

# 20180426图像特征提取三大法宝

原文：<http://www.open-open.com/lib/view/open1440832074794.html>

## (一) HOG特征

### 1、HOG特征：

方向梯度直方图 ( Histogram of Oriented Gradient, HOG ) 特征是一种在计算机视觉和图像处理中用来进行物体检测的特征描述子。它通过计算和统计图像局部区域的梯度方向直方图来构成特征。Hog特征结合 SVM分类器已经被广泛应用于图像识别中，尤其在行人检测中获得了极大的成功。需要提醒的是，HOG+SVM进行行人检测的方法是法国研究人员Dalal 在2005的CVPR上提出的，而如今虽然有很多行人检测算法不断提出，但基本都是以HOG+SVM的思路为主。

#### (1) 主要思想：

在一副图像中，局部目标的表象和形状 ( appearance and shape ) 能够被梯度或边缘的方向密度分布很好地描述。( 本质：梯度的统计信息，而梯度主要存在于边缘的地方 )。

#### (2) 具体的实现方法是：

首先将图像分成小的连通区域，我们把它叫细胞单元。然后采集细胞单元中各像素点的梯度的或边缘的方向直方图。最后把这些直方图组合起来就可以构成特征描述器。

#### (3) 提高性能：

把这些局部直方图在图像的更大的范围内 ( 我们把它叫区间或block ) 进行对比度归一化 ( contrast-normalized )，所采用的方法是：先计算各直方图在这个区间 ( block ) 中的密度，然后根据这个密度对区间中的各个细胞单元做归一化。通过这个归一化后，能对光照变化和阴影获得更好的效果。

#### (4) 优点：

与其他的特征描述方法相比，HOG有很多优点。首先，由于HOG是在图像的局部方格单元上操作，所以它对图像几何的和光学的形变都能保持很好的不变性，这两种形变只会出现在更大的空间领域上。其次，在粗的空域抽样、精细的方向抽样以及较强的局部光学归一化等条件下，只要行人大体上能够保持直立的姿势，可以容许行人有一些细微的肢体动作，这些细微的动作可以被忽略而不影响检测效果。因此HOG特征是特别适合于做图像中的人体检测的。

