法律声明

- 本课件包括：演示文稿，示例，代码，题库，视频和声音等，小象学院拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利。



关注 小象学院

第4讲　未有深度学习之前

计算机视觉

主讲人：叶梓

上海交通大学博士
主要研究方向：机器学习、深度学习、人工智能

本章内容

- 图像分割
  - 基于阈值、基于边缘
  - 基于区域、基于图论
- 人脸检测
  - Haar-like特征+级联分类器
- 行人检测
  - HOG+SVM
  - DPM

图像分割

- 所谓图像分割指的是根据灰度、颜色、纹理和形状等特征把图像划分成若干互不交迭的区域，
- 并使这些特征在同一区域内呈现出相似性，而在不同区域间呈现出明显的差异性。
- 经典的数字图像分割算法一般是基于灰度值的两个基本特性之一：不连续性和相似性。

传统图像分割的方法

基于阈值的分割方法

- 阈值法的基本思想是基于图像的灰度特征来计算一个或多个灰度阈值，并将图像中每个像素的灰度值与阈值相比较，最后将像素根据比较结果分到合适的类别中。

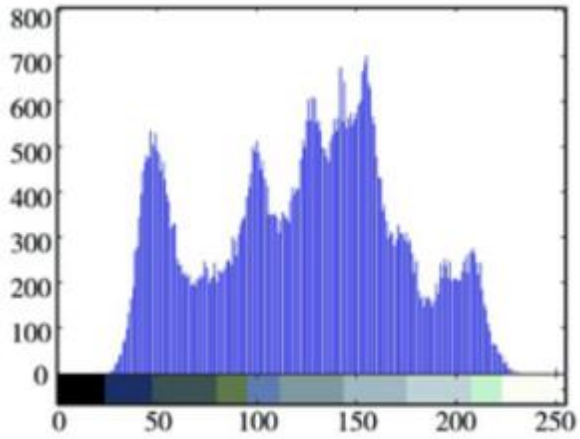
基于边缘的分割方法

- 所谓边缘是指图像中两个不同区域的边界线上连续的像素点的集合，是图像局部特征不连续性的反映，体现了灰度、颜色、纹理等图像特性的突变。

基于阈值的分割

- 设定某一灰度阈值T，能将图像分成两部分：大于T的像素群和小于T的像素群定一个合适的阈值就可准确地将图像分割开来。
- 算法的关键是确定合适的阈值。

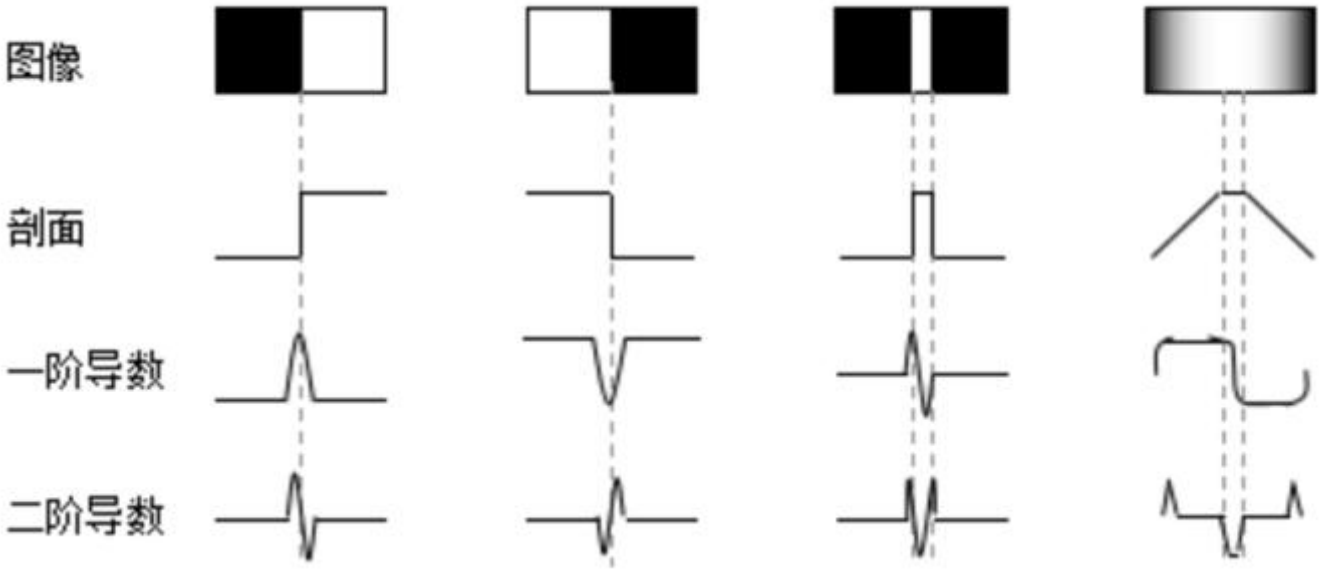






基于边缘的分割方法

- 基于边缘检测的图像分割方法的基本思路是先确定图像中的边缘像素，然后再把这些像素连接在一起就构成所需的区域边界。



传统图像分割的方法

- **基于区域的分割方法**
  - 此类方法是将图像按照相似性准则分成不同的区域，主要包括种子区域生长法、区域分裂合并法和分水岭法等几种类型。
- **基于图论的分割方法**
  - 此类方法把图像分割问题与图的最小割（min cut）问题相关联。首先将图像映射为带权无向图，图中每个节点对应于图像中的每个像素，每条边的权值表示了相邻像素之间在灰度、颜色或纹理方面的非负相似度。

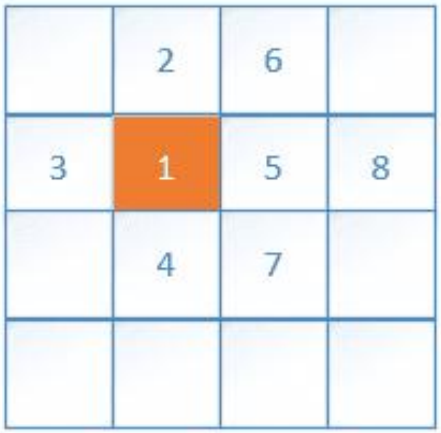
区域生长法

- 区域生长是根据一种事先定义的准则将像素或者子区域聚合成更大区域的过程，并且要充分保证分割后的区域满足以下条件：内部连通、互斥，且各子集的并集能构成全集。

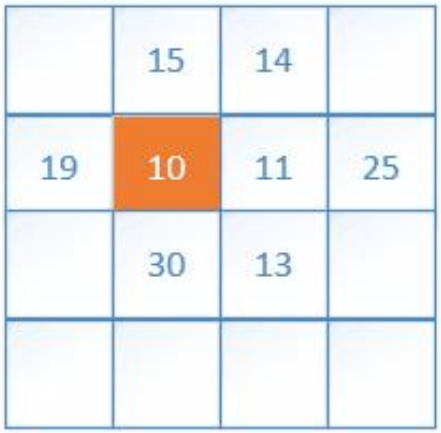
- 区域生长的条件实际上就是根据像素灰度间的连续性而定义的一些相似性准则，而区域生长停止的条件则是一个终止规则。

区域生长法

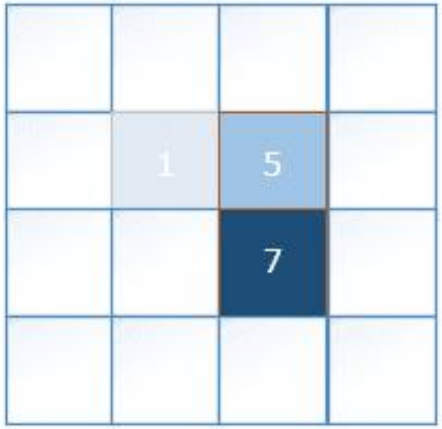
- 算法定义了最大像素灰度值距离，当待加入像素点的灰度值和已经分割好的区域所有像素点的平均灰度值的差的绝对值不大于最大像素灰度值距离时，该像素点加入到已经分割到的区域。相反，则区域生长算法停止。



像素点标号



对应像素点的灰度值



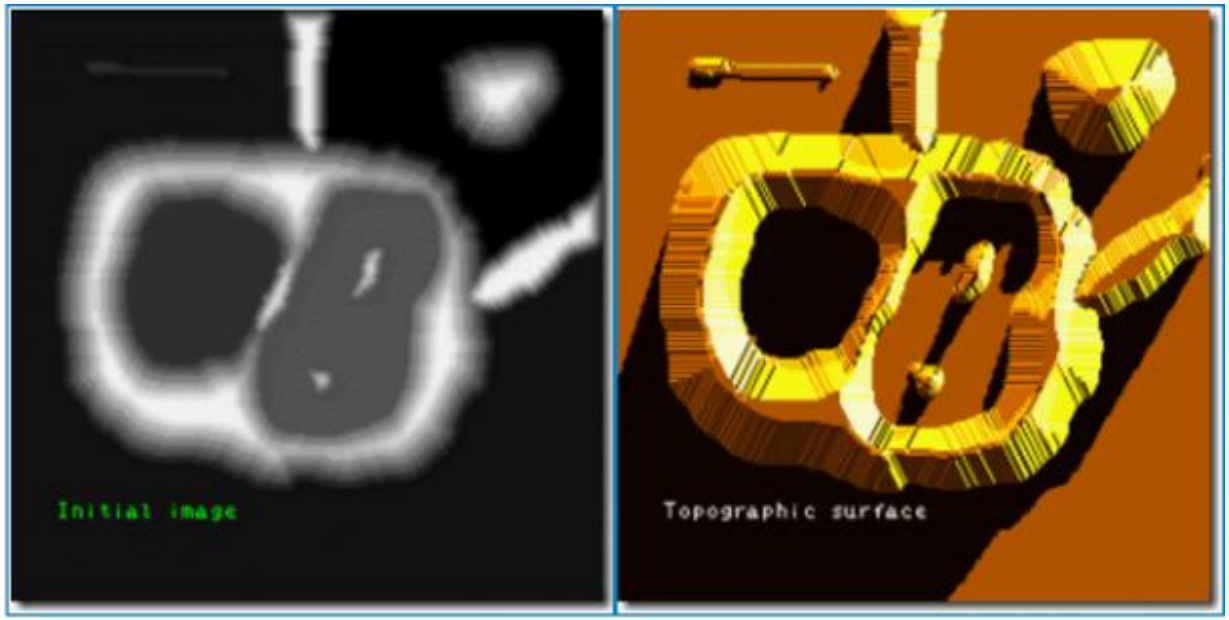
区域生长的方向

分水岭算法

- 分水岭算法是一种图像区域分割法，分水岭算法可以将图像中的边缘转化成"山脉"，将均匀区域转化为"山谷"，这样有助于分割目标。
- 在分割的过程中，它会把跟临近像素间的相似性作为重要的参考依据，从而将在空间位置上相近并且灰度值相近（求梯度）的像素点互相连接起来构成一个封闭的轮廓。

