

# AdaBoost在人脸检测上的应用

山世光

中科院计算所

山世光, [sgshan@ict.ac.cn](mailto:sgshan@ict.ac.cn)



中国科学院计算技术研究所  
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

# Demo

山世光, [sgshan@ict.ac.cn](mailto:sgshan@ict.ac.cn)



中国科学院计算技术研究所  
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

# 目标





# 人脸检测方法

- 基于规则/知识方法
  - 人脸模式的变化满足一定的规律，所以可以归纳描述人脸特征的规则，如灰度分布、比例关系、纹理信息等
- 基于模板的方法
  - 固定模板法，可变形模板法
- 基于不变特征的方法，如彩色信息
  - 人脸的肤色在彩色空间中的分布相对比较集中，所以可以用来检测和跟踪人脸。
- 基于表观分类器学习的方法
  - 将人脸检测视为区分“非人脸样本”与“人脸样本”的PR问题，通过对人脸样本集和非人脸样本集的学习产生分类器

# 基于肤色特征的检测

- 在很多人脸检测和手的跟踪应用中，人的肤色信息已被证明是一种非常有效的特征
- 有很多颜色空间可用来表征肤色，包括RGB, normalized RGB, HSV (or HSI), YIQ, YES, CIE XYZ, 和 CIE LUV 等
- 有许多建立肤色模型的方法
  - 最简单的方法是用Cr, Cb值定义肤色区域，选定阈值[Cr1,Cr2]和[Cb1,Cb2]
  - 一个像素点被归为肤色点，如果它的(Cr, Cb)值落入下面的范围： $Cr1 \leq Cr \leq Cr2$  and  $Cb1 \leq Cb \leq Cb2$

# 多种肤色模型

- J.L.Crowley and F. Berard, “Multi-model tracking of faces for video communications”
  - 将肤色区域的RGB颜色归一化，用其中的(r,g)值的颜色直方图 $h(r,g)$ 获取肤色变量的阈值
- M. H. Yang, N. Ahuja, “Detecting human faces in color images”
  - 认为人脸肤色区域的颜色值呈高斯分布，用高斯分布的均值和方差确定肤色变量的阈值
- T. S. Jebara and A. Pentland, “Parameterized structure from motion for 3D adaptive feedback tracking of faces”
  - 认为不同的种族和国家的人的肤色分布不同，在颜色直方图上形成多个聚类，可用高斯混合模型来表示

# 颜色空间

- RGB到“rg”空间

$$r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}$$

- RGB到YUV(YCrCb)空间，再转化到“FI”空间

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

$$F = \tan^{-1}(|V|/|U|)$$

$$I = (0.596, -0.274, -0.322) \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

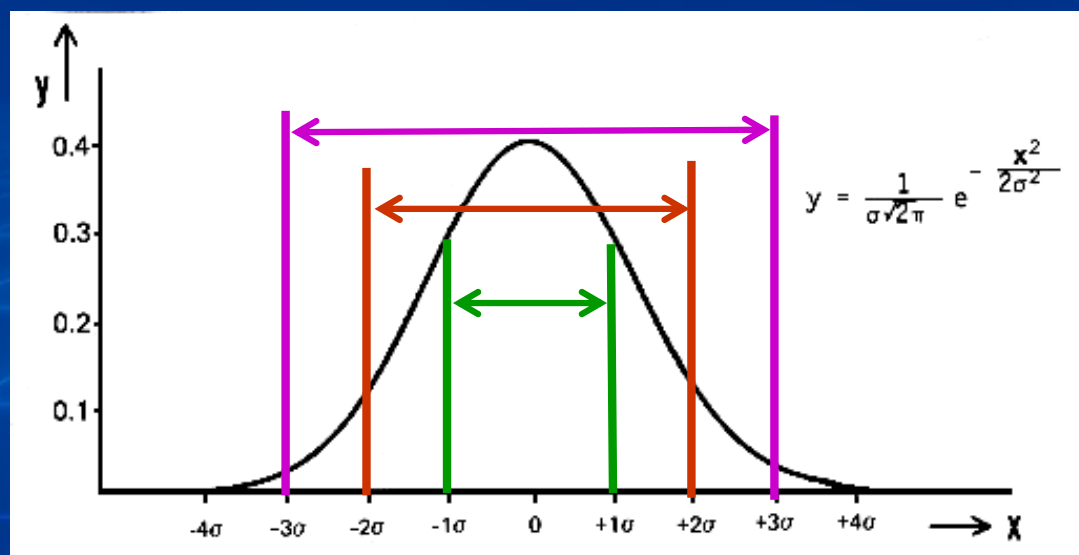


# 高斯肤色模型

- 一元正态分布肤色模型（以F颜色特征为例）

$$F \sim N(\mu, \sigma)$$

$$\mu = \frac{1}{N} \sum_{i=1}^N F_i, \quad \sigma^2 = \frac{1}{N-1} \sum_{i=1}^N (F_i - \mu)^2$$