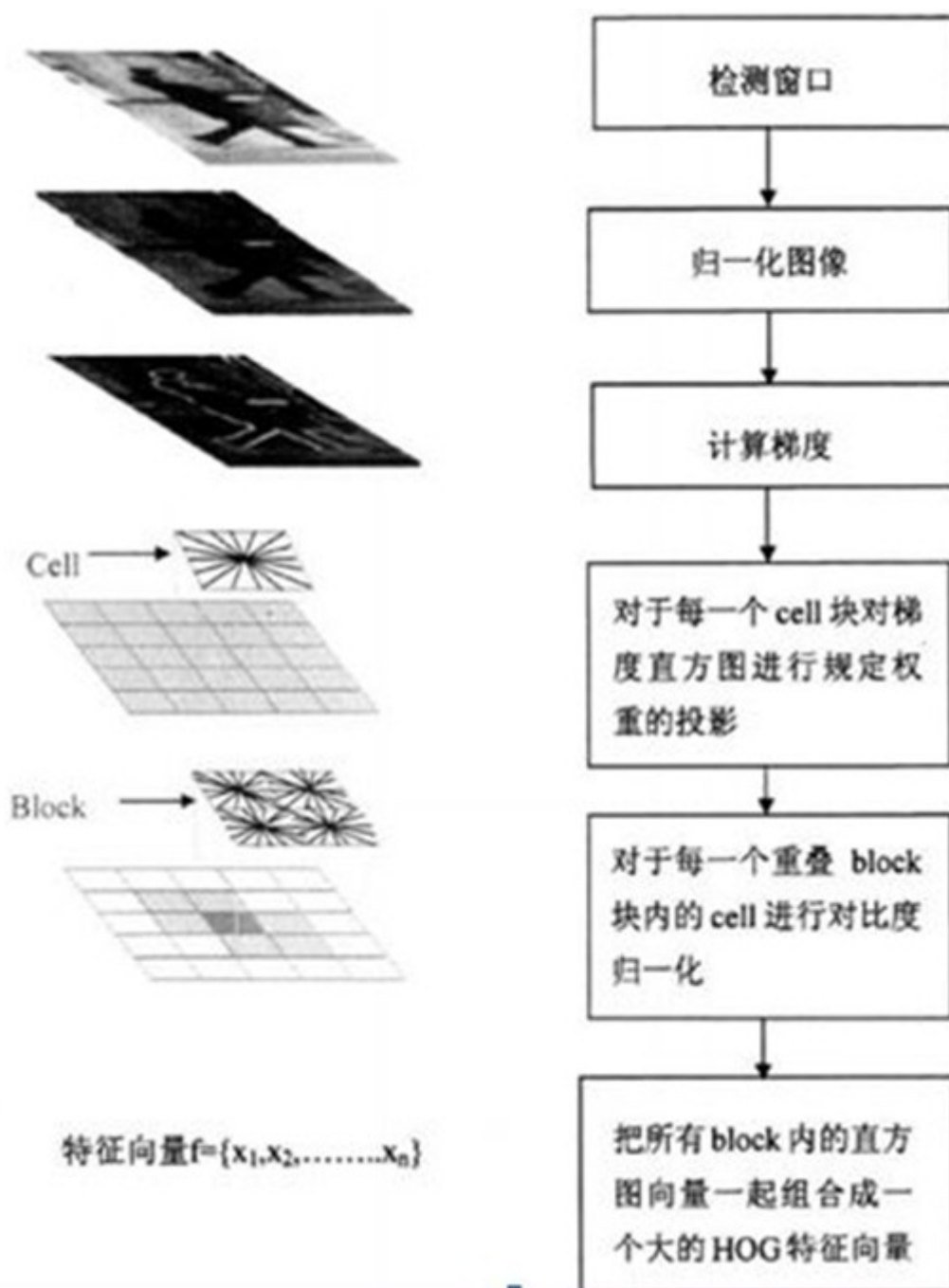
### 2、HOG特征提取算法的实现过程：

#### 大概过程：

HOG特征提取方法就是将一个image ( 你要检测的目标或者扫描窗口 )：

1) 灰度化 ( 将图像看做一个x,y,z ( 灰度 ) 的三维图像 )；

- 2) 采用Gamma校正法对输入图像进行颜色空间的标准化（归一化）；目的是调节图像的对比度，降低图像局部的阴影和光照变化所造成的影响，同时可以抑制噪音的干扰；
- 3) 计算图像每个像素的梯度（包括大小和方向）；主要是为了捕获轮廓信息，同时进一步弱化光照的干扰。
- 4) 将图像划分成小cells（例如6\*6像素/cell）；
- 5) 统计每个cell的梯度直方图（不同梯度的个数），即可形成每个cell的descriptor；
- 6) 将每几个cell组成一个block（例如3\*3个cell/block），一个block内所有cell的特征descriptor串联起来便得到该block的HOG特征descriptor。
- 7) 将图像image内的所有block的HOG特征descriptor串联起来就可以得到该image（你要检测的目标）的HOG特征descriptor了。这个就是最终的可供分类使用的特征向量了。



具体每一步的详细过程如下：

## (1) 标准化gamma空间和颜色空间

为了减少光照因素的影响，首先需要将整个图像进行规范化（归一化）。在图像的纹理强度中，局部的表层曝光贡献的比重较大，所以，这种压缩处理能够有效地降低图像局部的阴影和光照变化。因为颜色信息作用不大，通常先转化为灰度图；

Gamma压缩公式：

$$I(x, y) = I(x, y)^{\text{gamma}}$$

比如可以取Gamma=1/2；

## (2) 计算图像梯度

计算图像横坐标和纵坐标方向的梯度，并据此计算每个像素位置的梯度方向值；求导操作不仅能够捕获轮廓，人影和一些纹理信息，还能进一步弱化光照的影响。

图像中像素点(x,y)的梯度为：

$$G_x(x, y) = H(x+1, y) - H(x-1, y)$$

$$G_y(x, y) = H(x, y+1) - H(x, y-1)$$

式中  $G_x(x, y)$ ,  $G_y(x, y)$ ,  $H(x, y)$  分别表示输入图像中像素点(x,y)处的水平方向梯度、垂直方向梯度和像素值。像素点(x,y)处的梯度幅值和梯度方向分别为：