分水岭算法步骤

- 分水岭算法常用的操作步骤：
  - 彩色图像灰度化，然后再求梯度图，
  - 最后在梯度图的基础上进行分水岭算法，求得分段图像的边缘线。



前景背景分割

前景背景分割

- 前景一般包含物体
- 需要交互提供初始标记



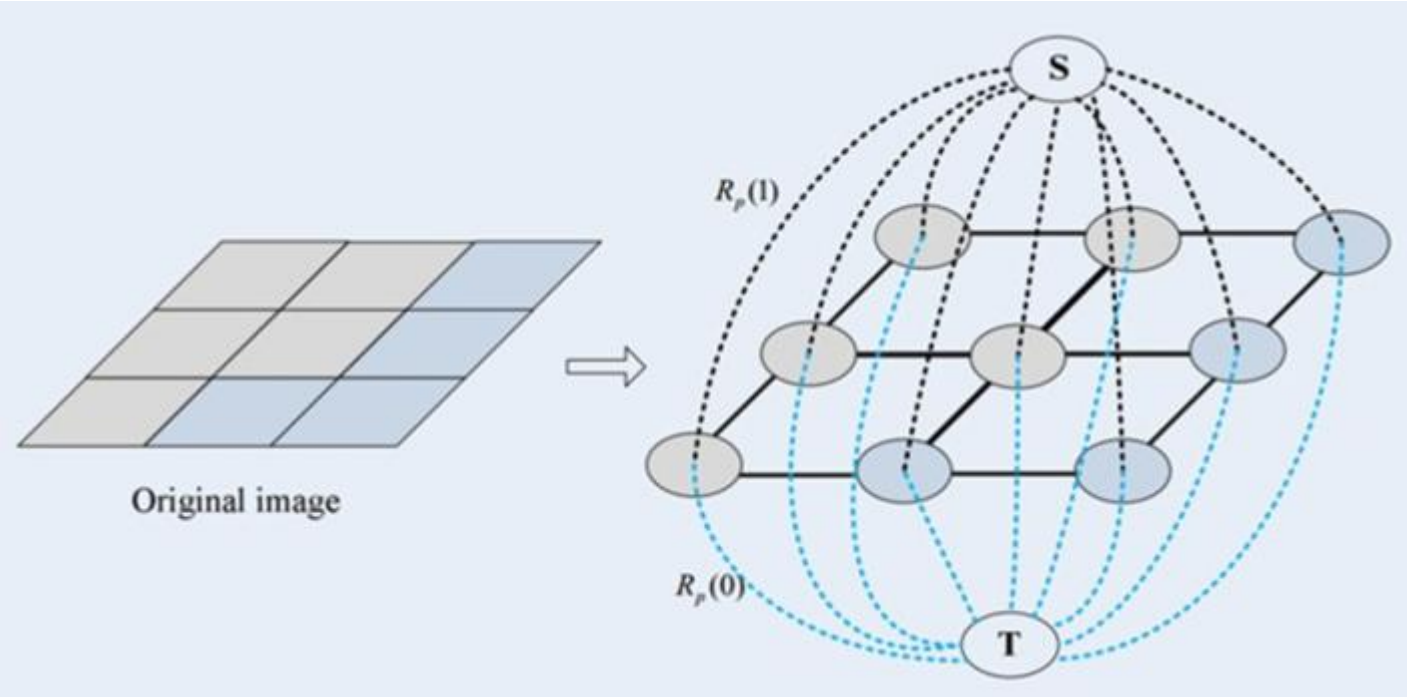


Graph Cuts分割

- Graph Cuts是基于图论的分割方法。
- 第一种普通顶点对应于图像中的每个像素。每两个邻域顶点（对应于图像中每两个邻域像素）的连接就是一条边。
- 另外两个终端顶点，叫S（source：代表前景）和T（sink：代表背景）。每个普通顶点和这2个终端顶点之间都有连接，组成第二种边。
- Graph Cuts中的Cuts是指这样一个边的集合，该集合中所有边的断开会导致残留"S"和"T"图的分开，所以就称为"割"。

Graph Cuts分割

- 如果一个割，它的边的所有权值之和最小，那么这个就称为最小割，也就是图割的结果。



Graph Cuts分割

基于能量的算法

- $E(A) = \lambda \cdot R(A) + B(A)$

$$R(A) = \sum_{p \in \mathcal{P}} R_p(A_p)$$

区域项

$$B(A) = \sum_{\{p,q\} \in \mathcal{N}} B_{\{p,q\}} \cdot \delta(A_p, A_q)$$

边界项

$$\delta(A_p, A_q) = \begin{cases} 1 & \text{if } A_p \neq A_q \\ 0 & \text{otherwise.} \end{cases}$$

R是为像素p分配标签l_p的惩罚：
当像素p的灰度值属于目标的概率大于背景，那么$R_p(1)$就小于$R_p(0)$，也就是说当像素p更有可能属于目标时，将p归类为目标就会使能量R(L)小。

B是像素p和q之间不连续的惩罚，一般来说如果p和q越相似（例如它们的灰度），那么B越大；如果他们非常不同，那么B就接近于0。

Graph Cuts分割

$$K = 1 + \max_{p \in \mathcal{P}} \sum_{q:\ \{p,q\} \in \mathcal{N}} B_{\{p,q\}}.$$

大数，代表绝对不可分割

$$B_{\{p,q\}} \propto exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right) \cdot \frac{1}{dist(p,q)}.$$

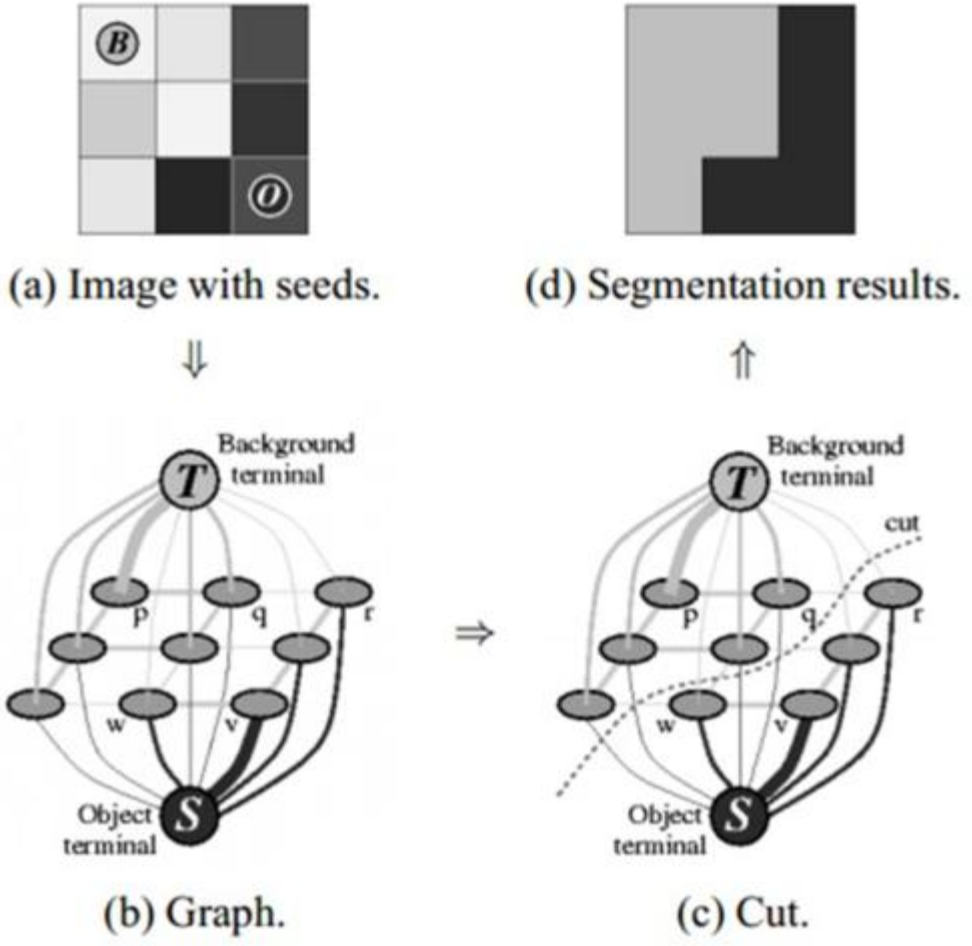
与像素灰度差、距离成反比。

$$R_p(\text{“obj”}) = -\ln \Pr(I_p|\mathcal{O})$$
$$R_p(\text{“bkg”}) = -\ln \Pr(I_p|\mathcal{B}).$$

edge	weight (cost)	for
$\{p,q\}$	$B_{\{p,q\}}$	$\{p,q\} \in \mathcal{N}$
$\{p,S\}$	$\lambda \cdot R_p(\text{“bkg”})$	$p \in \mathcal{P},\ p \notin \mathcal{O} \cup \mathcal{B}$
	K	$p \in \mathcal{O}$
	0	$p \in \mathcal{B}$
$\{p,T\}$	$\lambda \cdot R_p(\text{“obj”})$	$p \in \mathcal{P},\ p \notin \mathcal{O} \cup \mathcal{B}$
	0	$p \in \mathcal{O}$
	K	$p \in \mathcal{B}$

Graph Cuts分割

- 先取两个种子点
- 然后我们建立一个图，图中边的粗细表示对应权值的大小
- 然后找到权值和最小的边的组合
- 完成图像分割的功能。



GrabCut分割

只需要在目标外面画一个框，把目标框住，它就可以完成良好的分割









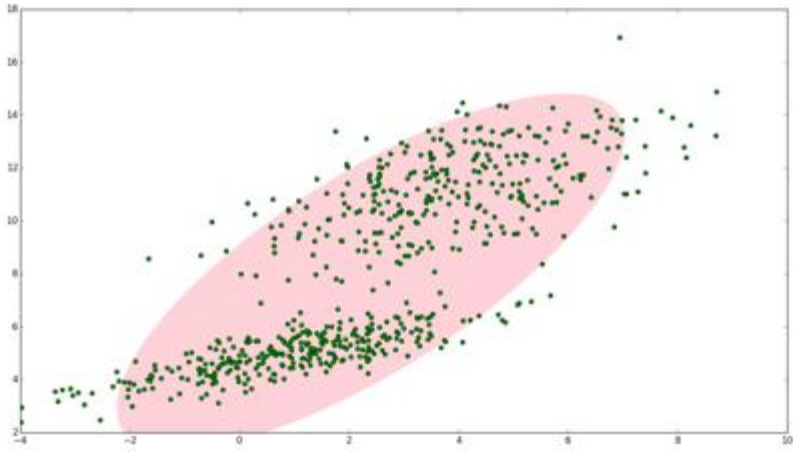
如果增加额外的用户交互（由用户指定一些像素属于目标），那么效果就可以更完美。

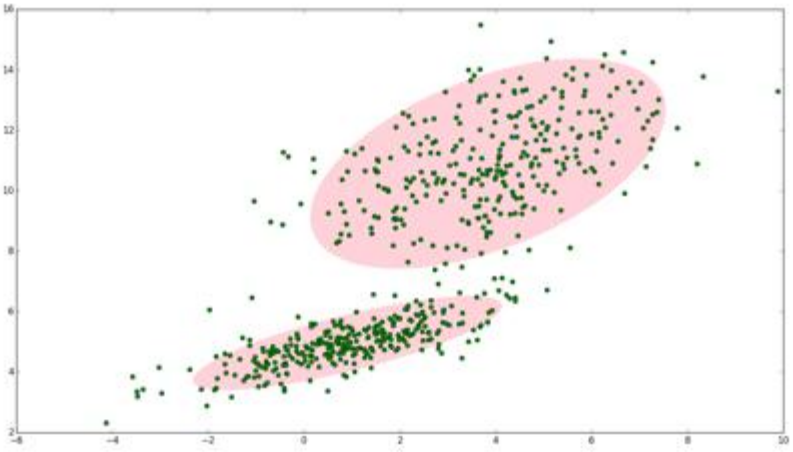
GrabCut分割

- 前景/背景的颜色模型
  - 假定前景和背景为高斯混合模型（GMM），
  - 对于每个像素，要么来自于目标GMM的某个高斯分量，要么就来自于背景GMM的某个高斯分量。
  - 用边界项来体现邻域像素之间不连续的惩罚，如果两邻域像素差别很小，那么它属于同一个目标或者同一背景的可能性就很大，反之则可能是边缘；
  - 通过K-means算法迭代获得。

GMM

- 两组点分别通过两个不同的正态分布随机生成而来。如果没有GMM，那么只能用一个的二维高斯分布来描述。
- 显然，这不太不合理。应该用两个不同参数的正态分布来描述。