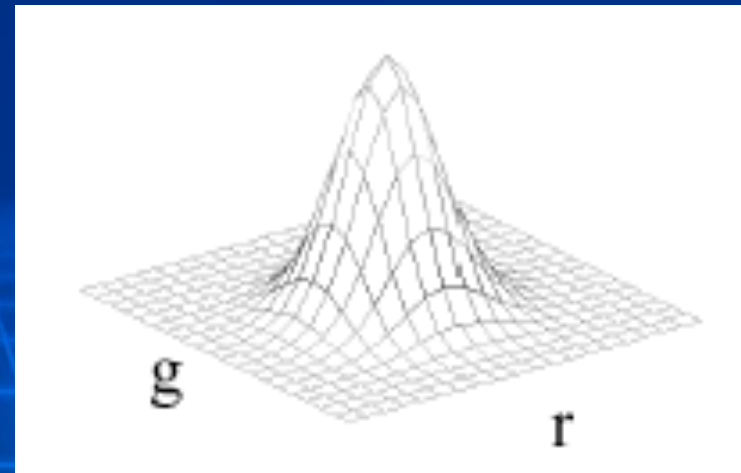
# 高斯肤色模型

- 二元正态分布肤色模型（以rg颜色特征为例）

$$N(\mu, \Sigma), \text{ where } \mu = (\bar{r}, \bar{g})$$

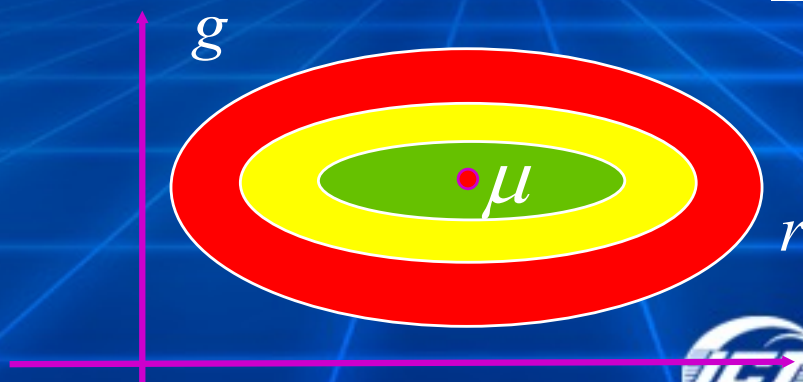
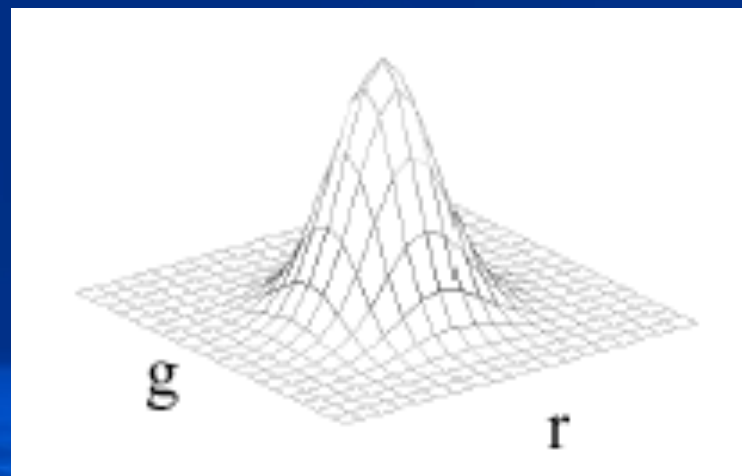
$$\bar{r} = \frac{1}{N} \sum_{i=1}^N r_i, \quad \bar{g} = \frac{1}{N} \sum_{i=1}^N g_i,$$

$$\Sigma = \begin{bmatrix} \sigma_{rr} & \sigma_{rg} \\ \sigma_{gr} & \sigma_{gg} \end{bmatrix}.$$



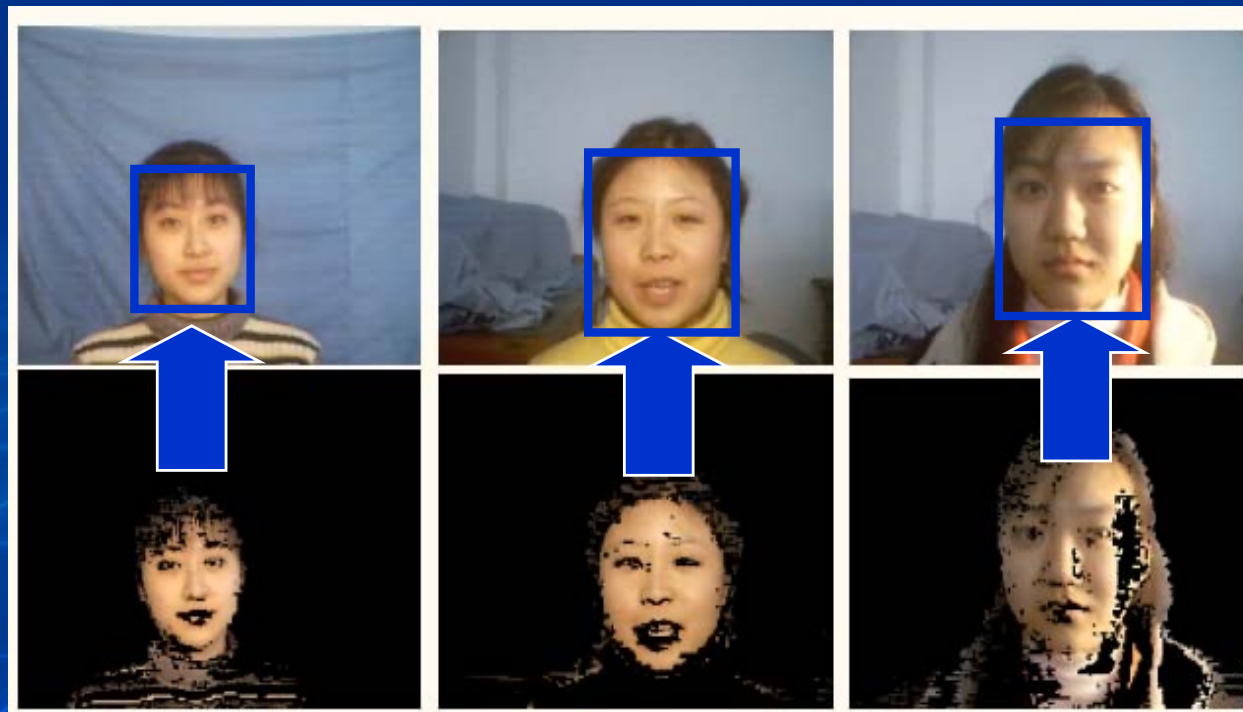
# 多人脸训练肤色模型

- 训练肤色模型
  - 手工标注部分人脸(肤色区域)人脸图像
  - 统计方法得到  $\mu, \Sigma$
  - 设置合适的阈值截断
- 测试阶段
  - 逐像素判断其是否在设定的肤色特征范围内



# 提取肤色区域

- 对检测到的肤色区域进行分析
  - 接近椭圆形
  - 有部分非肤色区域(五官、头发)



