> OpenCV 级联分类器训练

一、正样本准备

正样本的选取原则

1. 正样本的尺寸不是必须一致的，但是要和生成的正样本矢量文件中的宽高有相同的比例（训练过程中，会根据矢量文件中设置的宽高，自动对正样本进行缩放，比如我在用程序标注时，框选目标尺寸为30X30，但生产vec文件时填的尺寸为64X64，训练时会自动将框选图像做等比例缩放）；
2. 正样本图片应该尽可能包含少的干扰背景信息。在训练过程中这些背景信息也会成为正样本的一个局部特征，使得特征值的计算包含干扰信息。
3. 数据来源尽可能做到多样化，比如样本为车，车的姿态场景应稍丰富些。同一正样本目标的图像太多会使局部特征过于明显，造成这个目标的训练过拟合，影响检测精度，不利于训练器泛化使用。

1、采集正样本图片

正样本最后需要大小归一化，采集样本时可直接把它从原图里抠出来以便缩放；但是我是保留了原图，且在一张图上对多个目标做位置标注了，即保存了它的框个数和框位置信息；我在裁剪时将裁剪框做了比例约束，使得裁剪出来的框和最后需要归一化尺寸等比例。比如要归一化成64 X 64，在裁剪样本的时候，我按长宽等比例裁剪（如32X32，48X48）;也可以是非1：1的长宽比，具体看你目标对象的尺寸比例而定。

2、获取正样本路径列表

在你的图片文件夹里，编写一个.bat文件或者是一个.cmd文件（get route.bat，bat是避免每次都需要去dos框输入），如下所示：
内容也可更改为： dir pos /b > pos.txt

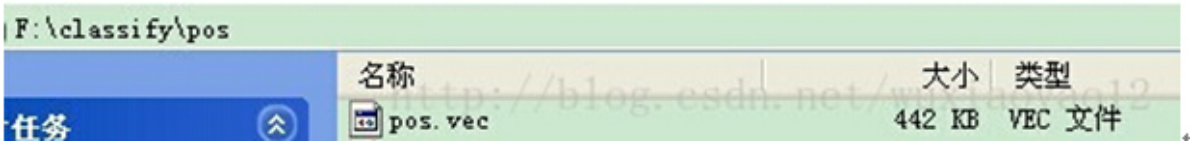
其中pos是文件夹名字，将pos文件夹中的所有文件名保存到pos.txt文件中；

运行bat文件，就会生成如下dat文件：

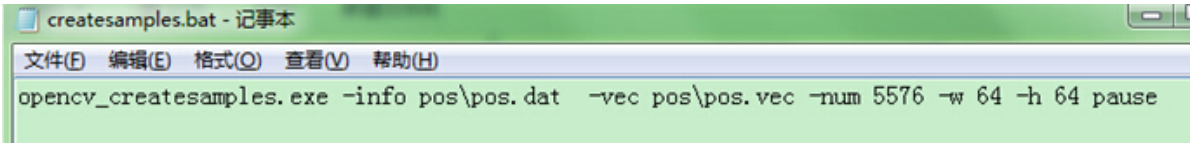
把这个dat文件中的所有非图片的路径都删掉，比如上图的头两行，由于我不是裁剪的，所以txt中的目标位置我都是通过一个objectMaker小程序生产的框和尺寸坐标位置：

（pos.txt文件内容中：图像格式后面的数字代表个数（可不只是1），后四个分别对应框的 left top width height，我不是把样本裁剪下来的，那txt中是xxx. jpg 3 1 3 24 24 26 28 25 25 60 80 26 26，xxx. jpg是原图,3为该图上的框数量，后面的数是每个框的信息）

3、获取正样本矢量集vec文件



利用opencv里的一个程序叫opencv_createsamples.exe，把它拷贝出来，放在当前目录下。针对它的命令输入也是写成bat文件，因为cascade训练的时候用的是vec。如下：可通过加-show显示某个正样本图像。可以根据自己需求更改num数，-w和-h



运行bat，就在我们得pos文件夹里生成了如下vec文件：

就此正样本东西准备结束。

注意：
样本矢量库的宽和高设置太大会使样本在训练过程中占用内存较大，可能出现内存分配失败的错误，根据实际情况适当更改。.