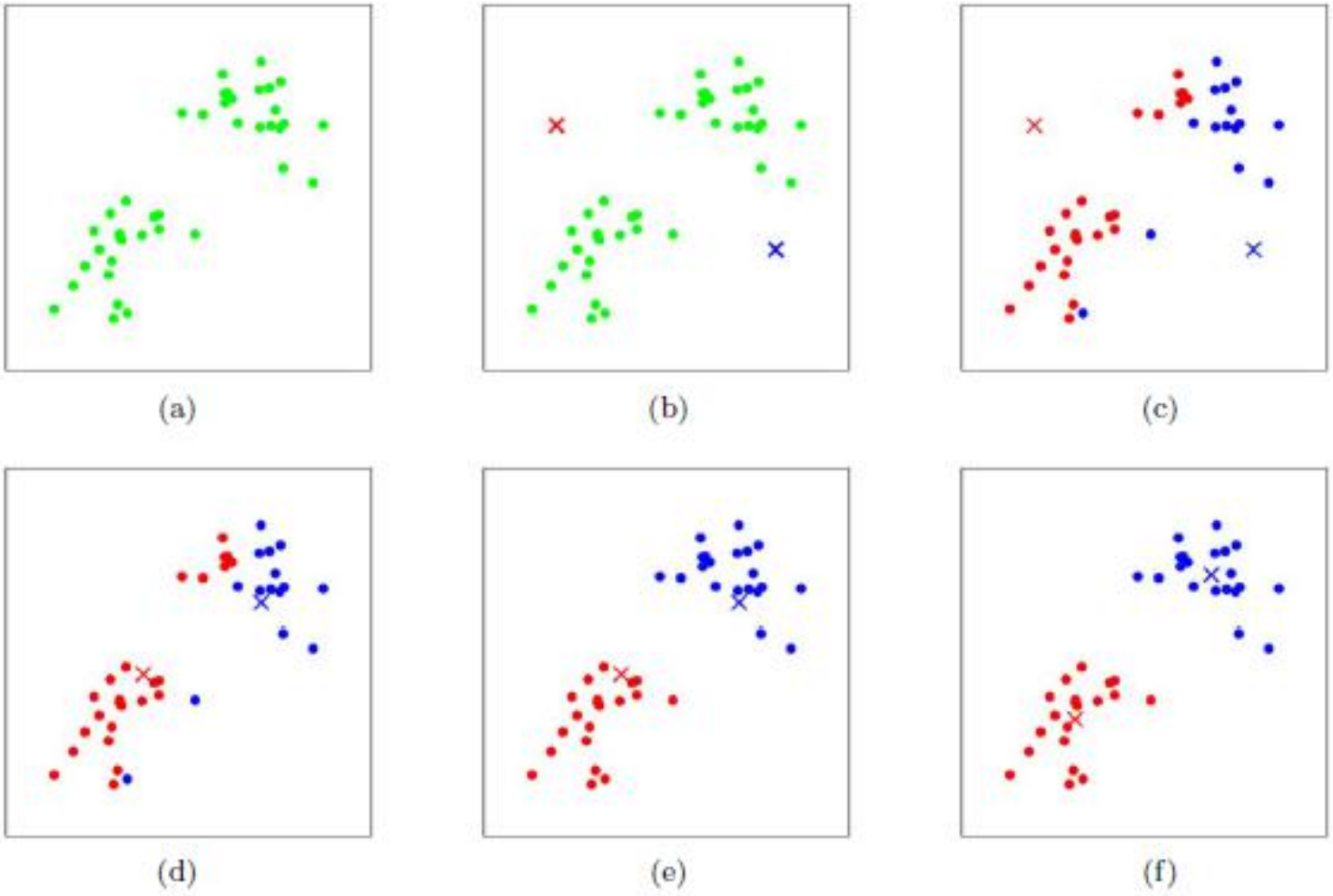
K-means

Kmeans

- 算法流程：
  1. 选取k个类中心
     - 首次随机选取
  2. 计算每个点跟k个类中心的距离
  3. 把数据点分配给距离最近的一个类中心
  4. 计算新的类中心→对该类中的所有点取均值
  5. 满足终止条件后终止迭代
     - 不再有重新分配
     - 最大迭代数
     - 所有类中心移动小于某一值

迭代

K-means聚类过程图示



关于K-Means的几个问题

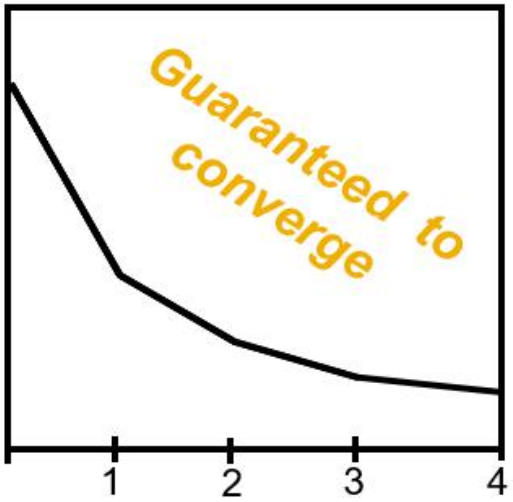
- K值怎么定？
  ——主要取决于经验，通常的做法是多尝试几个K值，看分成几类的结果更好解释，更符合分析目的等。
- 初始的K个质心怎么选？
  ——最常用的方法是随机选，初始质心的选取对最终聚类结果有影响，因此算法一定要多执行几次，哪个结果更合理，就用哪个结果。有一些优化的方法，例如：选择彼此距离最远的点。
- K-Means会不会陷入一直选质心的过程？
  ——不会，K-Means一定会收敛，大致思路是利用SSE的概念（也就是误差平方和），可证明这个函数是可以最终收敛的函数。

