 正面人脸样本集中不希望出现的和期望出现的一些样本

使用当前 Windows 操作系统对每个单独的图像文件进行大批量的样本检查是一个非常繁琐的工作。由于操作系统自身文件系统更新的原因，样本的翻页、删除都效率很低。另一方面，样本亮度、遮挡情况的不一致性导致了人检查过程中的负担，很多时候对于一幅比较暗的图片，我们需要仔细地观察才能定夺其作为人脸样本的质量。这在大批量样本检查时会给手工操作带来很大的麻烦。本文工作实现了一个独立的大批量样本检查工具，净化样本的程序包括了更好的显示方法（便于观看）、方便地去除非期望样本的方法，以降低人检查时的困难。

首先，根据特征点对齐、缩放、剪切后的样本集体存放于一个单独的文件中。这个文件的文件头记录了存放样本的数目、宽和高，然后紧接着文件头记录了从第一个样本开始的样本每个像素的原始数据。

对此文件进行样本去除的程序界面见图 2.20，使用程序中“初始化”可以打开任意一个样本文件，如图 2.20。“下一页”“上一页”工具栏按钮可以方便地进行样本的翻页，

左键点击一个样本就可以方便的从样本文件中去除一个样本，使用训练菜单上的子菜单“显示直方图或原始”即可进行选择显示原始样本或者是直方图均衡化后的样本，如图 2.21。图 2.21 中显示的样本是原始样本的显示，另外，在图 2.22 中，显示的样本是进行过直方图均衡化后的结果。最后，在程序界面的右边视图中显示了当前界面中的样本对应的文件的信息，以及文件中的位置，去除的样本个数等等，见图 2.23。

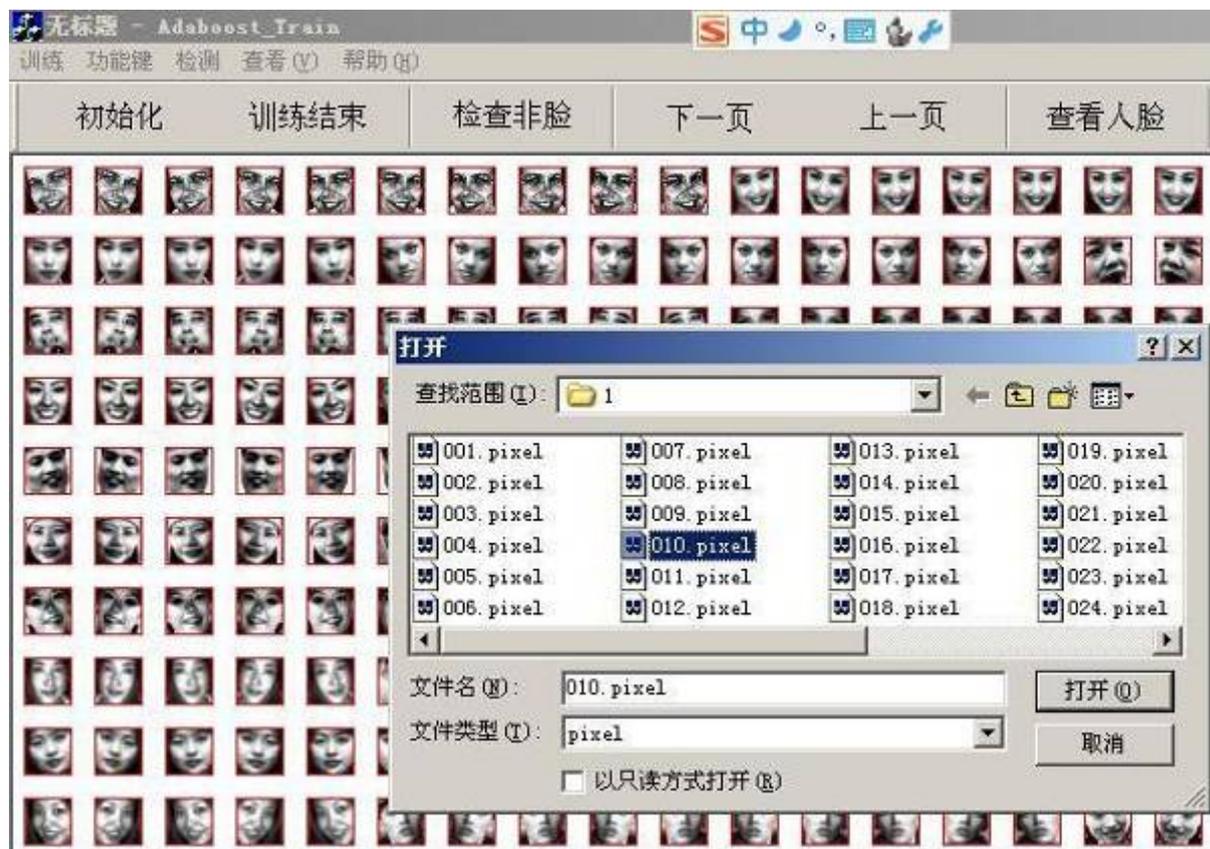


图 2.20 人工净化样本工具之打开文件



图 2.21 人工净化样本工具之直方图均衡化显示按钮

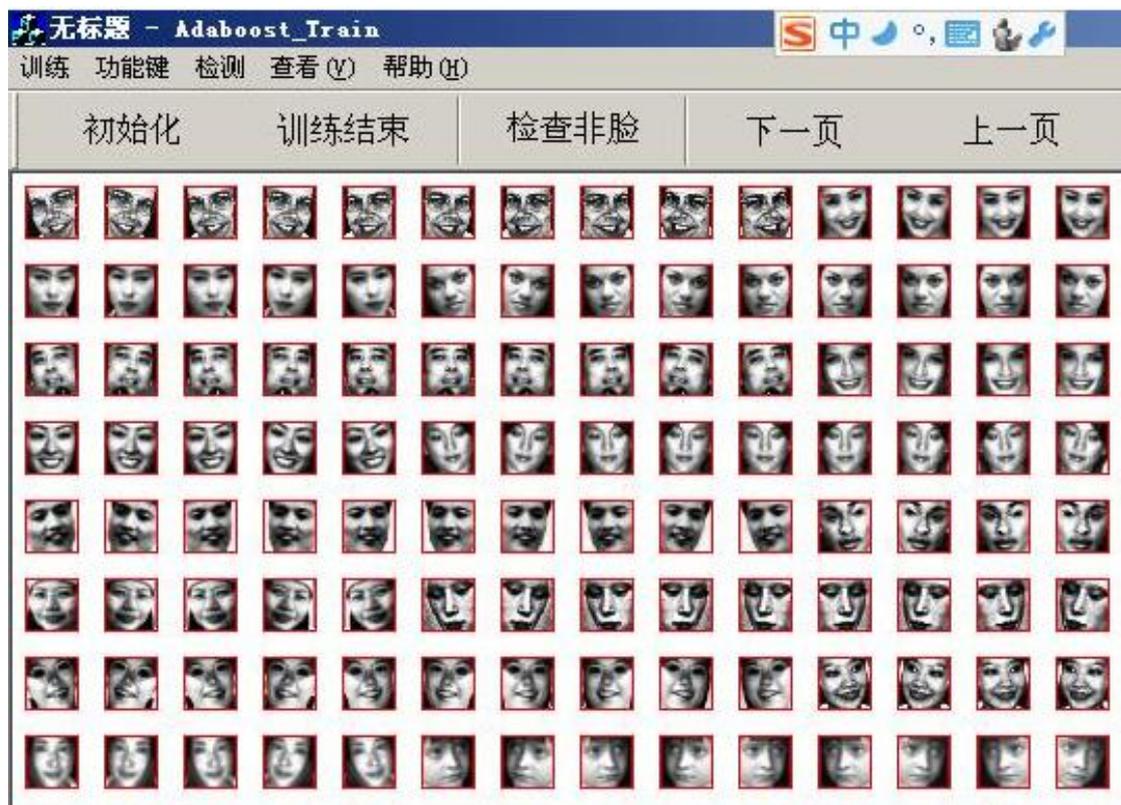


图 2.22 人工净化样本工具之直方图均衡化显示



图 2.23 人工净化样本工具之基本信息显示

## 2.3 反例数据集

用于学习的反例数据集应该尽可能的广泛，以使得分类器获得更好的分类性能。相对于正例人脸样本集的收集来说，反例样本只需要在确认不包含人脸样本的图像中剪切即可获得。这包含两个工作，其一是收集图像集合，其二是从收集到的图像中去除包含人脸的图像。如果人脸出现在了这些图像中，将会被选入训练集，导致学习得到的人脸检测器检测不到应该检测到的人脸，另外，由于增加了区分的难度，还会增加学习得到的分类器的复杂度。

和人脸样本的生成过程一样，非人脸样本依然可以生成虚拟样本。这可以通过对图像进行旋转、镜像等操作以获取更大的图像集合，也就是更多的反例集合。

本文工作收集了 15000 张图像，用于生成反例样本。这些图像都经过了严格的人工检查，去除了其中包含的人脸图像，另外，我们还采用了镜像的虚拟样本生成策略，生成了另外 15000 幅图像，所以提供给生成反例样本集合的大图像集合共包含了 30000 幅大图像。大图像集中一些手工去除人脸图像后的示例图片见图 2.24。

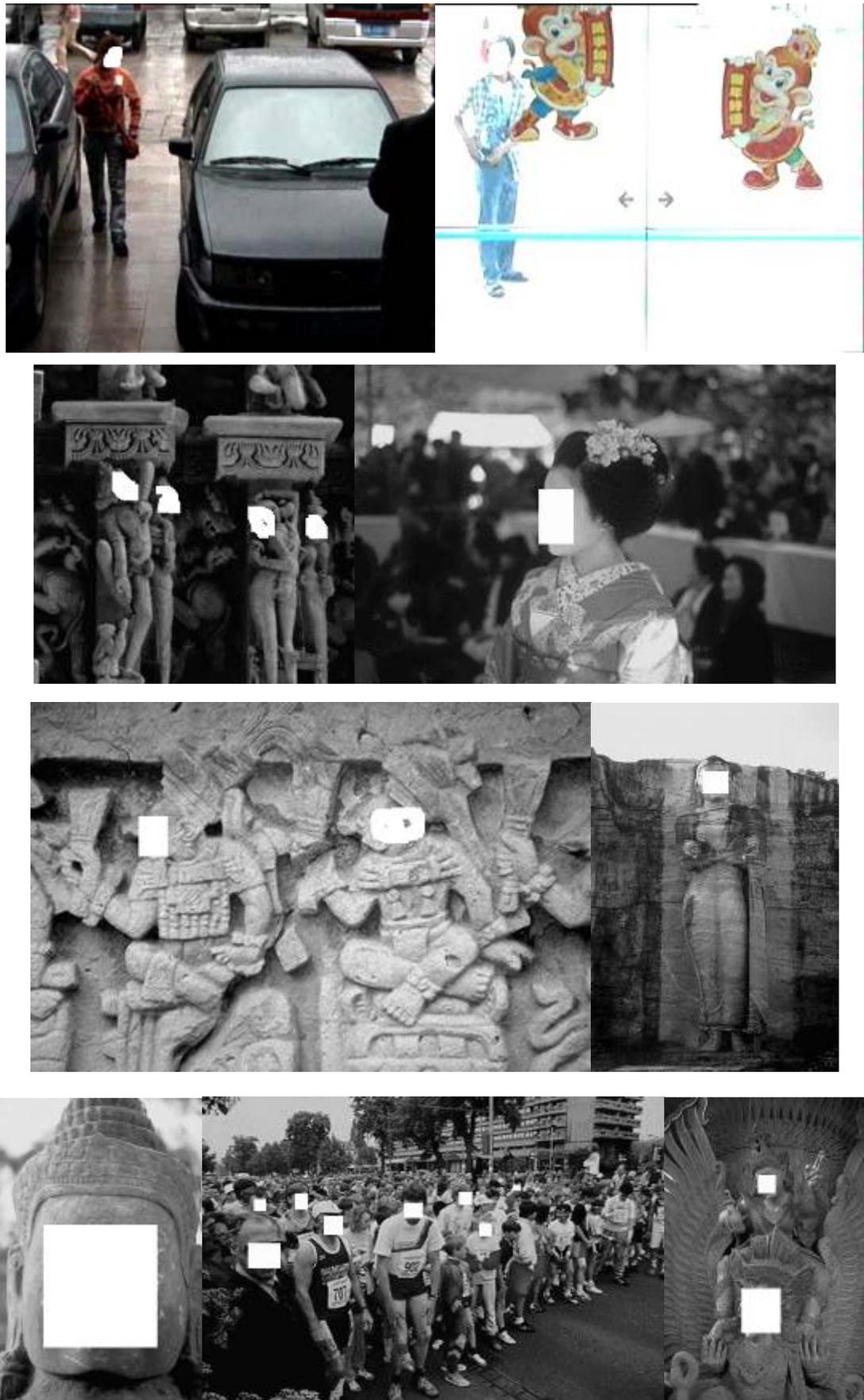


图 2.24 一些去除人脸图像区域后的图片

## 2.4 人脸检测算法的性能评估

对于一个已经检测到的人脸窗口来说，需要进行正确性的评测。窗口无论在尺度和位置上都不应该偏离真实人脸窗口过多。对于基于样本学习的分类器检测到的人脸窗口，理想情况下，一个正确的人脸检测窗口应该是和训练样本使用的特征点剪切参数获得的人脸窗口一致。但在实际的检测过程中，由于不同人脸模型本身存在的统计特性，最终检测到的人脸窗口也只是具有统计意义但并不精确。显然，人脸检测任务本身也必须允许一定检测到的人脸窗口与实际尺度和位置存在一定的偏差。图像中出现的人脸可以使用尺度和位置共三个参数来描述，中心横向位移  $x$ ，中心纵向位移  $y$ ，半径  $r$ 。真实人脸图像与检测到的人脸图像偏差可以使用如下模型进行表示。

1.  $rr_{\min} < r / r_g < rr_{\max}$
2.  $x_{\min} < x - x_g < x_{\max}$
3.  $y_{\min} < y - y_g < y_{\max}$

其中  $(x_g, y_g, r_g)$  表示手工标定的人脸位置和尺度，下标  $g$  是英文 *groundtruth* 的首字母。 $rr_{\min}$ 、 $rr_{\max}$  分别表示了尺度与手工标注的尺度在大小上分别允许的误差， $x_{\min}$ 、 $x_{\max}$ 、 $y_{\min}$ 、 $y_{\max}$  分别表示  $x$  方向和  $y$  方向中心位置允许向左右和上下偏移的距离。

以上对于人脸窗口精确程度的判定只是一个简单的模型。每个参数都只考虑了自身允许的偏差。显然，尺度和位置参数之间的偏差不是相互独立，而是存在一定的联系的。我们希望当检测到的人脸窗口位置和制作样本时相应的位置参数比较一致时，尺度上允许的偏差也相应可以较大；反之，当检测到的人脸窗口位置偏差较大，但还在允许范围内的時候，尺度上允许的偏差也相应收到限制，缩小尺度偏差范围。本文提出一种解决方案，就是采用一大一小两个矩形框来描述允许存在的正确检测窗口，当检测窗口中所有的像素存在于两个矩形框之间时，我们判定检测到的人脸窗口是正确的人脸检测窗口；反之则判定为误检。这两个矩形框的中心位置与人工标定的人脸窗口一致，可以表示为：

1.  $x_{\text{interRec}} = x_{\text{outerRec}} = x_g$
2.  $y_{\text{interRec}} = y_{\text{outerRec}} = y_g$

而尺度允许限制条件如下：

3.  $r_{\text{interRec}} = a * r_g$
4.  $r_{\text{outerRec}} = b * r_g$

在本文以后所有的实验中， $a$  取值为 0.6， $b$  取值为 1.5。此时人脸检测结果判定为正确时检测框允许存在的范围如图 2.25 中阴影区域所示。其中存在两个一大一小矩形框，

为规定的范围。蓝色矩形框为人工标注获得的人脸窗口。当且仅当检测到的窗口被大矩形框所包含，并且包含小矩形框时，是一个正确的检测结果；否则，认定为一个误检。

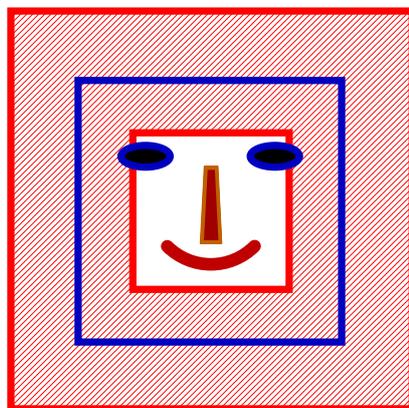


图 2.25 正面人脸正确检测窗口所在的区域（阴影显示）

以上所述内容是关于单个人脸检测窗口的准确性的一个度量。在单个人脸能够判断为正确检测或者错误检测的基础上，为了衡量一个人脸检测算法的统计特性，需要在—个测试集上对更多的人脸进行检测精度的测试。衡量一个人脸检测器检测精度需要考虑两方面的因素：检测率和误检率，定义如下：

检测率 = 检测到的人脸数 / 图像中所有的人脸数，

误检率 = 误检个数 / 图像中所有非人脸扫描窗口数。

当然，通常期望的是检测率越高越好，误检率越低越好，但这两个方面却是相互矛盾的。对于一个检测器，通过调整其中的一些参数可以获得在不同检测水平下的各个误检率，比如降低分类器的决策阈值可以提高检测器的检测率，与此同时更多的非人脸窗口被判定为—人脸，从而增加了误检率。因此，全面的评价—个检测器的检测性能需要考察其在不同检测水平上的误报率变化情况。将这些数据点绘制在横轴为误检率，纵轴为检测率的二维图中并连接—条连线，就可以得到评价检测精度常用的 ROC 曲线(Receive Operating Characteristic curve)。在以后的人脸检测器评测中，ROC 曲线将是一个非常重要并且直观的评测手段。

以上所述内容针对的是正面人脸，对于除正面以外的其它姿态的人脸，除检验过程中的特征点有所不同之外，其它方面和正面人脸的评估方式是一样的。

## 2.5 小结

本章讨论了人脸数据集的收集、样本的制作及净化和人脸检测结果的评价问题，介绍的内容在后面章节的实验中都将—被应用，如无特殊之处，将不再做额外的描述。



### 第三章 面向大规模训练样本集的矩阵结构瀑布型分类器学习

一般地，可以认为，学习算法的样本集越广泛，学习得到的分类器性能越好。所以，增大统计学习中使用的样本集，是提高最终学习得到的分类器分类性能的好方法。

在上一章中，讨论了样本集收集问题，并介绍了收集并生成的数十万大小的人脸训练样本集的情况。对于收集样本本身这个问题来说，通常情况下，收集样本会耗费一些人力物力，但是显然，这项工作随着各种数码设备的出现，互联网图片的共享，存储设备容量增加以及成本下降，正在变的越来越容易，所以，大规模样本集情况下如何应用当前存在的人脸检测算法的研究成为一个很有意义的问题。

本章将讨论如何在已经拥有一个大规模人脸样本集和非人脸样本集的情况下，进行瀑布型人脸检测器学习的问题，以使得在使用大规模样本集提高检测器的检测精度的同时，降低训练负担，提高训练速度。本章介绍的工作建立在当前最流行的基于 Haar 特征、Boosting 算法、瀑布型分类器的基础之上。本章提出的算法在瀑布式分类器的学习过程中，同时对大规模正例样本集进行自举的算法，由于在正例和反例交叉自举的过程中，得到的分类器呈矩阵形式的布局，本章提出的算法被叫做叫瀑布式分类器的矩阵式学习算法，矩阵式学习可以非常有效地提高训练效率。

本章下面的章节安排如下：3.1 节中回顾了基于 Boosting 瀑布型分类器的方法和理论，然后在 3.2 中描述了基于大规模样本集的矩阵式瀑布型分类器学习算法，3.3 节给出了另外一种计算复杂度比较高的算法及其分析，3.4 节中进一步给出了矩阵式学习中进一步提高学习效率可供采用的技术——特征继承，3.5 节中给出了实验与分析，最后在 3.6 节中给出了小结。

#### 3.1 Boosting 瀑布型分类器检测框架基础

近年来的人脸检测工作几乎都借鉴了 Viola 等人在 2001 年提出的基于 Haar 特征、Boosting 算法、瀑布型分类器框架结构的人脸检测方法，这是一个非常具有影响力的算法，最重要的是在检测速度上获得了突破性的进展，在检测精度和当时最好的结果具有可比性的同时，将正面人脸检测速度提高到近实时的检测速度。

本章所述的内容也是基于此框架下提出的。Boosting 瀑布型分类器包含三个层次，从上到下分别是瀑布型分类器，Boosting 子分类器和 Haar 特征。

下面我们首先介绍瀑布型分类器。瀑布型分类器框架并不是一个新概念，在几十年以前的就存在。在上个世纪七十年代和八十年代的自动目标识别系统中就已经被采用 [Bha86]。瀑布型分类器由多个子分类器构成，这些子分类器从前到后顺序排列，只有通过前面所有的子分类器，才能进入下一个子分类器。每个子分类器都进行一次二值判决，决定输出窗口是人脸还是非人脸。本质上，瀑布型分类器大大对非人脸空间中样本进行

序贯决策，简单的样本在靠前位置被排除，而复杂的样本在靠后的位置被排除。这符合客观世界中的非人脸样本的分布规律，大部分非人脸样本分布集中在简单非人脸样本中，而复杂的非人脸样本只占据了整体非人脸样本的一小部分，所以，瀑布型分类器能够很好的提高分类器的运行效率。瀑布型分类器的检测流程图见 1.6。

下面介绍 Boosting 算法。瀑布型分类器的每个子分类器都是一个由 Boosting 算法学习获得的强分类器，Boosting 是一类能够有效提升已有学习算法精度的方法 [FS99]。它最早的理论模型来源于 Valiant 所提出的 PAC (Probably Approximately Correct) [Val84, KV94]。将 Boosting 方法更加形式化来源于 Kearns 等人提出的这样一个问题：PAC 框架下，一种学习能力稍强于随机猜测的弱学习算法能否被转化为一个精度很高的强学习算法。这个问题的解决方案之一就是 Boosting 算法。Schapire 在 1989 年首先给出了一个多项式复杂度的 Boosting 算法 [Sch90]，而几年后，Freund 又提出了另一种效率更高的 BBM 算法(Boost By Majority)[Fre95]。其后，Freund 等人进一步提出了 BBM 算法的升级版本——AdaBoost [FS95]，不同于以前的 Boosting 算法，AdaBoost 算法解决了许多实际操作中的困难，能够自行适应到不同弱分类器的错误率。这也是 AdaBoost 其中 Ada 的由来 (Adaptive:适应)。由于其简单易用、性能出众，被应用于各种领域 [SS00, ILS00, FISS98, ASS99, Col00, EMR00, HSO00, MLR01, WRR01, TV00, MFLS01, MWGJ00, ORM00]。1999 年，Schapire 进一步将 AdaBoost 中弱学习得到的分类器从二值域扩展到了实数域，扩展后的算法叫做 Real AdaBoost 算法 [SS99]。后来，研究者从统计学的角度对 AdaBoost 进行分析，进一步提出了 LogitBoost, GentleBoost, BrownBoost 等等。

本文的重点并不在于统计学习算法本身，更为详细的描述可以参考各相关文献。本文采用的 Boosting 学习算法是 Boosting 家族中的 Real AdaBoost 学习算法，Real AdaBoost 学习算法见图 3.1 所示。

### 3.2 面向大规模训练样本的矩阵式学习

在基于 Boosting 瀑布型分类器的检测算法结构中，即使在样本集不是很大的情况下，训练起来就需要花费较多的时间，精确地说 Viola 等人的正面人脸检测器花了 2 个星期进行训练(采用了正面人脸样本五千多个)。

另一方面，为了提高检测器的分类性能，自然的会想到去增加训练集中样本的数目。在现在的条件下，数码产品、网络的发展，使收集更多的样本变成一个并不是特别困难的工作。另外，针对原始样本，还可以进行小范围内的仿射变换获得更多的样本，比如说，现在我们拥有正面人脸图片 10000 个，那么我们可以通过镜像一次、平面内小幅度旋转 5 次，获得 100000 个样本，这在第二章中已经有所陈述。

现在，在已经拥有数十万或者更多正例样本的条件下，考虑到原来瀑布式分类器学习的速度，直接在大规模样本集上进行学习的话，效率会非常低，几乎成为一个不可完成的任务，那么，怎样提高训练效率，同时还要保证大样本集带来的好处——提高分类

器的检测性能?