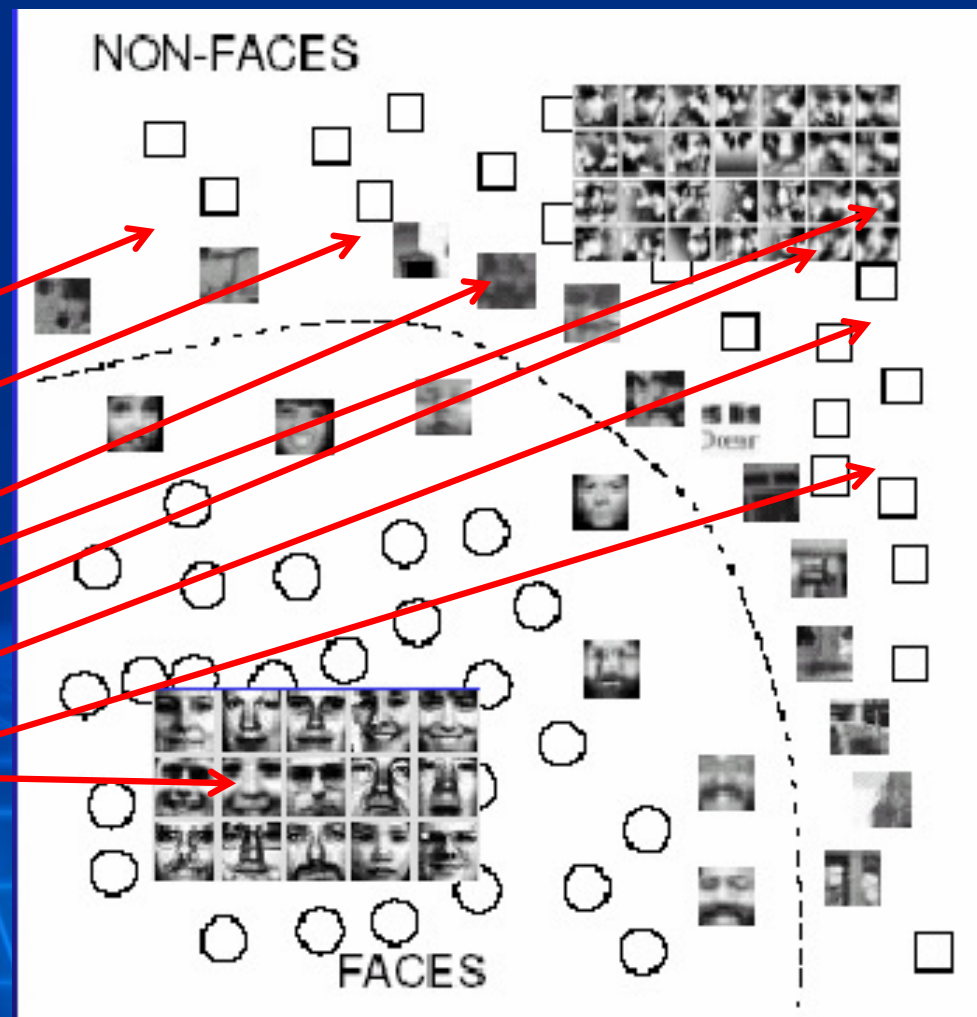
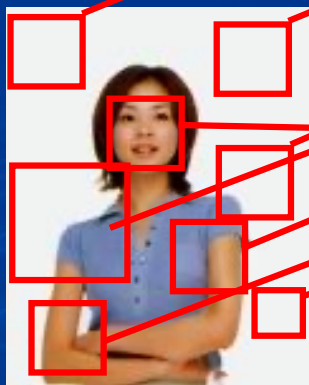
# 肤色模型的缺点

- 肤色模型难以适应各种环境光照变化
  - 对于背景和前景的光照变化，肤色通常不稳定
  - 单纯的肤色信息对于人脸检测通常是不充分的
- 如何适应在不同光照下的人脸跟踪
  - 如果环境光照有变化，原有的肤色模型可能不再适用，如何建立一个自适应的肤色模型需要进一步的研究



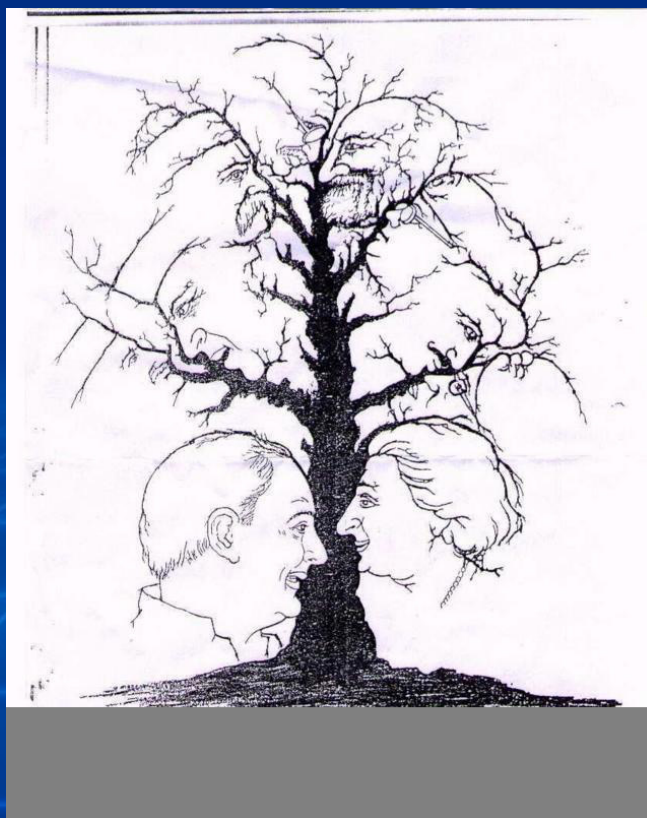
# 基于Appearance的方法

- ANN
- SVM
- Naïve Bayes Classifier
- AdaBoost



遍历所有可能的“矩形窗口”，判断每个小窗口是否人脸？

# 基于AdaBoost的快速人脸检测



山世光, sgshan@ict.ac.cn

中国科学院计算技术研究所  
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

# 算法简介

- 在2001年的ICCV上，当时在Compaq的研究员Paul Viola和Michael J. Jones发表的文章介绍了一个实时人脸检测系统
  - 图像大小为384x288时，其速度是平均每秒15帧
  - 第一个准实时的(准正面)人脸检测系统
  - 可以较容易的扩展到多姿态人脸检测
- 系统在技术上的三个贡献：
  - 1.用简单的Haar-like矩形特征作弱特征，可快速计算
  - 2.基于AdaBoost的分类器设计(弱特征→弱分类器)
  - 3.采用了Cascade(分级分类器)技术提高检测速度