物体分割（Object Segmentation）

GrabCut分割

- 迭代进行Graph Cuts
  - 优化前景和背景的颜色模型
  - 能量随着不断迭代变小
  - 分割结果越来越好

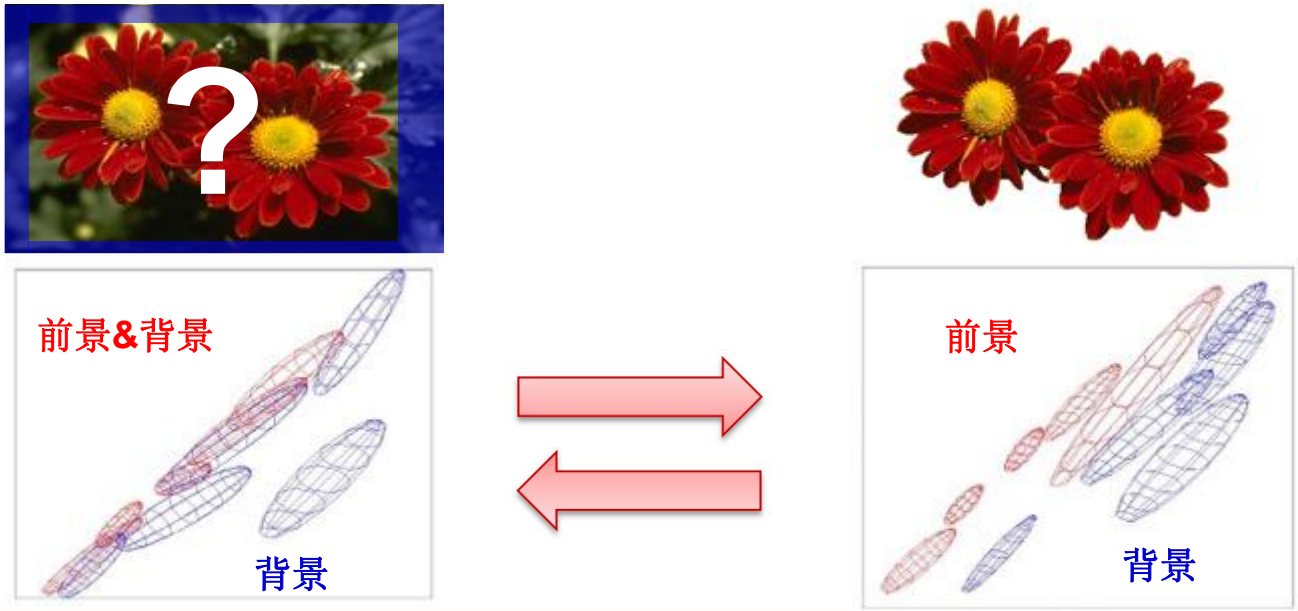




GrabCut分割

GrabCut分割

- 算法流程
  - 使用标记初始化颜色模型（K=5）
  - 迭代进行Graph Cut

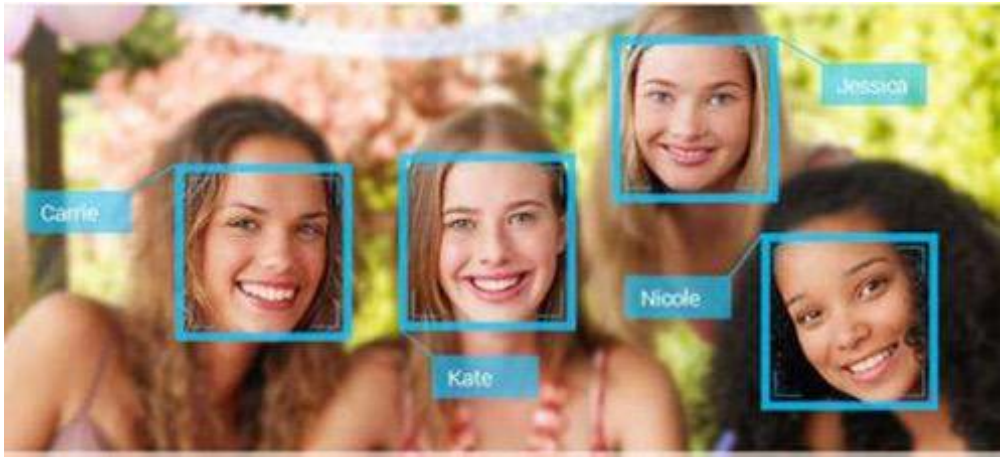


人脸检测



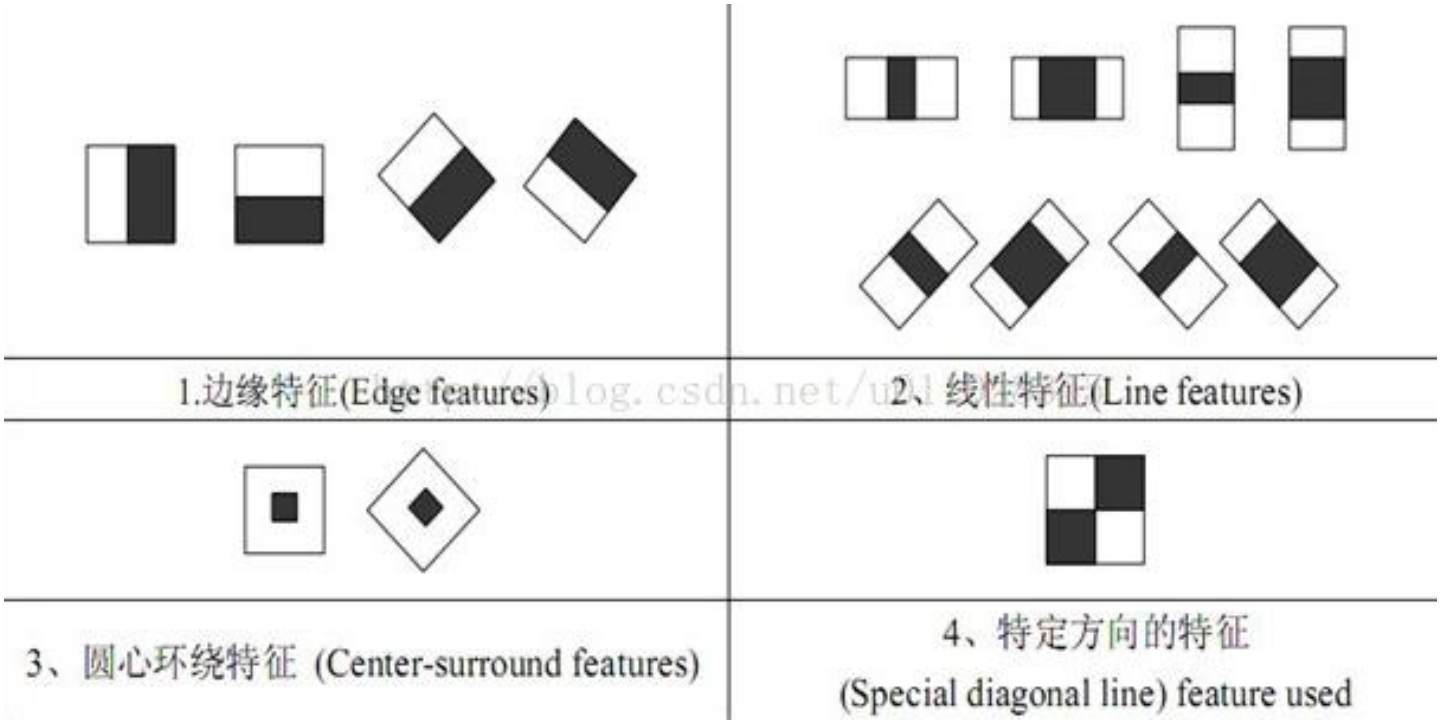
人脸检测

人脸识别



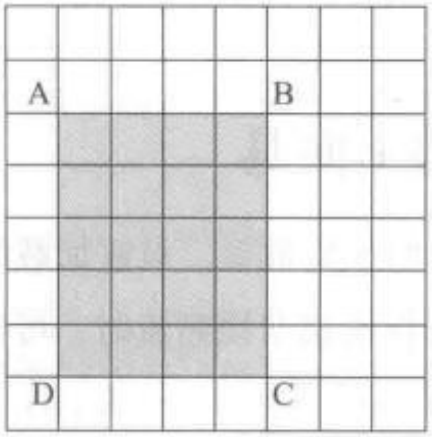
Haar-like特征

- 特征模板内有白色和黑色两种矩形，并定义该模板的特征值为白色矩形内的像素和减去黑色矩形内的像素和。

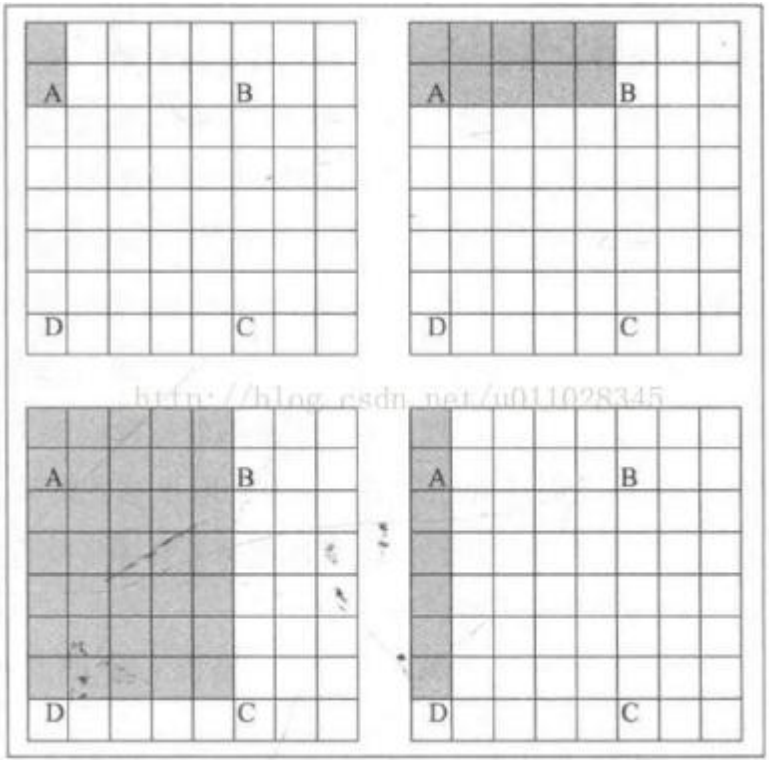


Haar-like特征

- 同一个像素如果被包含在不同的Haar-like的特征模板中，会被重复计算多次。
- 积分图是根据四个角点就能计算区域内的像素之和的方法。

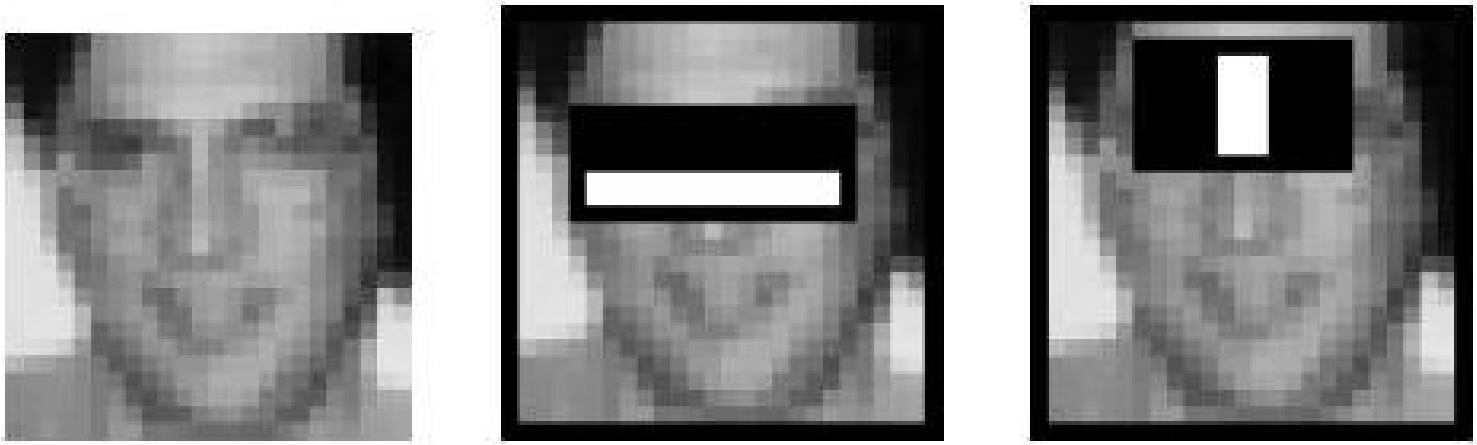


$$A_{ABCD}=A_C-(A_B+A_D-A_A)$$



Haar-like特征模板

- Haar-like模板可表示出人脸的某些特征。
- 例如：中间一幅表示眼睛区域的颜色比脸颊区域的颜色深；右边一幅表示鼻梁两侧比鼻梁的颜色要深。



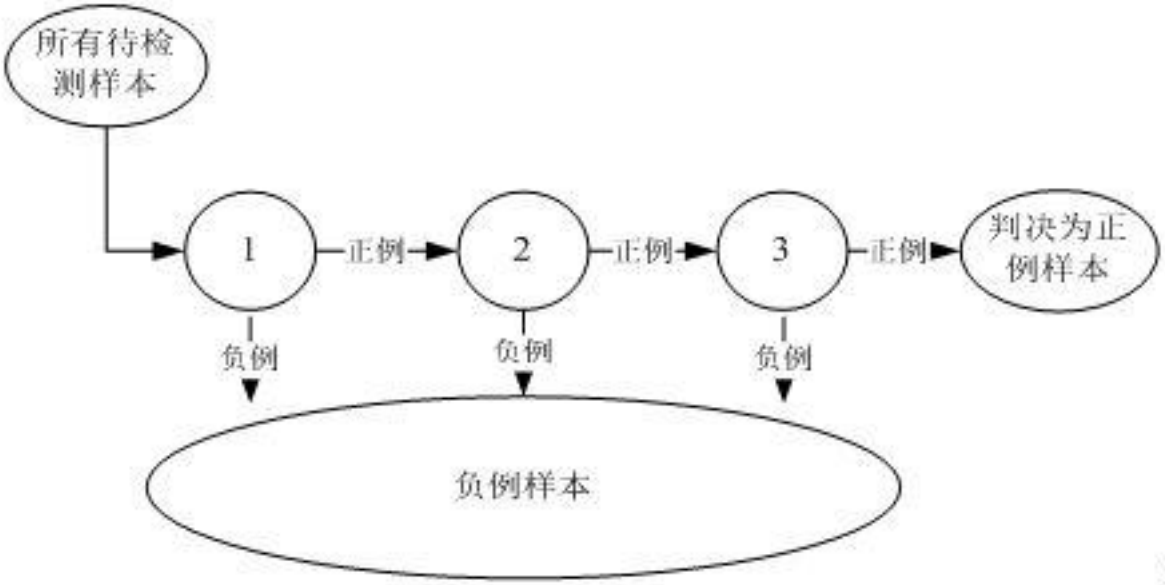
Haar-like特征的数量

- 要选多少个Haar-like特征?
  - 不同的模板（15种）
  - 不同的位置
  - 不同的缩放（长宽比固定或可变的）
  - 总之，非常多、非常多……

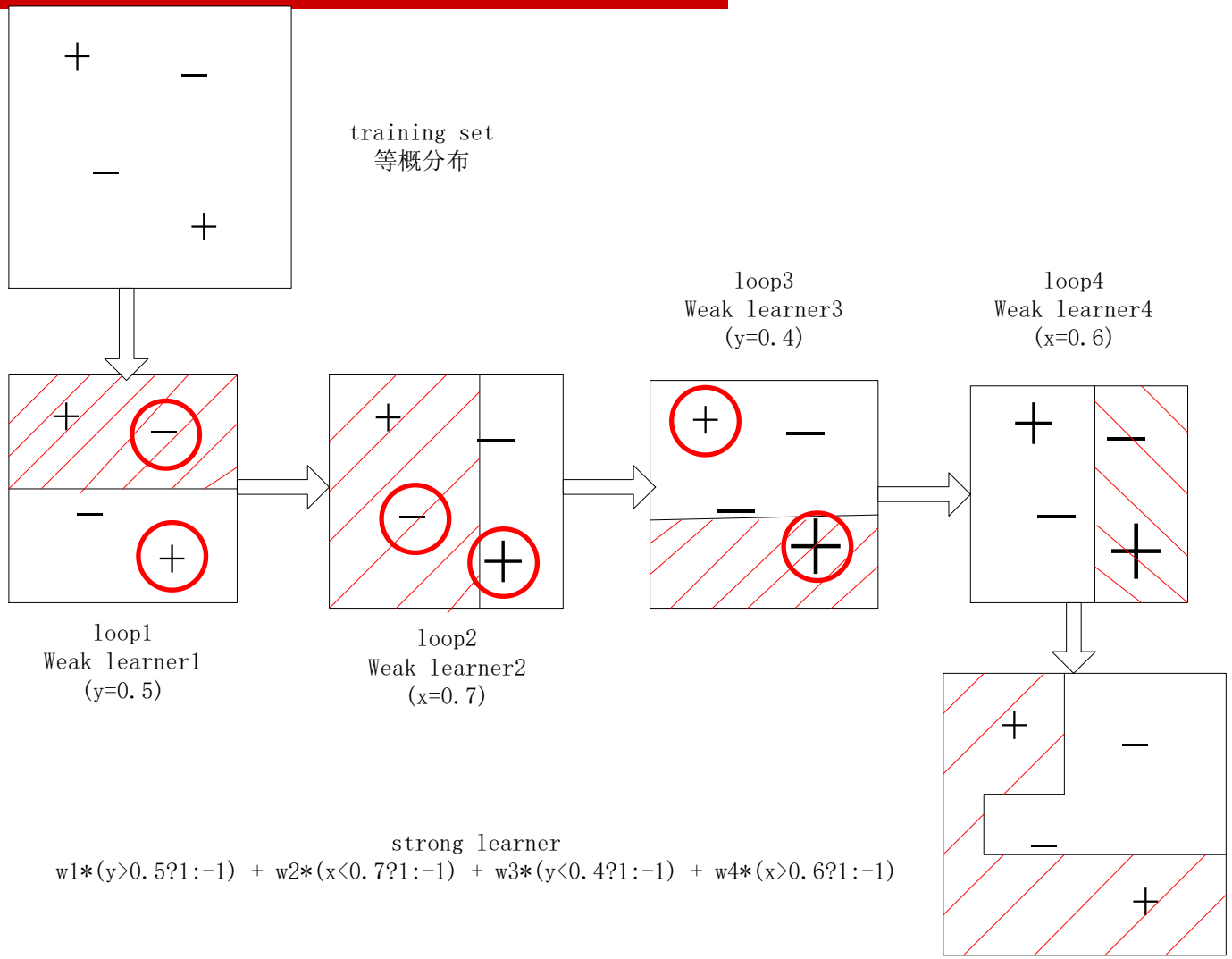
窗口大小	36×36	30×30	24×24	20×20	16×16
特征数量	816,264	394,725	162,336	78,460	32,384