给定:  $(x_1, y_1), \dots, (x_m, y_m); x_i \in X, y_i \in \{-1, +1\}$

初始化  $D_1(i)=1/m$ 。

For  $t=1, \dots, T$ :

- 1) 使用分布  $D_t$  训练每个弱分类器  $h_t: X \rightarrow \mathbb{R}$ ;
  - a) 使用弱学习算法对样本空间进行划分, 得到  $X_1, \dots, X_n$
  - b) 在训练样本的概率分布  $D_t$  下, 计算:

$$W_t^j = \sum_{i: x \in X_j \wedge y_i = l} D_t(x) |l = +1 \text{ 或 } -1$$

- c) 设置弱分类器在这个划分上的输出:

$$\forall x \in X_j \quad h(x) = \frac{1}{2} \ln \left( \frac{W_+^j + \varepsilon}{W_-^j + \varepsilon} \right) \quad j = 1, \dots, n$$

其中  $\varepsilon$  为一个小的正常数。

- d) 计算归一化因子:

$$Z = 2 \sum_j \sqrt{W_+^j W_-^j}。$$

- 2) 选择使得  $Z$  值最小的  $h_t$ 。
- 3) 更新训练样本概率分布

$$D_{t+1}(x_i) = D_t(x_i) \frac{\exp[-y_i h_t(x_i)]}{\widetilde{Z}_t}, i = 1, \dots, m$$

其中  $\widetilde{Z}_t$  为归一化因子, 使得  $D_{t+1}$  为一个概率分布

- 最终, 强分类器为:

$$H(x) = \text{sign} \left( \sum_{t=1}^T h_t(x) \right)$$

图 3.1 Real AdaBoost 学习算法

其实, 在瀑布式分类器结构中, 已经包含了对于大规模反例训练集的解决办法。这种方法的思想来自于 Sung 等人[SP98, Sun96]提出的叫做自举(Bootstrap)的样本采样方法。自举算法是一种大规模样本集的下采样办法, 它的核心思想是: 既然样本集很大, 那么费时的训练操作并不直接在原始的大规模样本集上进行, 而是通过从中采样出一个小的样本集替代大规模样本集, 并进行分类器的学习。为了保证在大规模样本集上分类器的性能, 自举过程中的重采样采取了回补样本操作, 具体来说, 就是对使用当前训练样本集训练得到的分类器, 在大规模样本集上进行测试, 对分类器测试错误的样本, 选择一部分重新加到当前的训练集中, 扩大样本集, 增加其从训练分类器的角度对大样本集的代表性。如此循环, 直到训练出来的分类器满足预期的目标为止。图 1.16 给出了瀑布式分类器的结构。瀑布式分类器在每个子分类器学习之前, 先进行自举反例样本, 所

以每增加一个子分类器后都对反例的处理能力变得更强。

基于上述自举思想,针对同时拥有大规模正例样本集和反例样本集的情况,本文提出了在瀑布式分类器的学习过程中,同时对大规模正例样本集进行自举的算法,如图 3.5 所示,由于在正例和反例交叉自举的过程中,得到的分类器呈矩阵式的分布,所以此算法被称作瀑布式分类器的矩阵式学习算法。

在本章后面的介绍中,3.2.1 小节介绍了矩阵式瀑布式分类器学习算法如何为传统瀑布使分类器学习算法引入正例样本前人提出的自举方法;然后,在此基础上 3.2.2 小节提出了一种针对矩阵式学习算法而设计的累加式正例样本自举方法。另外,由于本章以后的描述中,将会使用很多符号,为了方便阅读时参考,这些符号被集中在图 3.2 中进行了解释。

$C(i,j)$	矩阵式学习中,第 $i$ 层子分类器,第 $j$ 次正例样本自举学习得到的分类器
$C'(i,j)$	线式学习中,第 $i$ 层子分类器,第 $j$ 次正例样本自举学习得到的分类器
$N(i,j)$	$C(i,j)$ 学习使用的反例用本
$N'(i)$	$C'(i,j)$ 学习使用的反例用本
$P(i,j)$	$C(i,j)$ 学习使用的正例用本
$P'(i)$	$C'(i,j)$ 学习使用的正例用本
$F_j$	$C(i,j)$ 包含特征的数目
$f_{jk}$	$C(i,j)$ 中第 $k$ 个特征
$H_i$	矩阵式学习中第 $i$ 次正例自举的迭代次数
$B$	瀑布式分类器的总层数
$M_i$	矩阵式学习中第 $i$ 层子分类器的正例自举迭代次数
$M$	线式学习中正例样本自举迭代次数

图 3.2 本章中用到的符号及涵义

### 3.2.1 自举正例样本

自举算法是由 Sung 等人[SP98, Sun96]提出的,提出的目的是对大规模的样本集进行下采样,以收集到可以代表大规模样本集的一个小规模样本集来进行分类器的学习,Sung 等人提出的自举算法见图 3.3。训练初始的时候,分类器的训练从一个从大规模样本集中随机挑选的小规模样本集开始,得到第一个分类器,然后使用这个分类器,对大规模样本集进行测试,对于那些测试失败的样本,随机选择一小部分到当前训练样本集,如此反复迭代,直到分类器达到了目标设定值或者预先设定的最大自举迭代次数,退出。自举算法的一个特点就是在收集真正用于训练的小样本集时,直接以分类器在大规模样本集上的分类性能为指导,这样,自举算法在避免大规模样本集直接参与训练、提高训练效率的同时获得了在大规模样本集上性能可以保证的分类器。

- 1) 初始从一个从大规模样本集中随机挑选出来的样本构成的小规模样本集;
- 2) 使用当前样本集训练一个分类器;
- 3) 在大样本集上运行分类器。搜集分类错误的样本。加入当前训练集。
- 4) 返回 2), 继续运行, 直到分类器达到目标性能或者样本迭代此处超过指定数目。

图 3.3 Sung 等人提出的自举算法

在 Viola 等人提出的瀑布式分类器学习算法中, 自举算法只是被用来进行对大规模反例样本集进行收集, 他们对自举算法的使用和 Sung 等人的方法有些不同, 在每次反例自举之后, 为了构建一个具有层次结构的瀑布型分类器, 会产生一个使用当前反例样本集训练得到的子分类器, 并且对于用于训练分类器的反例样本中被本子分类器分类正确的反例样本会被抛弃, 这一点可以通过下面对传统瀑布式分类器学习算法的介绍理解的更加清楚。

Viola 等人提出的瀑布式分类器学习算法中, 每个子分类器由 AdaBoost 学习算法从 Haar 特征候选集合中学习得到, 每次反例自举后, 使用自举得到的反例样本集, 结合整个训练过程中始终保持不变的正例训练样本集, 训练子分类器。在一个子分类器的总体训练过程中, 反例样本集和正例样本集都保持不变, 训练的终止条件是训练获得的子分类器在一个预先收集好的校验集上的检测性能, 如果测试后, 检测性能未能满足目标性能, 则增加特征, 使得 AdaBoost 学习得到的强分类器增强, 以期进一步提高在校验集上的检测性能; 如果测试后, 检测性能满足目标性能, 则使用自举方法更新反例样本, 进入瀑布型分类器下一个子分类器的训练。

从以上描述可以看出, 在 Viola 的方法中, 正例样本集始终保持不变, 显然在正例样本集规模很大的情况下, 由于效率的问题往往不可能将所有大规模样本集中的样本都用来直接参与训练。

瀑布型分类器的矩阵式学习算法采用了正例样本自举来降低大规模正例样本集同时参与训练时的计算代价问题。瀑布式分类器的矩阵式学习算法的整体框架如图 3.5 所示, 图中  $N_i$  和  $P(\cdot)$  分别表示自举过程中的正例样本集和反例样本集,  $C(\cdot)$  表示子分类器, 具体意义如图 2 中所描述。从图中可以看出, 图 3.5 中水平方向的带箭头的直线表示正例样本的自举流程, 垂直方向的带箭头的直线表示了反例样本的自举流程, 箭头的方向表示自举迭代的推进, 正例样本自举和反例样本自举轮流交叉进行。椭圆内包含的子分类器构成了最终的瀑布式分类器, 椭圆内包含的正例样本集和反例样本集是最终训练瀑布型分类器子分类器使用的正例训练样本集和反例训练样本集。

当然, 在矩阵式学习中使用的正例样本自举方法可以直接将 Sung 等人提出的方法搬用, 过程如图 3.4 中描述算法所示, 图 3.5 所示的矩阵式学习中每行都进行一次如图

3.4 所示的正例样本自举过程, 最终收集到一个训练人脸样本集能够训练出一个子分类器来满足预先设定的目标性能。此过程在图 3.5 中使用带箭头的蓝线来示意。另外, 反例样本自举过程使用带箭头的紫色线来示意。

- 1) 开始于一个从大规模正例样本集中随机挑选的小规模正例训练集;
  - 2) 使用当前的训练样本集训练能够在其自身上达到检测率  $d_{min}$  误检率  $f_{max}$  目标的强分类器;
  - 3) 在大规模正例样本集上测试分类器, 得到检测率  $d$ 。
- 如果  $d$  超过  $d_{min}$ , 当前分类器训练结束; 否则, 搜集一部分分类错误的样本加入到当前小规模正例样本训练集; 返回 2), 继续。

图 3.4 子分类器学习中的正例样本自举

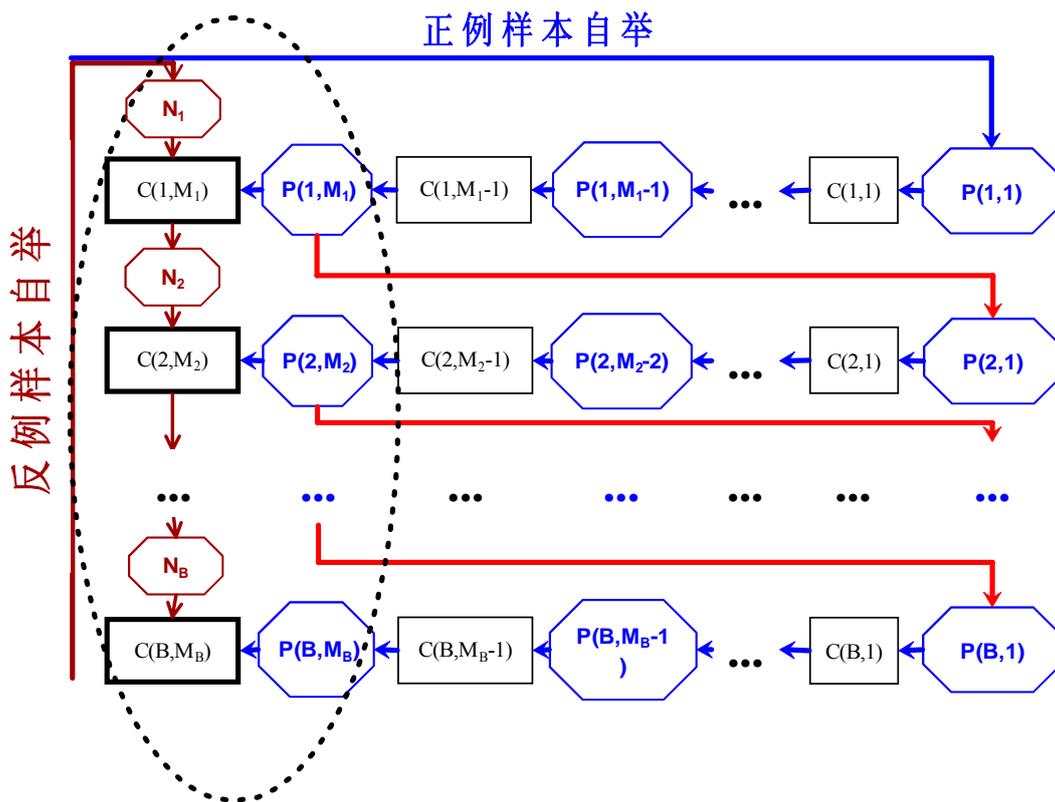


图 3.5 矩阵式学习

从正例样本自举过程可以看到, 为了保证小规模样本集上训练得到的分类器能够在大规模样本集上获得预期检测性能, 训练得到的分类器首先在小规模训练集上达到预期目标性能。然后, 反复进行正例样本自举, 直到训练得到的分类器在大规模样本集上也达到预期目标性能为止。在正例样本自举过程中, 相邻的第  $j$  个和第  $j+1$  个小规模正例样本训练集之间存在如下关系:

$$P(i, j) \subset P(i, j+1) \quad 0 < i < B+1, 0 < j < M_i$$

其中  $P(i, j)$  表示第  $i$  个分类器训练过程中, 第  $j$  次自举样本后的正例训练样本集,  $B$  是瀑布式分类器总的层数,  $M_i$  是第  $i$  个子分类器的训练共进行的正例样本自举迭代次数。

### 3.2.2 累加式样本自举方法

上节介绍的瀑布式分类器学习算法中, 正例样本自举使用了 Sung 等人提出的原始自举方法, 但是这并不是最有效率的方法, 针对矩阵式学习, 样本自举的效率可以被进一步提高, 本节介绍一种专门为瀑布型分类器的矩阵式学习算法设计的自举方法。

在矩阵式学习算法中, 正例样本自举的过程使得小规模样本集的对大规模样本集的代表能力逐步提高。现在考虑瀑布型分类器中两个相邻子分类器训练使用的正例样本集的关系。假设在第  $i$  个分类器训练完成后, 最终使用的训练正例训练样本集为  $P(i)$ , 然后对反例样本集进行扩充, 以采集更加稠密的样本。然后进行分类器的训练。考虑到每个顺序在后的子分类器的反例样本都是对前面分类器的稠密采样, 对于当前的反例训练集来说, 前面层使用的小规模正例样本训练集仍然对大规模正例样本集具有一定的代表能力。这也就意味着每个子分类器的正例样本自举过程不必重新从一个随机挑选的样本集开始, 这是 Sung 等人提出的自举算法中采用的策略, 而是将上一个子分类器的训练使用的正例样本集直接作为下一个紧接其后的子分类器样本自举过程的初始样本集, 这可以减少自举过程迭代的次数, 从而加速训练过程。正式的说, 两个正例训练样本集之间的关系可以表示为:

$$P(i, 1) = P(i-1, M_{i-1}) \quad 1 < i < B+1$$

在整个瀑布式分类器训练过程中, 对于从前到后不同子分类器之间自举过程中的正例样本集, 存在顺序在前的样本集被顺序在后样本集包含关系。显然, 正例样本训练集规模从前到后逐步累加, 所以, 新的自举方法称作累加式自举方法。在图 3.5 示意了累加式自举方法, 前一个子分类器的训练向下一个子分类器的训练转换的时候, 正例样本被保留, 使用了带箭头的红线来表示此操作。除第一个子分类器之外, 累加式样本自举方法大大降低了的每个子分类器正例样本自举的迭代次数, 对于第一个子分类器的训练, 也就是累加式正例样本自举的开始的时候, 和 Sung 等人方法一样, 初始训练集仍然随机挑选得到。

### 3.3 讨论: 另一种计算复杂度高的算法

对比于矩阵式学习, 对于同时拥有大规模正例样本集和大规模反例样本集的情况下, 存在着另外一种可供选择的瀑布型分类器训练方法, 这种方法也采用了正例样本自举的方法来进行小规模正例样本集的收集, 反例样本的自举和矩阵式方法一样。在矩阵式学习中, 正例样本自举是建立在对每个瀑布型分类器的子分类器上的, 也就是说, 判断一

个样本集是否满足要求，能够获得目标性能的分类器是每个子分类器。不同于矩阵式学习，本节将要对比的方法中的正例样本自举，并不建立在某个子分类器上，而是将整个瀑布型分类器作为一个整体，来进行正例样本的自举，如图 3.6 所示。图中每个瀑布型分类器被包含在一个直立的矩形框中，正例样本自举从右向左按照水平带箭头线段方向执行，箭头是执行的方向。由于本方法中正例样本自举过程将瀑布型分类器串联成了一条横线，为了方便讨论，这种方法被称为线性结构学习，图 3.6 中的符号含义见图 3.2。

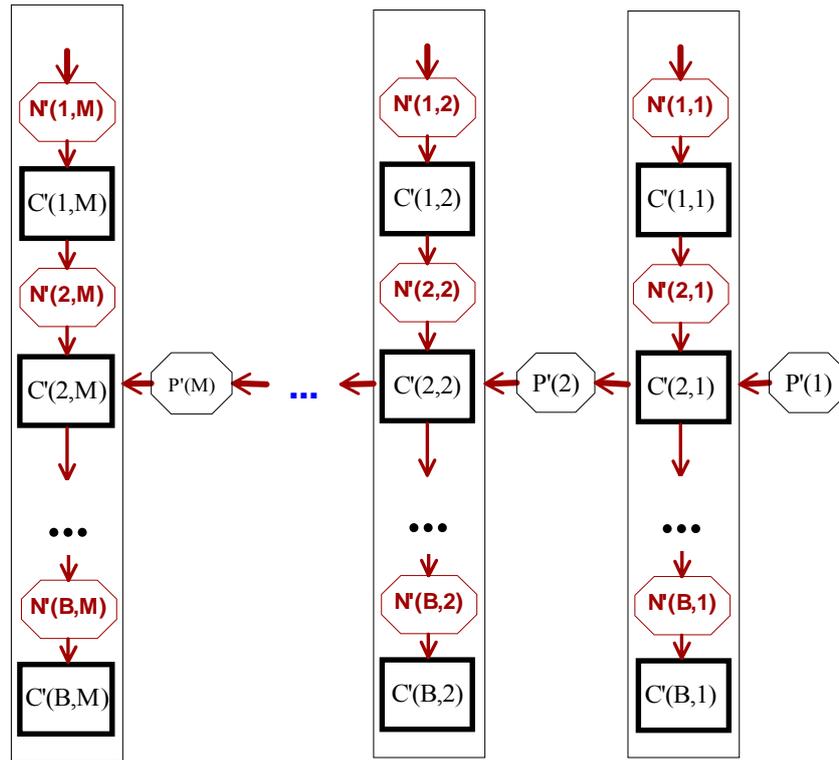


图 3.6 线性结构学习

下面对两种不同的方法：矩阵式学习和线性结构学习，给出一个粗略的计算复杂度上的简单分析。

分析之前，我们先将瀑布型分类器的训练过程划分为两个不同的运行过程并讨论其计算代价。瀑布型分类器的学习可以分为两个过程，第一个过程是使用自举的方式从大样本集中收集部分有代表性的正例样本，并使用收集到的正例样本作为实际使用的训练集进行瀑布型分类器子分类器的学习。为了方便讨论，这个过程被称为 CL。在 CL 中，从大规模样本集中使用自举方式收集正例样本的过程占用的时间很少，大部分时间都被瀑布型分类器子分类器的学习所占用，它几乎占用了 CL 过程的所有时间。第二个过程是使用自举方法从大规模范例样本集中收集反例样本，这个过程表示为 CN。

下面我们分别就 CN 过程和 CL 过程对矩阵式学习和线性结构学习这两种方法进行计算上的分析。首先，对于 CN 过程，在矩阵式学习过程中，只需要  $B$  次执行，其中  $B$  表示瀑布型分类器中包含的子分类器的数目。但是在线性结构学习中，CN 需要执行  $M \times B$  次，其中  $M$  表示线性结构学习中自举正例样本的次数，这比矩阵式学习多了  $M$  倍。

对于 CL 过程，矩阵式学习需要  $\sum_{k=1}^B M_k$  次执行，而线性结构学习需要  $M \times B$  次执行。

假设自举算法的参数，比如正例样本初始样本数目、每次迭代最大增加样本数目等等的设置等对两种算法都一样，那么，在线性结构学习中，根据反例样本的难易区分程度，可以近似认为自举正例样本的迭代次数是  $M$ ，在矩阵式学习中，相应的数目是  $\sum_{k=1}^B M_k$ 。由于使用相同的大规模反例样本集和正例样本集，其它训练参数一样，分类器学习的难度从样本的角度来说是一样的，可以近似认为  $M$  和  $\sum_{k=1}^B M_k$  在一个数量级上。进一步，如果矩阵式学习和线性结构学习的训练正例样本集也认为是接近的，那么在对于 CL 过程，线性结构学习所需要的计算代价是矩阵式学习的  $B$  倍。这导致矩阵式学习的效率要比线性结构学习的效率高很多。

### 3.4 特征继承

在矩阵式学习中，正例样本自举过程中的两个相邻“临时”分类器由于样本之间的相互关联，所以学到的特征也具有关联性。实际实验中，累加式正例样本自举算法中每次新一轮的自举获得的正例样本集，相对于上一轮获得的正例样本集来说，新增加的正例样本数目都很少。两个正例样本集合大量相同样本的存在造成两个样本集的高度关联性，这种样本集之间的高关联可以被进一步利用来提高训练效率。由于训练样本集的高度关联，所以训练得到的分类器也具有很强的关联性，这就给我们一个提示，我们可以在使用新样本集进行下一个子分类器的学习时，直接将前边子分类器学到的特征使用新样本集进行分类器的重新训练，具体来说，就是对前面子分类器的特征顺序不变，但是对每个特征重新使用 Boosting 算法使用新样本集对每个特征更新弱分类器，由于此过程去除了 Boosting 算法中特征选择时扫描所有候选特征的过程，而通过一个特征更新其弱分类器的过程耗时又非常少，所以总体需要时间很少。由于只有前面分类器的特征被新的分类器拿过来使用，前面分类器的弱分类器则被抛弃，所以我们称此技术为特征继承。为了更好的理解特征继承技术，图 3.7 图示了在瀑布型分类器的矩阵式学习中采用了特征继承之后，第  $i$  个子分类器的样本自举产生的每个“临时”子分类器的特征与其它子分类器的特征之间的关系。

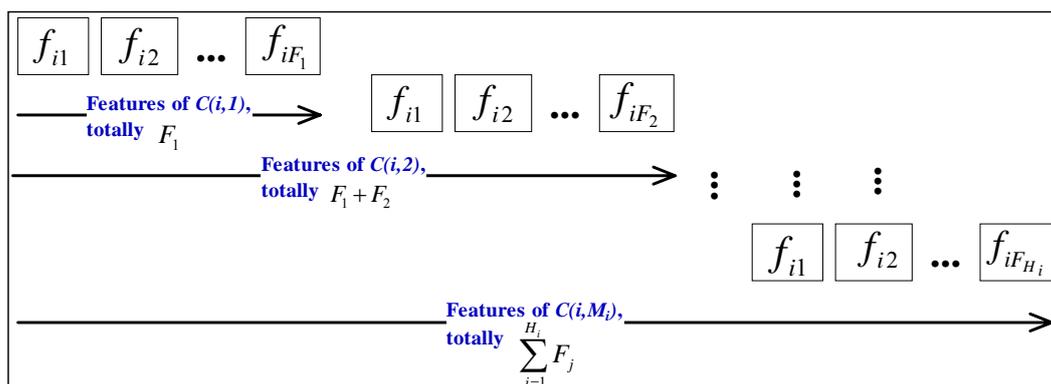


图 3.7 使用特征继承技术后第  $i$  个子分类器的特征构成

与以前的训练方法一样，当所有前面的特征继承过来并对相应弱分类器进行更新之后，训练得到的强分类器如果不能满足目标分类性能指标，那么仍然需要对所有候选特征进行扫描挑选新的特征加入并进行弱分类器的训练，以进一步增强分类器，最终达到目标性能。

正式地，对于特征继承技术，如果使用  $FS(C)$  表示分类器  $C$  中包含的特征集合，第  $i$  个子分类器的第  $j$  次正例样本自举和第  $j+1$  次自举所生成的分类器之间的样本集有如下关系：

$$FS(C(i, j)) \subset FS(C(i, j+1)) \quad 0 < i < B+1, 0 < j < M_i+1$$

### 3.5 实验

上面描述了引入大样本集之后的矩阵式学习算法和特征继承技术，下面通过实验来验证算法的有效性。

#### 3.5.1 实验设置

训练使用的 Haar 特征包含了五种类型，见图 3.8 所示。相对于 Viola 等人使用的 4 种特征，增加了一种上中下三个矩形的特征，这是为了增强特征的描述能力。在位置上进行穷举筛选后特征个数为 31728 个。

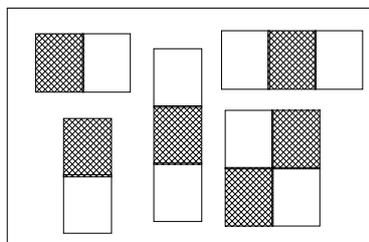


图 3.8 本文实验使用的五种 Haar 特征类型

矩阵式学习算法中的每个子分类器使用 Real AdaBoost 算法来进行学习，每个子分类器的目标训练性能设置为：检测率至少 0.9999，误检率不高于 0.4。

瀑布型分类器每个子分类器的训练过程中，自举得到的反例样本数目设置为 10000。矩阵式学习的正例样本自举过程中，初始训练样本集设置为 3000 个训练人脸样本，这些样本从如第二章所述的大规模样本集中随机挑选获得，每次正例样本自举最多增加训练人脸样本的数目设置为 500。

检测图像时，为了检测不同大小的人脸，金字塔扫描图像过程中缩放图像缩放系数设置为 0.8。

如果没有特别的声明，所有实验都在一个普通奔腾 3.2GHz，1GB 内存的个人电脑上进行。

### 3.5.2 正面人脸检测

#### 正面人脸检测上的训练效率调查

首先，除了人脸训练样本集不同之外，限定所有其它训练条件（比如反例样本集大小，特征集的大小）同传统学习算法一样。这样，训练时间只受到训练人脸样本集大小的影响。表 3.1 中给出了部分矩阵式学习中瀑布型分类器的某些子分类器最终训练使用的训练样本集的大小，从表中可以看出，第一个子分类器最终只使用了 3612 个人脸样本，对比传统的方法，训练使用的人脸样本大大减少。随着瀑布型分类器层级的增加，训练样本也随之相应增加。这是因为反例样本集在自举的过程中，变得更加复杂，所以也需要更多的人脸样本才能代表大规模人脸样本集来和非人脸样本集进行区分。相对于传统的固定训练人脸样本集的方法，矩阵式学习过程中，训练人脸样本数目随着反例的可区分程度动态调节，提高了分类器训练的效率。

表 3.1 一些分类器最终自举得到的人脸训练集大小

层序号	1	2	8	11	15	19
最终人脸训练集大小	3612	3695	6629	8260	10498	12899

表 3.2 累加自举方法和原始自举方法达到相同性能指标需要的自举迭代总次数对比

层序号	1	2	4	6	8	10	12	14	16
累加自举方法	5	4	4	4	5	5	7	7	6
原始自举方法	5	6	7	8	11	17	20	22	24

表 3.3 第 15 个子分类器自举过程中，不使用特征继承技术，训练各子分类器需要新添加特征的数目和相应的训练样本数目

自举迭代序号	1	2	3	4	5	6	Total
新添加特征数目	38	49	39	42	41	46	216
人脸训练样本数目	10087	10291	10392	10482	10533	10602	——

表 3.4 第 15 个子分类器自举过程中，使用特征继承后，训练各子分类器需要新添加特征的数目和相应的训练样本数目

自举迭代序号	1	2	3	Total
新添加特征数目	26	13	1	40
人脸训练样本数目	10087	10447	10498	—

第二，我们调查了累加式正例样本自举方法相对于 Sung 等人提出的自举方法对训练效率的影响。累加式样本自举方法和 Sung 等人提出的样本自举方法相比自举样本效率的提高主要表现在样本自举迭代次数减少，表 3.2 给出了瀑布型分类器中的一些子分类器上两种样本自举方法训练一个满足相同目标性能的子分类器的分别需要的自举迭代次数，表中“原始自举方法”表示 Sung 等人提出的自举方法，从表中可以看出，累加式正例样本自举方法的迭代次数明显少于原始自举方法。随着层数的增加，原始方法迭代次数不断增加，但是累加式自举方法基本稳定在 4 到 8 次，这充分表明累加式样本自举方法能够有效地提高训练效率。

第三，我们调查了特征继承技术对训练效率的影响，特征继承对训练效率的提高主要体现在每个瀑布型分类器的训练过程中新挑选特征的数目上，新挑选的特征越多，算法需要越多的时间去训练。由于挑选特征时正例样本训练样本数目不同，我们同时对正例训练样本数目也进行了调查。表 3.3 和表 3.4 以瀑布型分类器的第 15 层子分类器作为例子分别给出了每次自举时使用特征继承技术之后和未使用时需要新添加特征的个数和人脸训练样本数目。为了更好的对比，新添加的特征总数目也在最后一列给出。由表中可以看出，不使用特征继承方法新增特征 216 次，但是使用特征继承方法之后，新增特征次数降低为 40 次，增加特征数目大大减少。最终训练得到的子分类器包含特征在不使用特征继承和使用的情况下分别为 46 和 40 个，相对于采用特征继承技术之后的训练方法，特征数目在不使用此方法的时候居然更多，这是一个很有趣的现象，可能是由于 Real AdaBoost 学习算法本身并不是一个最优的分类器学习算法导致的。

最后，我们调查了本文提出的特征继承方法与 Huang 等提出的嵌入式瀑布型分类器 [HAWL04] 方法的对于继承能力方面的对比。由于重新训练了每个前面层拥有的特征并生成新的弱分类器，所以本文提出的特征继承方法在利用以前特征的分辨能力上更胜一筹，图 3.9 给出了瀑布型分类器训练过程中的各个子分类器使用两种方法需要在候选特征集中重新挑选特征次数的对比图，显然，我们的方法需要重新挑选特征的次数更少，效率更高。而图 3.10 给出了第十层开始训练之前由文献 [HAWL04] 中所述方法对前面层所有的特征生成的嵌入式弱分类器和本文方法形成的强分类器的人脸样本和非人脸样本的在分类器信度上的直方图统计曲线，显然我们的方法能够更好的区分人脸样本和非人脸样本，为了更精确的描述两个分布的区分性，我们分别计算了两种方法中人脸分布和非人脸分布之间的 Bhattacharyya 距离。Bhattacharyya 距离越大，说明两个分布之间的可区分性越好。本文方法获得的人脸与非人脸分布之间的 Bhattacharyya 距离为 243.8，嵌