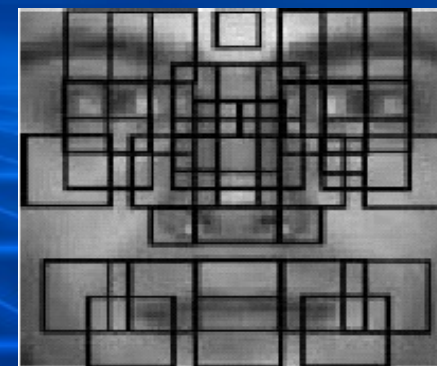
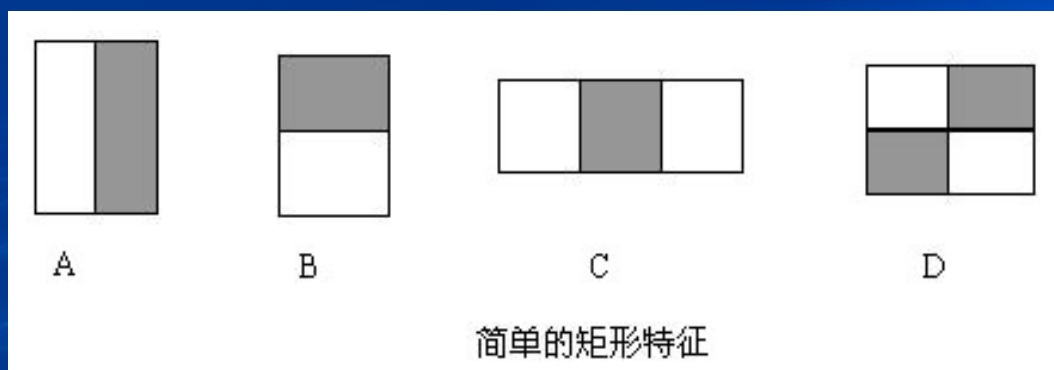
# 人脸的特征表示方法

- 矩形特征（Haar-like特征）

- 矩形特征

- 所有白色矩形中像素的亮度值的和，减去所有灰色矩形中点的亮度值的和，所得到的差即为Haar-like特征

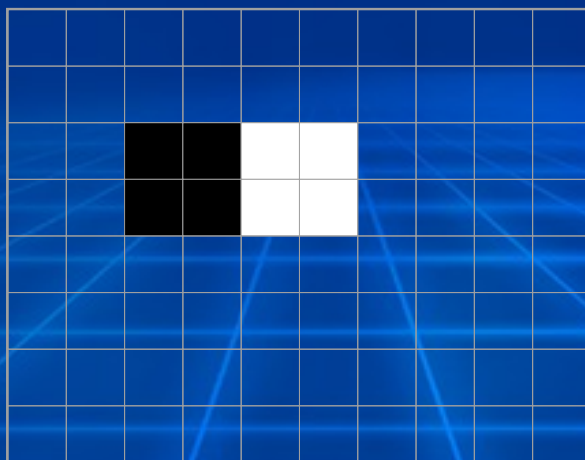
- 有4种类型的矩形特征





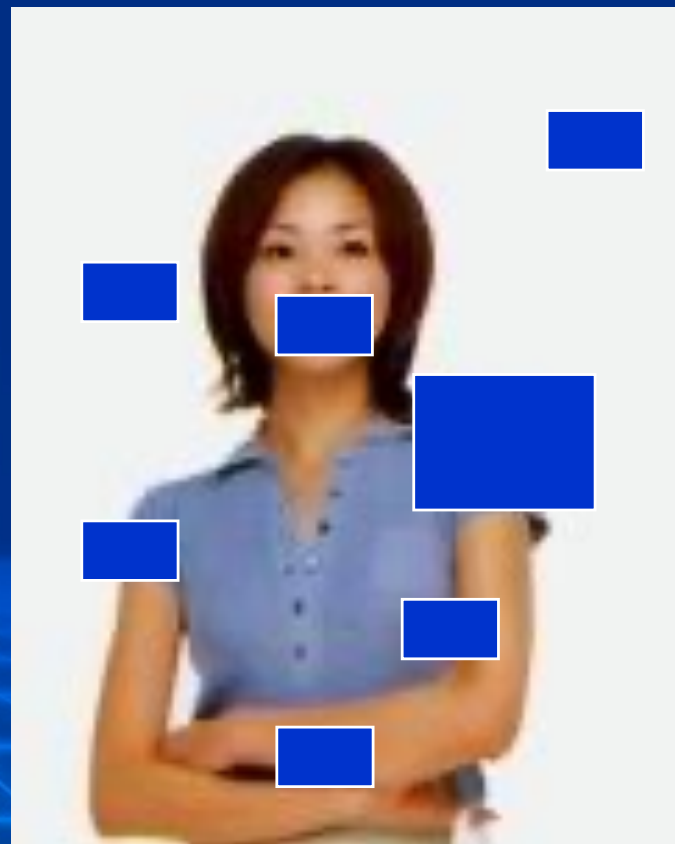
# Haar-like特征的表示

- 具体特征可以用一个五元组表示  
 $r(x,y,w,h,style)$ 
  - 比如:  $r(2,2,4,2,A)$ 表示下面的特征
    - 特征值即为白色四个像素与黑色四个像素的差值