Haar级联分类器

- Adaboost是一种基于级联分类模型的分类器。
- 级联分类器就是将多个强分类器连接在一起进行操作。
- 每一个强分类器都由若干个弱分类器加权组成。



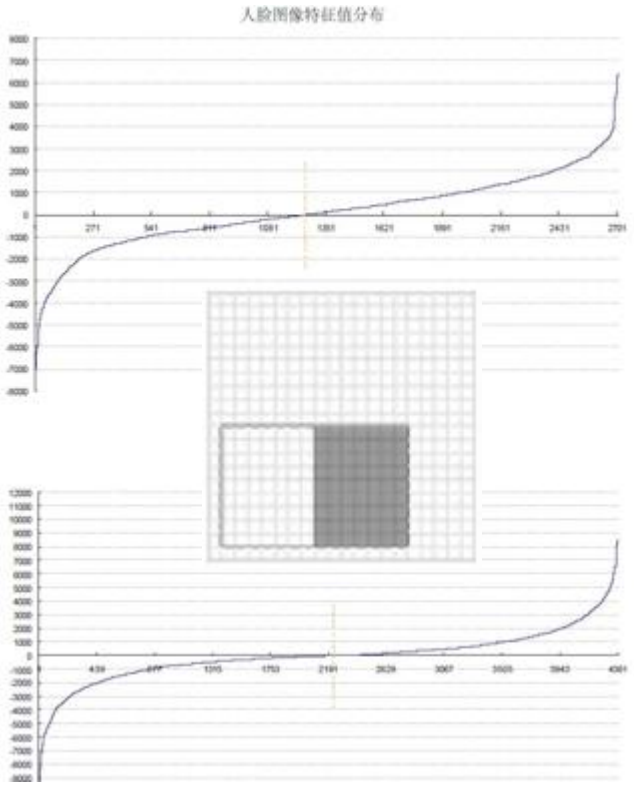
Boosting分类器示意

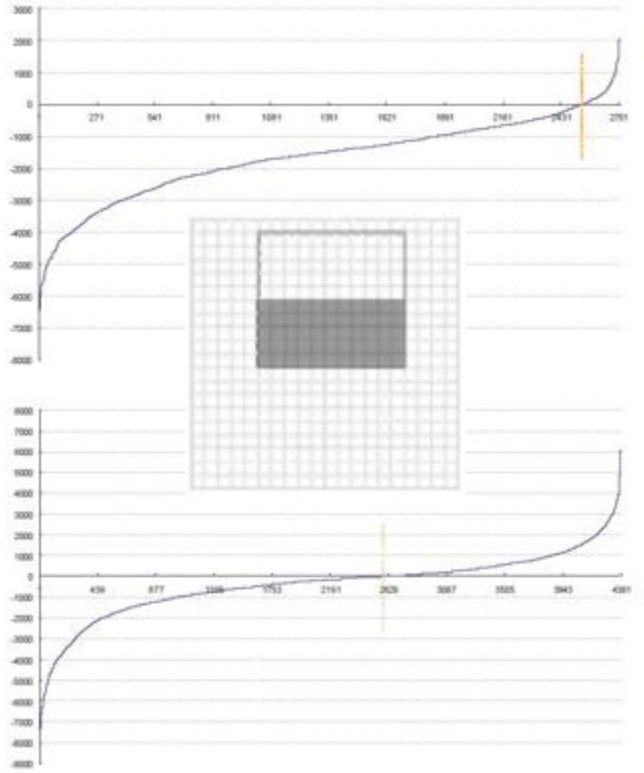


小象学院 ChinaHadoop.cn

弱分类器和强分类器

- 弱分类器，就是在这海量的特征中选取一个特征，用这个特征能够区分出人脸or非人脸，且错误率最低。





弱分类器和强分类器

- 一个弱学习器的要求仅仅是：它能够以稍低于50%的错误率来区分人脸和非人脸图像。

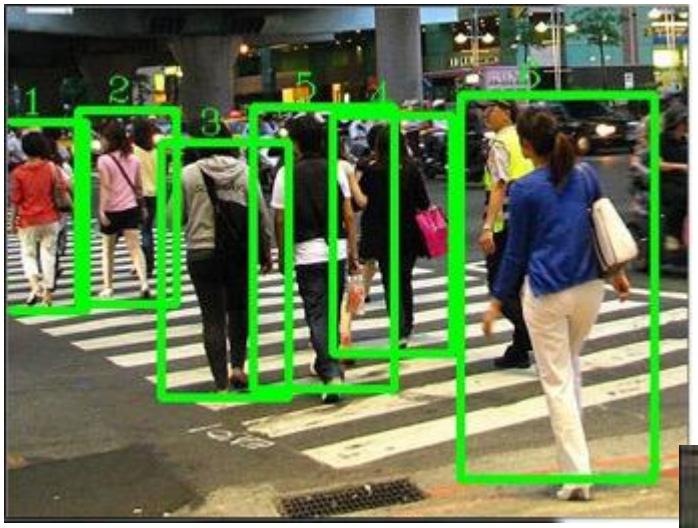
- 训练一个弱分类器就是在当前权重分布的情况下，确定f的最优阈值，使得这个弱分类器对所有训练样本的分类误差最低。

- 最后将每轮得到的最佳弱分类器按照一定方法提升（Boosting）为强分类器。

级联分类器的检测机制

- 级联分类器中每一个强分类器都是对于“非人脸”（即负样本）更敏感，使得每次被强分类器拒绝的，都几乎一定不是人脸。经过所有强分类器考验的，才是“人脸”。
- 一幅图像中待检测的区域很多都是负样本，只有正样本才会送到下一个强分类器进行再次检验。这样由级联分类器在分类器的初期就抛弃了很多负样本的复杂检测。
- 所以，级联分类器的速度是非常快的。

行人检测



应用在自动驾驶、人流监控、人体行为分析等领域。



梯度

- 梯度是一个矢量

$$\nabla f(x,y)=\left[G_x G_y\right]^T=\left[\frac{\partial f}{\partial x}\frac{\partial f}{\partial y}\right]^T$$

- G_x是沿x方向上的梯度，G_y是沿y方向上的梯度，梯度的幅值及方向角可表示如下：

$$\begin{cases}\left|\nabla f(x,y)\right|=mag(\nabla f(x,y))=(G_x^2+G_y^2)^{1/2}\\ \phi(x,y)=\arctan(G_y/G_x)\end{cases}$$

- 一维离散微分模板在将图像的梯度信息简单、快速且有效地计算出来，其公式如下：

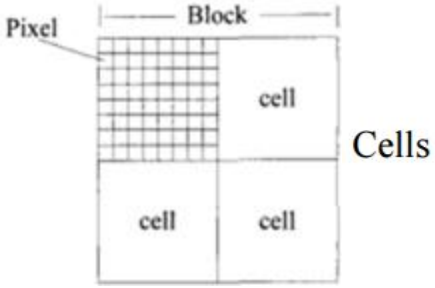
$$\begin{cases}G_x(x,y)=H(x+1,y)-H(x-1,y)\\ G_y(x,y)=H(x,y+1)-H(x,y-1)\end{cases}$$

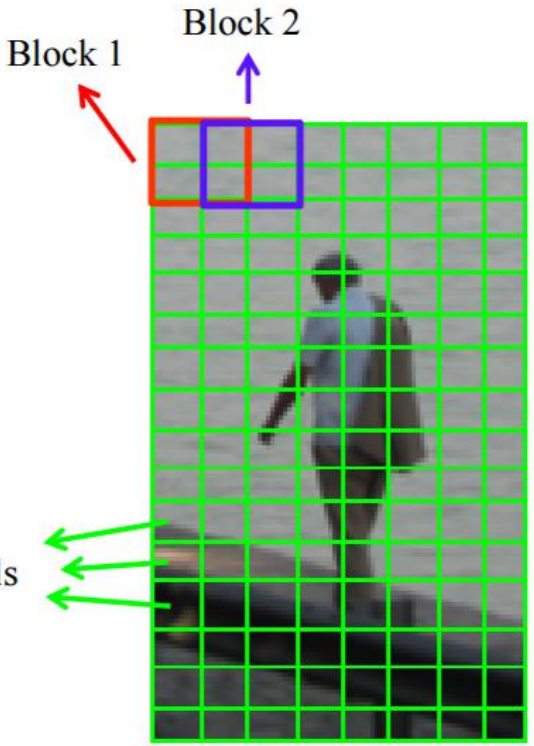
HOG

HOG（方向梯度直方图）

- 梯度计算
  - 分别计算水平、垂直梯度
    - 梯度幅值、方向 $s=\sqrt{s_x^2+s_y^2}$ $\theta=\arctan(\frac{s_y}{s_x})$
    - 无平滑
  - 对于彩色图，选取梯度幅值最大的通道
- Block拆分
  - 16x16的Block，步长是8，50%的重合
  - 包含2x2个Cell
  - 每个Cell是8x8



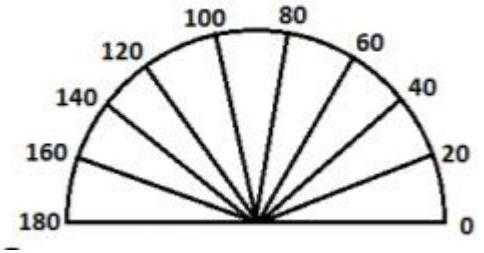


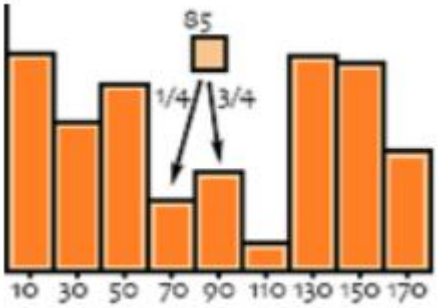


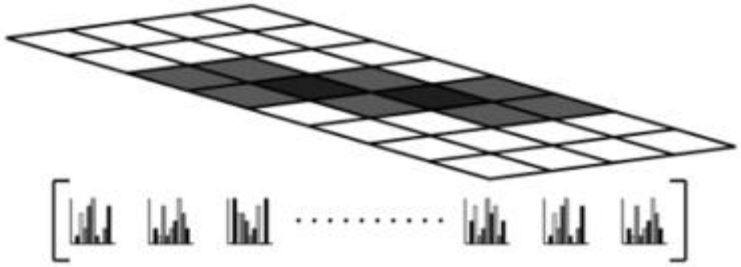
HOG

HOG（方向梯度直方图）

- 计算Cell的梯度方向直方图
  - 9个方向bin（0-180）
  - 积累梯度幅值
  - 使用位置高斯加权
  - 相邻bin使用线性插值
- 串接所有Block直方图
  - 64x128图片
    - 7x15 = 105 Block
    - 105 x (2x2) x 9 =3780 维

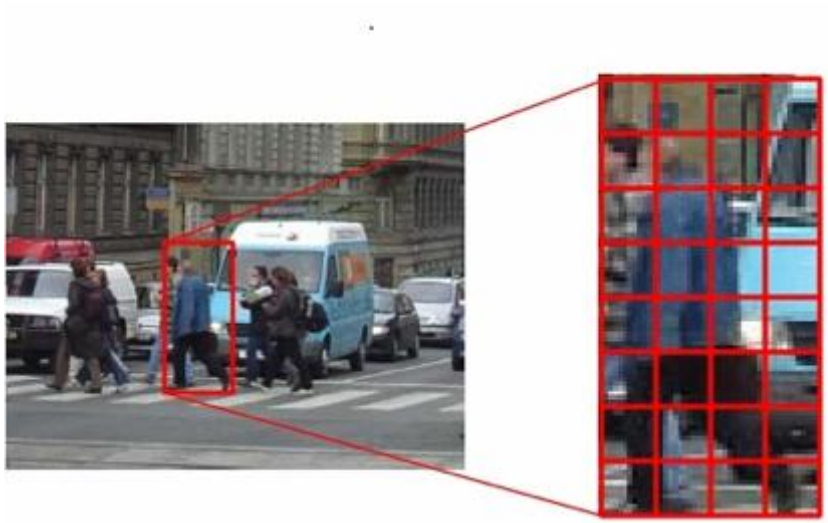




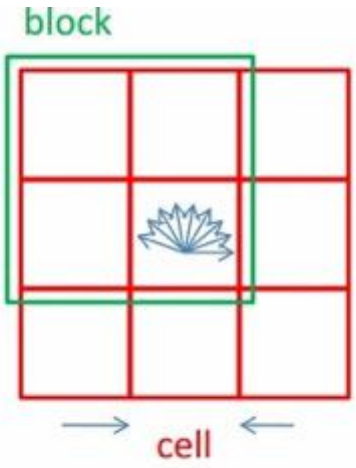


HOG：对比度归一化

- 梯度幅值绝对值的大小容易受到前景与背景对比度及局部光照的影响，要减少这种影响，还必须对局部细胞单元进行对比度归一化处理。



Feature vector
f = [...,...,..., ,...]