入式瀑布型分类器方法获得的人脸与非人脸分布之间的 Bhattacharyya 距离为 201.56, 我们的方法人脸分布非人脸分布距离更远, 区分性更好。

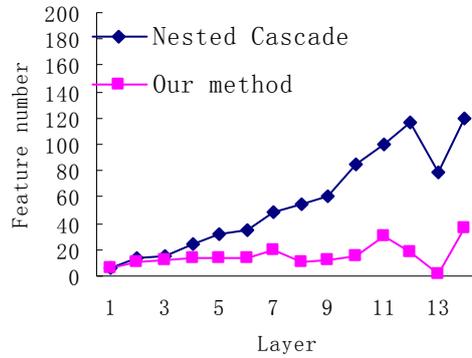


图 3.9 重新挑选特征数目对比

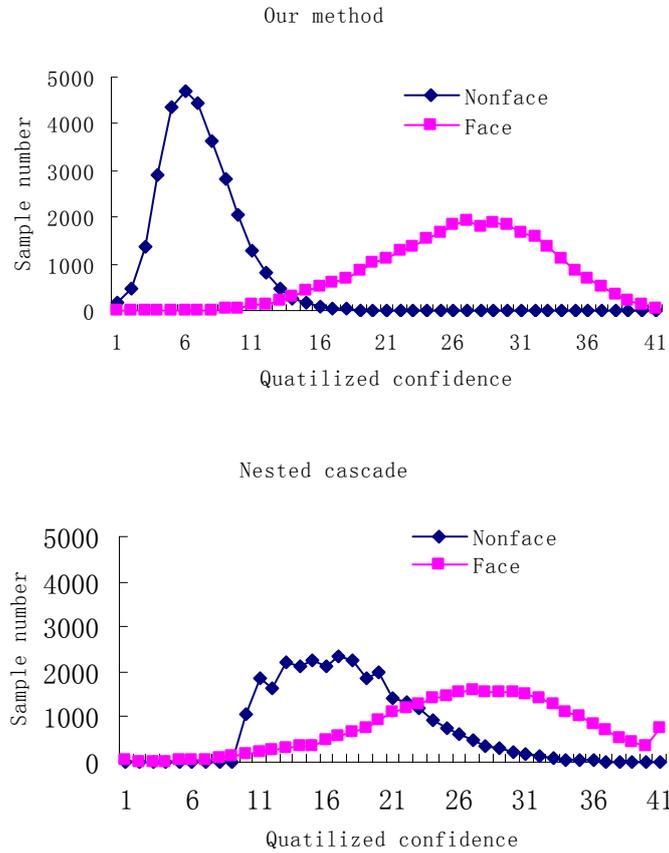


图 3.10 两种方法对已有特征获得的分类器分类判别能力对比

### 正面人脸检测上的总体性能

我们训练了一个误检率低于百万分之一误检率的瀑布式检测器, 训练共耗时 12 小时, 考虑到多于 23 万个正例样本参与了训练, 这个时间消耗非常小, 为了与传统的训练方法对比, 我们还使用一个从 23 万个正例样本中随机挑选获得的拥有 10000 个人脸样本作为训练集, 使用传统方法训练了一个瀑布型分类器。在其它训练条件、实验配置完全一样

的情况下,传统方法训练瀑布型分类器的训练时间为 17 小时。可以看出,即使在训练样本集远远高于传统方法的基础上 (230000 vs. 10000), 矩阵式学习算法所需的训练时间仍然少于传统算法, 这充分证明了矩阵式学习在训练上的高效性。

为了对训练得到基于瀑布型分类器的人脸检测器进行分类性能方面的评估, 我们在 CMU+MIT 正面公开人脸测试集上进行了测试。这个测试集包括 130 个图像, 507 张人脸。它可以从网上下载获得。图 3.11 给出了检测器的 ROC 曲线, ROC 曲线上的点通过逐步去掉瀑布式分类器的最后一层得到。另外, 图中还给出了一些其它比较具有代表性的研究者报告的最好的结果。值得注意的是, 本文结果好于文献[HAWL04]中的结果, 因为文献[HAWL04]中采用的方法是本文方法引入大规模样本集之前的基准方法, 除了本章中涉及的创新点, 这两个方法有很多相似之处。这可以归功于本文方法中大规模人脸样本集的使用。文献[BB05]和[Sch04]中的结果可以与本文的结果接近, 但是, 文献[BB05]中使用了一种关于 ROC 曲线调节的最优化策略, 相信如果本文方法使用了这些优化策略, 检测器的性能也可以进一步提高。另外需要注意的是, 文献[Sch04]的测试结果是在去掉包含 5 个拥有笔画脸的测试图像后得到的。

最后, 在 CMU+MIT 正面公开测试集上的部分检测结果图片见图 3.12 到图 3.14。

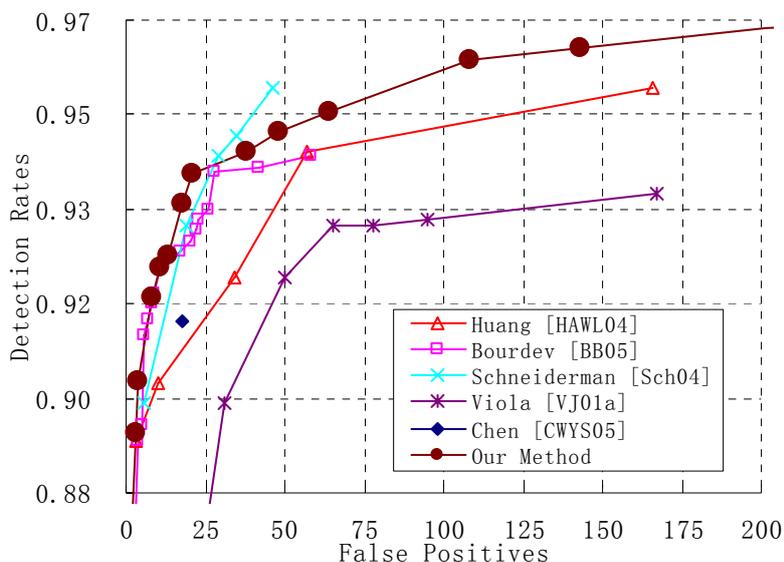


图 3.11 矩阵式学习在 CMU+MIT 正面人脸测试集上的 ROC 曲线

### 3.5.3 多姿态人脸检测

使用如第二章所示的人脸多姿态分类类别, 对拥有 120000 张和 300000 张的全侧面和半侧面人脸训练样本进行了矩阵式学习的训练时间情况如下, 每个姿态的瀑布型分类器的训练时间都在 18 个小时之内, 最小的只需要 8 个小时。此训练时间代价相对于 120000 和 300000 的样本数目来说, 是完全可以接受的。

训练获得的所有姿态类别的检测器同时应用每幅图片的候选子窗口, 构成多姿态检测程序。学习得到的多姿态人脸检测器在 CMU 侧面人脸公开测试集上进行了测试, 这

个测试集包含 208 幅图像，其中有 441 张各种姿态和场景的人脸。这个测试集人脸模式复杂，非常具有挑战性。测试得到的 ROC 曲线见图 3.15。可以看出，本文方法取得比较好的结果。考虑到更多场景变化的侧面人脸训练集仍需进一步收集，这个结果完全可以接受。部分检测结果实例图片见图 3.16 到图 3.18。

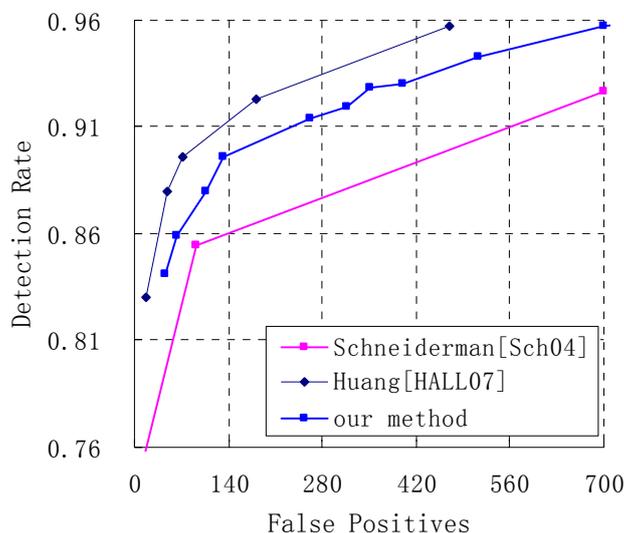


图 3.15 CMU 侧面人脸测试集上的 ROC 曲线

### 3.6 总结

本章提出了矩阵式学习算法，一种针对瀑布式分类器训练设计的训练方案，能够缓解物体检测中同时存在大规模正例样本集和反例样本集导致的训练负担过大问题。矩阵式学习训练分类器效率高，有利于现实情况中的分类器学习及验证等相关操作。同时，由于大规模正例样本集的引入，学习得到的人脸检测器性能获得保证。在正面人脸检测和多姿态人脸检测上进行了的实验表明本文方法解决了大规模正例样本集存在的情况下的训练时间问题，在公开人脸测试集上的测试结果本文方法在分类性能上也表现卓越。相信本章提出的方法具有很大的实用价值和理论价值，并且可以用于人脸之外的其它物体检测领域。

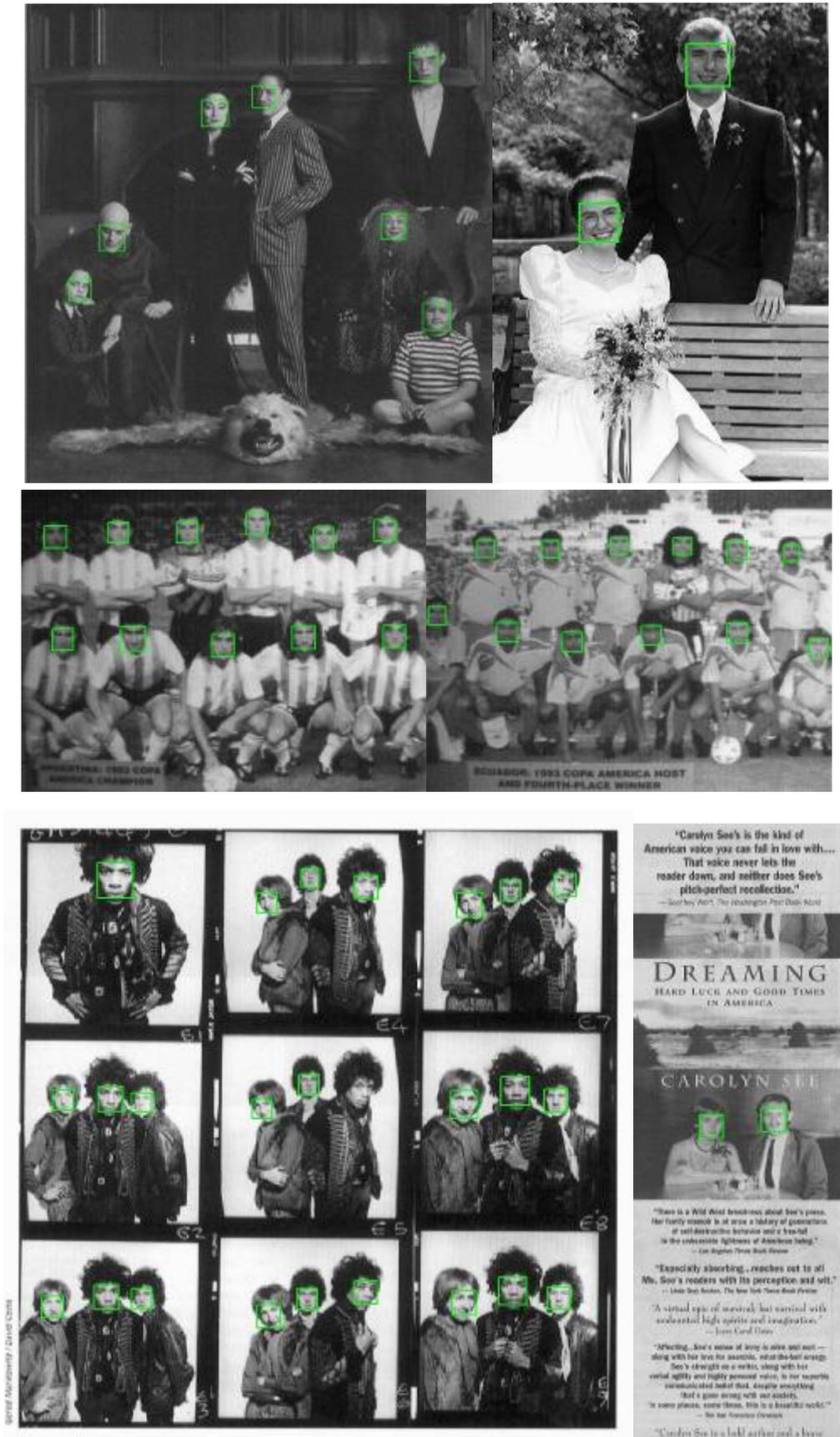


图 3.12 矩阵式学习正面人脸检测结果图片 1

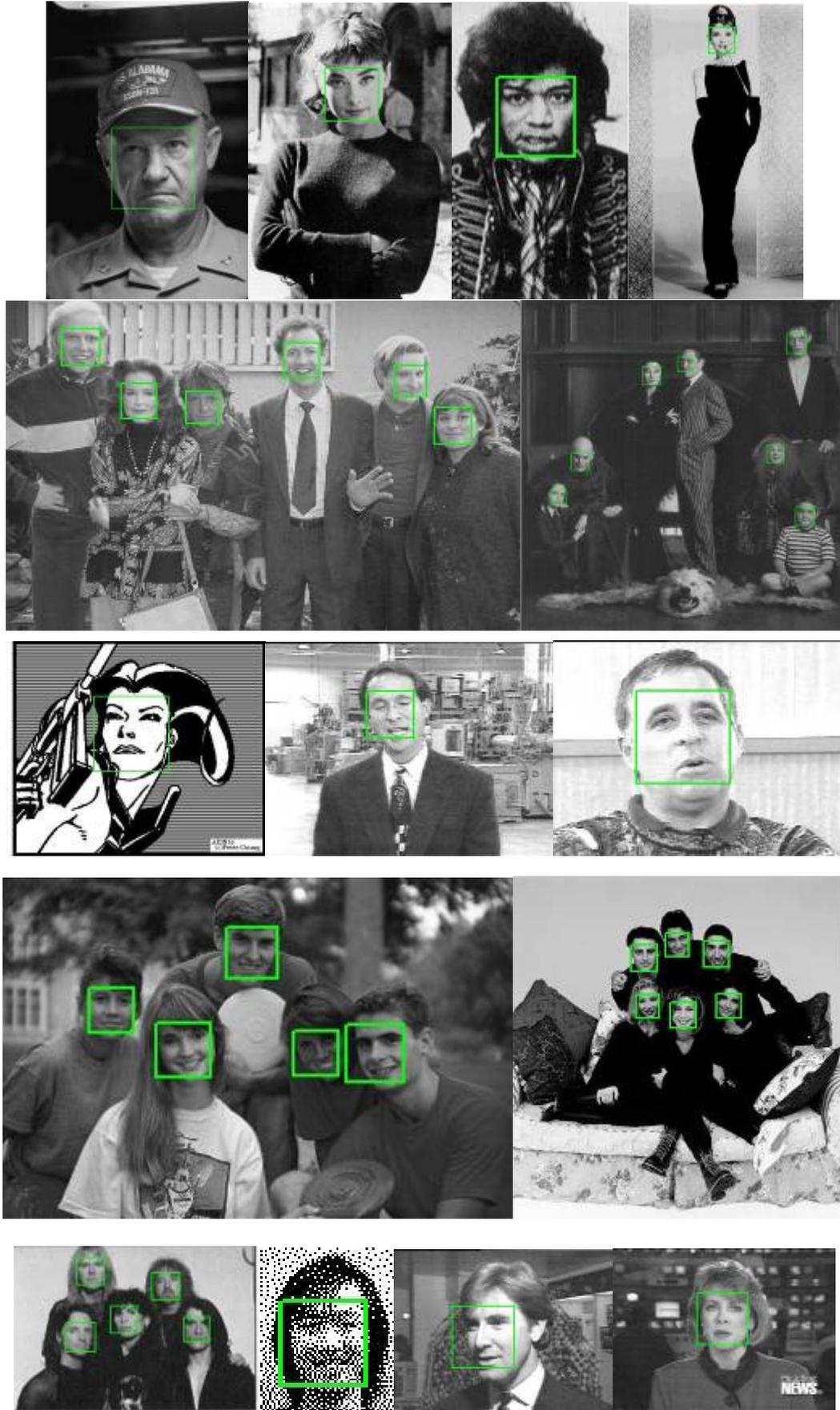


图 3.13 矩阵式学习正面人脸检测结果图片 2

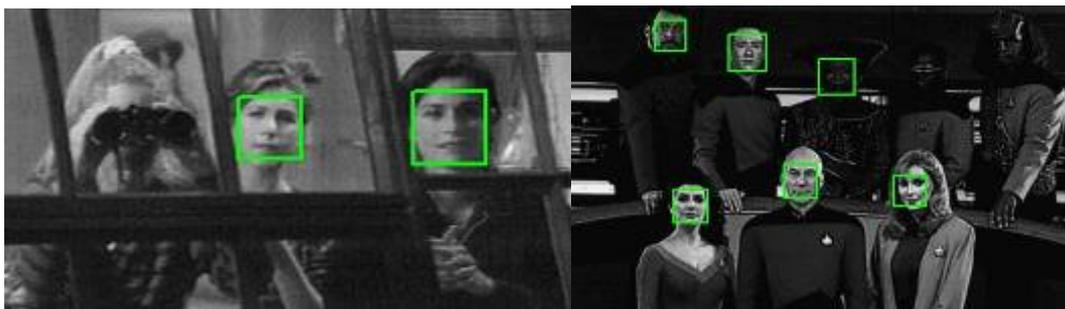
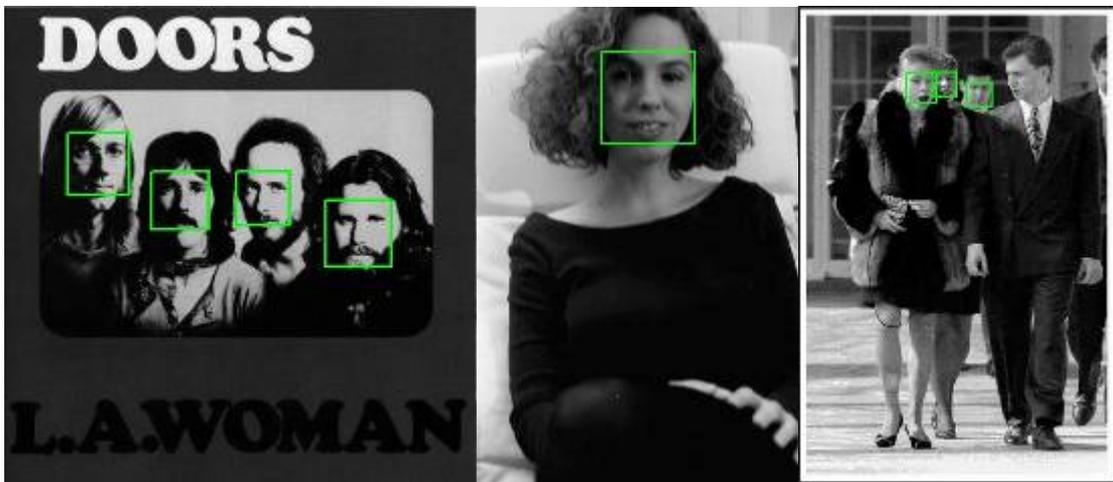


图 3.14 矩阵式学习正面人脸检测结果图片 3



图 3.16 矩阵式学习多姿态人脸检测结果图片 1

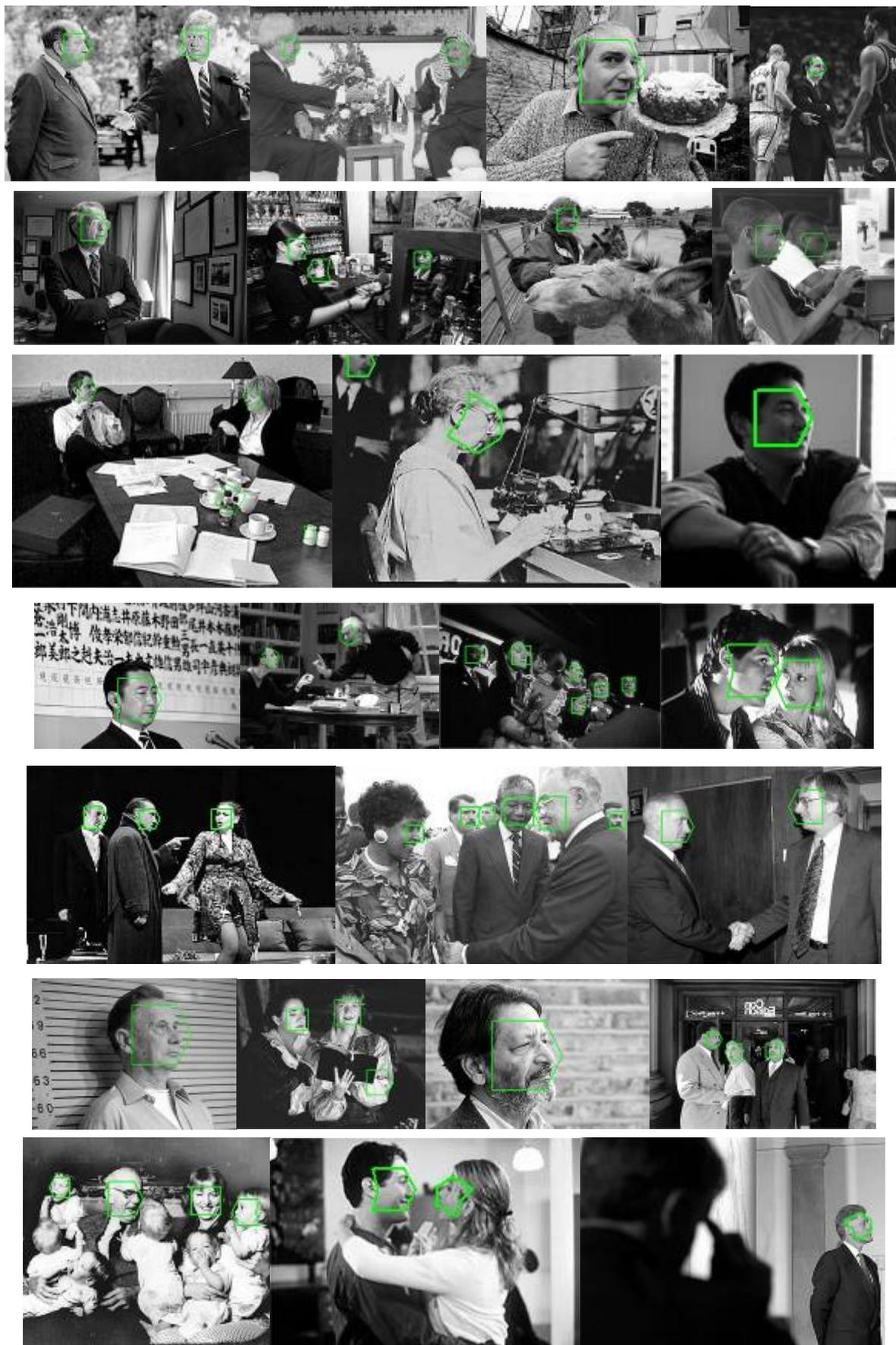


图 3.17 矩阵式学习多姿态人脸检测结果图片 2



图 3.18 矩阵式学习多姿态人脸检测结果图片 3



## 第四章 局部组合二值特征及快速预处理分类器

在第三章中，提出了基于大规模样本集的瀑布型分类器训练算法，训练获得的人脸检测器获得很好的分类性能。但是，随着大规模样本集的引入，样本变得更加复杂，学习获得的瀑布型分类器的计算复杂度也相应提高，检测一幅图像所需的时间相应增加。从技术细节上来说，不能同时使得检测性能和检测时间同时推进的原因是多方面的。首先是特征的代表能力依然不够有效，特征计算快速、简单的同时，描述能力却在下降；其二，检测框架依然存在问题，某些可以节省的计算存在于传统方法中，比如 Viola 等人的检测方法。其三，瀑布型分类器可以做的更加精细，更加高效。本章中试图通过对上述三点进行改进，解决此问题。本章提出一种解决人脸检测算法获得的检测器计算复杂度过高问题的方法，具体包括了提出了一种分辨能力更强、能够进行高效计算的特征，以及使用此特征进行高效人脸检测的方法。

本章提出的检测方法实现中体现在对一个计算复杂度高的检测器前端添加了一个更高效的预分类器。在添加的预分类器中使用了一种新的特征，局部组合二值特征。局部组合二值特征具体包括二值特征和二值特征的组合。二值特征建模的是图像上相邻区域间灰度差异的二值信息，即两个区域的更“黑”更“白”二值关系。对多个二值特征进行共生信息进行类似局部二值模式 (local binary pattern) 的编码建模构成局部组合二值特征。其次，预分类器使用一次性收集的人脸样本和非人脸样本和 Real AdaBoost[SS99] 学习算法学习得到一个整体分类器，对这个分类器进一步构建瀑布型分类器。最后，在使用预分类器检测过程中，采用了以特征为中心的瀑布型分类器检测方法。

本章下面的小节安排如下：首先在 4.2 节中介绍了本章提出的预分类器与第二章中提出的瀑布型分类器的结合总体结构；其次 4.3 节提出了在预分类器中使用的一种新的特征——局部组合二值特征以及基于局部组合二值特征的快速瀑布型分类器构建和检测算法；然后 4.4 节给出了相关的实验和结果；最后，在 4.5 节给出了小结。

### 4.1 由粗到细的人脸检测系统框架

瀑布型检测器总体分为两个层次。第一层是预处理层，预处理层的预处理瀑布型分类器（简称预分类器）计算效率高，能够在短时间内排除大部分简单非人脸。第二层是深入处理层，深入处理层的分类器计算复杂度高，用来排除通过了预分类器后剩下的较为复杂的非人脸。预分类器和深入处理分类器都是基于 Real AdaBoost 学习得到的瀑布型分类器。如图 4.1 篮框中所示，瀑布型分类器由上到下逐步运行每个子分类器，每个子分类器位置进行一次人脸和非人脸的判断。一旦候选窗口被判断为非人脸，则将被拒绝并退出。通过所有  $N$  个子分类器的候选窗口最终被判断为人脸，否则，判定为非人脸。

检测阶段，采用传统的窗口扫描模式，预分类器和深入处理分类器对输入图像上的

每个候选窗口进行逐步深入的扫描，并对本窗口做出人脸与非人脸的判断。为了检测不同大小的人脸，输入图像以一定的缩放系数被缩放为多个大小的图像。扫描过程图示见图 1.4，图中最左上角的图像是原始输入图像，向右下角方向给出了不同缩小尺度下的图像。

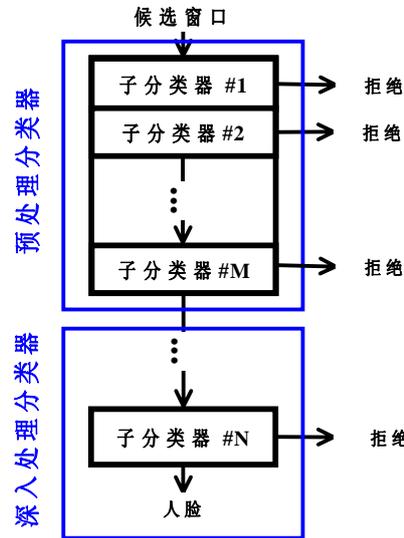


图 4.1 两阶段瀑布型分类器

预分类器是通过一次性收集人脸样本和非人脸样本并经验地指定分类器中包含的特征数目一次性由 Real AdaBoost 算法学习得到。得到之后，根据每个特征期待的用户指定的性能指标对每个特征位置建立瀑布型分类器。预分类器使用的特征是本文提出的局部组合二值特征。相对于 Haar 特征，局部组合二值特征天生对光照变化具有鲁棒性，并且保持计算简单的特性。深入处理分类器的结构和 Viola 等人提出的检测方法一样，每个子分类器包含多个特征的强分类器，每个子分类器学习的非人脸样本由前面的子分类器误判的非人脸样本组成。

预处理层和深入处理层的分类器都是瀑布型分类器，但是构建方法又不同。预分类器是一次性使用 Real AdaBoost 学习算法得到的集成(ensemble)分类器。训练非人脸样本在学习算法运行之前一次性收集得到。预分类器中包含的弱分类器数目，也就是 Real AdaBoost 算法的迭代次数，由先验人为决定。在学习得到多个弱分类器构成的强分类器之后，每个弱分类器位置都根据要求的检测率和误检率设置阈值，如图 4.1 所示。图 4.1 中示例的预分类器是一个包含 M 个弱分类器的强分类器，分为是 M 个层级。

深入处理分类器的学习和第三章中提出的方法一样，通过设置瀑布分类器的目标检测率和误检率训练每个包含多个特征的层级分类器，这些层级分类器的非人脸训练样本由自举方法收集得到。