$$G(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2}$$

$$\alpha(x, y) = \tan^{-1}\left(\frac{G_y(x, y)}{G_x(x, y)}\right)$$

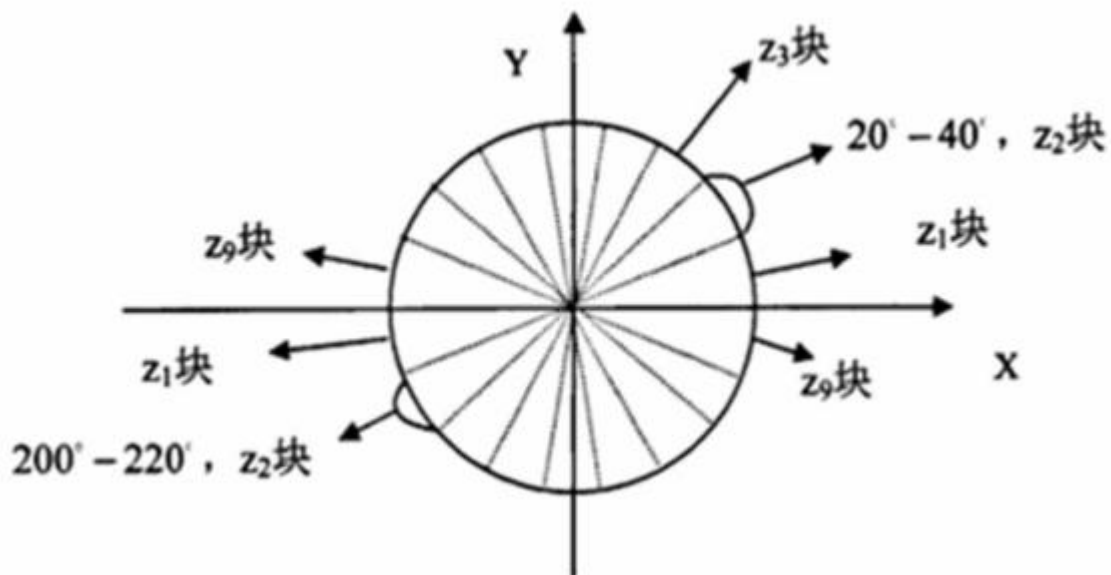
最常用的方法是：首先用[-1,0,1]梯度算子对原图像做卷积运算，得到x方向（水平方向，以向右为正方向）的梯度分量gradscalx，然后用[1,0,-1] T 梯度算子对原图像做卷积运算，得到y方向（竖直方向，以向上为正方向）的梯度分量gradscaly。然后再用以上公式计算该像素点的梯度大小和方向。

## (3) 为每个细胞单元构建梯度方向直方图

第三步的目的是为局部图像区域提供一个编码，同时能够保持对图像中人体对象的姿势和外观的弱敏感性。

我们将图像分成若干个“单元格cell”，例如每个cell为66个像素。假设我们采用9个bin的直方图来统计这66个像素的梯度信息。也就是将cell的梯度方向360度分成9个方向块，如图所示：例如：如果这个像素的梯度方向是20-40度，直方图第2个bin的计数就加一，这样，对 cell内每个像素用梯度方向在直方图中进行加权投影（映射到固定的角度范围），就可以得到这个cell的梯度方向直方图了，就是该cell对应的9维特征向量（因为有9个bin）。

像素梯度方向用到了，那么梯度大小呢？梯度大小就是作为投影的权值的。例如说：这个像素的梯度方向是20-40度，然后它的梯度大小是2（假设啊），那么直方图第2个bin的计数就不是加一了，而是加二（假设啊）。

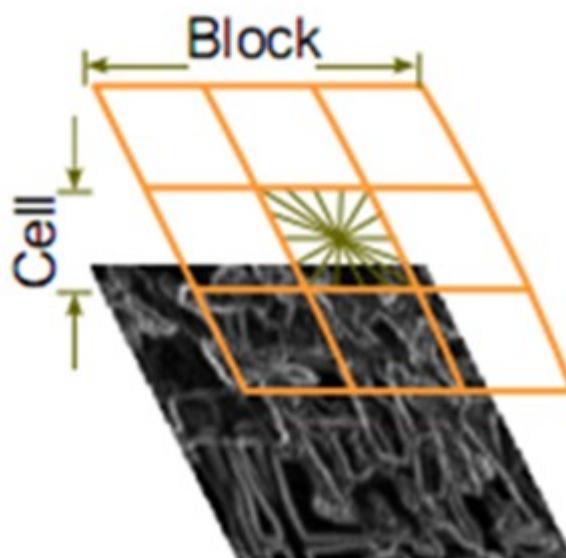


细胞单元可以是矩形的 ( rectangular ) , 也可以是星形的 ( radial ) 。

#### ( 4 ) 把细胞单元组合成大的块 ( block ) , 块内归一化梯度直方图

由于局部光照的变化以及前景-背景对比度的变化, 使得梯度强度的变化范围非常大。这就需要对梯度强度做归一化。归一化能够进一步地对光照、阴影和边缘进行压缩。

作者采取的办法是: 把各个细胞单元组合成大的、空间上连通的区间 ( blocks ) 。这样, 一个block内所有cell的特征向量串联起来便得到 该block的HOG特征。这些区间是互有重叠的, 这就意味着: 每一个单元格的特征会以不同的结果多次出现在最后的特征向量中。我们将归一化之后的块描述符 ( 向量 ) 就称之为HOG描述符。