9 orientation bins
0 - 180° degrees

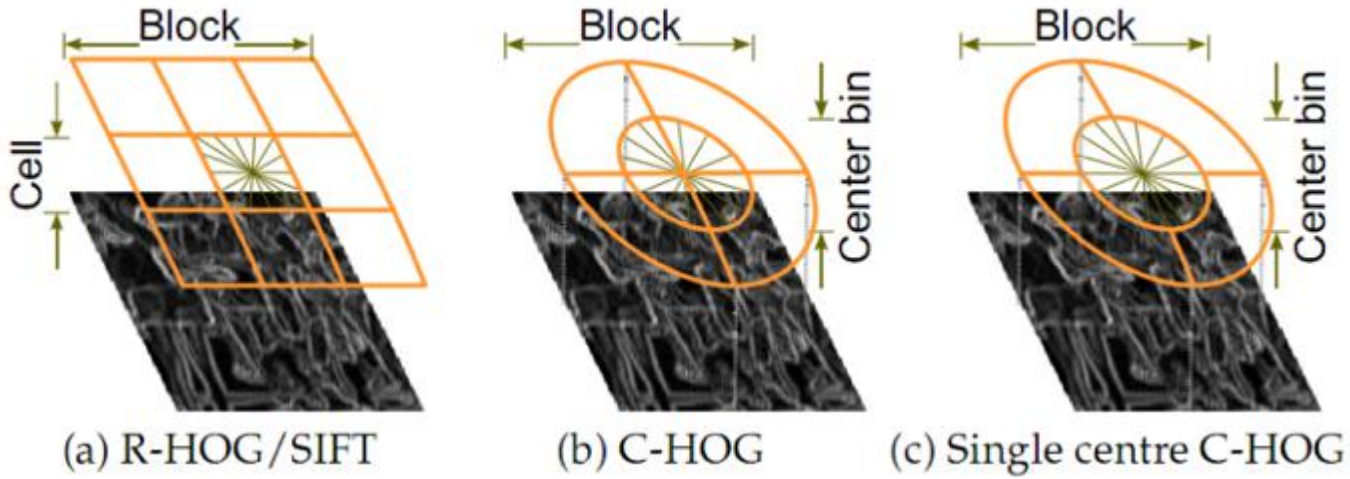
normalize

9x4 feature vector per cell

归一化可以是L2范式的，也可以是L1范式的。

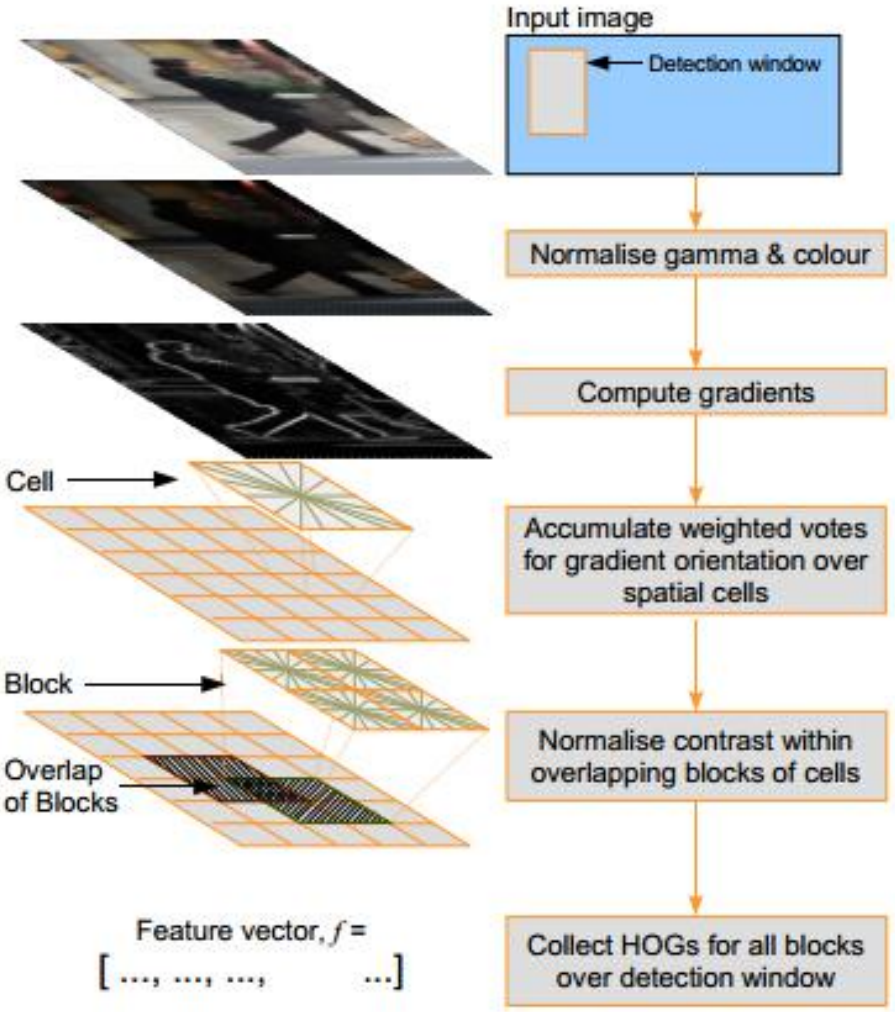
HOG：对比度归一化

- 对比归一化办法是：把各个细胞单元组合成大的、空间上连通的区间block，再归一。
- 这样，HOG描述符就变成了由各区间所有细胞单元的直方图成分所组成的一个向量。
- 这些区间是互有重叠的，这就意味着：每一个细胞单元的输出都多次作用于最终的描述器。

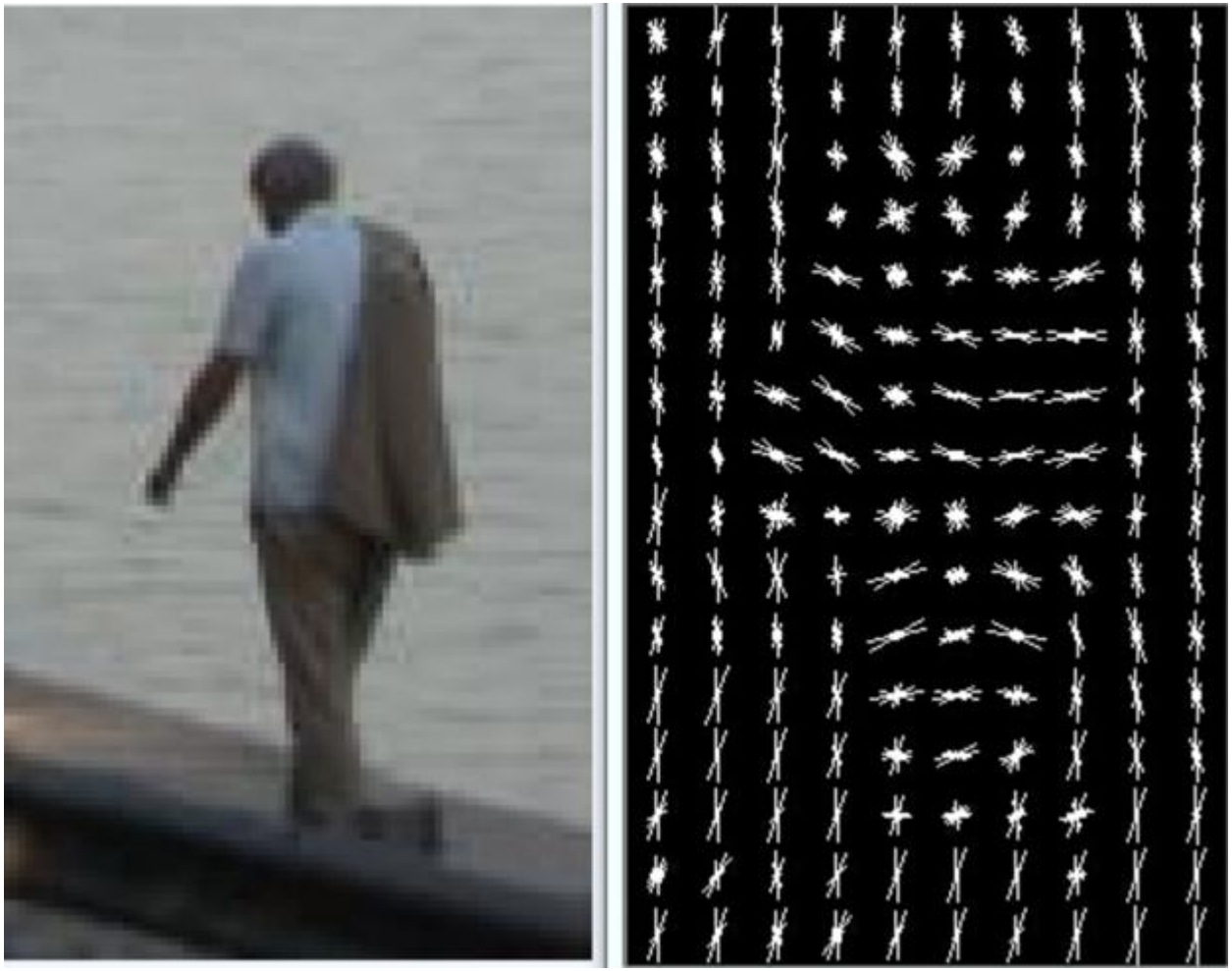


HOG的步骤

- 将整个图像进行Gamma空间、颜色颜色归一化
- 计算图像梯度
- 构建方向的直方图
- 将细胞单元组合成大的区间
- 收集HOG特征



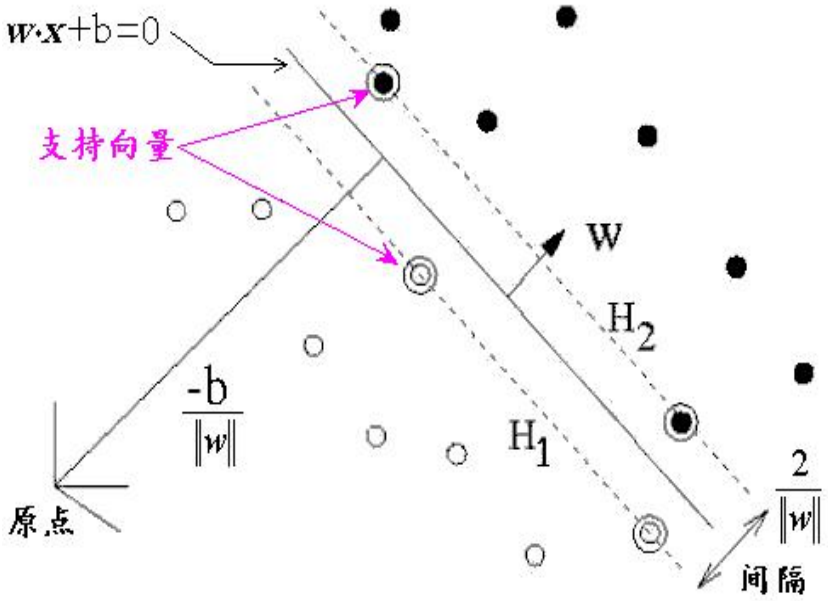
行人的HOG特征



人体轮廓内的HOG特征，与人体轮廓外的显著不同

SVM基本模型

- 支持向量机基本模型是针对线性可分情况下的最优分界面提出的。在这一条件下，正类和反类训练样本可用超平面完全正确地分开。



满足上述条件的分界面就是最优分界面，经过两类样本中距离最优分类面最近的点，且平行于最优分界面的超平面H1、H2（边界超平面）上的训练样本称为支持向量，即图中带圈的点。

带约束条件的极值问题

- 根据最近距离之和最大以及正确分离两类样本这两个条件，可以构造约束极值问题：见下式。

$$\min_{w,b} \frac{1}{2}\|w\|^2$$

$$st. \quad y_i(w \cdot x_i + b) \geq 1$$

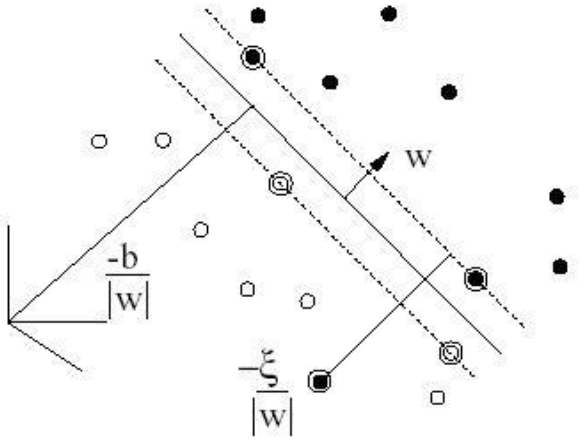
- 通过拉格朗日乘数法并引入拉格朗日乘数，该约束极值问题就可以转化成一个求解较为简单的对偶问题，通过寻求该对偶问题的最优解，就可以得到原问题的最优解。

松弛变量

- 由于噪声或其他因素的影响，两类数据可能有少数的融合或交叉。引入松弛变量x使得分类器在训练后仍可以存在一些错分样本，不但要使两类样本之间的间隔尽量大，同时还要使错分的样本的松弛变量之和尽可能的小，即