关于预分类器的学习值得一提的是，它使用了一个尺度下的特征在训练样本上不同位置滑动得到的特征作为候选特征集合，所以学习得到的预分类器中包含的特征都相应也是一个尺度下的特征。实验结果表明，即使只适用一个尺度下的局部组合二值特征集

来进行训练，就能获得一个性能较好的预分类器，获得的预分类器在检测率接近 100% 的同时，误检率可达到万分之一以下。在检测阶段，输入一幅图像，由于所有预分类器中的特征都是一个尺度，所以，输入图像中每个像素点的特征值可以一次性扫描得到，形成一幅特征值图像。窗口滑动的过程中，分类器运行需要的特征值只需要查找相应特征值图像获得，并进一步进行 Real AdaBoost 分类器相关的简单操作即可。这使得相邻窗口包含多个本特征的窗口扫描过程共享了特征值的计算，极大的提高了检测过程的运算效率。特征计算共享的示意图见图 3。图中特征  $j$  被窗口和  $x_1$  和  $x_2$  所共享。

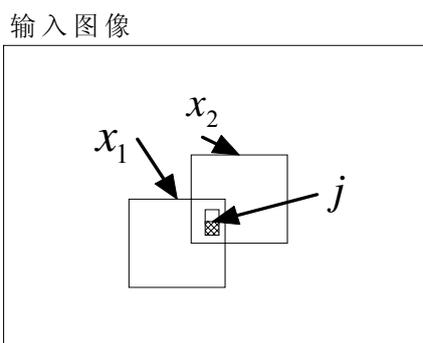


图 4.2 一个特征被多个窗口共享示意图

## 4.2 局部组合二值特征

本节描述提出的局部组合二值特征。局部组合二值特征中表示，有很明显的层次关系。最底层的构成是计算简单的二值 Haar 特征。然后，在此基础上，形成描述能力更强的组合二值特征。最后，为了方便更有效率的训练，我们使用了局部组合二值特征来进行检测器的学习。下面的描述中就从二值特征到组合二值特征，再到局部组合二值特征逐步陈述。

### 4.2.1 二值 Haar 特征

在介绍二值 Haar 特征之前，首先回顾一下 Haar 特征的计算方法以及其优缺点。一个 Haar 特征表示的是图像上两个相邻区域的灰度差异。Viola 等人提出的 Haar 特征的形式布局如图 1.8 所示。当然，更一般的 Haar 形式布局变化可以更加随意。Haar 特征的特征值是图中填充黑色的矩形框和未填充矩形框内所有像素的像素和的差值。这些矩形框内的像素和可以通过积分图的辅助从而快速计算得到，具体可以参考文献[VJ01a]。Haar 特征的计算包括了多次加减法运算。举例说明，一个如图 1.8(a)所示的包含两个矩形的 Haar 特征的计算可以形式化为：

$$f_j(x) = (s_1)_j - (s_2)_j \quad (1)$$

其中， $(s_1)_j$  和  $(s_2)_j$  表示填充矩形和未填充矩形内部所有像素的和， $j$  表示相应的 Haar

特征,  $x$  是输入图像窗口。

Haar 特征对全局光照是不具有鲁棒性的。全局光照的变化会导致邻域区域间的灰度差的变化, 也就是 Haar 特征值的变化。所以在 Viola 等人的检测方法中, 对如图 4.2 所示的每个窗口都要进行全局光照的归一化, 然后再在此光照归一化后的窗口上进行 Haar 特征值的计算。更具体的来说, 光照归一化方法就是对每个候选检测窗口的方差进行归一化处理。在使用由特征构成的分类器对候选窗口进行扫描的过程, 其实主要归一化的是 Haar 特征值的归一化。根据表示候选窗口内部像素变化情况的方差, 对特征值进行归一。归一化后的 Haar 特征值在某个特定窗口上的计算可以形式化为:

$$f'_j(x) = \frac{(s_1)_j - (s_2)_j}{\sigma_x} \quad (2)$$

其中  $\sigma$  表示候选检测窗口  $x$  的方差。

由于每个窗口方差不同, 所以对于如图 1.8 所示的 Haar 特征  $j$ , 属于不同窗口的同一个特征会存在不同的特征值。这在 Viola 等人的传统方法中, 导致了当同一个 Haar 特征存在于不同方差的候选窗口时, 存在不同的特征值, 也就意味着每个窗口都要进行特征值的计算, 这会带来多次重复的 Haar 特征值计算。

所以, 为了让同一个特征在为属于不同窗口的分类器的计算过程中提供相同的输入特征值, 本文引入了二值 Haar 特征。二值 Haar 特征只保留 Haar 特征的符号信息, 形式化如下:

$$b_j(x) = \begin{cases} 1 & ((s_1)_j - (s_2)_j) > 0 \\ 0 & \text{其它} \end{cases} \quad (3)$$

可以看出, 二值 Haar 特征中, 两个相邻矩形区域的绝对灰度差别被抛弃, 只有它们的相对是“亮”或是“暗”两种信息。因此二值特征对全局光照变化具有不变性。对于属于不同交叠窗口的同一个二值 Haar 特征, 所有窗口的本特征拥有相同的特征值。在本文所述的检测方法中, 此点被利用来提高检测过程的计算效率, 详细描述见 4.3 节。

#### 4.2.2 组合二值 Haar 特征

二值 Haar 特征相对于 Haar 特征来说具有了很好的光照鲁棒性, 对于不同窗口的同一个特征拥有相同的特征值, 但是每个特征的分辨能力相对于 Haar 特征来说却由于相对灰度信息的抛弃大大下降, 这会给学习算法的学习带来巨大的困难, 学习算法不能从这些分辨能力过弱的特征集合学到一个性能良好的分类器。

为了增加特征的描述能力, 本文进一步提出将多个二值特征组合的方法。组合后的新特征表达的主要是多个二值 Haar 特征的共生(co-occurrence)信息。组合的方法是将多个二进制的二值 Haar 特征值作为码字进行多位的编码。为了方便, 进一步将二进制编码转换为十进制的编码。图 4.3 给出了一个 3 个二值 Haar 特征进行组合的示例, 其中 a 表

示组合特征的计算函数， $b_1$ 、 $b_2$ 、 $b_3$  是三个二值特征， $(\cdot)_2$  表示从二进制编码到十进制编码的操作。示例中 3 个二值 Haar 特征的特征值分别为 1, 1 和 0，组合后得到的十进制的特征值为 6。

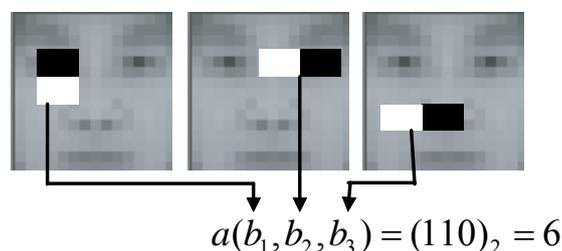


图 4.3 组合多个二值特征

### 4.2.3 局部组合二值特征

对于如 4.2.2 节中所述的组合特征，可以选择的组合方式非常多，可以在候选图像窗口的任意位置、任意尺度上和任意数量的二值 Haar 特征进行组合。这样穷举所有组合，会产生一个巨大的组合特征集合。巨大的候选特征集合会给学习算法带来巨大的计算和存储负担，使得学习困难不能完成。很幸运，我们发现了一种组合，使用这种组合方法可以生成一个对于人脸表示非常有效的特征子集。我们称之为局部组合二值特征。局部组合二值特征的组合方法将如图 4.4 所示的 8 种特征的黑色矩形限制为相同大小、并且限制用于组合的特征共享黑色矩形区域。最终得到一个由 8 个特征组合而成的新特征。局部组合二值特征保持了二值特征的光照鲁棒性，描述能力强，并且只有数千个，非常适用于实际的人脸检测器的学习和应用。

局部组合二值特征可以表示为一个四元组， $l(x,y,w,h)$ ，这里特征在图像上  $x$  坐标和  $y$  坐标的值， $w$  和  $h$  表示特征中每个矩形的宽和高。图 8 给出了两个局部组合二值特征示例的示例。

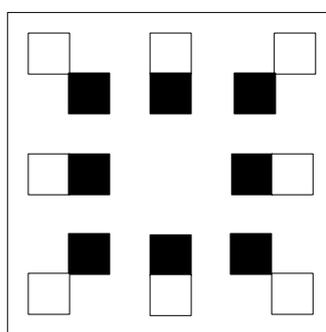


图 4.4 组合使用的二值 Haar 特征

局部组合二值特征的计算需要计算 8 个 2-矩形 Haar 特征值并进行符号提取以及编码的操作，相对于一个 Haar 特征，计算量有所增加。但是有两个因素使得局部组合二值特征能够更加高效的用于人脸检测器中。其一是局部组合二值特征比一个 Haar 特征具有更强的分辨能力，关于这点在 4.4.2 节中有专门的实验来验证；其二是同一个局部组合二

值特征对于在多个窗口的分类器特征值可以进行共享，解决如图 4.2 所示的共享特征在不同窗口存在不同特征值的问题，所以能够节约整体人脸检测过程的计算时间，这个节约是巨大的。

某种意义上，局部组合二值特征与局部二值模式（locally binary pattern）相似。局部二值模式已经在纹理识别中取得了很好的应用[OPH96]。但局部二值模式是局部组合二值特征在每个矩形的大小都是一个像素时的特例。

与局部二值模式一样，局部组合二值特征的特征值空间为 $\{0, \dots, 255\}$ ，每个特征值代表了一幅图像局部的特定结构。

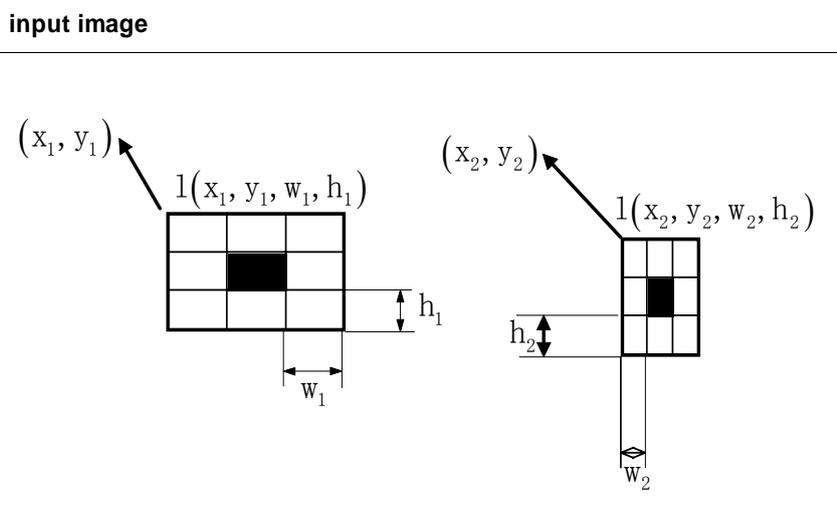


图 4.5 局部组合二值特征

### 4.3 使用局部组合二值特征进行人脸检测

上面讲述了局部组合二值特征，下面描述使用局部组合二值特征进行人脸检测的过程。本文方法采用了瀑布型的分类器检测结构，本文使用的瀑布型分类器检测结构对于预分类器和深入处理分类器检测结构采用了不同的检测技术，其中预分类器采用了一种以特征为中心(feature-centric)收集检测和学习方法[Sch04]。而深入处理分类器采用了如第三章中矩阵式学习中提出的方法，这种检测方法可以被归结为以窗口为中心(window-centric)的检测方法。

本节组织如下，首先 4.3.1 节描述了以特征为的检测方法。然后 4.3.2 节基于对以特征为的检测方法的分析，描述了以特征为中心的瀑布型的分类器的构建。4.3.3 节描述了以窗口为的检测方法。最后，本文提出的方法被扩展到多姿态人脸检测的情况。

#### 4.3.1 以特征为的检测方法

在描述以特征为的检测结构之前，我们先介绍传统的以窗口为的检测方法。

传统方法中，为了检测每个窗口，需要将学好的分类器在本窗口上运行，运行一次分类器可以分为特征值的计算和分类器的计算。以常用的 Boosting 学到的线性加和分类器来说，一个 Real AdaBoost 算法学到的分类器表示如下：

$$H(x) = \text{sign} \left( \sum_{t=1}^T h_t(f_t(x)) \right) \quad (4)$$

其中  $H$  表示强分类器， $h$  是弱分类器， $t$  表示某个特征， $T$  是强分类器中包含的弱分类器的总数目。这其中特征的计算操作是  $f$ ，分类器相关的计算包括除了  $f$  之外的操作。

传统方法中，首先滑动到某个窗口位置，然后计算运行如上所述的分类器，包括每个特征值计算和基于特征值的分类器相关的计算。对于一个特征，在窗口的滑动过程中，它必会被多个相邻交叠的窗口所包含，如图 4.2 中示例，那么特征值也相应的也会被相应多次重复计算。

为了避免重复计算，Schneiderman 提出了以特征为中心的检测方法[Sch04]。

首先，以特征为中心的检测方法构建的分类器里包含的特征只拥有少数的特征尺度。更明确的说，对于局部组合二值特征  $l(x,y,w,h)$ ， $x$  和  $y$  可以随意变化， $w$  和  $h$  只拥有少数的固定值。为了便于理解，首先只使用一个尺度的特征进行本方法的解释，也就是只有一个固定的  $w$  和  $h$  在不同位置下的特征构成的特征集。在理解了一个尺度的情况之后，增加多个尺度的情况会很容易理解。假设特征尺度为  $3 \times 3$ ，也就是  $w=3$  和  $h=3$ 。训练分类器的阶段，使用  $3 \times 3$  的特征在样本窗口不同位置穷举，进行学习得到一个分类器。检测阶段，由于这些特征都是一个尺度的特征，可以在初始的时候将待检测图像上所有位置的  $3 \times 3$  尺度特征值全部预先计算出来。图 4.6 中间图像就是一幅图像就是计算  $3 \times 3$  局部组合二值特征得到的特征图像。在此特征图像的基础上，再运行上面训练的分类器，那么对于每个窗口的扫描过程，只需要根据窗口位置和特征在窗口中的位置在特征图像上进行特征值的查找，以及基于特征值作为输入的分类器相关的操作。这使得分类器中的每个特征值的计算都只是预先进行一次，而在分类器的扫描过程中，只是进行多次的查表。相对与特征值的计算，查找表的计算代价小很多，所以计算效率提高很多。表现在 Real AdaBoost 获得的分类器上，分类器计算方法表示为：

$$H(x) = \text{sign} \left( \sum_{t=1}^T h_t \left( LUT_{(t,x)}(I_f) \right) \right) \quad (5)$$

其中， $LUT$  是根据特征  $t$  和当前检测窗口位置  $x$  所进行的查找特征值的操作， $I_f$  是已经计算得到的特征值图像。

当以特征为中心的分类器包含多个特征尺度的时候，显然，只要在初始预先计算这些多个特征尺度的特征图，并依据分类器中的不同尺度特征图上执行查找即可。

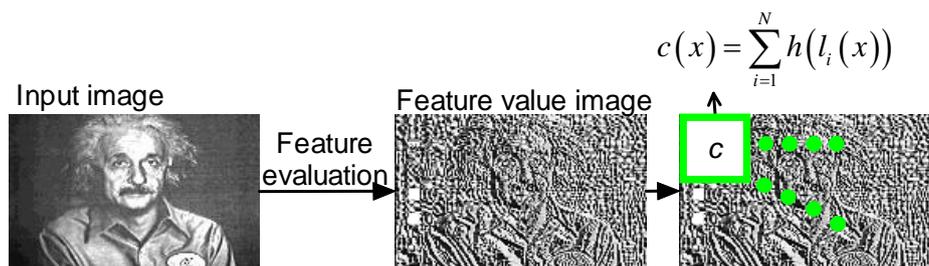


图 4.6 以特征为中心的检测方法

当然，特征尺度多少的选择会对计算效率带来直接影响，更多的特征尺度，学习得到的分类器分辨能力更强，但同时也会增加越多次数的预先的特征图像的计算，归根结底，这是一个折中问题。在本文的实验中，对于局部组合二值特征，只使用了一个尺度下的特征，使用 Real AdaBoost 学习已经能够获得性能非常优良的预分类器。我们使用穷举所有尺度来进行特征模板的挑选，具体操作是对每个尺度下的特征集合都训练相应的预分类器，并根据各预分类器展现的性能情况进行最优挑选，为以特征为中心的分类器挑选特征模板的过程见图 4.7。最终实验挑选出来的分辨能力最强的尺度是  $3 \times 3$  大小。

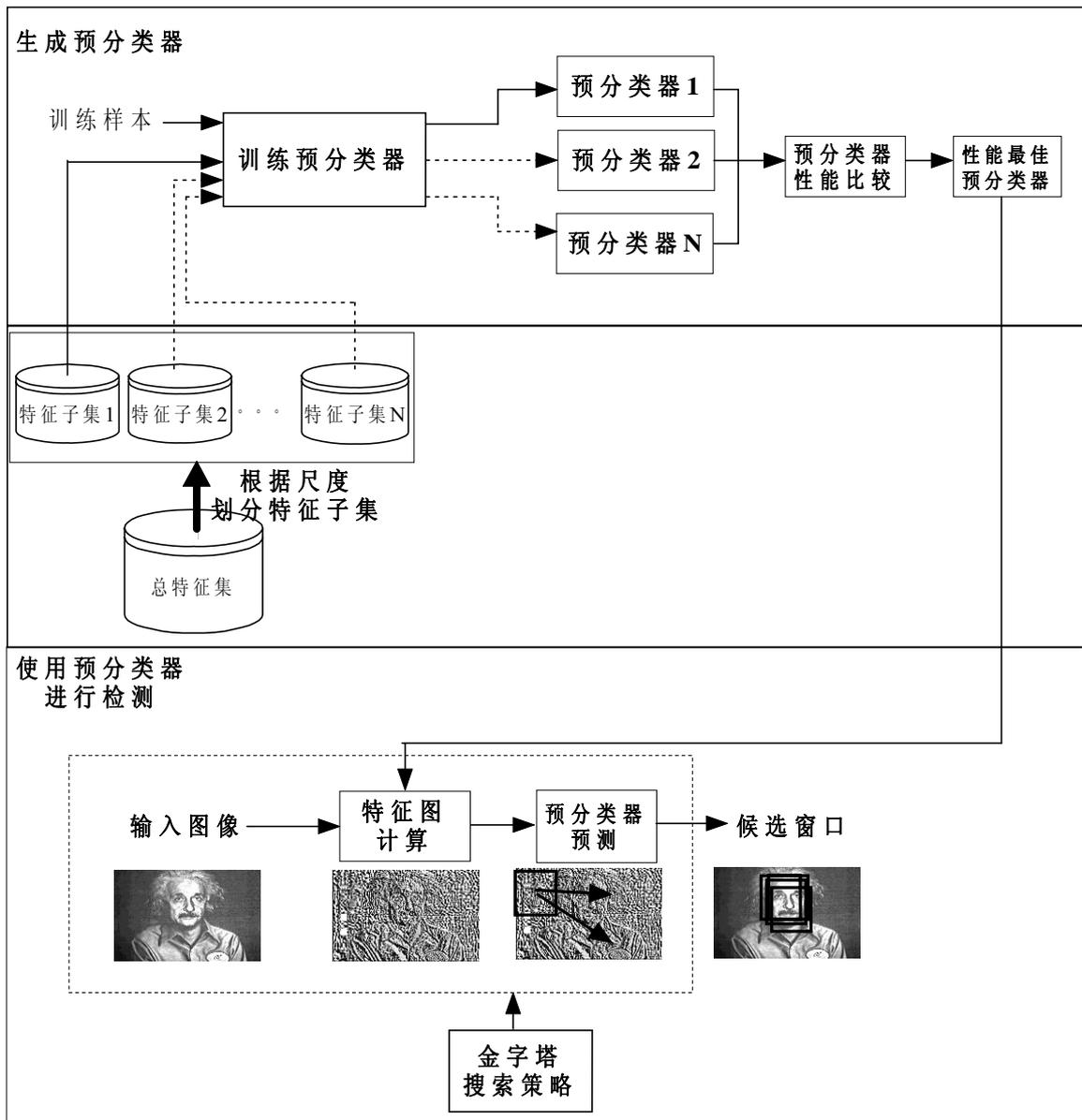


图 4.7 为以特征为中心的分类器挑选尺度的过程

### 4.3.2 以特征为中心的瀑布型预分类器

4.3.1 节中介绍了以特征为中心的检测方法，在以特征为中心的检测方法中，H. Schneiderman 提出的时候使用所有特征构建一个整体分类器，这要求每次分类器的运行要对所有的特征进行查表，这种做法计算效率低，尤其是当一个特征的分辨能力已经较强的时候。局部组合二值特征就是这样的分辨能力强的特征，表 4.1 给出了一个使用 Boosting 学到的正面人脸强分类器在每个特征处设置一个层级时的性能情况。性能的判断首先是固定每个层级处的检测率为 0.99984，这样每个特征位置处的阈值也就确定了。然后在一个非人脸测试集上测试得到当前阈值下的误检率，在每个位置都进行阈值设置和测试，每个位置处的误检率在表 4.1 中给出。从表中可以看出，每增加一个特征，整体分类器的误检率都会下降，这说明了特征的有效性。设置检测率的人脸样本有约 23

万个，非人脸样本 20 万个。

基于局部组合特征的有效性，以特征为中心的检测方法得到的整体分类器为基础，我们进一步将这个整体分类器在每个特征构建一个层级，构成瀑布型分类器。这样对于复杂程度不同的候选窗口并不需要同时进行所有特征的查表，提高了分类器检测效率。

为了说明计算效率的提高，这里我们讨论对于一个瀑布型分类器的计算复杂度的计算问题。一个瀑布型分类器在一个窗口上的期望计算量如下：

$$C_{Total} = \sum_{i=1}^M \left( \prod_{l=0}^{i-1} (P_{l-1}) \times (F_i - F_{i-1} + CC_i - CC_{i-1} + C) \right) \quad (6)$$

其中， $C_{Total}$  表示整体瀑布型分类器的计算复杂度， $P_l$  表示在  $l$  位置处获得前面所有  $l$  个特征所需要的计算量， $P_{l-1}$  表示在  $l-1$  位置处非人脸通过的概率，显然，初始时， $P_0=1$ ， $CC_l$  表示由  $l$  个特征构成的强在获得  $l$  个特征值之后分类器相关的操作， $C$  为在一个层级位置由于分层所带来， $M$  是瀑布型分类器中所包含的总的层级数目。在 Real AdaBoost 学习得到的以特征为中心的分类器中，由于已经得到特征图像，所以  $F$  对于一个特征来说具体的操作是一次查找表操作，分类器相关的操作  $CC$  具体就是一次根据特征值进行的信度查找和查找得到信度后的信度累加（浮点数加法）， $C$  就是一次浮点数对比计算。

下面根据表 4.1 所示的 20 个特征构成的整体分类器，给出一个具体的计算实例。如果使用 Schneiderman 提出的原始的方法，没有进行分层，那么需要  $20 \times 2=40$  次的查找表（查找特征值和查找信度）和 20 次的加法（信度累加）和一次对比计算  $C$ （阈值对比）。但是如果在每个位置处设置分层，近似地将表中的误检率认为是本特征位置处非人脸通过的概率，那么只需要  $6.35857 \times 2$  次查找表和 6.35857 次加法，和 6.35857 次的对比计算。可以看出虽然增加了  $6.35857-1=5.35857$  次对比计算，但是却节约了  $20 \times 2-6.35857 \times 2=13.64143 \times 2=27.28286$  次查找表操作，节约加法  $20-6.35857=13.64143$  次。总体来说，这大大降低了计算代价。分类器中的特征数目如果增加，节约的计算量将更大。

表 4.1  $3 \times 3$  大小局部组合二值特征的分类器对每个特征处设置层级，检测率设定为 0.99984，误检率的变化列表

特征位置	1	2	3	4	5	6	7	8	9	10
误检率	.91	.63	.64	.54	.45	.35	.30	.24	.20	.17
特征位置	11	12	13	14	15	16	17	18	19	20
误检率	.16	.14	.12	.10	.09	.08	.07	.060	.060	.050

### 4.3.3 以窗口为中心的深入处理分类器

虽然以特征为中心的瀑布型预分类器可以排除大部分简单非人脸，但由于只使用了少数尺度下的特征，描述能力有限，当面对复杂的非人脸时就会无能为力。所以，在预分类器运行之后，我们使用如第三章中所描述的矩阵式算法，并利用所有尺度下的 Haar 特征训练了一个针对复杂非人脸的深入处理分类器，这个工作在第三章已经描述。

### 4.3.4 多姿态人脸检测上的应用

到现在为止，本章描述了在一种单个姿态下人脸检测器的训练和测试方法，下面，本节进一步将单个姿态的人脸检测扩展到多姿态人脸检测。为了进行多姿态人脸检测，人脸姿态空间被分成如图 2.1 所示的 15 种姿态，具体关于多姿态的分类可以参考第二章所述。对于每种姿态，分别进行预分类器的学习和深入处理分类器的学习。检测时，在每个候选窗口位置同时运行 15 种姿态下的分类器。

需要注意的是，对于所有姿态的以特征为中心的特征值，使用了同一个尺度下的特征进行各姿态预分类器的训练。那么在多姿态的人脸检测过程中，对于所有姿态预处理分类器的运行过程，特征值图像可以被所有姿态预分类器所共享，这能够节约计算消耗。多姿态检测如图 4.8 所示，图中箭头方向即为检测过程运行方向。

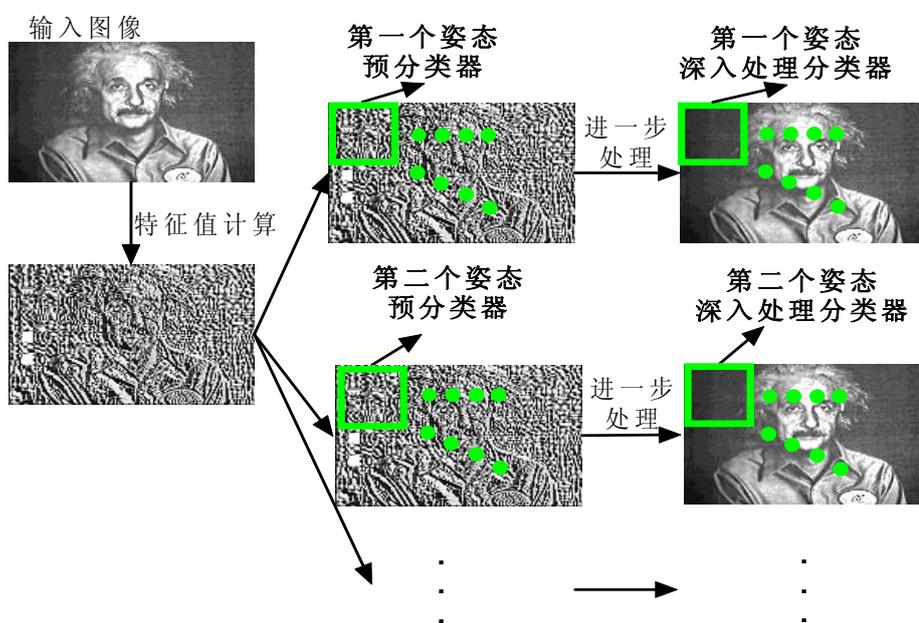


图 4.8 多姿态人脸检测流程

## 4.4 实验配置和实验结果

本节中，我们设计多个实验对本文所提出的方法进行验证。这主要包括局部组合二值特征的分辨能力量化的调查，本文方法所构建的正面人脸检测系统和多姿态人脸检测系统在分类性能和检测速度两方面的调查。本节内容安排如下：首先，4.4.1 节给出了实验过程中的配置；然后，4.4.2 节给出了一个对于局部组合二值特征相对于 Haar 特征的分辨能力的实验结果；最后，4.4.3 和 4.4.4 节分别给出了最终得到的正面人脸检测系统和多姿态人脸检测系统在国际公开测试集上的分类性能实验结果，以及检测速度情况。