区间有两个主要的几何形状——矩形区间 ( R-HOG ) 和环形区间 ( C-HOG ) 。R-HOG区间大体上是一些方形的格子, 它可以有三个参数来表征: 每个区间中细胞单元的数目、每个细胞单元中像素点的数目、每个细胞的直方图通道数目。

例如: 行人检测的最佳参数设置是: 3×3细胞/区间、6×6像素/细胞、9个直方图通道。则一块的特征数为: 339 ;

#### ( 5 ) 收集HOG特征

最后一步就是将检测窗口中所有重叠的块进行HOG特征的收集，并将它们结合成最终的特征向量供分类使用。

### (6) 那么一个图像的HOG特征维数是多少呢？

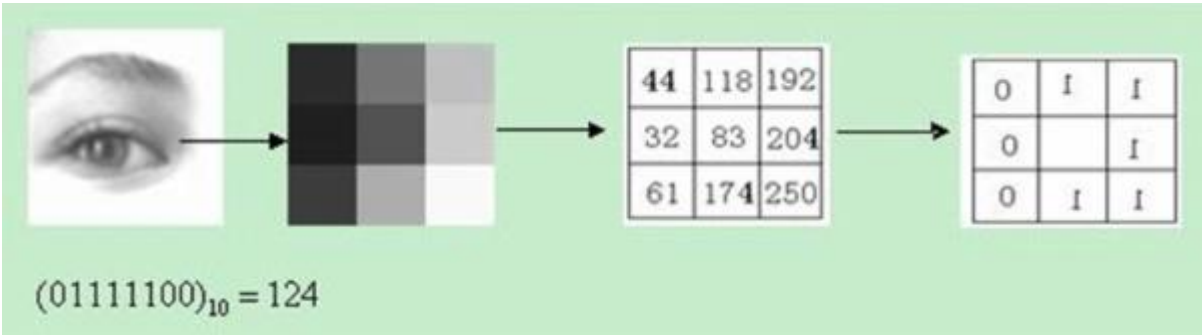
顺便做个总结：Dalal提出的Hog特征提取的过程：把样本图像分割为若干个像素的单元（cell），把梯度方向平均划分为9个区间（bin），在每个单元里面对所有像素的梯度方向在各个方向区间进行直方图统计，得到一个9维的特征向量，每相邻的4个单元构成一个块（block），把一个块内的特征向量联起来得到36维的特征向量，用块对样本图像进行扫描，扫描步长为一个单元。最后将所有块的特征串联起来，就得到了人体的特征。例如，对于64128的图像而言，每1616的像素组成一个cell，每22个cell组成一个块，因为每个cell有9个特征，所以每个块内有49=36个特征，以8个像素为步长，那么，水平方向将有7个扫描窗口，垂直方向将有15个扫描窗口。也就是说，64128的图片，总共有367\*15=3780个特征。

## (二) LBP特征

LBP（Local Binary Pattern，局部二值模式）是一种用来描述图像局部纹理特征的算子；它具有旋转不变性和灰度不变性等显著的优点。它是首先由T. Ojala, M.Pietikäinen, 和D. Harwood 在1994年提出，用于纹理特征提取。而且，提取的特征是图像的局部的纹理特征；

### 1、LBP特征的描述

原始的LBP算子定义为在33的窗口内，以窗口中心像素为阈值，将相邻的8个像素的灰度值与其进行比较，若周围像素值大于中心像素值，则该像素点的位置被标记为1，否则为0。这样，33邻域内的8个点经比较可产生8位二进制数（通常转换为十进制数即LBP码，共256种），即得到该窗口中心像素点的LBP值，并用这个值来反映该区域的纹理信息。如下图所示：



