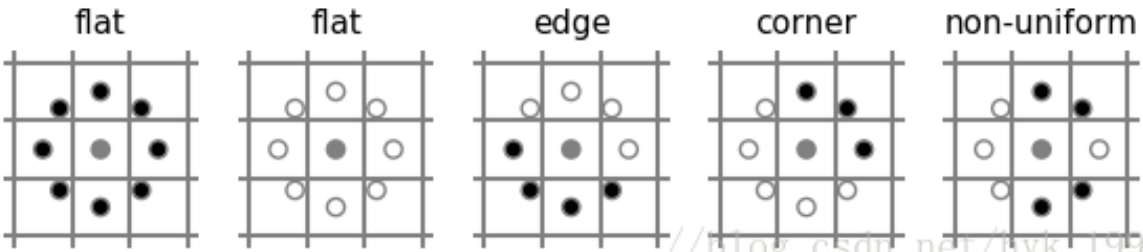
一. LBP特征

LBP（Local Binary Pattern），局部二值模式，主要用于提取纹理特征，根据文献[1]我们可以了解到LBP及其变体。一般的使用方法是，先将图像转换为灰度图，接着计算LBP特征图，最后计算其**直方图**作为特征向量。

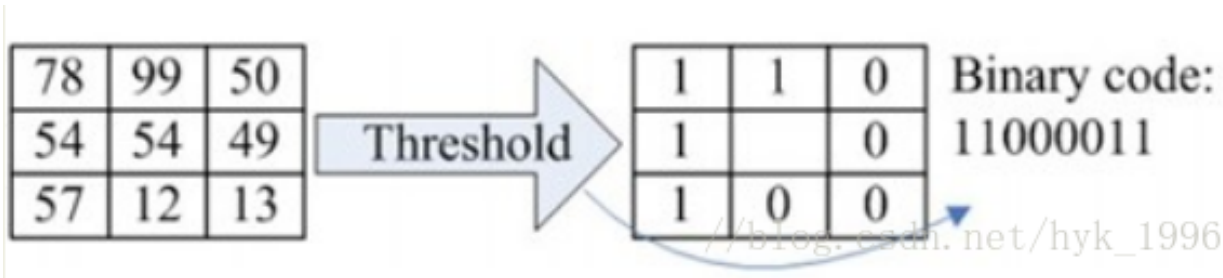
0.如何描述纹理信息

不多说，看图。LBP特征可以表示平坦、边缘、角点等区域。



1.original LBP

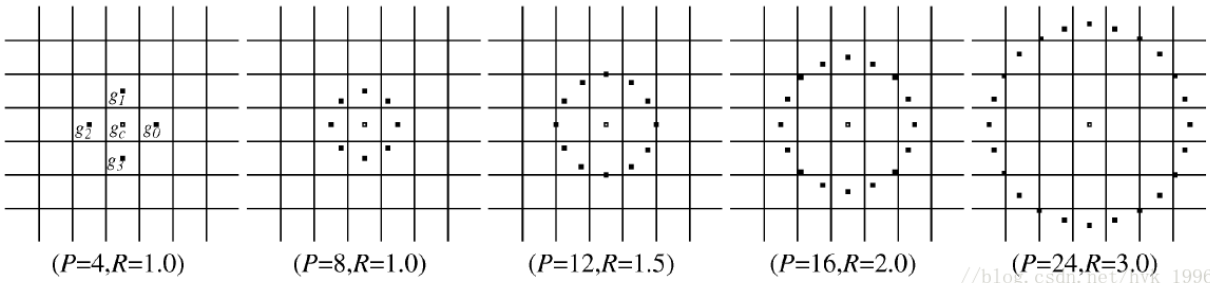
最经典的LBP特征，对于某像素的8-邻域，大于该像素值的置1，小于该像素值的置0，最后组成八位二进制码。



2.采样方式及角空间

圆形邻域的像素采样方式会比8-邻域的方式要更灵活，可以通过改变圆形的半径R来调整邻域大小。但半径R越大，采样点的数目P也越多。对于没有落到像素格子中央的点的灰度值，一般采用插值法得到。

除此之外，通过对采样角度设置更高的量化等级，可以得到更精细的角度分辨率。



3.灰度不变性

由于各邻域像素的灰度值需要减去中心像素的灰度值，可以实现针对平均光源的灰度不变性。

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p.$$

4.旋转不变性

由于编码的起始点是一定的，每一种二值编码模式经旋转（循环位移）后会产生不同的编码结果。为了形成旋转不变的编码模式，我们让有同一编码模式经旋转后产生的编码结果编码为同一值，定义二值编码为这些旋转结果中的最小值。