# 问题

- 如何快速计算任意矩形内所有像素的亮度之和？
  - 请大家设计一个算法
  - 要求计算量尽可能的小

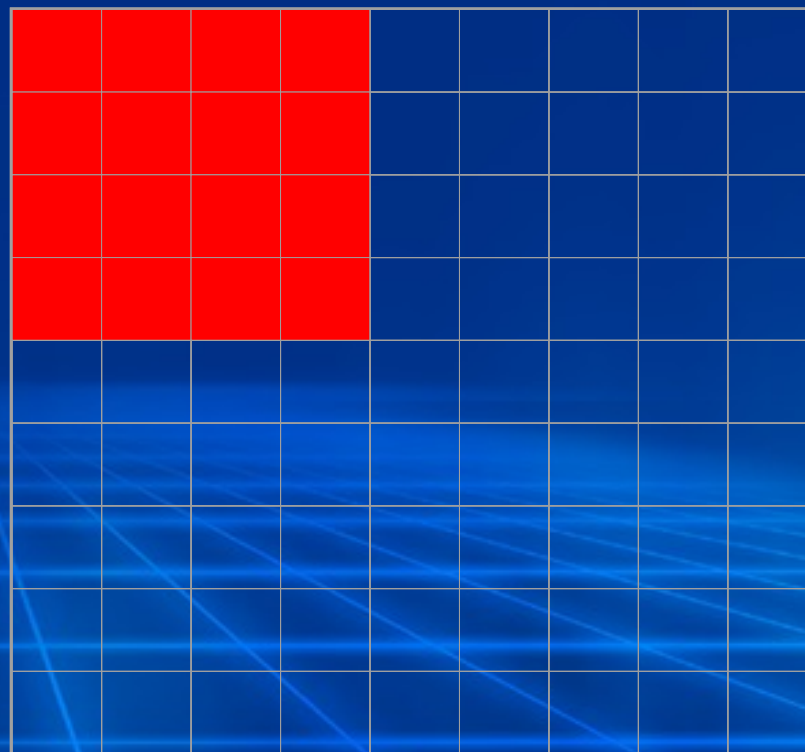


# 特征的快速计算

- 需要一幅辅助图像（积分图）如下图所示：

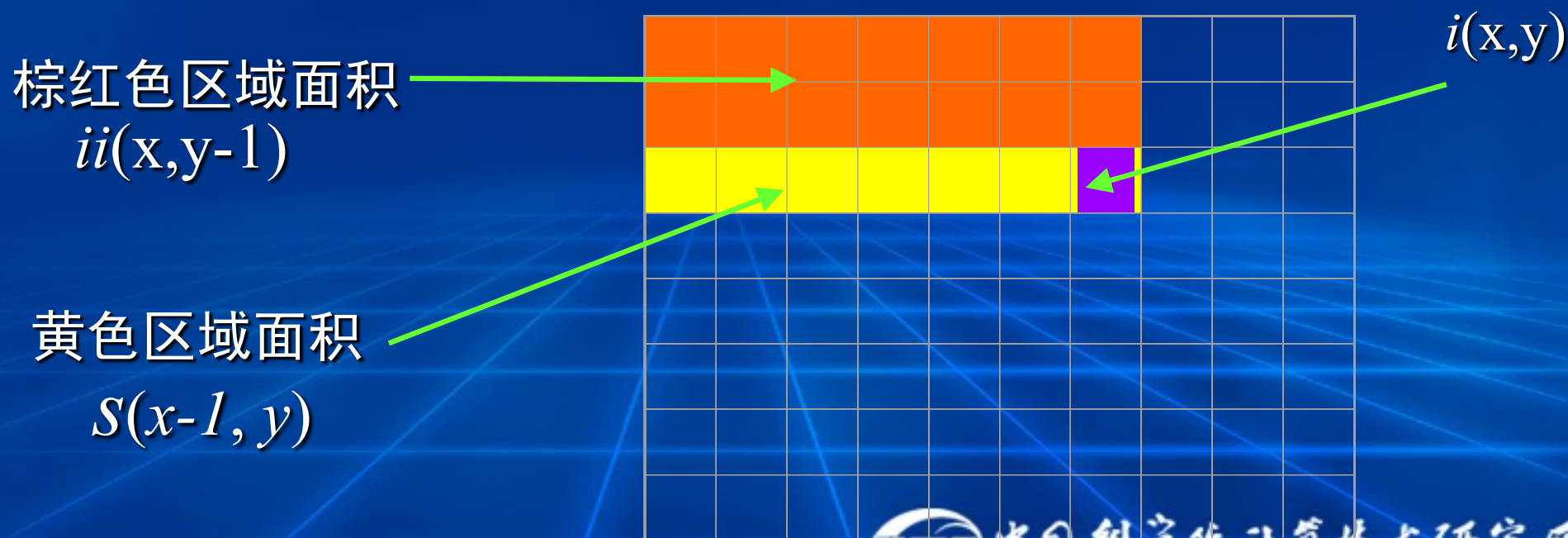
$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

$ii(x, y)$  = 该点左上面区域(红色)内所有像素值的和！



# 积分图的快速计算

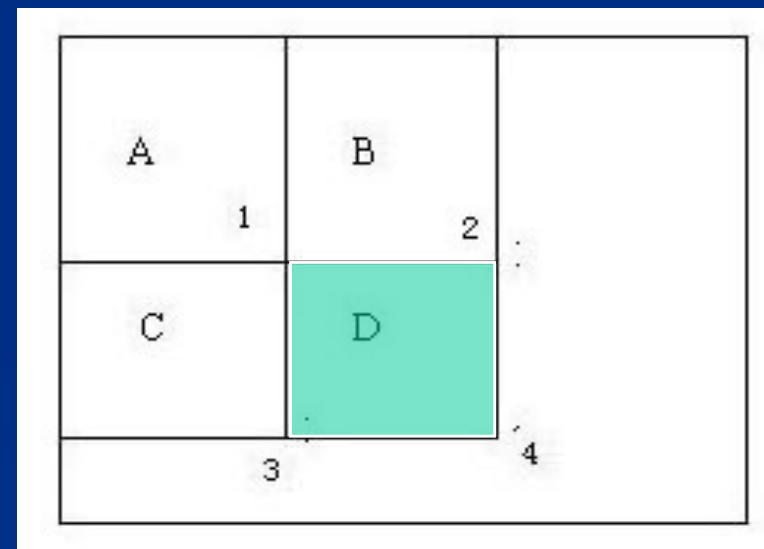
- 从左到右、从上到下扫描一遍图像，对图像进行如下操作，得到每个像素位置的积分图像  $ii(x,y)$ 
  - $ii(x,y) = ii(x,y-1) + s(x,y)$
  - $s(x,y) = s(x-1,y) + i(x,y)$





# Haar-Like特征的快速计算

- 矩形D内像素亮度积分值：  
 $= ii(4) - [ii(2) + ii(3)] + ii(1)$



- 推导
  - 像素点1的积分值是矩形A所有点的亮度值的和
  - 像素点2的积分值是A+B中...
  - 像素点3的积分值是A+C中...
  - 像素点4的积分值是A+B+C+D中...