创建vec命令行参数：

> -vec 输出文件，内含有用于训练的正样本
>
> -img 输入图片文件名
>
> -bg 背景描述文件，文件中包含一系列的图像文件名，这些图像将被随机选作物体的背景
>
> -num 生成正样本的数目
>
> bgcolor 背景颜色（目前为灰度图），背景颜色表示透明颜色。因为图像压缩可造成颜色偏差，颜色的容差可以由 -bgthresh 指定。所有处于 bgcolor-bgthresh 和bgcolor+bgthresh 之间的像素都被设置为透明像素。
>
> -bgthresh
>
> -inv 如果指定该标志，前景图像的颜色将翻转。
>
> -randinv 如果指定该标志，颜色将随机地翻转。
>
> -maxidev 前景样本里像素的亮度梯度的最大值。
>
> -maxxangle X轴最大旋转角度，必须以弧度为单位。
>
> -maxyangle Y轴最大旋转角度，必须以弧度为单位。
>
> -maxzangle Z轴最大旋转角度，必须以弧度为单位。
>
> -show 很有用的调试选项。如果指定该选项，每个样本都将被显示。如果按下 Esc 键，程序将继续创建样本但不再显示。

> -w 输出样本的宽度（以像素为单位）。
>
> -h 输出样本的高度（以像素为单位）。

创建样本的流程如下：

输入图像沿着三个轴随机旋转。
旋转的角度由 -max?angle 限定。然后像素的亮度值位于 [bg_color-bg_color_threshold;bg_color+bg_color_threshold]范围的像素被设置为透明像素。将白噪声加到前景图像上。
如果指定了-inv ，那么前景图像的颜色将被翻转。如果指定了-randinv ，程序将随机选择是否将颜色进行翻转。
任选背景图像，将获得的前景图像放到背景图像上，并将图像调整到-w 和-h 指定的大小。最后将图像存入vec文件，vec文件名由命令行参数-vec 指定。
正样本也可从一系列事先标记好的图像中创建。标记信息可以存储于一个文本文件，与背景描述文件类似。文件中的每行对应一个图像文件。每行的第一个元素为图像文件名，后面是物体的数目，最后是物体位置和大小的描述 (x, y, width, height)。

二、关于负样本的准备

负样本的选取原则：

在负样本图片中不能包含正样本目标；

每个负样本之间是各不相同的，即确保负样本的多样性；

每个负样本的尺寸不是必须相同的，但负样本的尺寸不能小于正样本矢量集图像的宽和高，本文设置的尺寸都大于等于64*64；

这个特别简单，直接拿原始图，不需要裁剪抠图，也不需要保存框，只要保证比正样本尺寸大就可以了，只要把路径保存下来。同正样本类似，同理也可以是.cmd文件，内容如：dir neg /b > neg.txt　其中neg为负样本所在的图像文件夹名。neg.txt中内容不含框和位置信息。图如下：

至此有关负样本的也准备完成。

进行训练的话，创建一个classify20.bat
我写的.bat内容为：

```
@echo off
opencv_traincascade.exe -data data -vec pos.vec -bg neg\neg.txt -numPos 20000 -numNeg 50000 -numStages 16 -featureType HOG -w 64 -h 64
pause
```

