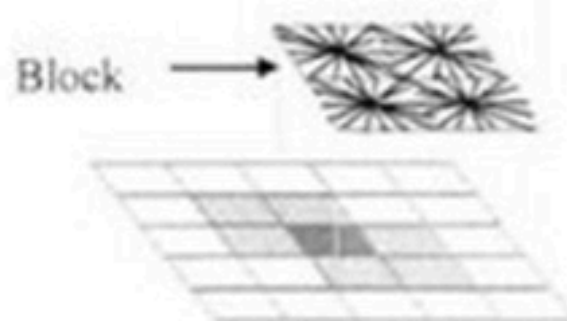
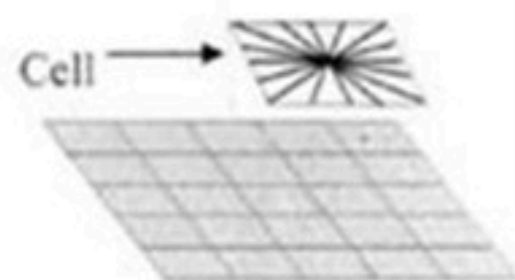
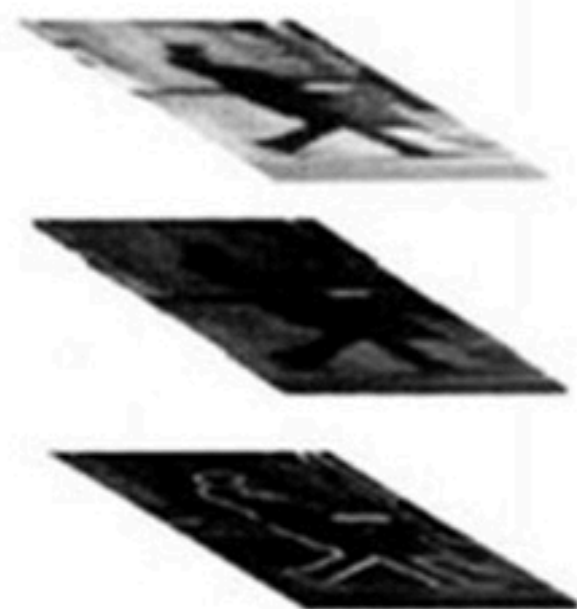