输入图像



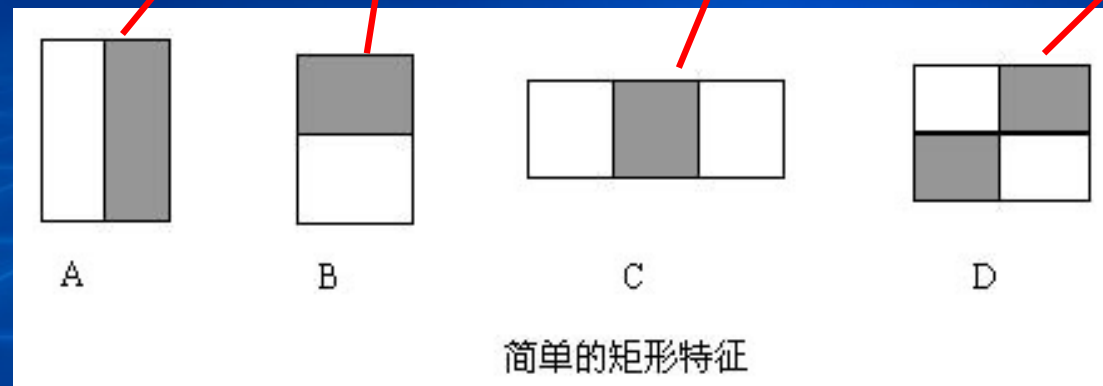
积分图像



基于积分图像的  
Haar-like特征计算

# 何以快速？

- 全部为定点加/减法操作！
  - 矩形区域和：2次加法1次减法
  - 矩形特征计算：1次减法，或2次减法，或者2次加法1次减法
- 没有乘法！



# AdaBoost分类器

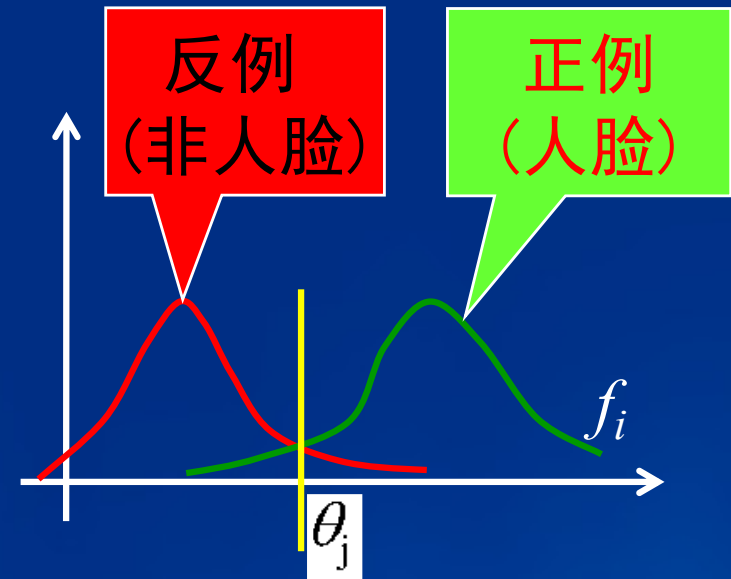
- AdaBoost分类器
  - Adaboost学习算法是用来提高简单分类算法的性能的
  - 通过对一些弱分类器的组合来形成一个强分类器
- 功能
  - 将分类性能不好的弱分类器提升为分类性能好的强分类器的学习算法
- 思想
  - 学习一系列分类器，在这个序列中每一个分类器对它前一个分类器导致的错误分类样例给予更大的重视



# AdaBoost用于人脸模式分类

- 弱分类器

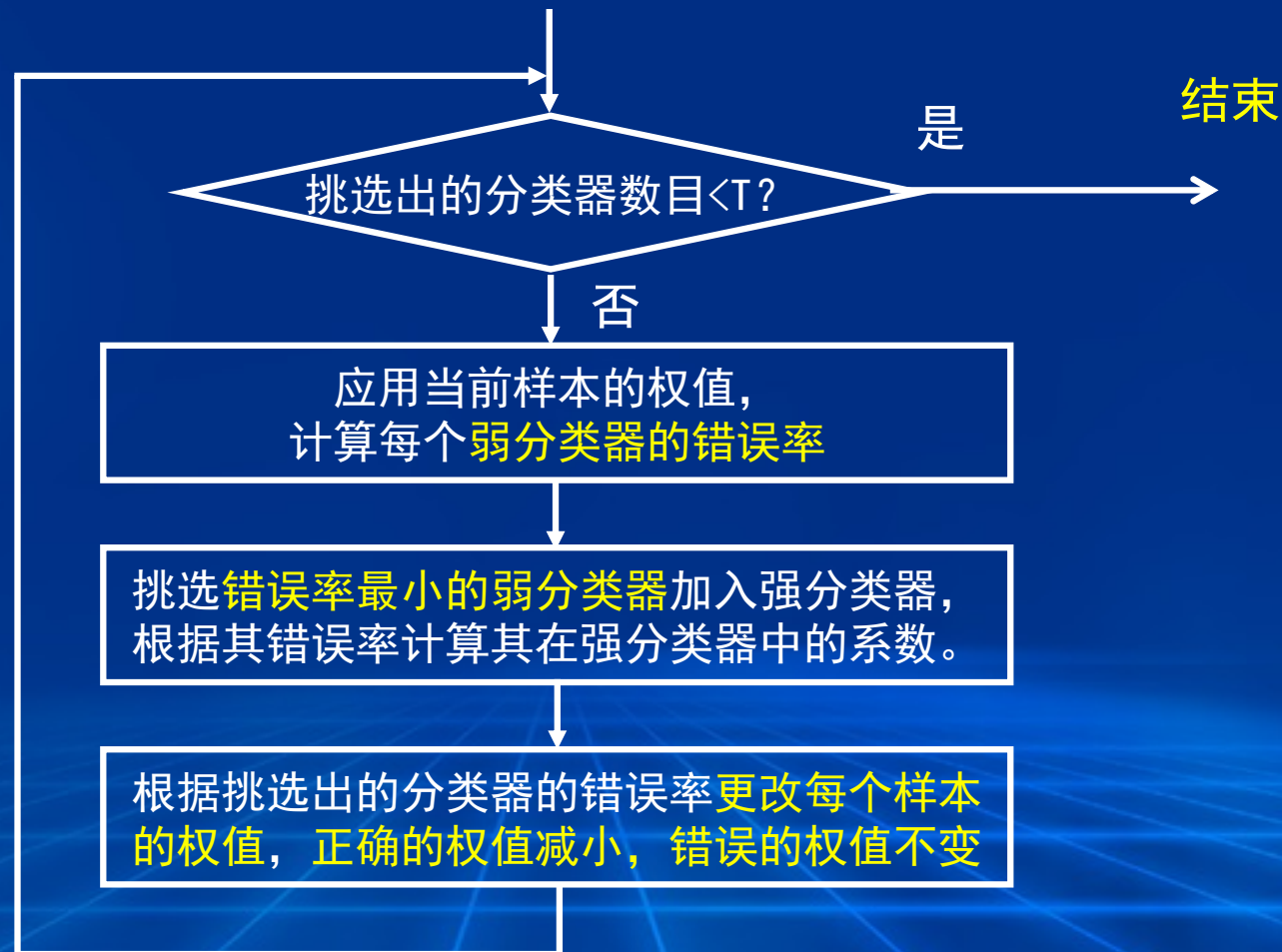
$$h_j(x) = \begin{cases} 1, & \text{if } f_j(x) < \theta_j \\ 0, & \text{otherwise} \end{cases}$$