会在data文件夹中生产一个xml文件，即分类器。

双击运行即可开始训练。

注意:numPos *2* < *numNeg* < *numPos*4

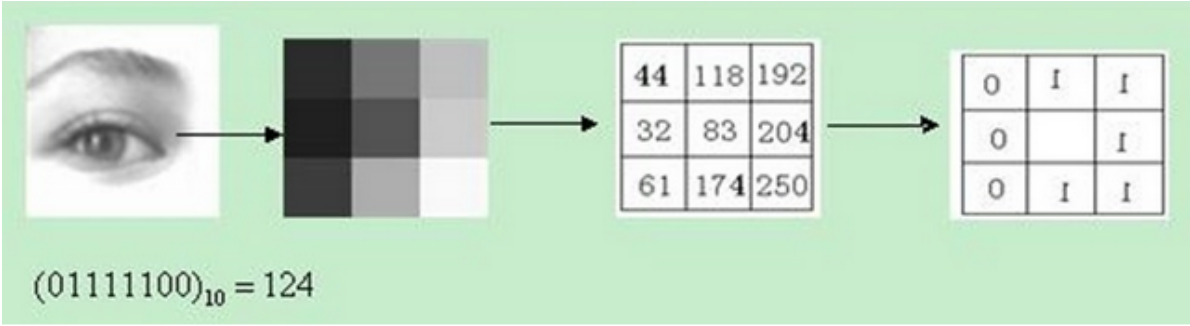
需要的文件都在这里列出：

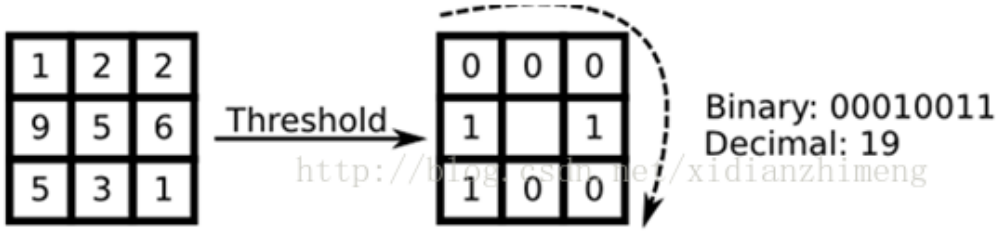
没有看LBP之前觉得它很神秘，看完了之后也就那么回事，不过提出LBP的人确实很伟大！！

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*3*3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，*3*3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：





用公式表示就是：

$$LBP(x_c, y_c) = \sum_{p=0}^{P-1} 2^p s(i_p - i_c)$$

其中(x_c, y_c)是中心像素，i_c是灰度值，i_n是相邻像素的灰度值，s是一个符号函数：

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{else} \end{cases} \qquad (1)$$