#### 4.4.1 实验配置

对于基于单个特征的以特征为中心的瀑布型预分类器，每个子分类器的目标检测率设置为 0.99984。用于训练以特征为中心的总的集成分类器的非人脸数目为 60000 个，这些非人脸样本随机挑选得到。设置预分类器内包含的总的特征个数为 150 个，这是根据经验设置的。实验显示所有姿态的预分类器，在设置目标检测率接近百分之百的情况下，包含 150 个局部组合二值特征的各姿态强分类器都可以排除所有训练使用的非人脸样本。

检测过程中为了检测不同尺度的人脸，测试图像进行了缩放，缩放系数为 0.8。在以下所有实验中，如果没有特别声明，实验都是在一个普通奔腾 3.2GHz，内存 1Gb 的个人电脑上进行。

#### 4.4.2 局部组合二值特征在正面人脸上的分辨能力调查

表 4.2 等分类错误率情况下不同尺度模板局部组合二值特征分类器的特征数目

高 宽	1	2	3	4	5
5	44	36	41	42	67
4	37	31	31	30	44
3	30	26	25	26	35
2	28	28	26	28	36
1	28	29	31	40	48

首先使用各个尺度的局部组合二值特征集合进行强分类器的训练，根据达到相同分类性能的情况下，分类器包含的特征数目确定使用哪个尺度的特征集作为以特征为中心的分类器候选特征集，具体每个尺度下的特征数目表格见表 4.2。最终  $3 \times 3$  的尺度被挑选出来。

其次，为了对比基于  $3 \times 3$  的局部组合二值特征子集和 Haar 特征的对比分类性能，我们分别使用正面人脸测试集学习得到了这两个特征集进行了两个集成分类器的学习。其中  $3 \times 3$  的局部组合二值特征子集包含特征 256 个，而 Haar 特征包含所有尺度的特征 31728 个，远远多于局部组合二值特征的集合大小。分类器使用 Real AdaBoost 学习算法进行学习，Haar 特征的 Bin 数目设置为 40 个，这个数目已经被实验证明能够学习一个性能非常好的分类器。

在分类器性能确定相同的情况下，强分类器中包含的特征数目反映了计算复杂度，并在某种意义上反映了特征的代表能力。越少的特征表示特征表述能力越强。限定检测器性能为 0.9999，不同误检率情况下需要的 Haar 特征数目和  $3 \times 3$  局部组合二值特征数目列于表 4.3 中。从表 4.3 可以看出来， $3 \times 3$  局部组合二值特征所需要的特征数目在不

同的误检率情况下总是少于 Haar 特征数目。这充分证明了  $3 \times 3$  局部组合二值特征在针对简单非人脸样本时的有效性。

表 4.3  $3 \times 3$  尺度下局部组合二值特征集成分类器在对每个特征处设置层级，检测率设定为 0.99984，

误检率的变化情况

误检率	0.4	0.1	0.05	0.01	0.005	0.001	0
Haar	13	34	48	68	72	80	89
局部组合二值特征	7	18	22	30	31	35	39

#### 4.4.3 正面人脸检测

使用本文提出的方法，训练了一个正面人脸检测器，包含预分类器和深入处理分类器。深入处理分类器就是第三章中获得的人脸检测器。为了方便对比增加预分类器和未增加预分类器情况下深入处理分类器的性能，深入处理分类器的训练独立于预分类器的训练。

在使用预分类器对图片进行检测之后，非人脸窗口大量被排除，图 4.12 到图 4.16 给出了一些预检后的结果图片实例，其中每幅图中上面的一幅图片为原始未检测的图片，下面的图片为预检后的结果，结果使用绿色的矩形框画出。

学习得到的正面人脸检测器在 CMU+MIT 正面人脸公开测试集上进行了测试，测试得到的 ROC 曲线见图 4.9。为了对比，近年来其它比较具有代表性的方法所报告的结果也被列出。可以看出，我们的方法取得了更好的结果。

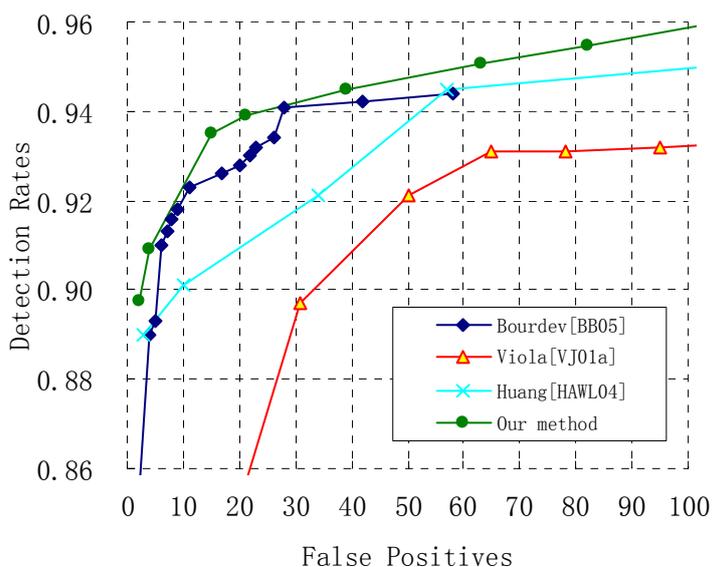


图 4.9 CMU+MIT 正面人脸测试集上的 ROC 曲线

为了对比相对于后处理分类器来说，增加预处理分类器对性能的影响，图 4.10 和表 4.4 给出了单独的后处理分类器和加上预处理分类器后在 CMU+MIT 上的测试结果。可

以看出来，预处理分类器并没有带来系统检测性能的下降，这充分说明了本章所提出的预分类器构建方法的有效性。

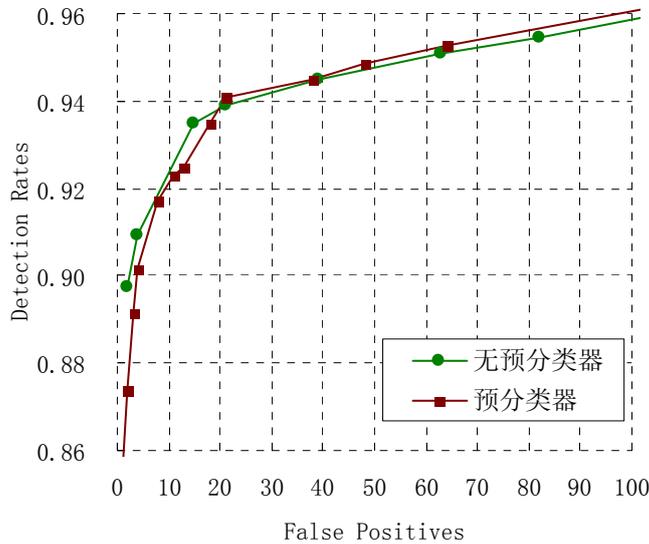


图 4.10 CMU+MIT 正面人脸测试集上的性能曲线

表 4.4 后处理分类器(无预分类器)和及其加上预处理分类器之后(预分类器)的检测结果对比

误检个数	2	4	15	18	38	39	63	64	82	108
无预分类器	0.874	0.909	-	0.935	0.945	-	-	0.953	0.945	0.963-
预分类器	0.897	0.909	0.935	-	-	0.945	0.951	-	0.955	0.961

另外我们还收集了一个拥有各种饰物、种族、遮挡变化但是图像质量比较高的测试集。这个测试集共收集图像 138 张，其中人脸共 142 个。

总体检测率为  $137/142 = 96.5\%$ 。共有 1 个误检。

人脸具体情况和检测结果见表 4.5。

表 4.5 正面人脸检测器在一个自选测试集上的测试结果

图像分类	图像数	误检数	成功检测数目
白色人种	38张	1个误检	全检测到
褐色人种	8张	无误检	全检测到
黑人人种	19张	无误检	漏2张
黄人人种	34张	无误检	全检测到
口罩	5张	无误检	漏3张
帽子	26张	无误检	全检测到
墨镜	12张	无误检	全检测到

当然，本章提出的局部组合二值特征预分类器的分类方法更大的优势在速度。对训

练得到的局部组合二值特征+Haar 分类器的检测时间进行了观察。观察是在  $320 \times 240$  图像上进行的, 检测过程对图像中每个像素进行扫描, 对图像进行每次 0.8 倍的缩放, 以进行不同大小人脸的处理, 实际检测的人脸大小是  $24 \times 24$  到  $240 \times 240$ 。在 Pentium 4 3.20MHz 个人计算机上, 不加局部组合二值特征预处理分类器的检测时间超过 1 秒。然而加了预处理分类器之后, 检测时间降到了 40 毫秒。足足快了超过 25 倍之多。

另外, 在这里对上述的基于 Haar 特征的分类器和基于局部组合二值特征预处理分类器+Haar 后处理分类器进行更精确地计算复杂度上的分析。如表 4.3 中所示, 局部组合二值特征分类器包含的特征数目通常是 Haar 特征分类器包含特征数目的一半。在我们的局部组合二值特征检测器中, 一个窗口大概需要 15 个局部组合二值特征相关的计算。对于这 15 个局部组合二值特征相关的计算, 它包括 15 次查表和 15 次加法(分类器计算信度)。再加上预处理计算局部组合二值特征图每个窗口所需的计算, 11 次加法, 4 次对比计算, 那么运行局部组合二值特征分类器总共需要 15 次查表、26 次加法和 4 次对比。

而相应的 Haar 检测器中, 大概需要 30 个 Haar 特征相关的计算。具体来说, 30 个 Haar 特征相关的计算包括: 30 个 Haar 特征值计算(通过积分图, 每个特征至少需要 7 次加法操作), 30 次除法(方差归一化), 30 次查表和 30 次加法(计算信度)。因此, 总共需要约 240 次加法、30 次除法和 30 次查表。

显然, 从计算复杂度上来讲, 局部组合二值特征所需要的计算要小很多, 上面的实验也充分的证明了这一点。

#### 4.4.4 多姿态人脸检测器

使用本文提出的方法, 还训练了多姿态人脸检测器, 多姿态人脸已经在第二章详细描述, 共 15 种人脸姿态, 针对每个姿态分别训练相应的预分类器和深入处理分类器。我们对训练得到的多姿态人脸检测器在  $320 \times 240$  大小的图像上进行测试, 每个像素都进行扫描, 处理的人脸大小为从  $24 \times 24$  到  $240 \times 240$ , 所需要的检测时间大约为 100 毫秒。可以看到, 多姿态检测器的检测时间并不是随着姿态数目的增加而线性增加, 这是由于占用大量处理时间的图像预处理被所有 15 种姿态分类器所共享(包括积分图计算,  $3 \times 3$  局部组合二值特征图像的计算等等)。

学习得到的多姿态人脸检测器在 CMU 侧面人脸公开测试集上进行了测试, 这个测试集包含 208 幅图像, 其中有 441 张各种姿态和场景的人脸。可以看出, 加入预分类器之后的多姿态人脸检测结果仍然可以接受, 我们的结果获得了与当前报告的最好结果是具有可比性的。

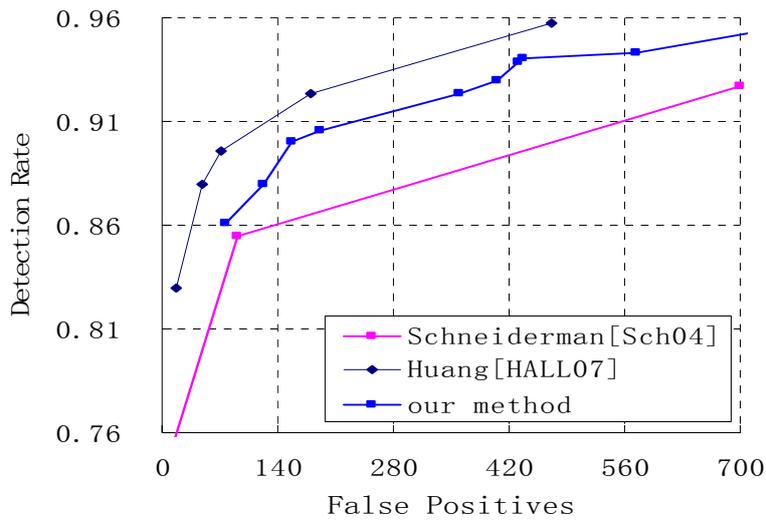


图 4.11 CMU 侧面人脸测试集上的 ROC 曲线

#### 4.5 小结

本章提出了一种新的特征——局部组合二值特征，并相应的提出了使用局部组合二值特征进行快速人脸检测预分类器的构建以及检测方法。在引入的人脸检测预分类检测过程中，采用以特征为中心的检测方法，并进一步提出了以特征为中心的瀑布型分类器检测结构。实验表明，提出的方法能够在保持高检测率的同时，排除大量的非人脸窗口，大幅度提高人脸检测算法的速度。本算法的提出能够大大缓解人脸检测算法中性能和速度不能兼得的问题。并且，局部组合二值特征的提出和单特征瀑布型分类器的提出具有很高的学术意义和应用价值。

当然，对于本章提出的预分类器构建框架中，考虑到搜索空间的计算代价，对于局部组合二值特征的使用只选择了一种尺度下的特征来进行快速预分类器的构建，虽然在此情况下人脸检测算法在速度上已经获得了巨大的进展，但是是否存在更加有效的特征尺度组合或者是另外一种二值特征依然是一个值得探讨的问题。另外，对于基于每个特征的瀑布型分类器，显然还存在更优化的层级调节方法。最后，本文提出的方法是否能够成功地应用于文字、汽车、人体等等，这些都是很有趣的应用，这都是下一步需要进行研究和探讨的工作。

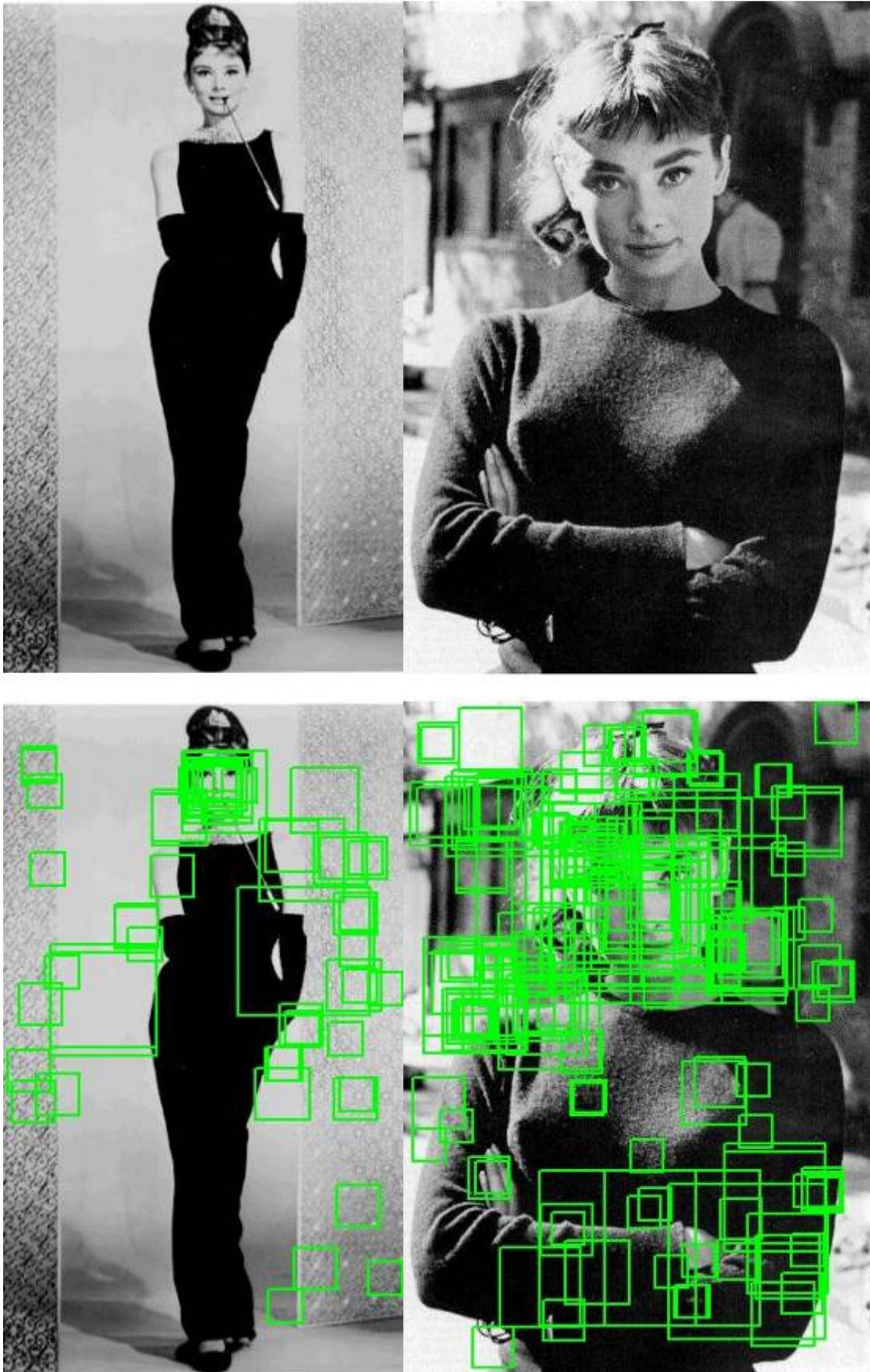


图 4.12 使用局部组合二值特征预检后的结果图片 1

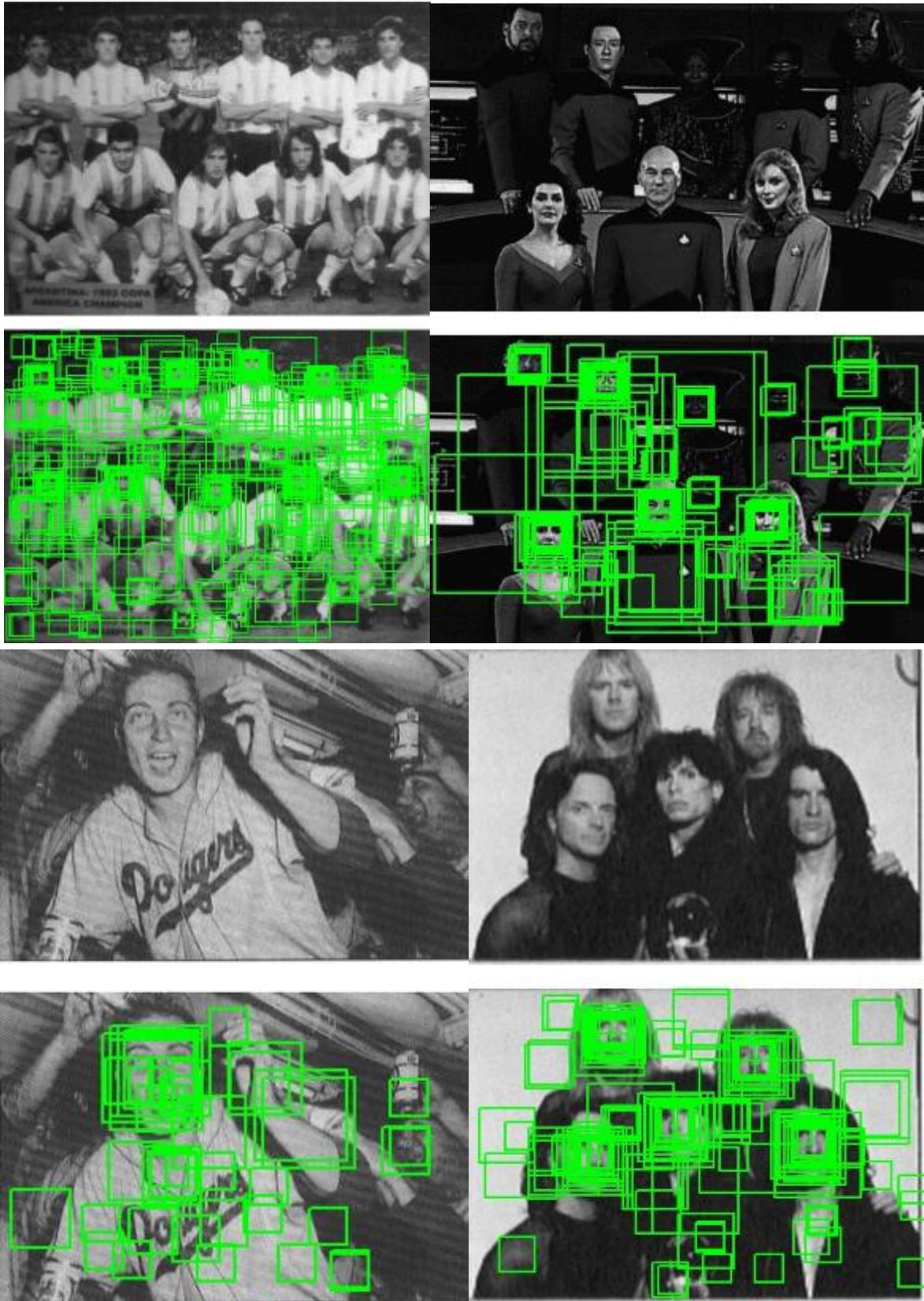


图 4.13 使用局部组合二值特征预检后的结果图片 2

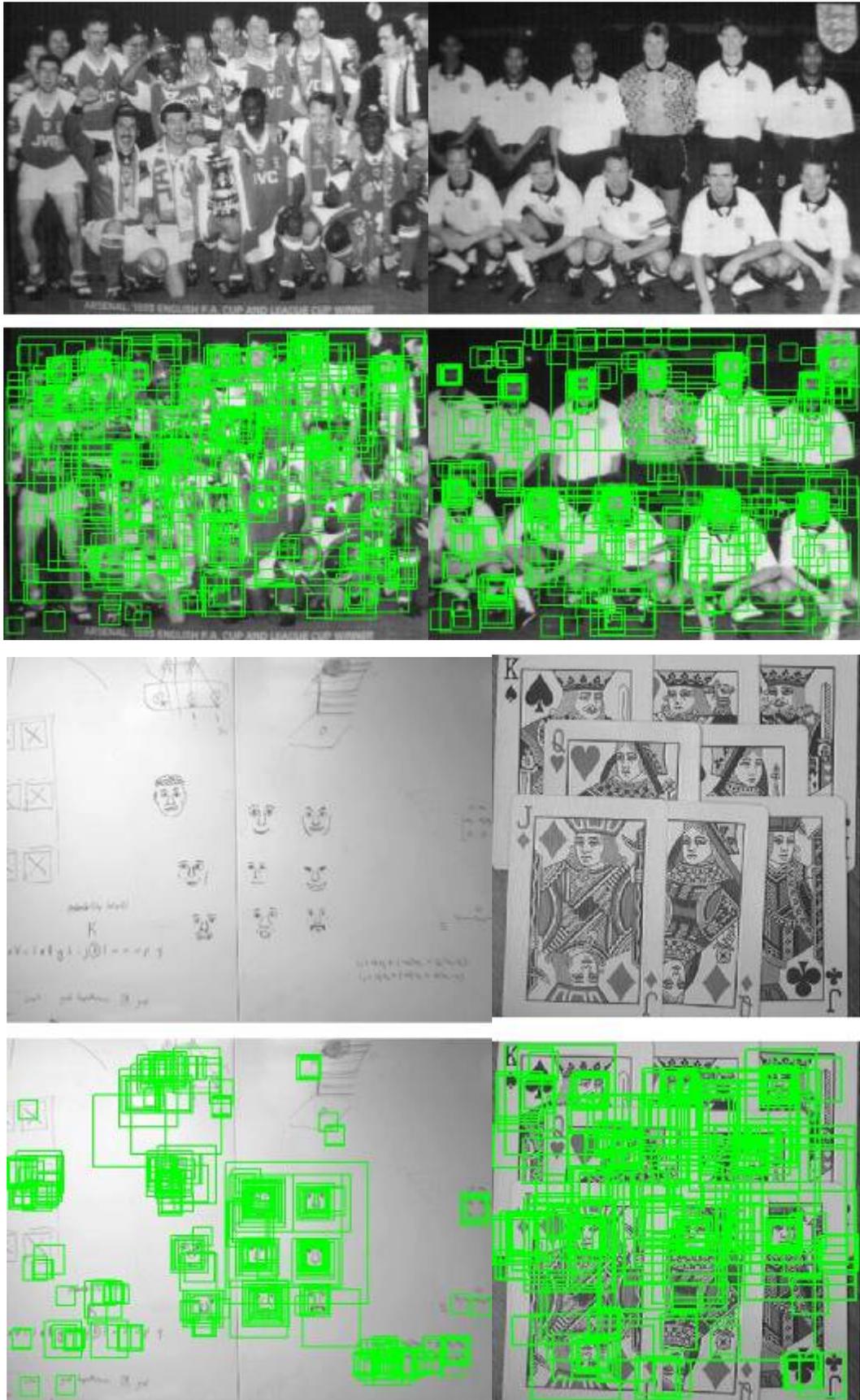


图 4.14 使用局部组合二值特征预检后的结果图片 3

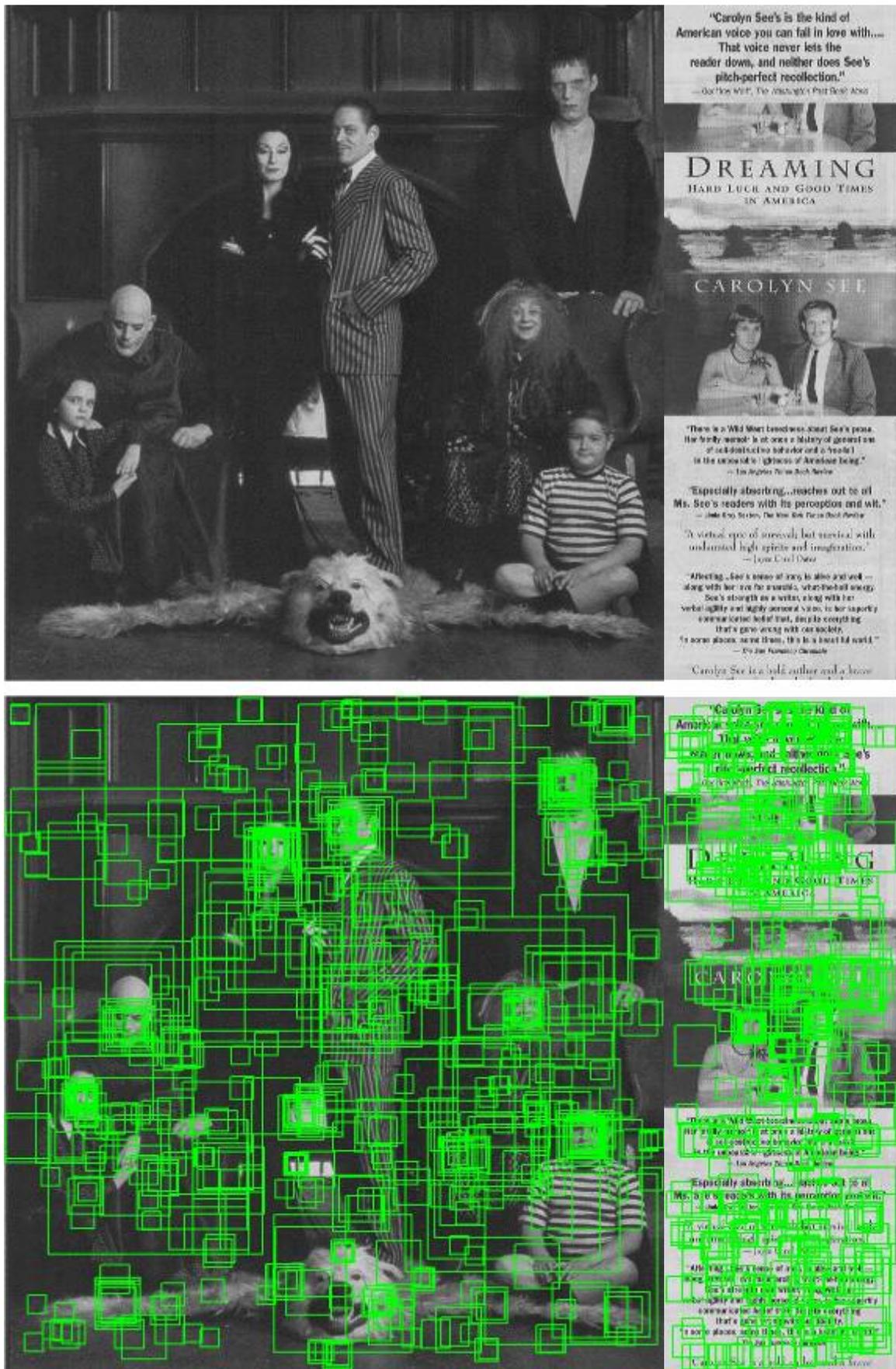


图 4.15 使用局部组合二值特征预检后的结果图片 4



图 4.16 使用局部组合二值特征预检后的结果图片 5



## 第五章 基于搜索的瀑布型分类器构建方法

近年来一些物体检测系统逐步进入产业化阶段，实际应用中，物体检测技术面临的一个重大挑战就是更加快速的物体检测系统。经常，一个应用中的物体检测系统需要在速度\精度之间进行折中，以获取用户需要的最好的整体性能。这就需要物体检测系统能够提供不同检测精度条件下的速度最快的检测系统，或者说是相同速度下精度最高的分类器。