在具体代码实现时，通过使用计算映射(mapping)的方式来将不同的旋转二值编码映射为旋转不变的二值编码。

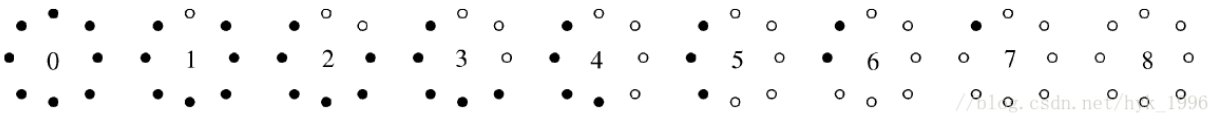
$$LBP^{ri}_{P,R} = min\{ROR(LBP_{P,R}, i) \quad | i = 0, 1, ..., P-1\}$$

5.uniform LBP

首先，我们将LBP的二值编码看作是头尾相接的，接着定义**计数U**为二值编码中0-1变化的次数(即0变成1,或者1变成0,记做一次)，公式表达如下：

$$U(LBP_{P,R}) = | s(g_{P-1} - g_c) - s(g_0 - g_c) | + \sum_{p=1}^{P-1} | s(g_p - g_c) - s(g_{p-1} - g_c) | .$$

对于U值小于等于2的二值编码，我们定义其为uniform pattern，例如半径为1，采样点为8的LBP特征有如下8种uniform pattern:



而对于其它的非uniform编码，我们将其归为一类。之所以这样定义，是因为这类编码易受噪声影响，变化频率非常大。

最终的uniform LBP计算公式如下：

$$LBP_{P,R}^{riu2} = \begin{cases} \sum_{p=0}^{P-1} s(g_p - g_c) & \text{if } U(LBP_{P,R}) \leq 2 \\ P+1 & \text{otherwise,} \end{cases}$$

总结一下，对于8个采样点的LBP，灰度不变LBP有256种编码输出，旋转不变LBP有36种编码输出，而uniform旋转不变LBP只有9种编码输出(8种uniform编码和剩余的非uniform编码)，非旋转不变的uniform LBP有58种输出。

6.结合对比度的LBP

根据灰度不变LBP的定义，我们直接舍弃了像素点间的灰度幅值差异，进而丢失了对比度的信息（即灰度值的方差）。因此，我们考虑结合对比度Var和LBP作为联合特征. 对比度计算公式如下：

$$VAR_{P,R} = \frac{1}{P}\sum_{p=0}^{P-1}(g_p - \mu)^2, \quad \text{where } \mu = \frac{1}{P}\sum_{p=0}^{P-1} g_p.$$

二. Python实现

在研究了网上LBP的代码实现后，我认为使用ski-image包的LBP特征提取函数比较好，因为它封装了多种LBP的方法，非常简单易用。官网上的参考文档如下：

Parameters:	image : (N, M) array Graylevel image. P : int Number of circularly symmetric neighbour set points (quantization ofthe angular sp R : float