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；

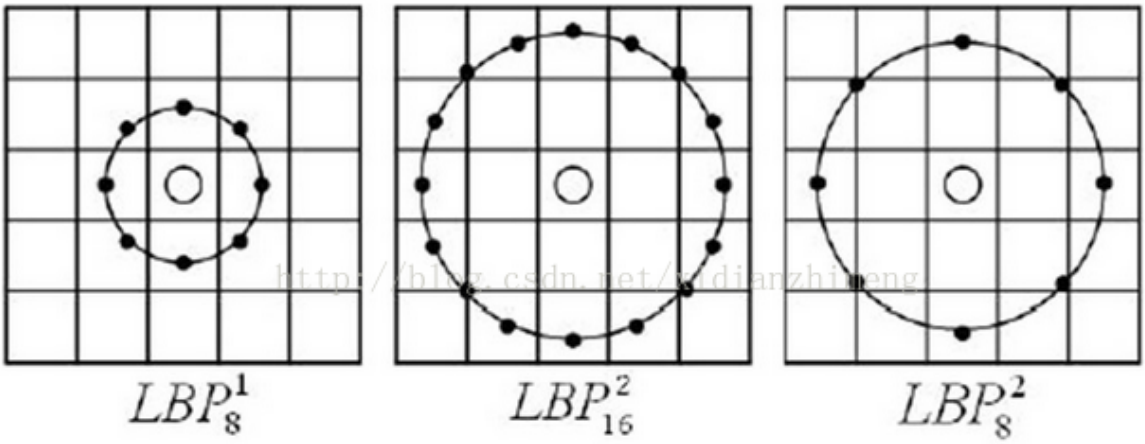


图2.2 几种LBP算子

（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。

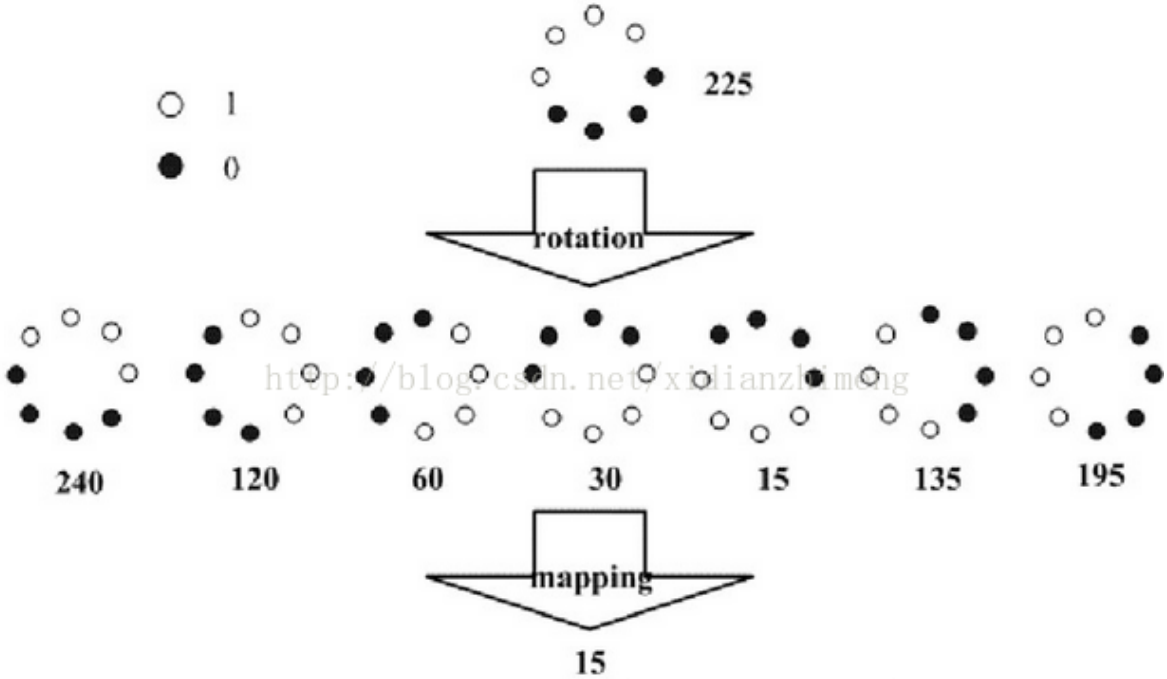


图 2.5 旋转不变的 LBP 示意

（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2P2P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220220 = 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

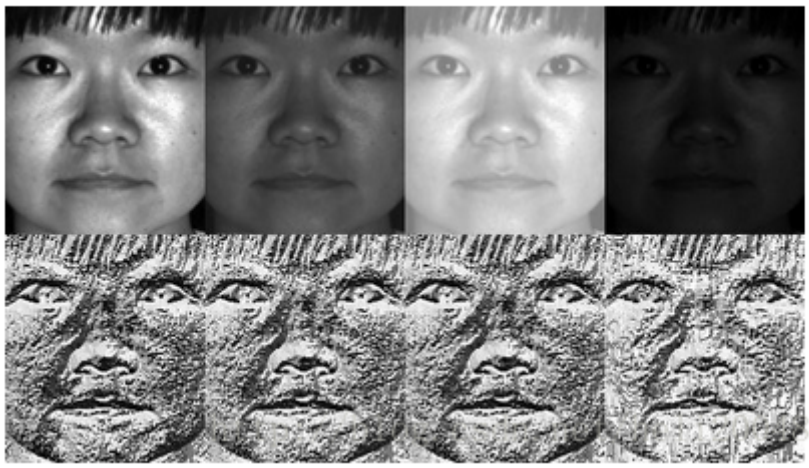
数据量减少的情况下能最好的代表图像的信息。

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2P2P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）。



从上图可以看出LBP对光照具有很强的鲁棒性

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100*100像素大小的图片，划分为10*10=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10*10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10*10个子区域，也就有了10*10个统计直方图，利用这10*10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16×16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*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。

Reference：

黄非非，基于 LBP 的人脸识别研究，重庆大学硕士学位论文，2009.5