这可以通过两种技术手段来实现。1、训练多个不同速度标准下的分类器，我们称之为多分类器方法。2、训练一个高精度的分类器，在此基础上对此分类器进一步调节获得不同速度下的最好的分类器参数，我们称之为单分类器多参数方法。这两种方法在应用系统中的执行流程如图 5.1 所示。多分类器方法的优点是可能能够获得速度/精度折中更好的分类器，但是缺点也非常明显。第一个缺点是分类器学习阶段需要学习多个分类器；第二个缺点是应用阶段物体检测系统中必须保存多各分类器的信息。当前能够实用的分类器学习算法普遍过程复杂、学习时间长，所以多分类器方法的第一个缺点被进一步放大。另外，一个应用系统往往对物体检测系统能够占用的内存有限制，这凸显了多分类器方法的第二个缺点。使得其在很多场合的不能成功应用。单分类器多参数方法不存在上述多分类器方法存在的两个缺点。在其分类器学习阶段首要目标是获得一个高精度分类器，训练时间短。并应用调节算法调节得到各速度标准下最好的分类器，应用阶段只存放调节得到的对应不同速度的参数。参数的大小要远远小于分类器本身所占用的内存空间。

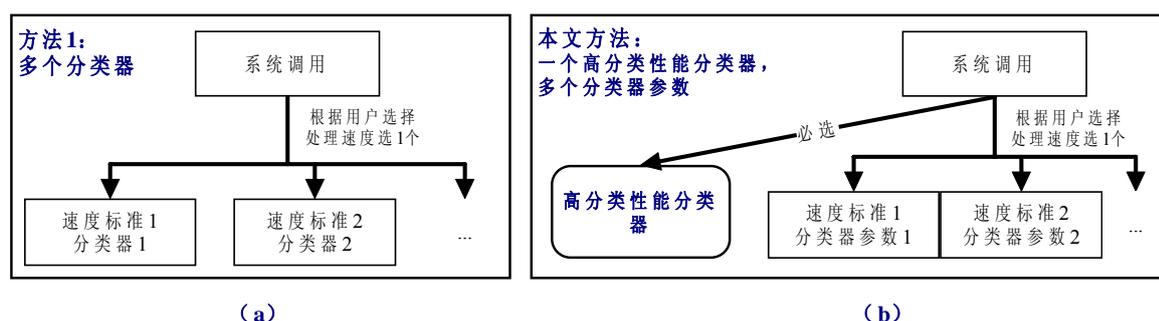


图 5.1 两种解决应用中不同速度需求的方案。(a)训练多个分类器，(b)训练一个高性能分类器，并调节得到多个速度下的分类器参数

对于单分类器多参数方法来说，可以使用很多现有技术来训练一个高精度分类器，这些技术在第一章中都有所介绍。针对 Boosting 瀑布型分类器方法获得的高精度分类器，本章提出了一种在限制检测精度的条件下，自动调节高精度分类器到指定精度的快速瀑布型分类器调节方法，以前没有发现类似工作。使用本方法，可以获得各个指定精度下的 Boosting 瀑布型分类器参数，并构建不同速度标准下的物体检测系统。

限定精度下的速度调节是通过构建不同的瀑布型分类器参数实现的。本章提出一种新的基于搜索的方法。搜索算法对可能存在的瀑布型分类器进行搜索，并获得计算代价更小的瀑布型分类器。瀑布型分类器空间很大，为了降低搜索算法的计算负担，搜索算法递进迭代运行。搜索开始于正例分类正确率为 1 的瀑布型分类器。每一轮的运行在当前瀑布型分类器基础上，搜索进一步损失小部分正例分类正确率条件下所有可能的瀑布型分类器，并获得当前计算代价最小的瀑布型分类器。然后，在当前获得的瀑布型分类器基础上，迭代运行，直到获得的瀑布型分类器正例分类正确率达到用户指定的正例分类正确率，算法终止，输出瀑布型分类器。我们在正面人脸检测问题上对基于多种不同特征的集成分类器进行了实验，实验表明了本文算法的有效性。将本算法应用于人脸检测系统中，在正例分类性能只损失约 5% 的情况下，系统需要处理特征数目下降为原来的一半。另外，我们还对多姿态人脸检测进行了实验，在多姿态人脸检测系统在  $320 \times 240$  图像上处理速度达到 80ms 时，系统人脸检测率比原来方法提高了近 3%，效果非常明显。

本章所述内容的技术创新之处有三个。首先第一次提出并形式化了基于搜索的瀑布型分类器调节算法。基于搜索的算法能够获得比以前人工干预调节的方法获得更好的效果。其次，提出的搜索算法考虑了所有可能的瀑布型分类器空间，这使得寻找到的瀑布型分类器更优化。第三，本文提出的搜索算法将构成瀑布型分类器的每个子分类器的计算复杂度区别考虑。这种瀑布型分类器调节算法可以应用于任何不同类型特征训练构成的待调节分类器中，此点区别于文献[BB05]中的方法，是一种更加通用的调节算法。

本文的整体结构安排如下：在第 5.1 节中，首先介绍了 Boosting 瀑布型分类器的概念，然后，第 5.2 节描述了本文提出的基于 Boosting 高精度瀑布型分类器构建限定精度瀑布型分类器的算法以及其更进一步的扩展；第 5.3 节给出算法在人脸检测系统中的表现和两个应用实例的实验分析；第 5.4 节给出小结。

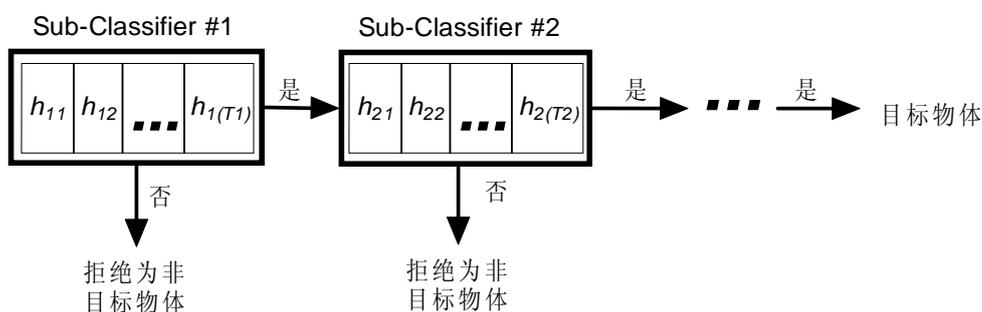


图 5.2 Boosting 瀑布型分类器

## 5.1 Boosting 瀑布型分类器中的特征

Boosting 瀑布型分类器如图 5.2 所示，它是顺序从前到后多个子分类器的串联，每个候选窗口只有顺序通过所有子分类器的决策，被每个子分类器都决策为物体窗口，才

能被瀑布型分类器决策为物体窗口。而一旦在某个子分类器决策为非目标物体，则放弃后边子分类器的进一步决策，被瀑布型分类器决策为非目标物体。

在物体检测应用阶段，所有尺度图像上的所有位置被瀑布型分类器所决策，并输入物体\非物体类别。此过程示意图参看图 1.4。原始输入图像在图 1.4 的左上角，缩放后的图像在右下方向，对每个尺度图像的每个位置存在的候选图像窗口运行分类器并找出目标物体窗口。扫描窗口的具体操作是在此窗口上运行分类器。

物体检测中的瀑布型分类器设计利用如下认知来提高系统处理速度，客观世界中大部分候选窗口可以通过简单而计算代价低的子分类器进行排除。所以简单而计算代价低的子分类器存在于瀑布型分类器的前端，复杂而计算代价高的层级分类器存在于后端。这种设计提高了系统的计算效率，大量简单而容易区分的非目标物体窗口被前端简单的子分类器排除，只有少数复杂“疑似”目标物体窗口通过前端分类器后触发后端复杂而计算代价高的子分类器的运行。

Boosting 瀑布型分类器中涉及的 Boosting 分类器形式如下：

$$H_T(x) = \sum_{t=1}^T h_t(x) \quad (1)$$

其中  $H_T(x)$  表示强分类器， $x$  表示输入图像窗口， $h$  是弱分类器， $t$  表示某个弱分类器的序号， $T$  是强分类器中包含的弱分类器的总数目。

集成分类器是多个弱分类器的线性组合，弱分类器从一个弱分类器假设空间由 Boosting 顺序挑选得到。挑选时的衡量标准是分类错误率，错误率的计算是在每个样本拥有权重的样本集上进行计算的。算法会根据已挑出的分类器更新样本的权重，对于分类正确的样本，增加权重，对于分类错误的样本，降低权重，以保证整体分类错误率是下降的。

集成分类器的分类判决过程是使用强分类器  $H(x)$  值和一个实数阈值对比的过程。如果强分类器值超过阈值，那么决策输入窗口  $x$  为物体，反之，如果强分类器的值小于阈值，则判断为非物体目标。决策函数可以表示为：

$$C_T(x) = \begin{cases} 1 & \text{if } (H_T(x) \geq \theta_T) \\ 0 & \text{if } (H_T(x) < \theta_T) \end{cases} \quad (2)$$

其中  $C$  是决策函数， $\theta$  是设定的阈值，决定分类器的分类边界位置。

检测率和误检率是物体检测分类器分类精度的重要标志。 $C(x)$  的检测率表示在已经确定为目标物体窗口的测试集上，分类器实际判定为目标物体窗口的个数与总测试样本个数的比例，表示如下：

$$d = \frac{\sum_{i=1}^n C(x_i)}{n} \quad (3)$$

其中  $n$  是目标窗口测试集中拥有的所有测试窗口数目。 $C(x)$ 的误检率表示在已经确定为非目标物体的窗口测试集上, 分类器实际判定为非目标物体窗口的比例, 表示如下:

$$f = \frac{\sum_{i=1}^m (1 - C(x_i))}{m} \quad (4)$$

其中  $m$  是非目标窗口集合中拥有的所有测试窗口数目。

不同的阈值  $\theta$ , 直接影响分类器  $C(x)$ 的检测率和误检率。 $\theta$  值越大, 检测率越低, 误检率也低;  $\theta$  值越小, 检测率越高, 误检率也越高。

瀑布型分类器学习阶段也相应从前到后进行学习, 针对收集到的正例样本集和大批量的反例样本集, 首先收集第一批反例样本进行第一个子分类器的学习, 子分类器的复杂度由指定的最小检测率和最大误检率所控制, 如果当前 Boosting 学习到的分类器未能达到指定目标性能, 则继续增加弱分类器, 直到达到目标性能或者弱分类器的数目超出某个最大值为止。然后使用“自举”的方法重新从大批量的反例样本集中收集误检的反例样本, 并进行第二个子分类器、第三个子分类器的学习, 直到瀑布型分类器总的误检小于一个指定值为止。使用这种分类器学习方法, 可以获得一个高精度(检测率接近 1, 误检率接近 0)的瀑布型分类器。

## 5.2 精度指定条件下瀑布型分类器速度优化

### 5.2.1 瀑布型分类器的计算复杂度

瀑布型分类器在一个测试集上的平均计算代价通过对测试集中所有测试图像窗口运行瀑布型分类器获得。平均计算代价表示如下:

$$CM = \sum_{l=1}^M p_l (c_l + v) \quad (5)$$

其中  $p_l$  为第  $l$  层决策之后通过本子分类器决策的图像窗口与所有图像窗口的比值,  $M$  是瀑布型分类器拥有的子分类器的数目,  $c_l$  是第  $l$  个子分类器的计算代价,  $v$  是子分类器的最后决策阈值对比带来的计算量。 $p_l$  可以通过调节子分类器阈值改变,  $c_l$ ,  $v$ ,  $M$  是确定的。

对于一个瀑布型分类器, 如果提高位置靠前的瀑布型分类器的子分类器阈值, 那么会有更多的输入图像窗口在本子分类器位置被排除, 所以在测试集上总计算代价会减小。

很明显, 升高某些子分类器的决策阈值只会使得瀑布型分类器的误检率进一步减小。这对实用中的检测系统是有利的。所以, 在瀑布型分类器的调节中, 不关注误检率的变化情况。但是对于检测率来说, 升高阈值却会带来检测率的下降。那么, 我们要解决的问题就是针对一个指定允许的检测率, 调节得到一个计算代价更小的瀑布型分类器。

## 5.2.2 不同检测精度计算复杂度优化

本节讨论在给定正例样本测试集和反例测试样本集的情况下，如何对一个拥有多个子分类器的阈值进行调节，在指定正例样本集上检测率条件下，获得在反例测试样本集上计算代价更小的瀑布型分类器。

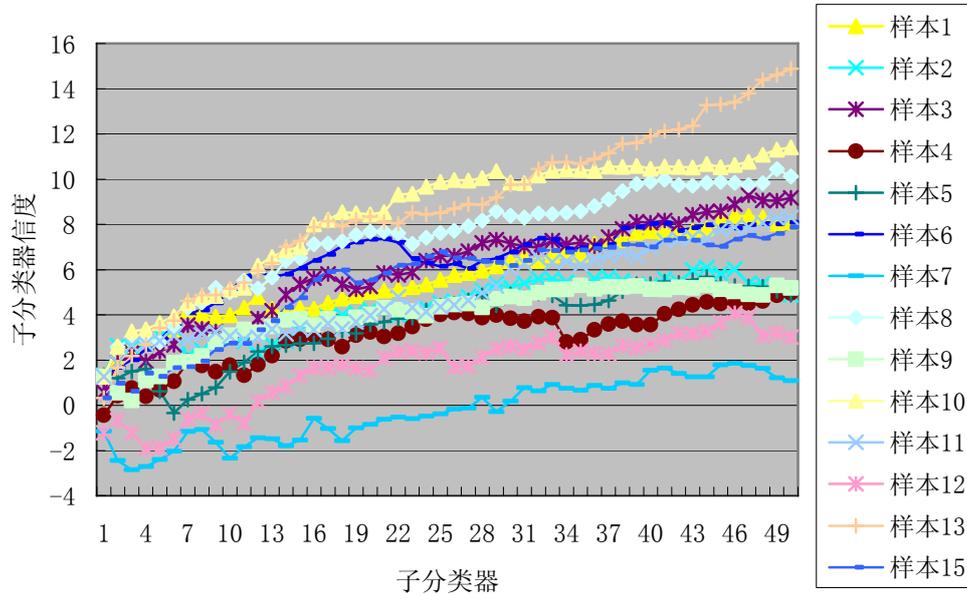


图 5.3 部分样本的信度曲线

在子分类器数目和顺序确定的情况下，针对指定检测率，怎样控制每个子分类器的阈值增加幅度是一个资源分配的问题：将待损失的检测率分配各子分类器并相应提升阈值获得最大的计算减少量。但是与普通的资源分配问题不一样的是，当部分损失检测率被分配到某个子分类器时，其它的子分类器可能与其共享部分损失检测率，所以也需要相应的升高阈值，以获得计算代价的更优化。为了解释这一点，我们引入有关子分类器的一个概念，强分类器值变化曲线。对于一个集成分类器，对于每个样本，我们可以获得此样本的强分类器值，如式(1)所示。强分类器值越大，此样本越可能属于物体类。瀑布型分类器的每个子分类器对应的强分类器，对每个样本都有其相应的值，将一个样本不同子分类器的强分类器值按子分类器顺序排列，就得到一个样本强分类器值的集合，以子分类器序号作为横坐标，相应的强分类器值为纵坐标，每个样本形成一条强分类器值曲线。图 5.3 显示了一个拥有 150 个子分类器的瀑布型分类器的每个可能子分类器的部分样本的强分类器值曲线。对于具有一定损失检测率的瀑布型分类器，测试正例样本集被分为两个集合，测试正确样本集合和测试错误样本集合。瀑布型分类器要使得每个需要测试正确的样本的在每个子分类器位置处都大于它，而允许损失的样本在某些或者全部子分类器位置的信度小于阈值。给定允许损失的样本数目，为了寻找计算最优的瀑布型分类器，通过提升阈值来达到减小  $p_1$  的目标，从而降低计算代价。无论在哪个子分类器进行样本损失，这些损失的样本都必须包含强分类器值最低的那个样本。以图 5.3 中的样本为例，如果目标是损失一个样本的话，那么损失的样本就是样本 7，因为其它

样本的强分类器值在每个子分类器位置都大于其强分类器值。

从样本的角度，当然可以将待求解的问题转换为寻找带来最优计算增益的指定大小的损失样本集。假定现在目标损失样本个数为  $N$ ，为了找到损失  $N$  个样本的最优瀑布型分类器，理论上需要对每个可能存在的损失样本集进行遍历才能找出计算最优的瀑布型分类器。但考虑到损失样本总是出现在相应强分类器值最小的样本，所以，可能尝试的样本最多有  $T$  个。同理，对每个去除一个样本后的路径，又有  $T$  个最小强分类器值样本可以尝试，如此反复  $N$  次，去掉  $N$  个样本就可获得所有可能的瀑布型分类器。以上过程的搜索的空间的数量级是  $TN$ 。计算复杂度非常高，假设  $N=100$  的话，即使  $T=2$ ，有  $2100=1030$  大小，这是不可接受的。

为了寻找任意指定检测率情况下的计算更优的瀑布型分类器，首先给出在一个指定集合上检测率为 1 的瀑布型分类器阈值调节算法。

#### (1)检测率为 1 的最优的瀑布型分类器

根据 5.1 节描述的瀑布型分类器的特点，将第 1 个子分类器表示为  $SUB_1(x)$ ，相应的强分类器函数表示为  $H_1(x)$ ，相应的阈值表示为  $\theta_1$ 。则相应可能的瀑布型分类器的变化参数为  $\{\theta_1, \dots, \theta_M\}$ 。

瀑布型分类器在正例样本测试集上测试损失的正例样本个数为 0，也就是说所有正例样本都被所有的子分类器正确分类。具体来说，每个样本的瀑布型分类器中的子分类器的强分类器值都大于本子分类器相应的阈值。所以，对于每个子分类器，其阈值必小于等于拥有最小样本子分类器对应强分类器的值。另一方面，每个子分类器的阈值应该尽量大，使得通过本子分类器的候选窗口的概率  $p_i$  尽可能小，以降低计算代价。对于一个正例样本测试集，子分类器的阈值唯一确定，求解阈值的算法见图 5.4。

---

输入： $T$ 个弱分类器构成的集成分类器；测试正例样本集  $S$ 。

初始化：

当前子分类器弱分类器位置  $i=1$

执行步骤：

- (1) 使用当前子分类器  $i$ ，对  $S$  内每个正例样本计算强分类器信度，获得最小的信度，并设置其为子分类器  $i$  的阈值；
  - (2)  $i=i+1$ ；如果  $i < T+1$ ，转 (1)；否则，结束算法，输出当前瀑布型分类器。
- 

图 5.4 构建一个正例测试集上检测率为 1 的瀑布型分类器算法

#### (2)构建任意检测率的瀑布型分类器

下面我们提出一个次优前向递进搜索算法来进行任意检测率情况下的瀑布型分类器搜索算法。搜索开始于损失样本数为 0 的瀑布型分类器，以后每一轮的运行尝试搜索在当前瀑布型分类器正确分类正确率基础上进一步损失小部分正例样本条件下所有可能的瀑布型分类器，并找到计算代价最小的瀑布型分类器，此分类器作为下一轮搜索的起

点，循环损失样本，直到损失的正例样本数目达到指定的损失正例样本数目，算法终止。搜索算法每次循环都使用损失一部分样本后最好的瀑布型分类器开始尝试进一步损失样本后的瀑布型分类器，并在此损失样本数目的条件下可能的迭代使用的瀑布型分类器都基于上次迭代得到的最优瀑布型分类器，并在此基础上进一步损失少量的样本并查找更优的瀑布型分类器。本文提出的算法见图 5.5。如果每次损失样本数目为 1，此算法的计算复杂度为  $M \times N$ ，是完全可接受的。另外算法中引入了一个变量每次迭代损失个数  $U$ ，它也可以控制算法的计算复杂度，算法的计算复杂度为  $M \times N / U$ 。 $U$  的引入也是为了避免在某个特征值处多个不同的样本在同一个子分类器出现同一个信度时，调节算法失去对此子分类器的调节能力。比如  $U=1$  的情况，如果在样本集中出现了 2 个相同的样本，那么至少 2 个样本在各个子分类器位置信度相同，此时搜索算法就会无能为力，因为所有的位置计算量下降都为 1。但是如果  $U$  设置的大于 1，就能发现此位置真实的对计算量减少带来的贡献。由于测试集是有限的，我们希望通过  $U$  来控制某些位置偶然出现的计算代价降低不多的情况，获得更有统计性的子分类器损失样本后的计算增益。

输入：

$T$  个子分类器构成的瀑布型分类器；测试正例样本集  $PosSet$  和测试反例样本集  $NegSet$ ；目标损失样本个数  $M$ ；每次迭代损失样本个数  $U$ ，每个弱分类器计算代价  $C_i$

初始化：

检测成功正例样本集  $S^c = PosSet$ ；检测失败正例样本集  $S = \Phi$ ；（其中  $S \cup S^c = PosSet$ ， $S \cap S^c = \Phi$ ）

使用图 1 算法， $S^c$  作为正例样本测试集获得在  $S^c$  上检测率为 1 的瀑布型分类器，设置为当前瀑布型分类器。

执行如下步骤：

- (1) 记录当前瀑布型分类器各阈值为  $PrevThresholds$ ， $i=1$ ，利用  $NegSet$  和  $\{C_i | 0 < i < T+1\}$  计算当前子分类器计算代价  $V_0$ ；
- (2) 对当前子分类器  $i$ ，在成功检测集合中信度第  $U+1$  大的样本，找到信度最小的  $U$  个样本，并构成集合，记录为  $S_j$ ，设置当前第  $i$  个子分类器阈值为倒数第  $U+1$  大样本的信度；
- (3) 循环查找所有除子分类器  $i$  外的子分类器在  $S$  内的最小信度样本，如果此样本在  $S_j$  中出现，更新相应子分类器阈值为新的此样本的信度，直到所有的子分类器最小信度样本都不在  $S_j$  中出现，转 (4)；
- (4) 计算当前阈值设置下瀑布型分类器的计算代价，记录此值为  $V_i$ ；记录当前瀑布型分类器阈值  $VVV_i$ ；
- (5) 更新当前瀑布型分类器各阈值为  $PrevThresholds$ ； $i=i+1$ ；如果  $i < T$ ，转 (2)；否则，转 (6)；
- (6) 从  $\{(V_i - V_0) | 0 < i < T+1\}$  数集中挑出最小值，假设为  $V_j - V_0$ ，那么更新瀑布型分类器阈值为  $VVV_j$ ； $S^c = S^c - S_j$ ， $S = S + S_j$ ，如果  $S$  中样本数目大于  $M$ ，结束算法，输出当前瀑布型分类器。否则，转 (1)。

图 5.5 构建损失指定个数正例样本情况下的瀑布型分类器算法

本算法中的每次迭代都找到计算量减少的局部最优点的瀑布型分类器进行进一步的调节，最终的结果可能并不是全局最优点，但是算法本身计算复杂度不高。相对于人为限定每个子分类器损失样本的方法，更有可能获得最优解。

### 5.2.3 扩展：基于每个弱分类器的瀑布型分类器调节算法

5.2 节提出针对子分类器进行不同精度瀑布型的分类器的调节算法。可以看出来，瀑布型分类器中的子分类器个数和顺序利用原有高精度分类器中的子分类器和排序。考虑到每个子分类器的计算复杂度有可能仍然偏大，调节算法的精细程度仍然不够，每个子分类器还可以进一步细分以获得更多可能的瀑布型分类器。比如对于每个子分类器的目标检测率 0.9999 和目标误检率是 0.5，高精度的分类器的第一个子分类器能够获得 13 个弱分类器，显然将这 13 个弱分类器作为一个整体进行速度调节出的速度有限，还可以进一步将它们细分为更多的子分类器。这些子分类器被称为 SubSub 子分类器。图 5.6 示例了这样的一个集成分类器，图中弱分类器被顺序排列，一个小矩形表示一个弱分类器，顺序从左到右共有  $T$  个弱分类器，它们线性加和即可得到整体的集成分类器。图 5.6 中同时示例了一个对一个 Sub 子分类器进一步细分为  $M$  个 SubSub 子分类器的瀑布型分类器，弱分类器的顶端使用弧线覆盖了每个子分类器所包含的弱分类器。每个子分类器依然如式(2)中表示，不过  $T$  值不同。可以看出来，每个子分类器是由第一个弱分类器开始，在某个弱分类器终止的所有弱分类器的线性组合。所以，只记录终止的弱分类器位置即可以确定一个子分类器。对于瀑布型分类器，可以在任何弱分类器位置设置阈值，形成一个子分类器。

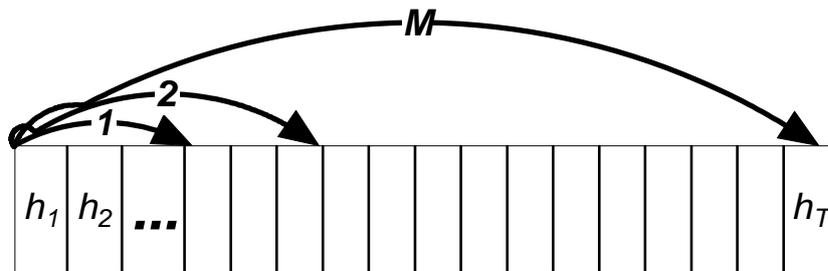


图 5.6 集成分类器到瀑布型分类器

而对由 SubSub 子分类器进行进一步调节的方法和对 Sub 子分类器调节的方法存在很多相同之处，因为 SubSub 子分类器的顺序也是训练阶段确定下来，然后调节算法如果将每个弱分类器位置都考虑存在一个 SubSub 子分类器的话，应用起来也有很多相同之处。这基于 Boosting 算法训练时的一个认识：更强的子分类器会被放置在集成分类器的更前位置。显然，如果每个弱分类器位置考虑存在一个 SubSub 子分类器，一个拥有  $T$  个弱分类器的集成分类器最多拥有  $T$  个子分类器。SubSub 子分类器和 Sub 子分类器不同的是子分类器的内部结构不同，所有相关的计算都要同时考虑其在 Sub 子分类器内的情况。

有了这些顺序排列的 SubSub 子分类器，应用算法 5.5，可以进一步调节得到一个速度更加优化的瀑布型分类器。

#### 5.2.4 反例样本集重采样

以上速度调节的算法中假设使用反例样本集来进行瀑布型分类器的计算复杂度估计是时间代价较小的。但是现实情况是反例样本多种多样，数量巨大。为了获得更好的具有通用性的分类器计算复杂度估计，测试一次瀑布型分类器计算代价很大。这会导致调节算法运行速度很慢。