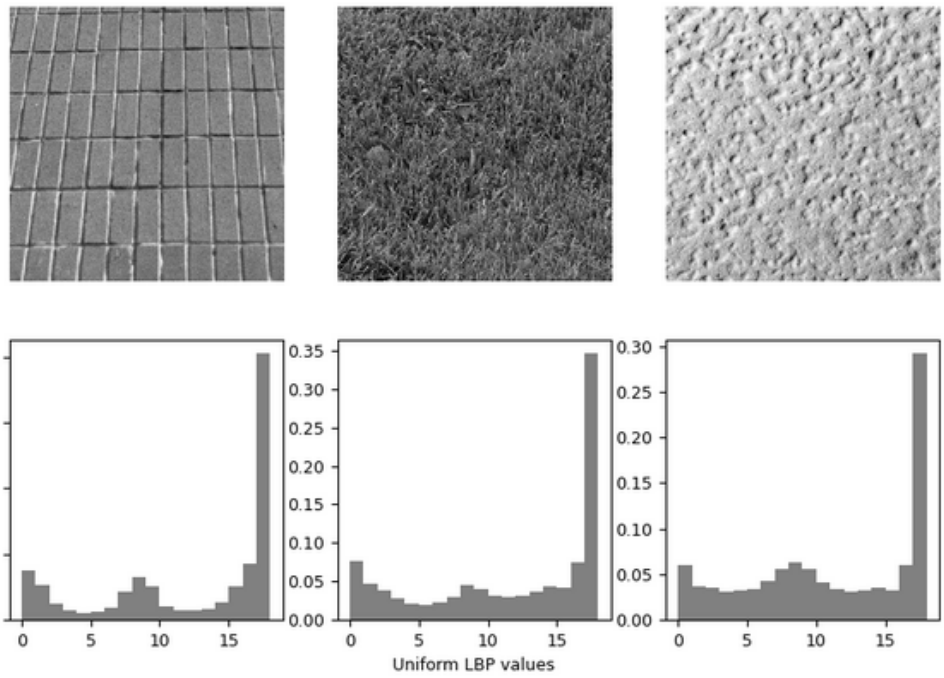
	Radius of circle (spatial resolution of the operator). method : {‘default’, ‘ror’, ‘uniform’, ‘var’} Method to determine the pattern. ‘default’: original local binary pattern which is gray scale but not rotation invariant. ‘ror’: extension of default implementation which is gray scale and rotation invarian ‘uniform’: improved rotation invariance with uniform patterns and finer quantizatio gray scale and rotation invariant. ‘nri_uniform’: non rotation-invariant uniform patterns variant which is only gray sc ‘var’: rotation invariant variance measures of the contrast of localimage texture wh scale invariant.
Returns:	output : (N, M) array LBP image.

代码例程：

```
1. # settings for LBP
2. radius = 2
3. n_points = 8 * radius
4. defkullback_leibler_divergence(p, q):
5. p = np.asarray(p)
6. q = np.asarray(q)
7. filt = np.logical_and(p != 0, q != 0)
8. return np.sum(p[filt] * np.log2(p[filt] / q[filt]))
9. defmatch(refs, img):
10. best_score = 10
11. best_name = None
12. lbp = local_binary_pattern(img, n_points, radius, METHOD)
13. n_bins = int(lbp.max() + 1)
14. hist, _ = np.histogram(lbp, normed=True, bins=n_bins, range=(0, n_bins))
15. for name, ref in refs.items():
16. ref_hist, _ = np.histogram(ref, normed=True, bins=n_bins,
```

```
range=(0, n_bins))
score = kullback_leibler_divergence(hist, ref_hist)
if score < best_score:
best_score = score
best_name = name
return best_name
brick = data.load('brick.png')
grass = data.load('grass.png')
wall = data.load('rough-wall.png')
refs = {
'brick': local_binary_pattern(brick, n_points, radius, METHOD),
'grass': local_binary_pattern(grass, n_points, radius, METHOD),
'wall': local_binary_pattern(wall, n_points, radius, METHOD)
}
# classify rotated textures
print('Rotated images matched against references using LBP:')
print('original: brick, rotated: 30deg, match result: ',
match(refs, rotate(brick, angle=30, resize=False)))
print('original: brick, rotated: 70deg, match result: ',
match(refs, rotate(brick, angle=70, resize=False)))
print('original: grass, rotated: 145deg, match result: ',
match(refs, rotate(grass, angle=145, resize=False)))
# plot histograms of LBP of textures
fig, ((ax1, ax2, ax3), (ax4, ax5, ax6)) = plt.subplots(nrows=2, ncols=3,
figsize=(9, 6))
plt.gray()
ax1.imshow(brick)
ax1.axis('off')
hist(ax4, refs['brick'])
ax4.set_ylabel('Percentage')
ax2.imshow(grass)
ax2.axis('off')
hist(ax5, refs['grass'])
ax5.set_xlabel('Uniform LBP values')
ax3.imshow(wall)
ax3.axis('off')
hist(ax6, refs['wall'])
plt.show()
```

结果如下：



Out:

```
Rotated images matched against references using LBP:
original: brick, rotated: 30deg, match result:  brick
original: brick, rotated: 70deg, match result:  brick
original: grass, rotated: 145deg, match result:  grass
```


网站参考：

doc文档

官方例程

参考文献：

[1] Ojala T, Pietikäinen M, Mäenpää T. Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns[C]// European Conference on Computer Vision. Springer-Verlag, 2000:404-420.

[2] Walt S V D, Schönberger J L, Nunez-iglesias J, et al. scikit-image: image processing in Python[J]. Peerj, 2014, 2(2):e453.