### LBP的改进版本：

原始的LBP提出后，研究人员不断对其提出了各种改进和优化。

### (1) 圆形LBP算子：

基本的 LBP算子的最大缺陷在于它只覆盖了一个固定半径范围内的小区域，这显然不能满足不同尺寸和频率纹理的需要。为了适应不同尺度的纹理特征，并达到灰度和旋 转不变性的要求，Ojala等对 LBP 算子进行了改进，将 3×3 邻域扩展到任意邻域，并用圆形邻域代替了正方形邻域，改进后的 LBP 算子允许在半径为 R 的圆形邻域内有任意多个像素点。从而得到了诸如半径为R的圆形区域内含有P个采样点的LBP算子；

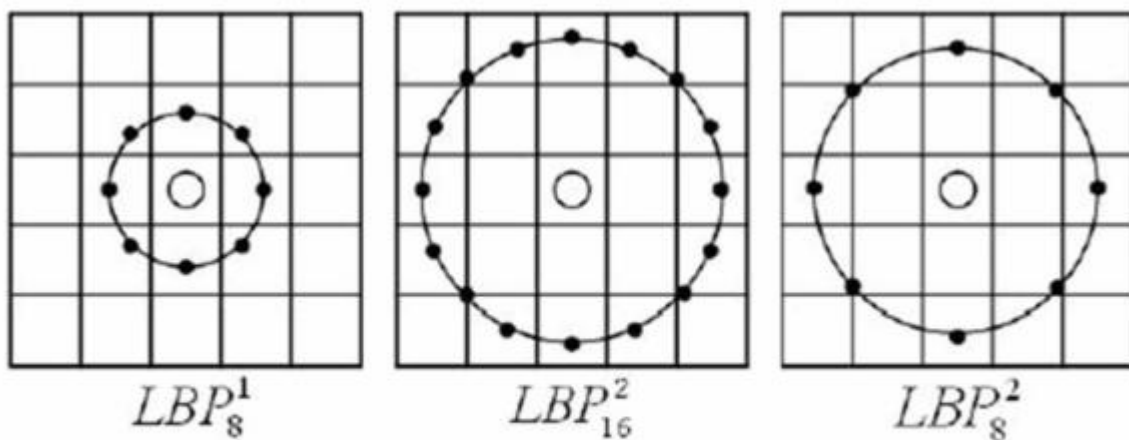


图2.2 几种LBP算子

## (2) LBP旋转不变模式

从 LBP 的定义可以看出，LBP 算子是灰度不变的，但却不是旋转不变的。图像的旋转就会得到不同的 LBP 值。

Maenpaa等人又将 LBP算子进行了扩展，提出了具有旋转不变性的 LBP 算子，即不断旋转圆形邻域得到一系列初始定义的 LBP值，取其最小值作为该邻域的 LBP 值。

图 2.5 给出了求取旋转不变的 LBP 的过程示意图，图中算子下方的数字表示该算子对应的 LBP 值，图中所示的 8 种 LBP 模式，经过旋转不变的处理，最终得到的具有旋转不变性的 LBP 值为 15。也就是说，图中的 8 种 LBP 模式对应的旋转不变的 LBP 模式都是 00001111。