其中， $h$ 表示弱分类器的值， $\theta_j$ 表示弱学习算法寻找出的阈值， $f_j(x)$ 表示特征值， $x$ 表示一个Harr-like特征。



# Adaboost学习算法流程



# AdaBoost用于人脸模式分类

- 输入
  - 1.训练用人脸和非脸样本
  - 2.指定要挑选出来的弱分类器的数目 $T$ 
    - 这也是程序循环的次数
  - 3.利用先验知识初始化权值向量
    - 一般可以平均设置

# AdaBoost学习算法

- 输入

- 训练样本集合  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$

其中:  $y_i \in \{1, 0\}$  1, 0 分别表示人脸、非人脸

- 每个样本固定大小

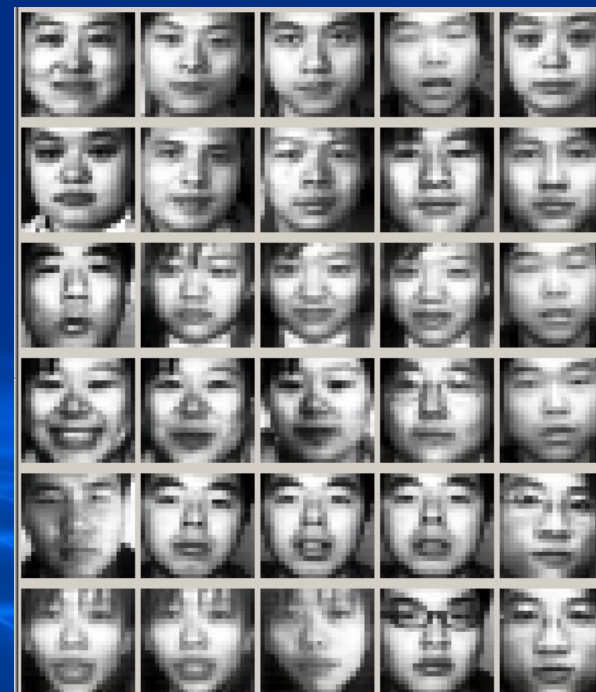
- 如20x20像素

- 计算全部可能的Haar-like特征 (数十万个)

- 训练样本可能多达100,000个

- 初始化T, 初始化每个样本的权重

$$w_{1,i} = \begin{cases} \frac{1}{2m} & \text{if } y_i = 1 \\ \frac{1}{2l} & \text{if } y_i = 0 \end{cases}$$





# AdaBoost学习算法

- 学习流程

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

- 1. 归一化权重, 使得 $w_t$ 为一个概率分布:  $w_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}}$
    - 2. 对每个特征 $j$ , 训练一个弱分类器 $h_j$ 并计算其带权重的错误率

$$\varepsilon_j = \sum_{i=1}^n w_{t,i} |h_j(x_i) - y_i|$$

- 3. 选择误差最小的弱分类器 $h_t$
    - 4. 更新每个样本的权重  $w_{t,i} = w_{t,i} \beta_t^{1-e_i}$ ,  $\beta_t = \frac{\varepsilon_t}{1 - \varepsilon_t}$   
其中:  $x_i$  被正确分类,  $e_i=0$ , 否则 $e_i=1$

正确分类则权重下降, 否则不变; 错误率越小, Beta也越小

# AdaBoost学习算法

- 最终得到的强分类器

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

弱分类器错误率越小， $\beta$  就越小， $\alpha$  就越大

$H(x)$ 为+，则为正例（人脸）

$H(x)$ 为-，则为反例（非人脸）



# 检测系统的构建——遍历所有位置

如何应用训练好的分类器？



INTER 2002/03





# 检测系统的构建——遍历不同大小





# 如何进一步加速？

- 遍历所有可能的位置、大小，需要做多少次分类？
  - 320x240图像，20x20模板，每次走1个像素，尺度缩小因子0.9，则大约有：
    - $300 \times 220 + 278 \times 196 + 240 \times 174 + 213 \times 155 + 190 \times 137 + 169 \times 122 + 150 \times 108 + 133 \times 95 + 118 \times 83 + 103 \times 73 + 91 \times 64 + 80 \times 55 + \dots = 30\text{多万}$
- 如果再考虑3个角度的旋转，则会超过100万次分类！
- 问题
  - 如何进一步加速？