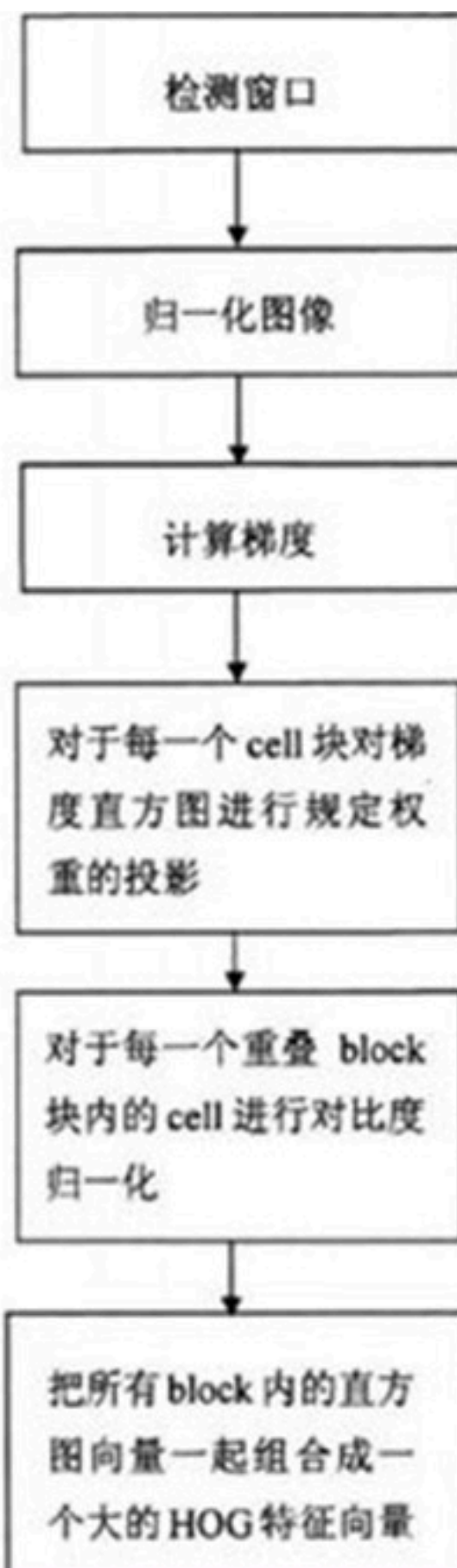
方向梯度直方图

Histogram of Oriented Gradient

[http://blog.csdn.net/zouxy09/
article/details/7929348](http://blog.csdn.net/zouxy09/article/details/7929348)



特征向量 $f = \{x_1, x_2, \dots, x_n\}$

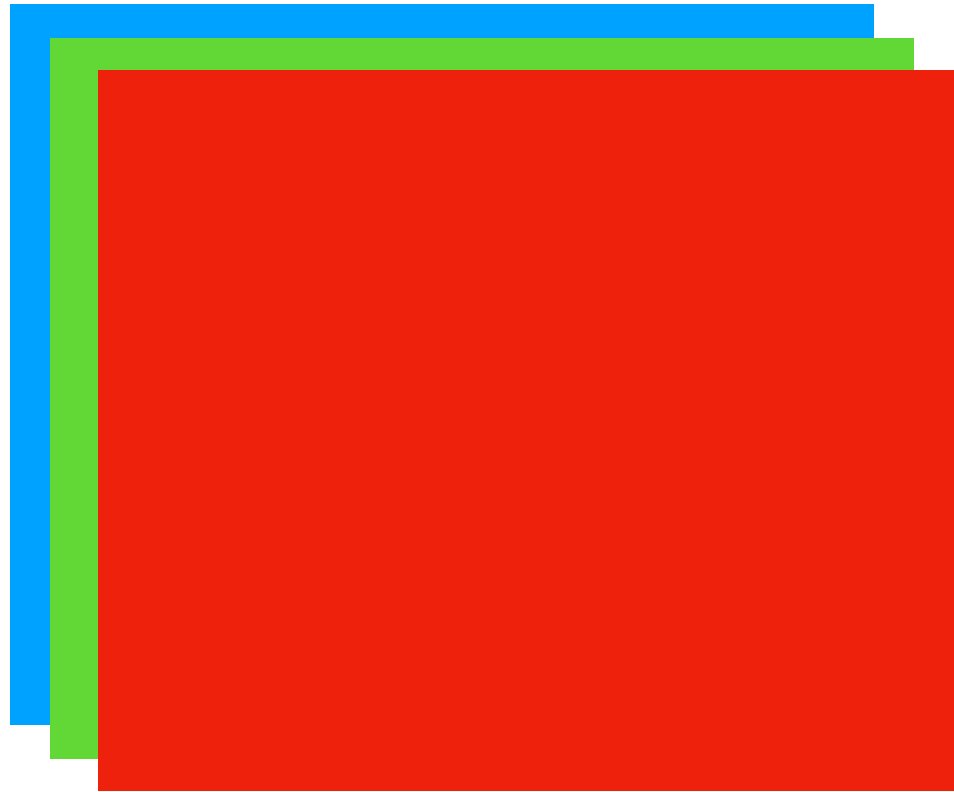


(1) 标准化gamma空间和颜色空间

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

Input



灰度化

1.分量法

```
im = im[:, :, 0]
```

2.最大值法

```
im = np.max(im, axis=2)
```

3.平均值法

```
im = np.average(im, axis=2)
```

4.加权平均法

```
im = np.dot(im[..., :3], [0.299, 0.587, 0.144])
```

```
plt.subplot(2,2,1)
plt.imshow(im)

plt.subplot(2,2,2)
plt.imshow(im[:, :, 0])

plt.subplot(2,2,3)
plt.imshow(np.max(im, axis=2))

plt.subplot(2,2,4)
plt.imshow(np.dot(im[:, :, 3], [0.299, 0.587, 0.144]))
```