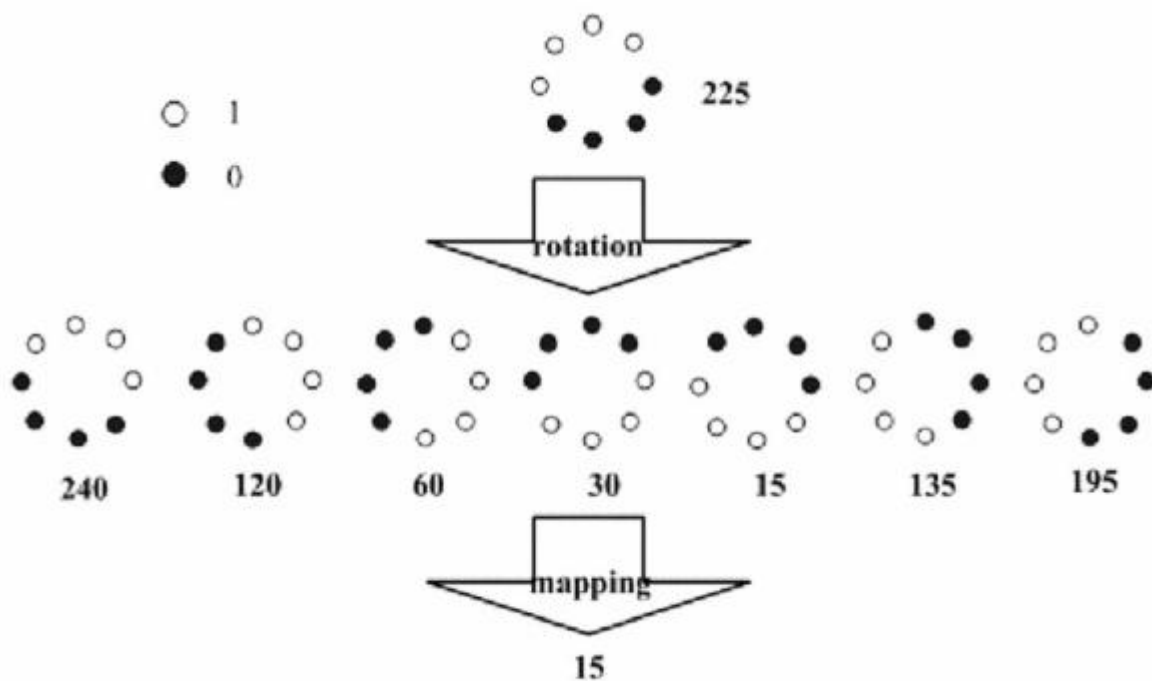


图 2.5 旋转不变的 LBP 示意

## (3) LBP等价模式



一个LBP算子可以产生不同的二进制模式，对于半径为R的圆形区域内含有P个采样点的LBP算子将会产生 $2^P$ 种模式。很显然，随着邻域集内采样点数的增加，二进制模式的种类是急剧增加的。例如：5×5邻域内20个采样点，有 $2^{20} = 1,048,576$ 种二进制模式。如此多的二值模式无论对于纹理的提取还是对于纹理的识别、分类及信息的存取都是不利的。同时，过多的模式种类对于纹理的表达是不利的。例如，将LBP算子用于纹理分类或人脸识别时，常采用LBP模式的统计直方图来表达图像的信息，而较多的模式种类将使得数据量过大，且直方图过于稀疏。因此，需要对原始的LBP模式进行降维，使得数据量减少的情况下能最好的代表图像的信息。

为了解决二进制模式过多的问题，提高统计性，Ojala提出了采用一种“等价模式”（Uniform Pattern）来对LBP算子的模式种类进行降维。Ojala等认为，在实际图像中，绝大多数LBP模式最多只包含两次从1到0或从0到1的跳变。因此，Ojala将“等价模式”定义为：当某个LBP所对应的循环二进制数从0到1或从1到0最多有两次跳变时，该LBP所对应的二进制就称为一个等价模式类。如00000000（0次跳变），00000111（只含一次从0到1的跳变），10001111（先由1跳到0，再由0跳到1，共两次跳变）都是等价模式类。除等价模式类以外的模式都归为另一类，称为混合模式类，例如10010111（共四次跳变）（这是我的个人理解，不知道对不对）。

通过这样的改进，二进制模式的种类大大减少，而不会丢失任何信息。模式数量由原来的 $2^P$ 种减少为 $P(P-1)+2$ 种，其中P表示邻域集内的采样点数。对于3×3邻域内8个采样点来说，二进制模式由原始的256种减少为58种，这使得特征向量的维数更少，并且可以减少高频噪声带来的影响。

## 2、LBP特征用于检测的原理

显而易见的是，上述提取的LBP算子在每个像素点都可以得到一个LBP“编码”，那么，对一幅图像（记录的是每个像素点的灰度值）提取其原始的LBP算子之后，得到的原始LBP特征依然是“一幅图片”（记录的是每个像素点的LBP值）。



(a) 原始图像      (b) LBP图谱

LBP的应用中，如纹理分类、人脸分析等，一般都不将LBP图谱作为特征向量用于分类识别，而是采用LBP特征谱的统计直方图作为特征向量用于分类识别。

因为，从上面的分析我们可以看出，这个“特征”跟位置信息是紧密相关的。直接对两幅图片提取这种“特征”，并进行判别分析的话，会因为“位置没有对准”而产生很大的误差。后来，研究人员发现，可以将一幅图片划分为若干的子区域，对每个子区域内的每个像素点都提取LBP特征，然后，在每个子区域内建立LBP特征的统计直方图。如此一来，每个子区域，就可以用一个统计直方图来进行描述；整个图片就由若干个统计直方图组成；

例如：一幅100100像素大小的图片，划分为1010=100个子区域（可以通过多种方式来划分区域），每个子区域的大小为1010像素；在每个子区域内的每个像素点，提取其LBP特征，然后，建立统计直方图；这样，这幅图片就有1010个子区域，也就有了1010个统计直方图，利用这1010个统计直方图，就可以描述这幅图片了。之后，我们利用各种相似性度量函数，就可以判断两幅图像之间的相似性了；