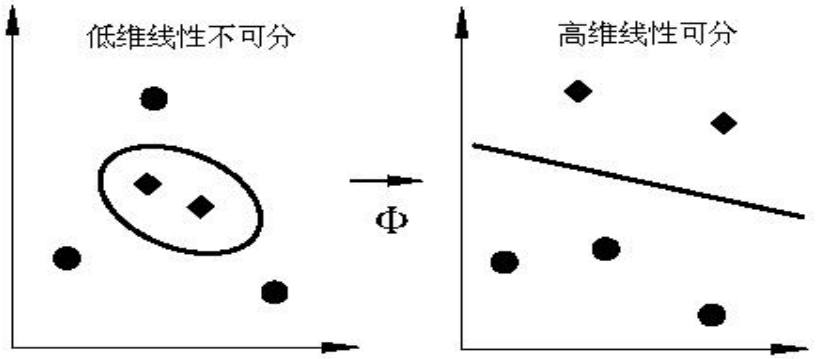
$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_i \xi_i \quad st. \quad y_i(w \cdot x_i + b) \geq 1 - \xi_i$$

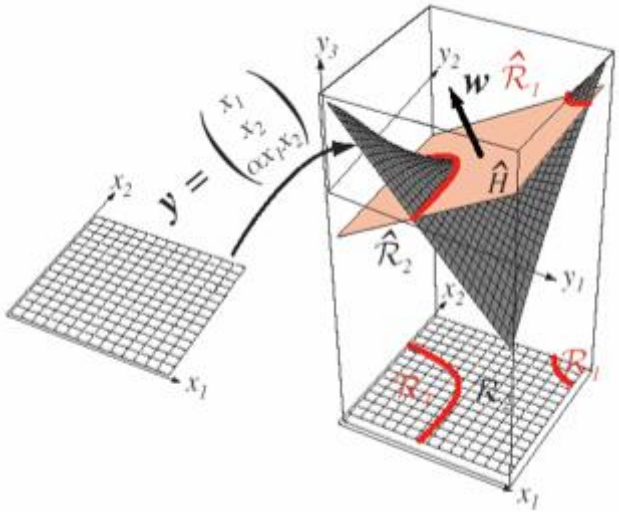
其中，x为松弛变量，满足xi≥0；C为大于零的折衷因子，它调和了间隔距离和错分样本数之间的关系。



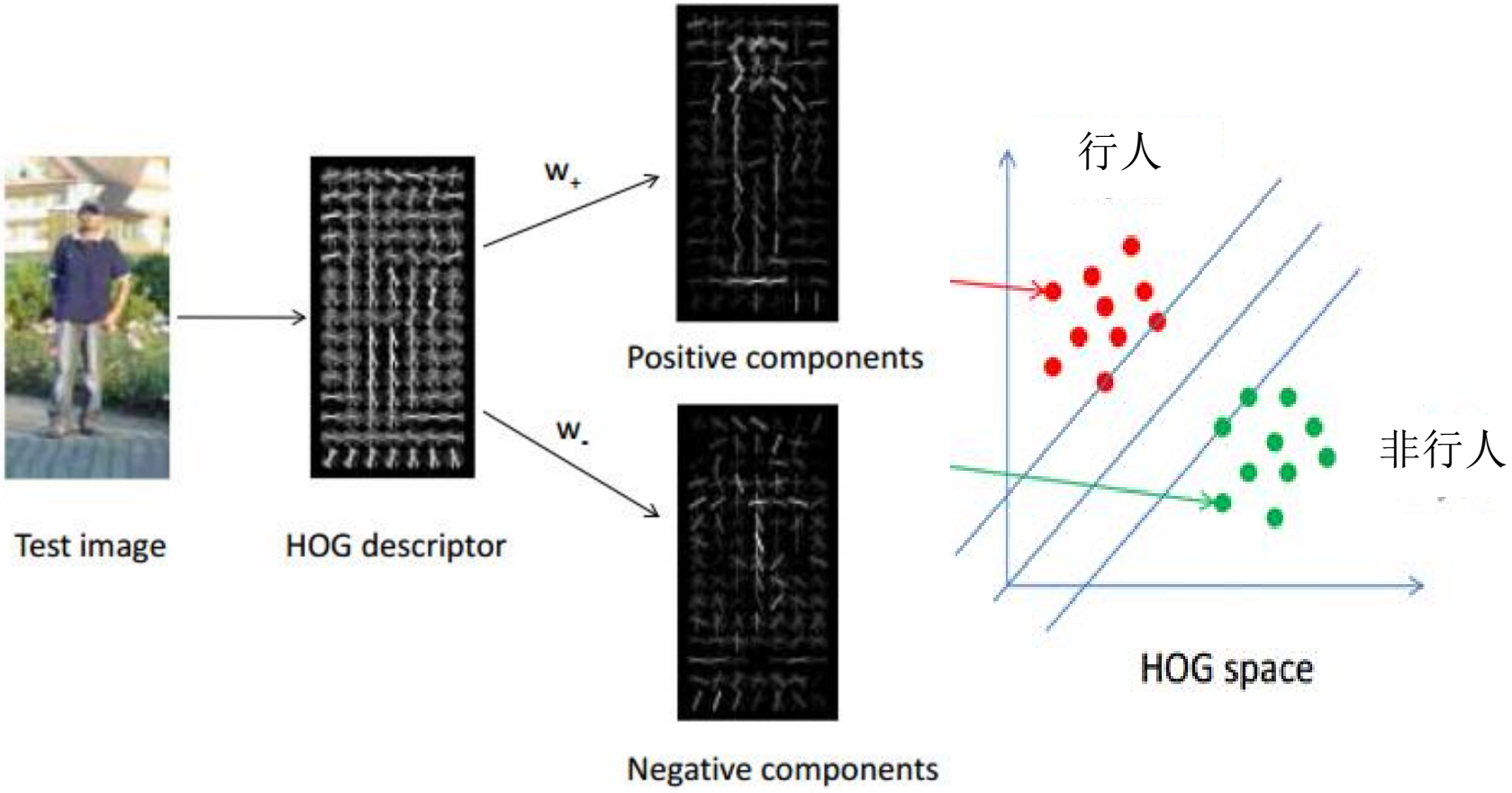
映射到高维空间

- 解决线性不可分数据问题的方法是将低维空间的线性不可分数据映射到高维的线性可分空间中。
- 支持向量机通过非线性映射$f(x)$把数据由低维空间向高维空间映射，在高维空间为低维数据构造线性分离超平面。
- 该分离超平面对应着原特征空间上的一个分割超曲面。





用SVM区分行人与非行人的HOG特征

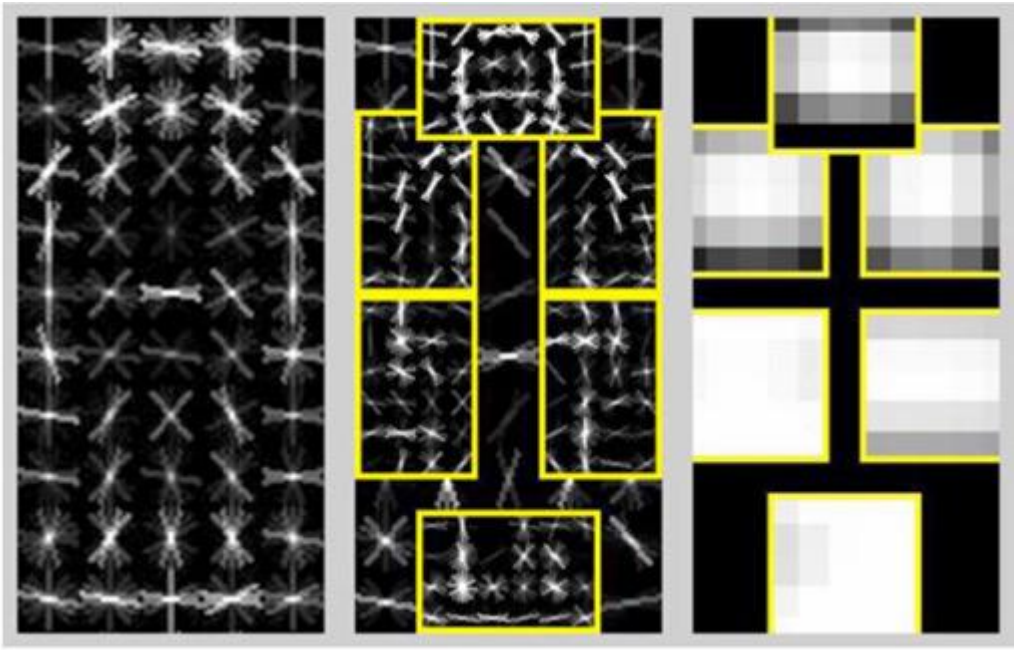


DPM

- Deformable Part Model（可变形的组件模型），是一种基于组件的检测算法。
- 该模型由Felzenszwalb提出，连续获得PASCAL VOC（Visual Object Class）07、08、09年的检测冠军。
- 为此， Felzenszwalb获得2010年PASCAL VOC的"终身成就奖"。

DPM

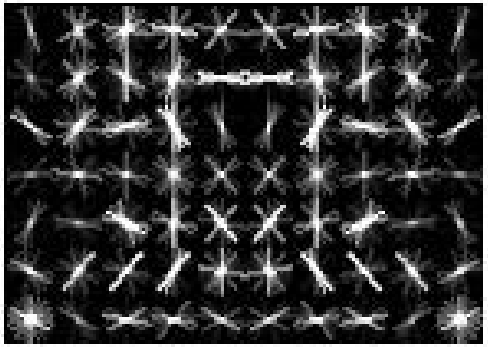
- 该模型包含了一个8*8分辨率的根滤波器（Root filter）（左）；
- 和4*4分辨率的组件滤波器（Part filter）（中）。
- 右图为其高斯滤波后的2倍空间模型。

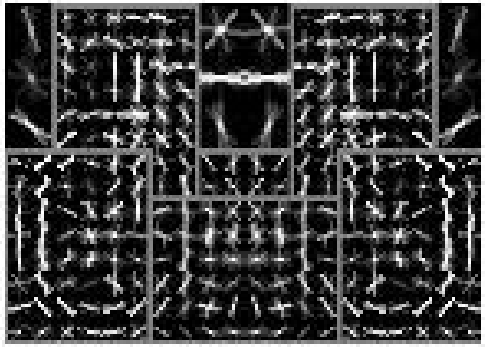


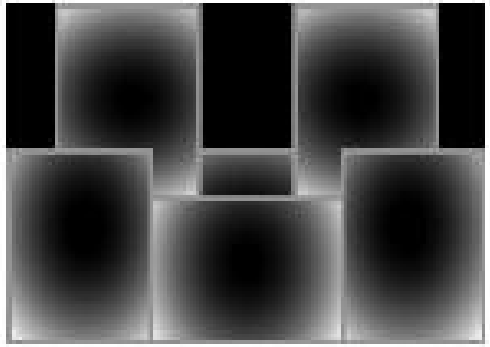
中间的分辨率是左图的两倍，梯度会更加精细。

自行车的DPM

- 左图：自行车整体模型的HOG；
- 中图：使用了6个子模型，白色矩形框出来的区域就是一个子模型；
- 右图：偏移Cost，圆圈中心就是子模型的理性位置，偏离的越远Cost越大。