算法 5.5 的调节每次迭代过程中，可以每次计算复杂度估计都在不同的随机采样挑的反例样本集上进行。具体来说，在算法中第(5)、(6)步骤的运行中使用不同的反例样本集，以达到目标性能的瀑布型分类器在面对不同现实情况样本能够获得相符合的计算代价。

### 5.3 实验

为了验证本文算法的有效性，我们在正面人脸检测和多姿态人脸检测问题上对算法进行了实验。本节安排如下，5.3.1 描述了输入的瀑布型分类器；5.3.2 节将算法应用于单个正面人脸检测器并给出实验结果，5.3.3 节中将算法应用于多姿态各个分类器后并给出实验结果。

#### 5.3.1 输入瀑布型分类器

我们使用如第四章中所述的技术首先获得一个高分类性能的瀑布型分类器。瀑布型分类器的最终检测率在训练集上达到 0.998。误检率的衡量标准是在我们收集的 30000 张确认不包含人脸的图像中没有误检为止，这其中存在上亿的非人脸样本。

如第四章中所述，瀑布型分类器第一个子分类器由 150 个  $3 \times 3$  尺度的局部组合二值特征构成。这些特征值预先计算。所以在瀑布型分类器运行过程中，获得特征值只需根据分类器运行时需要从预先计算的特征图像中查表获得即可。每个弱分类器的计算包括一次查表获得信度，并将信度累加。瀑布型分类器第一个子分类器以后的子分类器由 Haar 特征构成。每个弱分类器的计算包括特征计算和信度累加。

#### 5.3.2 正面人脸检测

本节调查一个真正实用的快速人脸检测系统，如何使用本文算法进行进一步的性能提高。本节所述的分类器是如第四章中所述方法获得的分类器。我们对  $3 \times 3$  尺度的局部组合二值特征构成的第一个 Sub 子分类器的所有弱分类器位置构建子 SubSub 分类器，对 Haar 特征瀑布型分类器我们只对原有的 Sub 子分类器的阈值进行进一步的调节。这是为了降低调节算法的计算复杂度，因为在实际调查中我们发现影响本检测系统检测速度的主要在于前面的 150 个局部组合二值特征构成的整体集成分类器上。原有的瀑布型分

类器除了第一个由局部组合二值特征构成的第一个 Sub 子分类器之外,拥有 37 个由 Haar 特征构成的 37 个 Sub 子分类器。所以调节的子分类器个数为 187 个。

原有的瀑布型分类器检测率为 0.998,应用本文的调节算法,在检测率从 0.998 下降到 0.993 时,速度从原来的约 50 毫秒下降到约 30 毫秒。考虑到系统中的预处理时间为约 15 毫秒,瀑布型分类器的计算时间从 35 毫秒下降为 15 毫秒,处理时间约减为原来的一半(从 35 毫秒到 20 毫秒)。测试在  $320 \times 240$  大小的图像上进行,每个像素都进行扫描,处理的人脸大小为  $24 \times 24$  到  $240 \times 240$ 。程序使用 VC++6.0 编写,没有做任何的优化。测试的机器是普通个人电脑,CPU 为奔腾 3.2GHz,内存 1Gb。同时,对调节后检测率为 0.993 的瀑布型分类器在 CMU+MIT 正面人脸公开测试集上进行了测试,获得了据我们所知报告的最好性能。CMU+MIT 正面人脸公开测试集包含 130 幅图像,其中有 507 张各种情况下的人脸。这些人脸模式变化大,有些质量非常差。ROC 曲线见图 5.7,ROC 曲线横轴表示检测率,纵轴表示误检个数。我们的 ROC 曲线通过调节后处理瀑布型分类器的层数获得。为了对比,图中同时给出了近年来其它研究者报告的好的结果。

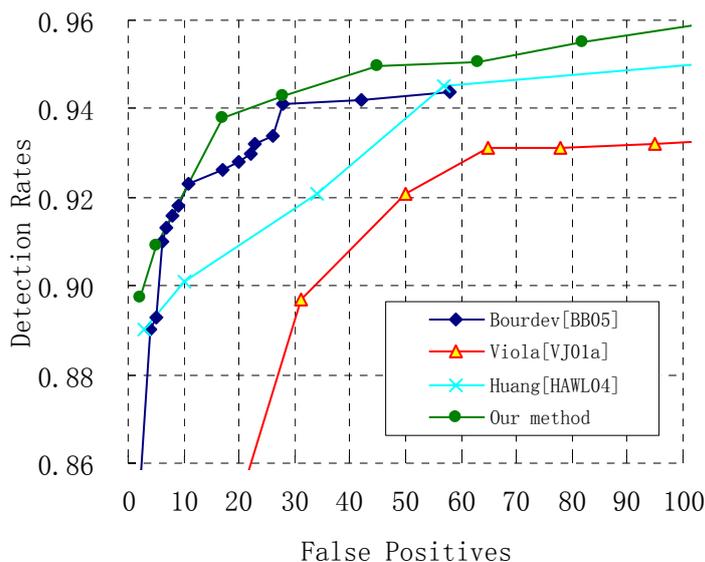


图 5.7 CMU+MIT 正面人脸测试集上的 ROC 曲线

### 5.3.3 多姿态人脸检测

本节调查一个真正实用的快速多姿态人脸检测系统。本节所述的分类器如第四章中所述的多姿态人脸检测器。和正面姿态人脸一样,我们对每种姿态下的  $3 \times 3$  尺度的局部组合二值特征构成的第一个 Sub 子分类器的所有弱分类器位置构建子 SubSub 分类器,对 Haar 特征瀑布型分类器我们只对原有的 Sub 子分类器的阈值进行进一步的调节。这是为了降低调节算法的计算复杂度,因为在实际调查中我们发现影响本检测系统检测速度的主要在于前面的 150 个局部组合二值特征构成的整体集成分类器上。

将原来的弱分类器位置排除在调节算法之外是为了使得新的瀑布型分类器误检率不

高于原来的瀑布型分类器，这也同时使得检测率不高于原有的瀑布型分类器。所有姿态的原瀑布型分类器检测率为 0.998，应用本文的调节算法，在检测率从 0.998 下降到 0.993 时，多姿态检测器处理速度从原来的约 150 毫秒下降到约 30 毫秒。考虑到系统中的预处理时间为约 80 毫秒，处理时间约减为原来的一半。测试在  $320 \times 240$  大小的图像上进行，每个像素都进行扫描，处理的人脸大小为  $24 \times 24$  到  $240 \times 240$ 。程序使用 VC++6.0 编写，没有做任何的优化。测试的机器是普通个人电脑，CPU 为奔腾 3.2GHz，内存 1Gb。同时，对调节后检测率为 0.993 的瀑布型分类器在 CMU 侧面人脸公开测试集上进行了测试，获得了与当前报导的性能可比的性能。CMU 侧面人脸公开测试集包含 208 幅图像，其中有 441 张各种姿态和场景的人脸。这个测试集人脸模式复杂，非常具有挑战性。ROC 曲线见图 5.8，ROC 曲线横轴表示检测率，纵轴表示误检个数。我们的 ROC 曲线通过调节后处理瀑布型分类器的层数获得。为了对比，图中同时给出了近年来其它研究者报告的好的结果。

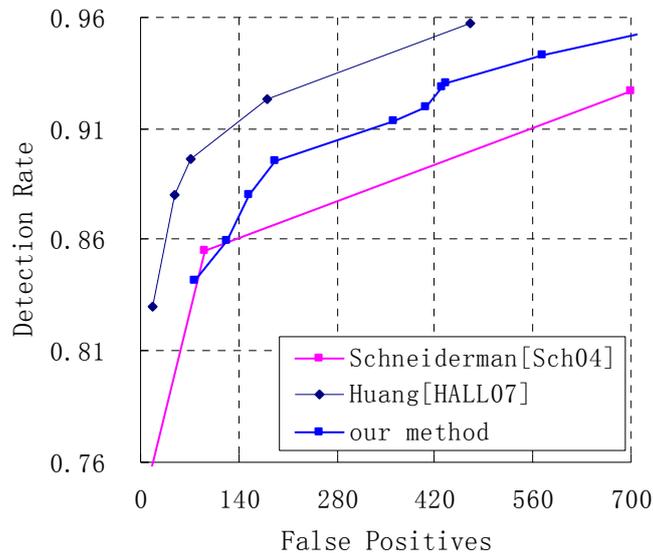


图 5.8 CMU 侧面人脸测试集上的 ROC 曲线

## 5.4 小结

本文提出了一种解决具有一定精度损失情况下快速瀑布型分类器调节算法，本文算法可以解决对应用中速度/精度更好的折中问题。提出一种搜索算法对可能存在的瀑布型分类器进行搜索，并寻找计算代价更低的瀑布型分类器。实验表明，在保持分类器分类性能的情况下，本文方法能够更好地降低所构建瀑布型分类器的计算代价。这对构建现实环境中常用的快速物体检测系统非常具有参考价值。



## 第六章 人脸检测原型及系统实现

本章描述了使用本文提出的算法构建的人脸检测原型系统。本章接下来的小节组织如下，首先 6.1 节描述为了方便使用本文算法训练得到的人脸检测器而实现的动态链接库模块；然后 6.2 节描述了一个人脸检测视频演示系统；最后 6.3 节中做出了小结。

### 6.1 动态链接库 JFD 模块实现

为了方便使用训练得到的人脸检测器，本文提到的多姿态人脸检测算法被实现为动态链接库以方便调用。动态链接库模块中有效地实现了 15 种姿态人脸检测分类器的灵活调用和检测过程中各种参数的方便调节，其中的函数包括：

FDInitialize	初始化检测函数
SetClassifiersPath	设置分类器文件在硬盘中存放的位置
GetClassifiersPath	获得分类器文件在硬盘中存放的位置
SetPanClassifier	设置调入内存分类器的平面内旋转角度范围
GetPanClassifier	获得内存中存在的分类器的平面内旋转角度范围
SetRollClassifier	设置调入内存分类器的平面外左右旋转角度范围
GetRollClassifier	获得内存中存在的分类器的平面外左右旋转角度范围
SetInputResolusion	设置输入图像大小
GetInputResolution	获得输入图像大小
SetInputBPP	设置输入图像模式（灰度图像或者是彩色图像）
GetInputBPP	获得输入图像模式（灰度图像或者是彩色图像）
SetMinDetectFaceSize	设置检测函数检测人脸的大小
GetMinDetectFaceSize	获得检测函数检测人脸的大小
SetPanDetectDegree	设置检测函数运行过程中平面内旋转角度范围
GetPanDetectDegree	获得检测函数运行过程中平面内旋转角度范围
SetRollDetectDegree	设置平面外左右旋转角度检测范围
GetRollDetectDegree	获得平面外左右旋转角度检测范围

SetSpeedAccuracyMode	设置检测的计算复杂度参数
GetSpeedAccuracyMode	获得检测的计算复杂度参数
SetInputParam	一次性设置所有参数
GetInputParam	一次性获得所有参数
FDFindFaces	检测函数
FDGetFacesInfo	获得检测结果

## 6.2 多姿态人脸检测视频演示系统

本文实现的多姿态人脸视频演示系统界面如图 6.1 所示，其中。其中“Settings”参数可以控制多姿态人脸检测的范围。范围包括第二章中所提到的 15 种姿态，使用两组参数来控制，分别是“Rotate in plane”和“Rotate out of plane”。其中“Rotate in plane”包含了平面内旋转的左倾斜(RIP11)、竖直(RIP12)和右倾斜(RIP01)人脸旋转角度范围；“Rotate out of plane”包含左全侧面(ROP 1)、左半侧面(ROP 2)、正面(ROP 3)、左全侧面(ROP 4)和左全侧面(ROP 5)。

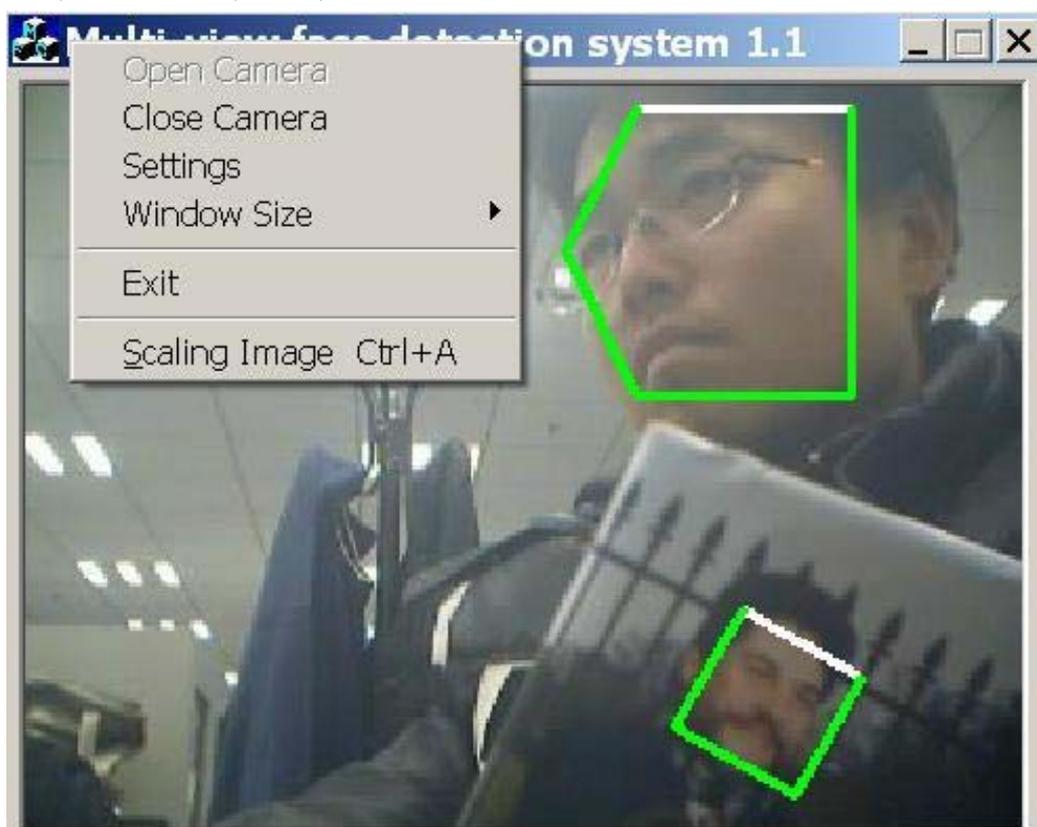


图 6.1 视频演示系统界面

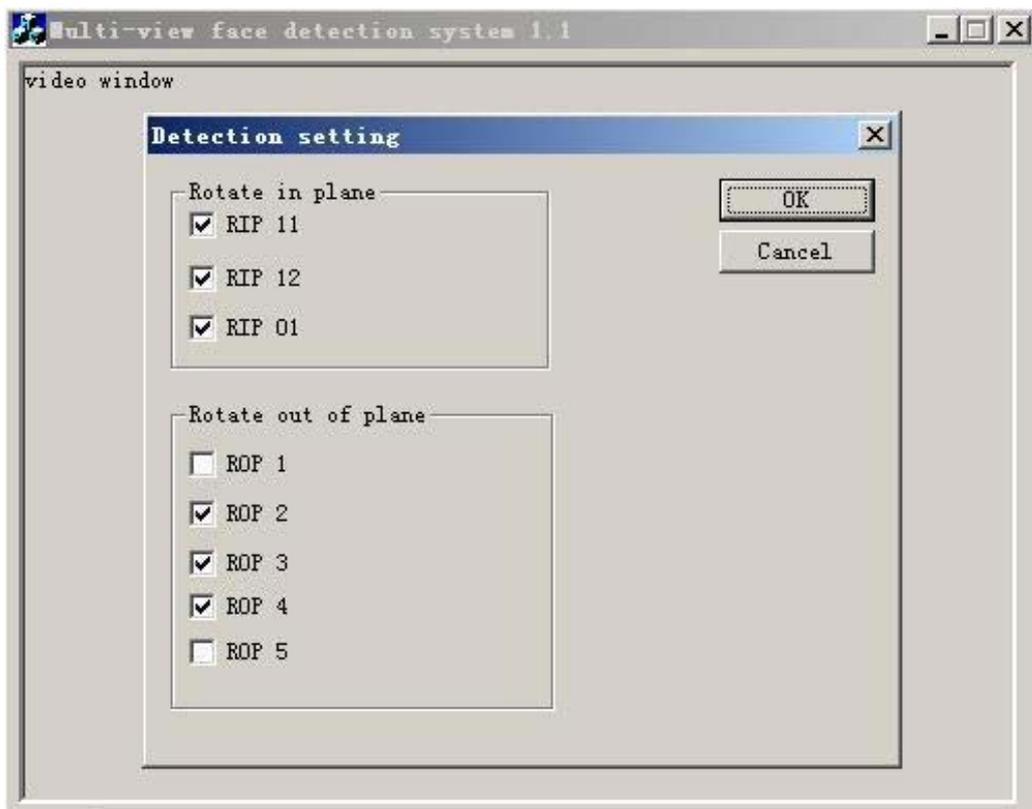


图 6.2 视频演示系统之检测姿态范围选择

### 6.3 小结

本节描述了前面章节中提出的检测算法的动态链接库实现和一个快速多姿态视频人脸检测演示系统，动态链接库方便了人脸检测算法的灵活设置和调用，视频演示系统方便地展现了当前人脸检测算法在实际场景中应用时的情况。



## 第七章 总结与展望

人脸检测研究具有重要的理论和应用价值，现已广泛应用到人机界面、基于内容的检索、数字视频处理、视觉监测等许多领域。目前已经开发出了一些实用的人脸检测系统，其中主流的人脸检测系统采用的分类器构建算法都是基于统计学习的算法构建的，目前看来如何进一步同时提高检测算法的精度和速度存在遇到瓶颈的趋势，针对此本文提出了一系列的相关算法，解决此问题，如前面章节中所述，本章对前面章节中的内容进行总结，并在此基础上探讨可能的后续扩展以及人脸检测的发展方向。

### 7.1 总结

本文针对瀑布型分类器的训练和检测框架提出一系列方法，主要的贡献包括：

- 1. 提出了一种基于大规模样本集的瀑布型分类器学习算法。**针对瀑布型分类器学习中同时存在大规模正例样本集和反例样本集的情况，提出一种叫做矩阵式学习的训练算法。在瀑布式分类器的训练过程中，矩阵式学习不仅在相邻的子分类器间对大规模反例样本集进行“自举”，而且，在每个子分类器自身的学习过程中，对大规模正例样本集也通过自举方式收集小部分具有代表性的正例样本进行最终强分类器的学习，矩阵式学习过程中的各个子分类器形成一个类似矩阵的结构，所以被称作矩阵式学习。更进一步，矩阵式学习不只简单地对大规模正例样本集和反例样本集进行自举，在自举过程中，相邻的子分类器的共享信息被用来进行更加快速的学习。这些共享信息包括两个方面，分类器训练过程中已经获得的特征和样本。矩阵式学习拥有非常高的学习效率，这非常有利于现实情况中的操作。同时，由于大规模正例样本集的引入，学习得到的检测器性能可以保证。我们在正面人脸检测和多姿态人脸检测上进行了详细的实验，实验表明本文方法解决了大规模正例样本集存在的情况下的训练时间问题。另外，在公开的人脸测试集上的测试结果验证了本文方法在分类性能上也表现卓越。相信本文提出的方法具有很大的实用价值和理论价值，并且可以于人脸之外的其它物体检测领域。
- 2. 提出了局部组合二值特征的概念以及使用局部组合二值特征构建快速预处理分类器的方法。**这包括了一种新的特征——局部组合二值特征，以及相应的以特征为中心的瀑布型检测算法。在提出的非人脸窗口预排除方法中，特征计算方式更加简单，避免可能重复的特征计算。局部组合二值特征的基本单元是二值特征，二值特征是图像上相邻区域间灰度差异的二值信息，即像 Haar 特征那样的多个区域间对比的更“黑”更“白”关系。局部组合二值特征是组合二值特征的一个有效子集，而组合二值特征表示的是多个不同二值特征值共生的信息。

在局部组合二值特征中，组合的方式被限制在类似局部二值模式组合像素点时的方式，形成局部组合二值特征。局部组合二值特征数目少，描述能力强，被用来进行预分类器的学习。预分类器采用了一种类似“注意机制”的以特征为中心的瀑布型分类器运行技术，以特征为中心的运行机制首先针对某个显著特征模板对图像每个位置计算出特征值，构成特征图像，然后在穷举得到的窗口上应用学习得到的本特征模板在各个位置的统计信息，并进一步判断窗口为人脸或者非人脸。无论是赖以计算的特征模板还是窗口内特征模板的统计信息都使用学习的方法得到，具体来说，模板由多个候选挑选得到的最优模板，统计信息是 Boosting 学习得到的瀑布型分类器。本方法在很大程度上缓解了性能速度不能同时兼得的难题。并且，局部组合二值特征的提出和单特征瀑布型分类器的提出具有很高的学术意义和应用价值。但是，对于局部组合二值特征，只是经验的选择了一种进行快速预分类器的构建，虽然在人脸检测上获得了巨大的进展，但是是否存在更加有效的特征依然是个很有意义的问题。另外，对于基于每个特征的瀑布型分类器，显然还存在更优化的层级调节方法。最后，本文提出的方法是否能够成功地应用于文字、汽车、人体等等，这些都是很有趣的应用，这都是下一步需要进行研究和探讨的工作。

- 3. 提出一种基于搜索的构建瀑布型分类器的算法。**基于一个已有的 Boosting 瀑布型分类器，我们提出一种构建多个更加快速的瀑布型分类器方法。新的瀑布型分类器允许损失一定的正例分类正确率，损失的大小由用户指定。提出了一种新的基于搜索的算法，对可能存在的瀑布型分类器进行搜索，并获得计算代价更小的瀑布型分类器。为了降低搜索算法的计算负担，搜索算法递进迭代运行。搜索开始于正例分类正确率为百分之百的瀑布型分类器。每一轮的运行在当前瀑布型分类器基础上，搜索进一步损失小部分正例分类正确率条件下所有可能的瀑布型分类器，并获得当前计算代价最小的瀑布型分类器。然后，在当前获得的瀑布型分类器基础上，迭代运行，直到获得的瀑布型分类器正例分类正确率达到用户指定的正例分类正确率，算法终止，输出瀑布型分类器。实验表明，本算法可以解决对应用中速度/精度更好的折中问题。在保持分类器分类性能的情况下，本文方法能够更好地降低所构建瀑布型分类器的计算代价。这对构建现实环境中常用的快速物体检测系统非常具有参考价值。

借助以上提出的针对瀑布型分类器的新方法，本文提出的人脸检测算法在速度和精度上都获得了显著的进步。借助贡献 1 提出的新的训练算法，大大缓解了反例样本集和正例样本集数量巨大的情况下，当前人脸检测算法计算代价过高的问题。另一方面大规模多变化的反例样本集和正例样本集保证了学到分类器的分类性能。这为进一步提高人脸检测性能提供了一个很好的解决方式，收集更多变化的样本，并应用矩阵式学习，而且，矩阵式学习是一个为单机运行（非多机器上分布式学习）设计，实现简单，方便使用。

借助于贡献 2 提出的算法, 本文构建了一个有效的快速预处理分类器作为传统 Haar 特征瀑布型分类器之外分类器的一部分。二值特征企图采用更为简单的计算来描述人脸的结构, 所有的计算都基于局部区域间的黑白二值特性, 进一步提出的预分类器检测方法基于这些二值特性直接对输入图像进行特性计算, 并最大化的利用每个像素点上获得的信息进行粗分类, 预分类器的加入能够大大降低检测器的计算复杂度。

不仅如此, 针对基于 Boosting 学到的瀑布型分类器, 本文进一步提出了对分类器进行计算复杂度调整的算法, 这使得人脸检测在拥有不同速度\精度需求的不同应用场景应用起来更加方便、灵活、高效。

## 7.2 展望

第一, 尽管本文所提出的多姿态人脸检测算法在  $320 \times 240$  的图像上, 在保证不错性能的同时, 达到了近实时的人脸检测速度, 在与同类算法的检测性能对比中, 也取得了一定的优势。但是, 在实际应用中, 这还远远不够, 速度问题始终还是一个急需解决的问题。我们需要在人脸姿态上覆盖更大的范围、在性能上更加鲁棒, 同时在速度上更快的鲁棒人脸检测系统。在多姿态人脸检测方面, 在本文的多姿态人脸检测算法中, 姿态范围并没有考虑图像外上下旋转的人脸姿态, 这导致在实际的应用中会出现露脸的情况。但是如果将此情况考虑进去, 仍使用现有的检测算法, 人脸检测速度却会下降。另外, 随着图像分辨率的增加, 本文提出的基于穷举扫描的人脸检测算法速度会相应线性增加。最后, 为了提高检测算法的检测速度, 本文提出了优化调高瀑布型分类器阈值的方法来获取最优的检测时间代价, 但是这个方法建立在牺牲部分检测率代价基础上, 从而没有保证在人脸检测器的精度。显然, 在未来的工作中, 对特征描述、统计学习算法, 穷举扫描窗口的检测框架的依然需要进一步改进。

对于检测速度和检测性能, 从某种意义上, 在研究者们努力下两者之间的总体关系是同时螺旋上升的, 虽然一般来讲, 效率与性能在某种程度上说是本质上对立的、不可调和的矛盾关系, 正确率、鲁棒性等方面更好的算法依然会花费更多的时间及系统消耗, 但应该在对立的前提下尽可能的寻找统一, 也即力争找到效率与性能的最佳结合点, 使检测系统从整体上达到最优。无论如何, 这两者结合后的总体系统表现目前所能达到的高度还远远不够, 还需要更进一步的研究和探索。

第二, 本文提出的检测算法针对的是一般意义上的人脸, 但是, 在现实情况中, 极端情况下如何更好的进行人脸检测依然存在问题, 比如: (1) 极端的光照条件 (光线极暗或者极亮的环境中); (2) 非常模糊的人脸检测, (3) 大面积遮挡的人脸检测等。如何采用本文所提出的算法提高这些极端情况下人脸检测系统性能, 仍需进一步研究和探索。

在极端情况下, 可以利用更多的上下文信息, 将人脸检测融合于人脸所处的大背景中, 通过对相关环境的分析, 获得人脸的信息。目前的人脸检测算法大多只是简单地考虑人脸模式与非人脸模式之间的区分, 而在人类的视觉中, 我们可以整体上把握图像的

结构和内容，利用不同物体之间在三维空间中的相互位置关系来验证我们的人脸检测结果。另外，人类还可以通过人类拥有的其它感知通道，比如听觉、嗅觉来验证人脸的存在。这就给我们提供一个思路，人脸模式并不是单独的存在一个场景之中，我们可以通过与之伴随出现的颈部、肩部、躯干、四肢的出现来进一步验证人脸。所以，我们可以将检测的对象从单一的人脸检测扩展到人体其它部分的检测，或者是场景的检测。一方面，利用人脸与人体其它部分之间的相互位置之间的关系进行人脸的重新估计，交叉验证，另一方面，利用人体整体所处的背景来重新评判进行人脸出现的可能性进行进一步验证。以进一步降低人脸检测系统的误检率，并提高人脸检测的整体性能。

无论如何，人脸检测发展至今，经历了从单纯的学术研究问题演变为成熟的供应应用的问题，人脸检测技术近年来越来越频繁地出现在以视频监控、身份验证、智能交互为代表的自动化信息处理系统之中。相信，以此为契机，作为计算机视觉领域的一个比较成熟的课题，在未来还会取得更大的发展和更加广泛的应用。

## 参考文献

- [YKA02] M. H. Yang, D. Kriegman, and N. Ahuja. Detecting faces in images: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):34-58, 2002.
- [YA98] M. H. Yang, N. Ahuja. Detecting Human Faces in Color Images. *Proceedings of International Conference on Image Processing*, Volume 1, 127-130, 1998.
- [HAJ02] R. L. Hsu, M. Abdel-Mottaleb, A.K. Jain. Face detection in color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):696-706, 2002.
- [MWSC99] J. Miao, B. Yin, K. Wang, L. Shen, X. Chen. A Hierarchical Multiscale and Multiangle System for Human Face Detection in a Complex Background Using Gravity-Center Template, *Pattern Recognition*, 32(7): 1237-1248, 1999.
- [Hje01] E. Hjelmas. Face Detection: A Survey. *Computer Vision and Image Understanding*, 83(3): 236-274, 2001.
- [LJ04] S. Li, A. K. Jain. *Handbook of Face Recognition*. Springer Verlag. New York. 2004.
- [YH94] G. Yang and T. S. Huang. Human Face Detection in Complex Background. *Pattern Recognition*, 27(1):53-63, 1994.
- [Kan73] T. Kanade, *Picture Processing by Computer Complex and Recognition of Human Faces*. PhD thesis, Kyoto University, 1973.
- [KP97] C. Kotropoulos and I. Pitas. Rule-Based Face Detection in Frontal Views. *International Conference on Acoustics, Speech and Signal Processing*, vol. 4, pp. 2537-2540, 1997.
- [KKSP00] H. Kim, W. Kang, J. Shin, S. Park. Face Detection Using Template Matching and Ellipse Fitting, *IEICE Transactions on Information and System*, E38-D(11): 2008-2011, 2000.
- [WT00] J. Wang, T. Tan. A New Face Detection Method based on Shape Information, *Pattern Recognition Letters*, 21(6-7): 463-471, 2000.
- [FE03] B. Fröba and A. Ernst. Fast Frontal-View Face Detection Using a Multi-Path Decision Tree, In *Proceeding of Audio- and Video-based Biometric Person Authentication*, 921-928, 2003.
- [OG97] E. Osuna, R. Freund and F. Girosi. Training support vector machines: An application to face detection, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 130–136, 1997.