### 3、对LBP特征向量进行提取的步骤

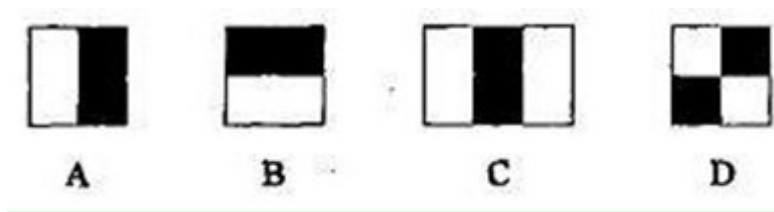
- (1) 首先将检测窗口划分为 $16 \times 16$ 的小区域 ( cell ) ；
  - (2) 对于每个cell中的一个像素，将相邻的8个像素的灰度值与其进行比较，若周围像素值大于中心像素值，则该像素点的位置被标记为1，否则为0。这样， $3 \times 3$ 邻域内的8个点经比较可产生8位二进制数，即得到该窗口中心像素点的LBP值；
  - (3) 然后计算每个cell的直方图，即每个数字（假定是十进制数LBP值）出现的频率；然后对该直方图进行归一化处理。
  - (4) 最后将得到的每个cell的统计直方图进行连接成为一个特征向量，也就是整幅图的LBP纹理特征向量；
- 然后便可利用SVM或者其他机器学习算法进行分类了。

## (三) Haar特征

### 1、Haar-like特征

Haar-like特征最早是由Papageorgiou等应用于人脸表示，Viola和Jones在此基础上，使用3种类型4种形式的特征。

Haar特征分为三类：边缘特征、线性特征、中心特征和对角线特征，组合成特征模板。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像素和减去黑色矩形像素和。Haar特征值反映了图像的灰度变化情况。例如：脸部的一些特征能由矩形特征简单的描述，如：眼睛要比脸颊颜色要深，鼻梁两侧比鼻梁颜色要深，嘴巴比周围颜色要深等。但矩形特征只对一些简单的图形结构，如边缘、线段较敏感，所以只能描述特定走向（水平、垂直、对角）的结构。



对于图中的A, B和D这类特征，特征数值计算公式为： $v = \text{Sum}_{\text{白}} - \text{Sum}_{\text{黑}}$ ，而对于C来说，计算公式如下： $v = \text{Sum}_{\text{白}} - 2 * \text{Sum}_{\text{黑}}$ ；之所以将黑色区域像素和乘以2，是为了使两种矩形区域中像素数目一致。

通过改变特征模板的大小和位置，可在图像子窗口中穷举出大量的特征。上图的特征模板称为“特征原型”；特征原型在图像子窗口中扩展（平移伸缩）得到的特征称为“矩形特征”；矩形特征的值称为“特征值”。

矩形特征可位于图像任意位置，大小也可以任意改变，所以矩形特征值是矩形模版类别、矩形位置和矩形大小这三个因素的函数。故类别、大小和位置的变化，使得很小的检测窗口含有非常多的矩形特征，如：在 $24 \times 24$ 像素大小的检测窗口内矩形特征数量可以达到16万个。这样就有两个问题需要解决了：（1）如何快速计算那么多的特征？—积分图大显神通；（2）哪些矩形特征才是对分类器分类最有效的？—如通过AdaBoost算法来训练（这一块这里不讨论，具体见 <http://blog.csdn.net/zouxy09/article/details/7922923>）

### 2、Haar-like特征的计算—积分图

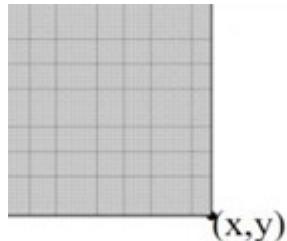
积分图就是只遍历一次图像就可以求出图像中所有区域像素和的快速算法，大大的提高了图像特征值计算的效率。

积分图主要的思想是将图像从起点开始到各个点所形成的矩形区域像素之和作为一个数组的元素保存在内存中，当要计算某个区域的像素和时可以直接索引数组的元素，不用重新计算这个区域的像素和，从而加快了计算（这有个相应的称呼，叫做动态规划算法）。积分图能够在多种尺度下，使用相同的时间（常数时间）来计算不同的特征，因此大大提高了检测速度。