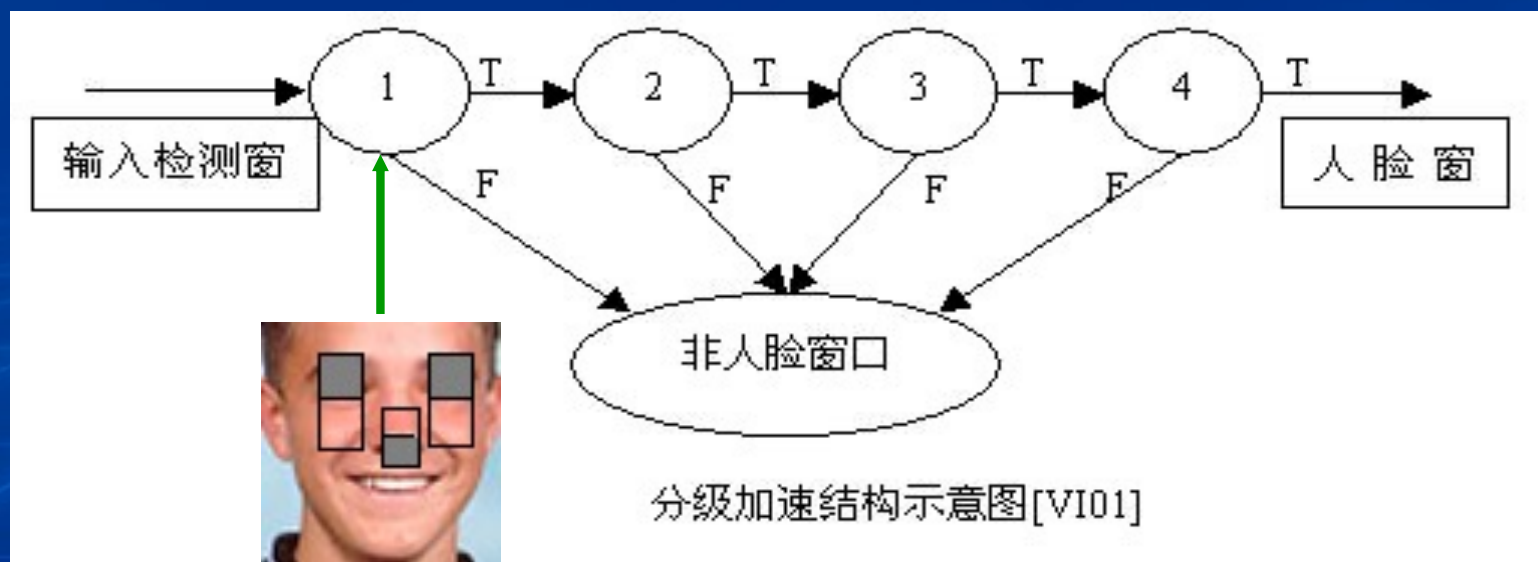
# 类比一下

- 分析
  - 非人脸模式里面大量与人脸极其不同！只有少量的模式很难与人脸分开！
- 类比筛沙子
  - 输入：巨石，大石头，小石头，沙砾，粗沙，细沙
  - 目标：只留下细沙（人脸）
  - 假设：大孔筛子容易做便宜，小细孔筛子难做昂贵
- 方法1
  - 只用一个筛子，网眼非常小，所有的都要过这个筛子
- 方法2
  - 用一系列筛子，网眼从大到小
  - 非常容易被截住的巨石首先被排除，然后是大石头，再是小石头，沙砾，粗沙，最后是细沙



# 基于AdaBoost的快速人脸检测

- 基于分级分类器的加速策略
  - 大量候选窗口可以利用非常少量的特征（简单快速的分类器）就可以排除是人脸的可能性！
  - 只有极少数需要大量的特征（更复杂的更慢的分类器来判别是否人脸）

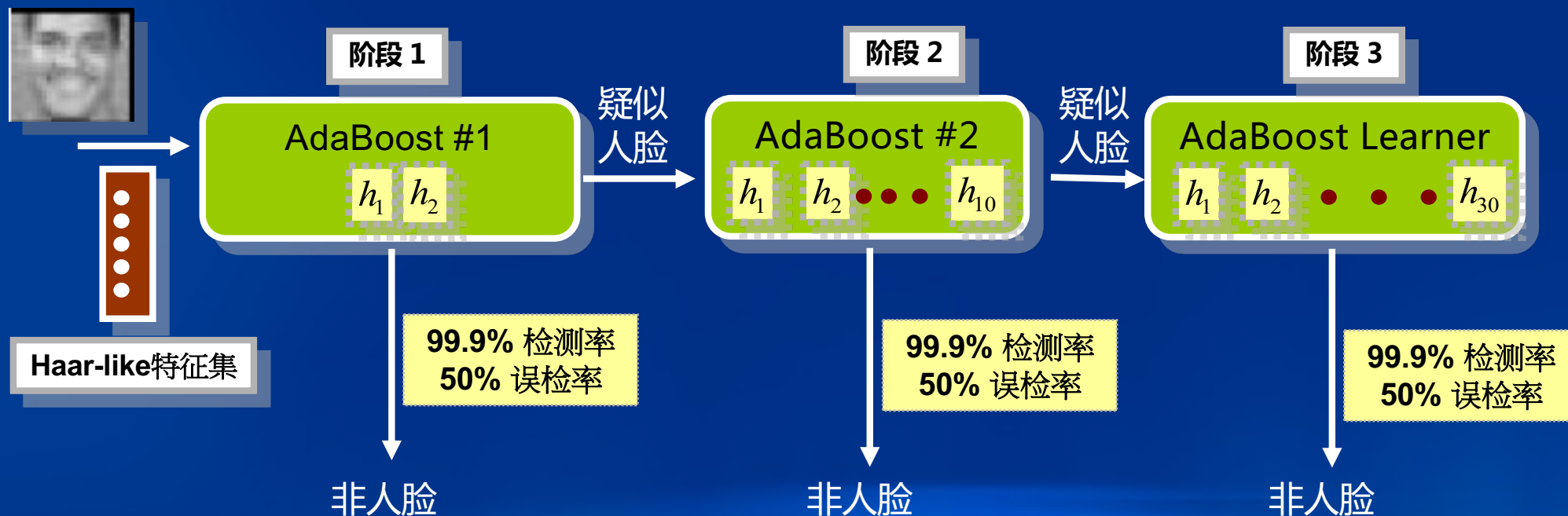


# 分级分类器的构建

- 采用由粗到细的思想（coarse to fine）
  - 将少量区分性好的特征构成的简单分类器置于前面若干层
    - 效果：放过检测绝大多数人脸的同时，排除大量非脸
  - 后面层包含更多次重要的特征对非脸进行进一步排除
- 训练方法关键思路
  - 每层训练用非脸样本使用前面层分类器误判为人脸的那些样本



# 由粗到精的多阶段人脸检测器



Reject as many negatives as possible → (minimize the false negative)