- [SP98] K. K. Sung, and T. Poggio. Example-Based Learning for View-Based Human Face Detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1): 39-51, 1998.
- [Sun96] K. K. Sung, Learning and Example Selection for Object and Pattern Detection, PhD thesis, MIT AI, Jan. 1996.
- [RBK96] H. A. Rowley, S. Baluja, and T. Kanade. Neural Network-Based Face Detection, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 203-208, 1996.
- [RBK98a] H. A. Rowley, S. Baluja, and T. Kanade. Rotation Invariant Neural Network-Based Face Detection, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*: 38-44, 1998.
- [RBK98b] 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.
- [Row99] H. A. Rowley. Neural Network-Based Face Detection. [PhD thesis]. Pittsburgh, PA, USA: School of Computer Science, Computer Science Department, Carnegie Mellon University, 1999.
- [FBVC02] R. Féraud, O. Bernier, J. Viallet, and M. Collobert. A Fast and Accurate Face Detector Based on Neural Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(1): 42-53, 2002.
- [SK98] H. Schneiderman and T. Kanade. Probabilistic modeling of local appearance and spatial relationships for object recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 45-51, 1998.
- [SK00] H. Schneiderman, and T. Kanade. A Statistical Method for 3D Object Detection Applied to Faces, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 746-751, 2000.
- [Sch00] H. Schneiderman. A statistical approach to 3D object detection applied to faces and cars. [PhD thesis]. Pittsburgh, PA, USA: School of Computer Science, Computer Science Department, Carnegie Mellon University, 2000.
- [Sch04] H. Schneiderman. Feature-centric evaluation for efficient cascaded object detection, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004: 29-36.
- [YRA00] M. H. Yang, D. Roth and N. Ahuja. A SNoW-Based Face Detector, *Advances in Neural Information Processing Systems*, MIT Press, 12: 855-861, 2000.

- [Liu03] C. Liu. A Bayesian Discriminating Features Method for Face Detection, IEEE Transactions On Pattern Analysis and Machine Intelligence, 25(6): 725-740, 2003.
- [HSPP03] B. Heisele, T. Serre, S. Prentice, T. Poggio. Hierarchical Classification and Feature Reduction for Fast Face Detection with Support Vector Machines, Pattern Recognition, 36(9): 2007-2017, 2003.
- [VJ01a] P. Viola and M. Jones. Rapid Object Detection Using a Boosted Cascade of Simple Features, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 511-518, 2001.
- [VJ01b] P. Viola and M. Jones. Robust real-time object detection. Proceeding of ICCV Workshop Statistical and Computational Theories of Vision, 2001.
- [VJ04] P. Viola and M. Jones. Robust real-time face detection. International Journal of Computer Vision, 57(2) 137-154, 2004.
- [BLFM03] M. S. Bartlett, G. Littlewort, I. Fasel, and J. R. Movellan. Real Time Face Detection and Facial Expression Recognition. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003.
- [LM02] R. Lienhart, J. Maydt. An Extended Set of Haar-like Features for Rapid Object Detection, IEEE International Conference on Image Processing, 1: 900-903, 2002.
- [WAHL04] B. Wu, H. Z. Ai, C. Huang, S. H. Lao: Fast Rotation Invariant Multi-View Face Detection Based on Real AdaBoost. IEEE International Conference on Automatic Face and Gesture Recognition: 79-84, 2004.
- [Li02] S. Z. Li, L. Zhu, Z. Q. Zhang, A. Blake, H. J. Zhang, and H. Y. Shum. Statistical learning of multi-view face detection. Proceedings of the 7-th European Conference on Computer Vision. Copenhagen, Denmark: Springer, 67-81, 2002.
- [LZSZ02] S. Z. Li, Z. Q. Zhang, H. Y. Shum and H. J. Zhang. FloatBoost learning for classification. Proceedings of the 16-th Annual Conference on Neural Information Processing Systems. Vancouver, Canada: MIT Press, 993-1000, 2002.
- [LZZBZ02] S. Z. Li, L. Zhu, and Z. Q. Zhang, A. Blake and H. J. Zhang. Learning to detect multi-view faces in real-time. Proceedings of the 2-nd International Conference on Development and Learning. Cambridge, MA, USA: IEEE Computer Society, 172-177, 2002.
- [LZ04] S. Z. Li and Z. Zhang. FloatBoost learning and statistical face detection, IEEE Transactions On Pattern Analysis and Machine Intelligence, 26: 1112 – 1123, 2004.
- [HAL06] C. Huang, H. Z. Ai, Y. Li, S. H. Lao: Learning Sparse Features in Granular Space for

- Multi-View Face Detection. IEEE International Conference on Automatic Face and Gesture Recognition, 401-407, 2006.
- [MCWW08] K. Masada, Q. Chen, H. Y. Wu and T. Wada. GA Based Feature Generation for Training Cascade Object Detector. 19th International Conference on Pattern Recognition, 2008.
- [LS03] C. Liu, H. Y. Shum. Kullback-Leibler Boosting, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 587–594, 2003.
- [MKH05] T.Mita, T. Kaneko, and O. Hori. Joint haar-like features for face detection. In IEEE International Conference on Computer Vision, 2005.
- [BSR04] S Baluja, M. Sahami and HA Rowley. Efficient face orientation discrimination. IEEE International Conference on Image Processing, 2004.
- [AS05] Y. Abramson and B. Steux. Real-time Object Detection. International Workshop on Automatic Learning and Real-Time, 2005.
- [OPH96] T. Ojala, M. Pietikainen, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29(1):51–59, 1996.
- [HPA04] A. Hadid, M. Pietikainen and T. Ahonen. A Discriminative Feature Space for Detecting and Recognizing Faces. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004.
- [ZZ04] H. M. Zhang, D. B. Zhao. Spatial Histogram Features for Face Detection in Color Images. *PCM (1)* 377-384, 2004.
- [ZGCZ06] H. M. Zhang, W. Gao, X. L. Chen, Debin Zhao. Object detection using spatial histogram features. *Image Vision Comput.* 24(4): 327-341, 2006.
- [SS99] R. E. Schapire, Y. Singer. Improved Boosting Algorithms Using Confidence-rated Predictions. *Machine Learning*, 37(3):297-336, 1999.
- [XZST07] R. Xiao, H. Y. Zhu, H. Sun, X. O. Tang: Dynamic Cascades for Face Detection. IEEE International Conference on Computer Vision, 2007.
- [LS06] D.D. Le and S. Satoh. Ent-Boost: Boosting Using Entropy Measure for Robust Object Detection. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006.
- [HLT06] X. Hou, C.-L. Liu, T. Tan. Learning Boosted Asymmetric Classifiers for Object Detection, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006.

- 
- [Vio01] P. Viola. Fast and Robust Classification using Asymmetric AdaBoost and a Detector Cascade, *Advances in Neural Information Processing Systems* 1311-1318, 2001.
- [WMR05] J. X. Wu, M. D. Mullin, and J. M. Rehg. Linear asymmetric classifier for cascade detectors. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 988–995, New York, NY, USA, 2005.
- [WBMR08] J. X. Wu, S. C. Brubaker, M. D. Mullin, James M. Rehg: Fast Asymmetric Learning for Cascade Face Detection. *IEEE Transactions Pattern Anal. Mach. Intell.* 30(3): 369-382, 2008.
- [HLT06] X. W. Hou, C. L. Liu, and T. N. Tan, Learning Boosted Asymmetric Classifiers for Object Detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006.
- [PC07a] M. T. Pham and T. J. Cham. Online Learning Asymmetric Boosted Classifiers for Object Detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.
- [PHC08] M. T. Pham, V-D. D. Hoang, T. J. Cham. Detection with multi-exit asymmetric Boosting. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008.
- [XZZ03] R. Xiao, L. Zhu, and H.-J. Zhang, Boosting chain learning for object detection. In *IEEE International Conference on Computer Vision*, pages 709–715, 2003.
- [XLZ04] R. Xiao, M. J. Li and H. J. Zhang. Robust Multipose Face Detection in Images, *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1): 31-41, 2004.
- [SM05] J. Sochman, J. Matas. WaldBoost—Learning for Time Constrained Sequential Detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.
- [HAWL04] C. Huang, H. Z. Ai, B. Wu, S. H. Lao: Boosting Nested Cascade Detector for Multi-View Face Detection. *The IEEE International Conference on Pattern Recognition*, 2004.
- [SRB04] J. Sun, J. M. Rehg, A. F. Bobick. Automatic Cascade Training with Perturbation Bias. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004.
- [Luo05] H. T. Luo. Optimization Design of Cascaded Classifiers, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 480-485, 2005.

- [BMR06] S. C. Brubaker, M. D. Mullin and J. M. Rehg. Towards Optimal Training of Cascaded Detectors, Proceeding of European Conference on Computer Vision, 325-337, 2006.
- [BWS08] S. C. Brubaker, J. X. Wu, J. Sun, M. D. Mullin, J. M. Rehg. On the Design of Cascades of Boosted Ensembles for Face Detection. International Journal of Computer Vision, 77(1-3): 65-86, 2008.
- [DB07] M. Dondur, J. B. Bi. An AND-OR Framework for Learning a Cascade Classifier. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Minneapolis, Minnesota, USA, 18-23 June 2007.
- [MQ03] Y. Ma and X. Q. Qing. Real-time rotation invariant face detection based on cost-sensitive AdaBoost. IEEE International Conference on Image Processing, 921-924, 2003.
- [JP03] M. Jones and P. Viola. Fast Multi-View Face Detection. Mitsubishi Electric Research Laboratories, Technical Report:MERL-2003-96, July 2003.
- [LL05] Y. Y. Lin and T. L. Liu. Robust Face Detection with Multi-Class Boosting. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 680-687, 2005.
- [ZGC06] H. M. Zhang, W. Gao, X. L. Chen, S. G. Shan, D. B. Zhao: Robust Multi-view Face Detection Using Error Correcting Output Codes. Proceeding of European Conference on Computer Vision, (4), 1-12, 2006.
- [HWAL04] C. Huang, B. Wu, H. Z. Ai, S. H. Lao: Omni-directional face detection based on real adaBoost. IEEE International Conference on Image Processing, 593-596, 2004.
- [HALL05] C. Huang, H. Z. Ai, Y. Li, S. H. Lao: Vector Boosting for Rotation Invariant Multi-View Face Detection. IEEE International Conference on Computer Vision, 2005.
- [HALL07] C. Huang, H. Z. Ai, Y. Li, S. H. High-Performance Rotation Invariant Multiview Face Detection. IEEE Transactions Pattern and Machine Intellegence, 29(4): 671-686, 2007.
- [WRM03] J. X. Wu, J. M. Rehg and M. D. Mullin. Learning a rare event detection cascade by direct feature selection. Advances in Neural Information Processing Systems, 2003.
- [PC07b] M. T. Pham and T. J. Cham. Fast training and selection of Haar features using statistics in Boosting-based face detection. IEEE International Conference on Computer Vision, 2007.

- 
- [FL06] Z. G. Fan and B. L. Lu. Fast Learning for Statistical Face Detection. *Face Analysis and Processing*. Springer Berlin / Heidelberg. Volume 4233:187-196, 2006.
- [BB05] L. Bourdev, J. Brandt. Robust Object Detection via Soft Cascade. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.
- [CY05] X. R. Chen and A. L. Yuille. An Time-Efficient Cascade for Real-Time Object Detection: with Applications for the Visually Impaired. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Volume 03, 2005.
- [HHD06] S. Hayashi, O. Hasegawa. A Detection Technique for Degraded Face Images, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1506-1512, 2006.
- [Sah03] H. Sahbi. Coarse-to-Fine Support Vector Machines for Hierarchical Face Detection. PhD Thesis, University of Versailles, April 2003.
- [Lyu05] S.W. Lyu. Infomax Boosting. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.
- [GD04] C. Garcia and M. Delakis. Convolutional Face Finder: A Neural Architecture for Fast and Robust Face Detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11): 408-1423, 2004.
- [BH03] S.M. Bileschi, B. Heisele. Advances in component based face detection. *IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, 2003. 149- 156.
- [HSP07] B. Heisele, T. Serre, T. Poggio. A Component-based Framework for Face Detection and Identification. *International Journal of Computer Vision* 74(2), 167–181, 2007.
- [BH02] S. M. Bileschi, B. Heisele. Advances in component-based face detection. In *Proceedings of Pattern Recognition with Support Vector Machines*, First International Workshop, SVM, Niagara Falls, pp. 135–143, 2002.
- [HSMP03] B. Heisele, T. Serre, S. Mukherjee, and T. Poggio.. Hierarchical classification and feature reduction for fast face detection with support vector machines. *Pattern Recognition*, 36(9):2007–2017, 2003.
- [LLK03] R. Lienhart, L. Liang, and A. Kuranov, A detector tree of Boosted classifier for real time object detection and tracking. *IEEE International Conference on Multimedia & Expo*, 2003.

- [LKP03] R. Lienhart, A. Kuranov, and V. Pisarevsky. Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection. In DAGM'03, 25th Pattern Recognition Symposium, pp. 297–304, 2003.
- [KS03] H. Kruppa, B. Schiele. Using Local Context To Improve Face Detection. British Machine Vision Conference, 2003.
- [CWYS05] J. Chen, R. Wang, S. Yan, S. Shan, X. Chen, and W. Gao. How to Train a Classifier Based on the Huge Face Database? IEEE International Workshop on AMFG, LNCS 3723, 84 – 95, 2005.
- [CWYS07] J. Chen, R. Wang, S. Yan, S. Shan, X. Chen, and W. Gao. Face Detection Based on the Example Re-sampling by Manifold. IEEE Transactions on System Man, and Cybernetics (Part A). 37(6):1017-1028, 2007.
- [WCSC06] R. Wang, J. Chen, S. Shan, X. Chen, and W. Gao. Enhancing Training Set for Face Detection. The IEEE 7th International Conference on Pattern Recognition, 477-480, 2006.
- [CCG04] J. Chen, X.. L. Chen, W. Gao. Expand Training Set for Face Detection by GA Re-sampling. The IEEE 6th International Conference on Automatic Face and Gesture Recognition, pp.73-79, 2004.
- [AAP05] A. Anelia, Y. Abu-Mostafa and P. Perona. Pruning Training Sets for Learning of Object Categories. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005.
- [Bha86] B. Bhanu. Automatic Target Recognition: a State of the Art Survey. IEEE Transactions on Aerospace and Electronic Systems, 22, pp 364-379. 1986
- [PMRR97] P. J. Phillips, H. Moon, S. A. Rizvi, P. J. Rauss . The FERET Evaluation Methodology for Face-Recognition Algorithms. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1997.
- [JKF01] O. Jesorsky, K. Kirchberg, R. Frischholz. Robust Face Detection Using the Hausdorff Distance. Audio and Video based Person Authentication, pages 90-95. Springer, 2001.
- [FS99] Y. Freund, and R. E. Schapire. A short introduction to Boosting. Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, 1999.
- [Val84] L. G. Valiant. A Theory of the Learnable. In: Communications of the ACM, 27(11):1134-1142, 1984.
- [KV94] M. J. Kearns and U. V. Vazirani. An Introduction to Computational Learning Theory.

- MIT Press, 1994.
- [Sch90] R. E. Schapire. The Strength of Weak Learnability. *Machine Learning*, 5(2):197-227, 1990.
- [FS95] Y. Freund, R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to Boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.
- [Fre95] Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256-285, 1995.
- [SSS00] R. E. Schapire, Y. Singer, and A. Singhal. Boosting and Rocchio applied to text filtering. *Proceedings of the Ninth International Conference on Information and Knowledge Management*, 2000.
- [ILS00] R. D. Iyer, D. D. Lewis, R. E. Schapire, Y. Singer and A. Singhal. Boosting for document routing. *Proceedings of the Ninth International Conference on Information and Knowledge Management*, 2000.
- [FISS98] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient Boosting algorithm for combining preferences. *Proceedings of the Fifteenth International Conference on Machine Learning*, 1998.
- [ASS99] S. Abney, R. E. Schapire, and Y. Singer. Boosting applied to tagging and PP attachment. *Proceedings of the Joint SIFDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999.
- [Col00] M. Collins. Discriminative reranking for natural language parsing. *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000.
- [EMR00] G. Escudero, L. Marquez, and G. Rigau. Boosting applied to word sense disambiguation. *Proceedings of the 12th European Conference on Machine Learning*. 129-141, 2000.
- [HSO00] M. Haruno, S. Shirai and Y. Ooyama. Using decision trees to construct a practical parser. *Machine Learning*, 34:131-149, 1999.
- [MLR01] P. J. Moreno, B. Logan, and B. Raj. A Boosting approach for confidence scoring. *Proceedings of the 7<sup>th</sup> European Conference on Speech Communication and Technology*, 2001.
- [WRR01] M. A. Walker, O. Rambow, and M. Rogati. SPoT: A trainable sentence planner. *Proceedings of the 2<sup>nd</sup> Annual Meeting of the North American Chapter of the Association for Computational Linguistics*, 2001.

- [TV00] K. Tieu and P. Viola. Boosting image retrieval. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2000.
- [MFLS01] S. Merler, C. Furlanello, B. Larcher, and A. Sboner. Tuning costsensitive Boosting and its application to melanoma diagnosis. In Multiple Classifier Systems: Proceedings of the 2<sup>nd</sup> International Workshop, pages 32-42, 2001.
- [MWGJ00] M. C. Mozer, R. Wolniewicz, D. B. Grimes, E. Johnson, and H. Kaushansky. Predicting subscriber dissatisfaction and improving retention in the wireless telecommunications industry. IEEE Transactions on Neural Networks, 11:690-696, 2000.
- [ORM00] T. Onoda, G. Ratsch, and K. R. Muller. Applying support vector machines and Boosting to a non-intrusive monitoring system for household electric appliances with inverters. In Proceedings of the Second ICSC Symposium on Neural Computation, 2000.
- [FK02] B. Froba and C. Kublbeck. Robust face detection at video frame rate based on edge orientation features, Fifth IEEE International Conference on Automatic Face and Gesture Recognition, 327-332, 2002.
- [陈 03] 陈锻生,刘政凯.彩色图像人脸高光区域的自动检测与校正方法, 软件报学,14(11): 1900-1906,2003.
- [梁 02] 梁路宏,艾海舟,徐光祐,张钺.人脸检测研究综述.计算机学报, 25(5):449-458, 2002.
- [刘 97] 刘明宝.人脸检测与跟踪的研究, 哈尔滨工业大学工学博士论文, 1997.
- [武 05] 武勃 黄畅 艾海舟 劳世竑 . 基于连续 AdaBoost 算法的多视角人脸检测. 计算机研究与发展, 42 (09): 1612-1621, 2005.
- [王 04] 王海川,张立明。一种新的 AdaBoost 快速训练算法。复旦学报(自然科学版), 43(1):28-33, 2004.
- [艾 03] 艾海舟,肖习攀,徐光祐.人脸检测与检索,计算机学报. 26(07): 874-881, 2003.
- [方 04] 方昱春,王蕴红,谭铁牛.融合人脸轮廓和区域信息改进人脸检测,计算机学报,27(4): 482-491, 2004.

## 致 谢

从事学术研究是一条不归之路。我深深感觉到，直到博士论文的完成，我才刚刚走上这条路的起点。数年的博士生活，在师长、亲友的多方大力支持下，走得辛苦却也收获颇丰。值此论文即将完成之际，在此由衷感谢所有在本人攻读博士期间对本人关心、帮助、支持我的老师、同学和家人们。正是老师们广博的科学实践经验、深厚的理论知识、严谨的治学态度和不倦的教导，引导我走向了探求科学真理的正确方向。正是同学之间的相互鼓励和无私帮助的团队生活，是我在前进的道路上得到了志同道合者的支持和鼓励。正是家人作为最有力的后盾，使我能够完成论文的撰写。

衷心感谢我的导师高文教授，高老师不仅教会了我如何做学问，更关键的是教会了我如何做人。在学业上，高老师鼓励我积极大胆创新，探索研究领域中的“金问题”，寻找前人没有涉足的领域。在生活中，高老师给我们创造了一个宽松而又积极的生活环境。

衷心感谢陈熙霖研究员对我的悉心培养，陈老师广播的学识、敏锐的思维、无私的科学精神和乐观的生活态度，是我一直钦佩和学习的榜样。每当我在工作中遇到疑难问题时，总是能够得到导师及时认真的指导和富有建设性的建议。

衷心感谢赵德斌教授，从进入实验室，赵老师就对我谆谆教导，告诉我如何做研究，如何在科学的道路上进行探索，如何发现问题并进入自己的课题。

衷心感谢山世光副研究员。山师兄在我进入实验室时还是我的师兄。山师兄是我平时工作中最贴心的兄长，科研中遇到的任何问题，我首先进行讨论的都是山师兄。每一次有了令人鼓舞的想法第一个进行讨论的也总是山师兄。山师兄和蔼可亲的性格，严谨的学风，都给我留下了深刻的印象，我所发表的所有论文都留下了山师兄修改的笔迹。

衷心的感谢苗军副研究员。苗师兄在我进入实验室时也是我的师兄。苗师兄对复杂问题的简单化非常有独到的见解，每每对我进行指导。另外，苗师兄还是一个厚道、朴实的大哥，生活中非常平易近人。

感谢贺思敏老师、王伟强老师、黄庆明老师、黄铁军老师、卿来云老师、梁路宏老师、常虹老师、宋裕庆老师在我博士其间对我的指导。

感谢同组的师兄师姐师弟师妹，我们曾经共同战斗。感谢张宏明、王建宇、张文超、张宝昌、曹波、陈杰、柴秀娟、刘心、杨鹏、李月敏、赵三强、刘珩、邓刚、马丙鹏、刘亚洲、王瑞平、谢术富、李博、牛志恒、苏煜、杨非、唐杰、索津丽、李祎、崔新毅、方芳、洪晓鹏、李安南、房佑寒、刘超然、韩琥、郑伟、王琳、阚美娜、路亮、马志国、王丹、翟德明、蒋长征。大家共同营造了团结友好、相互帮助、轻松愉快的良好环境。

向我的父母、姐姐、哥哥以及我的亲人致以最诚挚的谢意。他们多年来对我始终如一的爱和鼓励是我不断进取的动力。



## 作者简介

姓名：闫胜业      性别：男      出生日期：1978.6.23      籍贯：河南新乡

2003.9 – 2009.7      中科院计算所计算机应用技术专业博士研究生  
2000.9 -- 2003.7      北京工业大学计算机学院硕士研究生  
1996.9 -- 2000.7      南昌航空工业学院计算机应用专业本科生

### 【攻读博士学位期间发表的论文】

- [1]Shengye Yan, Shiguang Shan, Xilin Chen, Wen Gao. Locally Assembled Binary (LAB) Feature for Fast and Accurate Face Detection. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2008. (CVPR2008).
- [2]Shengye Yan, Shiguang Shan, Xilin Chen, Wen Gao, Jie Chen. Matrix-Structural Learning (MSL) of Cascaded Classifier from Enormous Training Set. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2007. (CVPR2007).
- [3]Jie Chen, Ruiping Wang, Shengye Yan, Shiguang Shan, Xilin Chen, Wen Gao. Enhancing Human Face Detection by Resampling Examples Through Manifolds. IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS, VOL. 37, NO. 6, NOVEMBER 2007.
- [4]Jie Chen, Shiguang Shan, Shengye Yan, Xilin Chen, Wen Gao. Modification of the AdaBoost-based Detector for Partially Occluded Faces. The IEEE 7th international Conference on Pattern Recognition, 516 – 519, 2006.
- [5]Jie Chen, Shiguang Shan, Peng Yang, Shengye Yan, Xilin Chen, Wen Gao. Novel Face Detection Method Based on Gabor Feature. The 5th Chinese Conference on Biometric Recognition, pp. 90 – 99, 2004. Springer-Verlag.
- [6]Ruiping Wang, Jie Chen, Shengye Yan, Shiguang Shan, Xilin Chen, Wen Gao. Face detection based on the manifold. Audio- and Video-based Biometric Person Authentication. (AVBPA 2005), LNCS 3546, pp. 208-218, Springer-Verlag.
- [7]Zhiheng Niu, Shiguang Shan, Shengye Yan, Xilin Chen, Wen Gao. 2D Cascaded AdaBoost for Eye Localization. Proceeding of International Conference on Pattern Recognition, Vol.2, pp1216-1219, 2006
- [8]Heng Liu, Shengye Yan, Xilin Chen and Wen Gao, Rotated Face Detection in Color Images Using Radial Template (RT), Proceedings of International Conference on Acoustics, Speech and Signal Processing, Vol. III, pp. 213-216, 2003.

### 【攻读博士学位期间申请的专利】

1. 闫胜业, 山世光, 陈熙霖, 高文。一种基于预分类器的物体检测方法。审查中。

### 【攻读博士学位期间申请的软件著作权】

1. 陈杰, 闫胜业, 苗军, 陈熙霖, 高文。FindU 人脸检测开发包法。受理号: 200713682。

### 【攻读博士学位期间参加的科研项目】

- [1]国家自然科学基金委项目 No. 60473043
- [2]国家自然科学基金委项目 No.60673091
- [3]国家自然科学基金委项目 No. 60332010
- [4]国家高技术研究开发计划项目 No.2006AA01Z122
- [5]中科院百人计划 No. NCET-04-0320
- [6]北京市自然科学基金 No. 40611001
- [7]北京市自然科学基金 No. 4072023
- [8] 2004 年, 建设银行某省银行 ATM 机和各营业网点, 智能视频监控系统
- [9] 2008 年, 某公司智能人脸检测与跟踪系统

### 【攻读博士学位期间的获奖情况】

- [1] 2006-2007 学年中科院研究生院“三好学生”
- [2] 2007-2008 学年中科院研究生院“三好学生”
- [3] 2007 年度计算所所长奖学金优秀奖
- [4] 2007 年度计算所优秀论文二等奖
- [5] 2007 年度 JDL 实验室优秀论文奖
- [6] 2008 年度 JDL 实验室优秀论文奖
- [7] 2008 年度计算所联想奖学金博士生奖