我们来看看它是怎么做到的。

积分图是一种能够描述全局信息的矩阵表示方法。积分图的构造方式是位置  $(i, j)$  处的值  $ii(i, j)$  是原图像  $(i, j)$  左上角方向所有像素的和：

$$ii(i, j) = \sum_{k \leq i, l \leq j} f(k, l)$$


### 积分图构建算法：

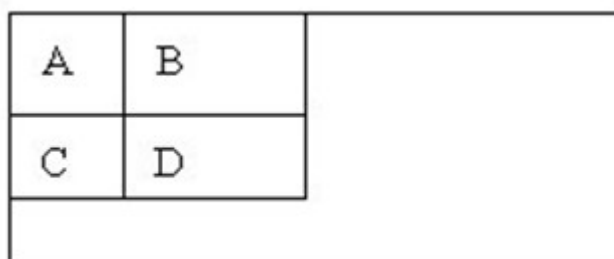
- 1) 用  $s(i, j)$  表示行方向的累加和，初始化  $s(i, -1) = 0$ ;
- 2) 用  $ii(i, j)$  表示一个积分图像，初始化  $ii(-1, i) = 0$ ;
- 3) 逐行扫描图像，递归计算每个像素  $(i, j)$  行方向的累加和  $s(i, j)$  和积分图像  $ii(i, j)$  的值

$$s(i, j) = s(i, j-1) + f(i, j)$$

$$ii(i, j) = ii(i-1, j) + s(i, j)$$

- 4) 扫描图像一遍，当到达图像右下角像素时，积分图像  $ii$  就构造好了。

积分图构造好之后，图像中任何矩阵区域的像素累加和都可以通过简单运算得到如图所示。




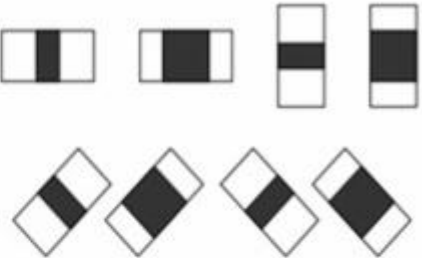


设  $D$  的四个顶点分别为  $\alpha$ 、 $\beta$ 、 $\gamma$ 、 $\delta$ ，则  $D$  的像素和可以表示为

$$D_{\text{sum}} = ii(\alpha) + ii(\beta) - ii(\gamma) - ii(\delta);$$

而 Haar-like 特征值无非就是两个矩阵像素和的差，同样可以在常数时间内完成。所以矩形特征的特征值计算，只与此特征矩形的端点的积分图有关，所以不管此特征矩形的尺度变换如何，特征值的计算所消耗的时间都是常量。这样只要遍历图像一次，就可以求得所有子窗口的特征值。

### 3、Haar-like 矩形特征拓展

Lienhart R. 等对 Haar-like 矩形特征库作了进一步扩展，加入了旋转 45° 角的矩形特征。扩展后的特征大致分为 4 种类型：边缘特征、线特征环、中心环绕特征和对角线特征：

	
1.边缘特征(Edge features)	2、线性特征(Line features)
	
3、圆心环绕特征 (Center-surround features)	4、特定方向的特征 (Special diagonal line) feature used

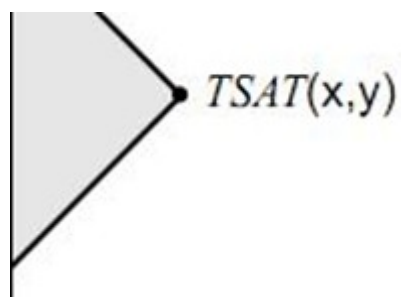
在特征值的计算过程中，黑色区域的权值为负值，白色区域的权值为正值。而且权值与矩形面积成反比（使两种矩形区域中像素数目一致）；

### 竖直矩阵特征值计算：

对于竖直矩阵，与上面2处说的一样。

### 45°旋角的矩形特征计算：

对于45°旋角的矩形，我们定义 $RSAT(x,y)$ 为点 $(x,y)$ 左上角45°区域和左下角45°区域的像素和。



用公式可以表示为：

$$RSAT(x,y) = \sum_{x' \leq x, x' \leq x - |y-y'|} I(x',y') .$$

为了节约时间，减少重复计算，可按如下递推公式计算：

$$RSAT(x,y) = RSAT(x-1,y-1) + RSAT(x-1,y) + I(x,y) - RSAT(x-2,y-1)$$

而计算矩阵特征的特征值，是位于十字行矩形 $RSAT(x,y)$ 之差。可参考下图：