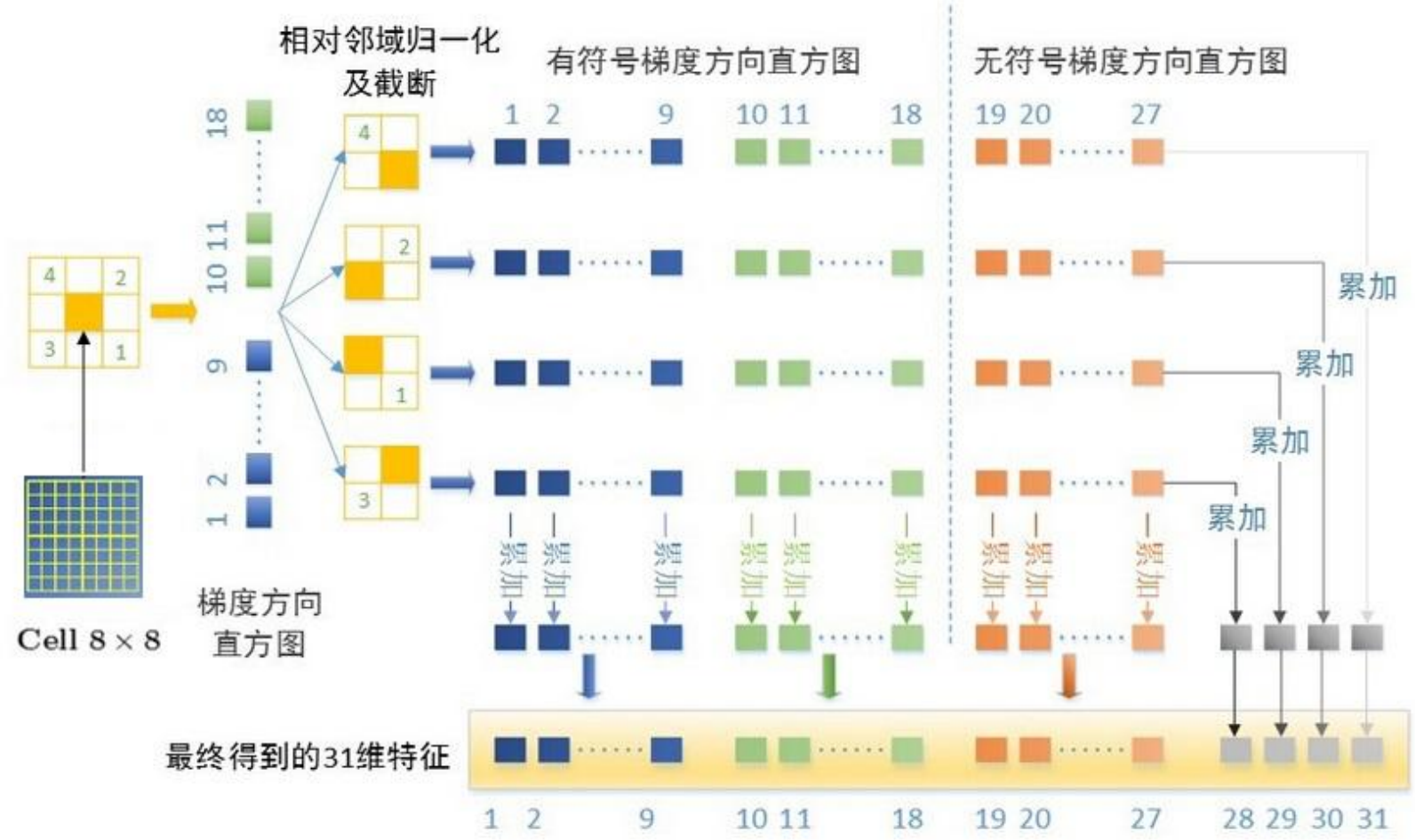


响应分数

$$\text{score}(x_0, y_0, l_0) = R_{0,l_0}(x_0, y_0) + \sum_{i=1}^{n} D_{i,l_0-\lambda}(2(x_0, y_0) + v_i) + b.$$

- x_0, y_0, l_0分别为锚点的横坐标，纵坐标，尺度。
- $R_{0,l_0}(x_0, y_0)$为根模型的响应分数
- $D_{i,l_0-\lambda}(2(x_0, y_0) + v_i)$为部件模型的响应分数
- b为不同模型组件之间的偏移系数，加上这个偏移量使其与跟模型进行对齐。
- $2(x_0, y_0)$表示组件模型的像素为原始的2倍

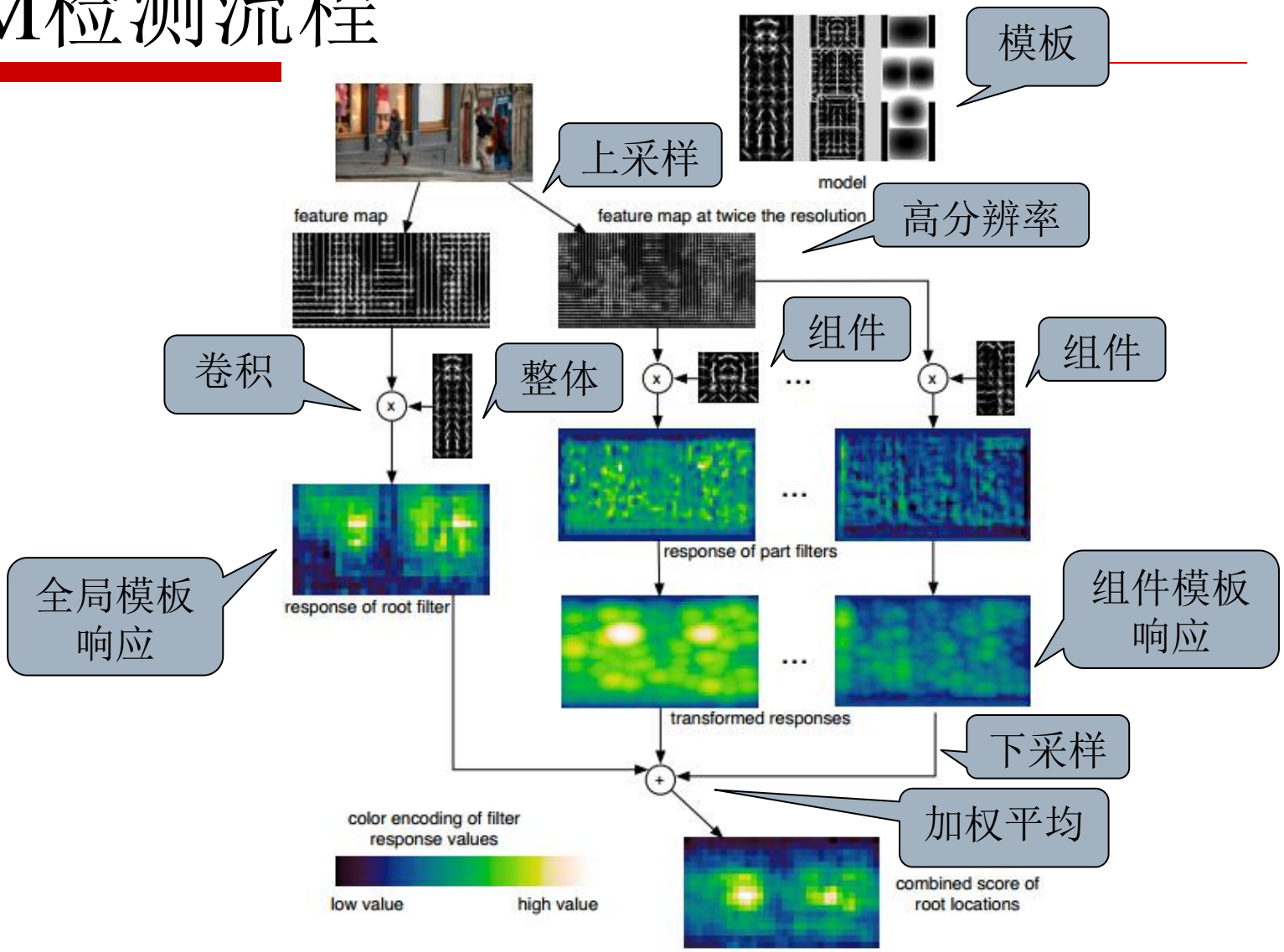
DPM的特征整理


相对邻域归一化
及截断
有符号梯度方向直方图
无符号梯度方向直方图
Cell 8 × 8
梯度方向
直方图
累加
累加
累加
累加
最终得到的31维特征
1 2 9 10 11 18 19 20 27 28 29 30 31

DPM检测流程

- 对于任意一张输入图像，提取其DPM特征图，然后将原始图像进行高斯金字塔上采样，然后提取其DPM特征图。
- 对于原始图像的DPM特征图和训练好的Root filter做卷积操作，从而得到Root filter的响应图。
- 对于2倍图像的DPM特征图，和训练好的Part filter做卷积操作，从而得到Part filter的响应图。
- 然后对其精细高斯金字塔的下采样操作，这样Root filter的响应图和Part filter的响应图就具有相同的分辨率了。
- 然后将其进行加权平均，得到最终的响应图。亮度越大表示响应值越大。

DPM检测流程


模板
model
上采样
feature map
feature map at twice the resolution
高分辨率
组件
组件
卷积
整体
response of part filters
全局模板
响应
response of root filter
组件模板
响应
transformed responses
下采样
加权平均
color encoding of filter
response values
low value
high value
combined score of
root locations

Multiple-instance SVM

- 思想是将标准SVM的最大化样本间距扩展为最大化样本集间距。
- 具体来说是选取正样本集中最像正样本的样本用作训练。
- 同样取负样本中离分界面最近的负样本作为负样本。因为目标是要保证正样本中有正，负样本不能为正。

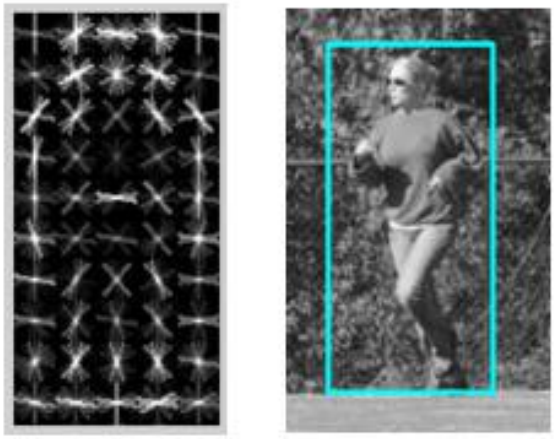
$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_I \xi_I$$

$$\text{s.t.} \quad \forall I: \; Y_I \max_{i\in I}(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 - \xi_I, \; \xi_I \geq 0.$$

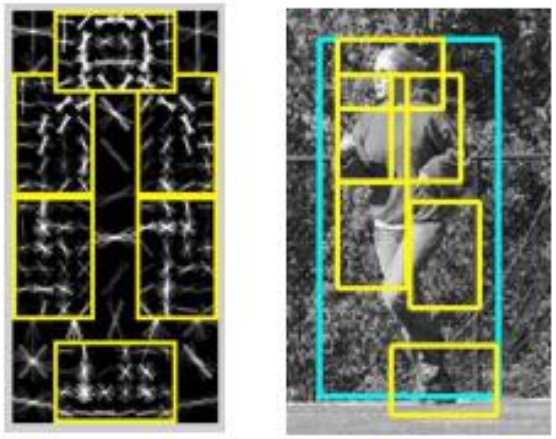
Latent SVM

- Latent-SVM实质上和MI-SVM是一样的。区别在于扩展了Latent变量。
- 首先解释下Latent变量，MI-SVM决定正样本集中哪一个样本作为正样本的就是一个latent变量。不过这个变量的取值只是正样本集中的序号而已。
- DPM中也是要选择最大的正样本，但是它的latent变量就很多。比如bounding box的实际位置，在HOG特征金字塔中的level，某样本属于哪一类component。
- 也就是说有了一张正样本的图片，标注了bounding box，我们要在某一位置，某一尺度，提取出一个最大正样本作为某一component的正样本。

Latent SVM



HOG & Linear SVM



Deformable Parts & Latent SVM

$$f_w(x) = w \cdot \Phi(x)$$

$$w = F_0$$

$$\Phi(x) = \phi(H(x), p_0)$$

$$w^* = \arg\min_w \lambda \|w\|^2 + \sum_{i=1}^{n} \max(0, 1 - y_i f_w(x_i))$$

$$f_w(x) = \max_{z \in Z(x)} w \cdot \Phi(x, z)$$

$$w = (F_0, \ldots, F_n, a_1, b_1, \ldots, a_n, b_n)$$

$$\Phi(x, z) = (\phi(H(x), p_0), \phi(H(x), p_1), \ldots, \phi(H(x), p_n), \tilde{x}_1, \tilde{y}_1, \tilde{x}_1^2, \tilde{y}_1^2, \ldots, \tilde{x}_n, \tilde{y}_n, \tilde{x}_n^2, \tilde{y}_n^2)$$

$$w^* = \arg\min_w \lambda \|w\|^2 + \sum_{i=1}^{n} \max(0, 1 - y_i f_w(x_i))$$

参考资料

- https://blog.csdn.net/dream_bin123/article/details/78633103
- http://www.360doc.com/content/18/0705/11/57490477_767879308.shtml
- https://blog.csdn.net/carson2005/article/details/22499565
- https://blog.csdn.net/qq_14845119/article/details/52625426
- https://www.xuebuyuan.com/2079028.html

试验链接

- https://github.com/Alarnti/dpm
- https://github.com/BUPTLdy/human-detector

问答互动

在所报课的课程页面，

1、点击“全部问题”显示本课程所有学员提问的问题。

2、点击“提问”即可向该课程的老师和助教提问问题。



联系我们

小象学院：互联网新技术在线教育领航者

– 微信公众号：**小象学院**