<matplotlib.image.AxesImage at 0x119f3d490>



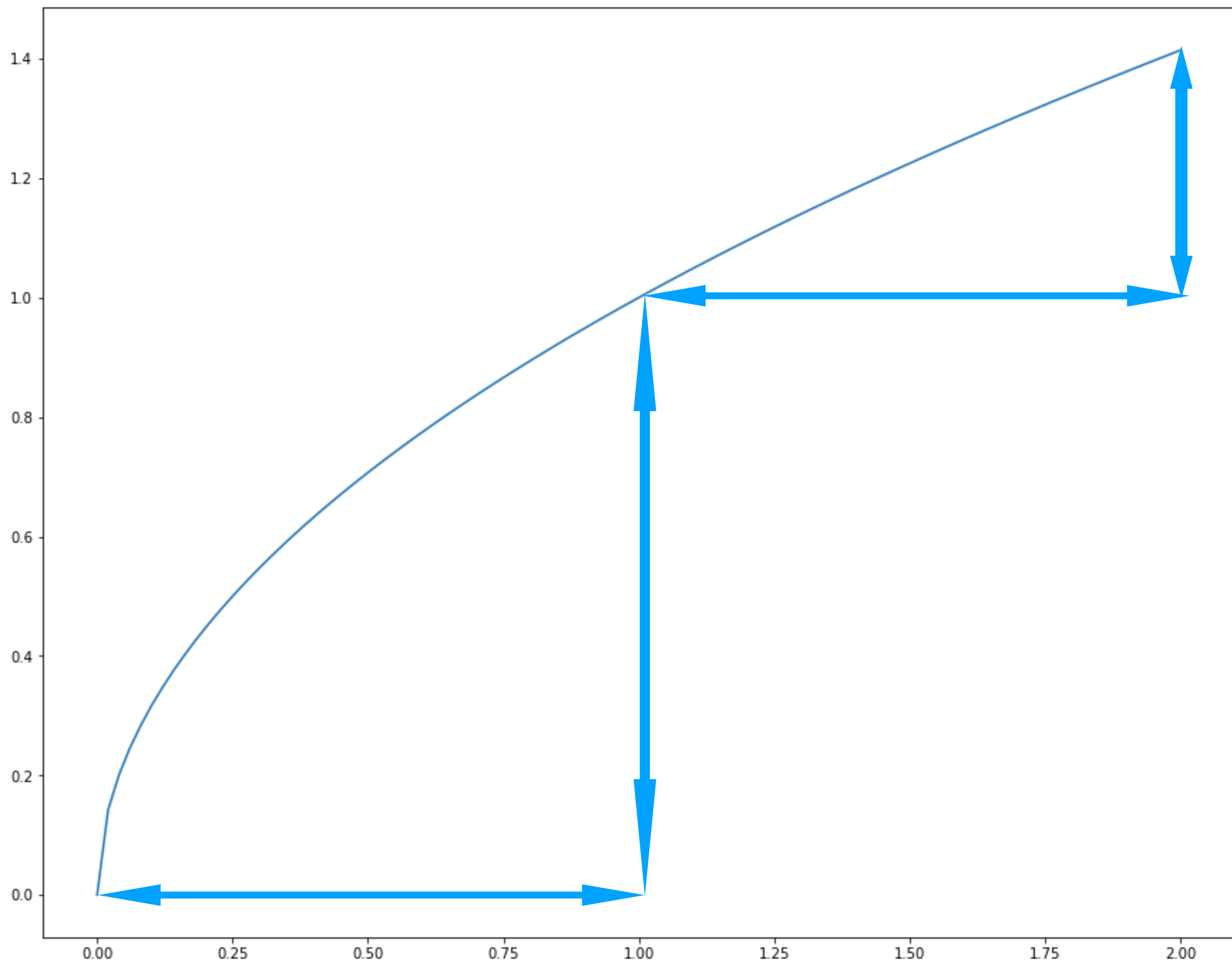


规范化

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

```
im = np.sqrt(im)
```





(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：

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

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

计算图像梯度

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

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

```
gx = np.zeros_like(image, dtype=np.double)
gx[:, 1:-1] = image[:, 2:] - image[:, :-2]
gy = np.zeros_like(image, dtype=np.double)
gy[1:-1, :] = image[2:, :] - image[:-2, :]
```

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

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

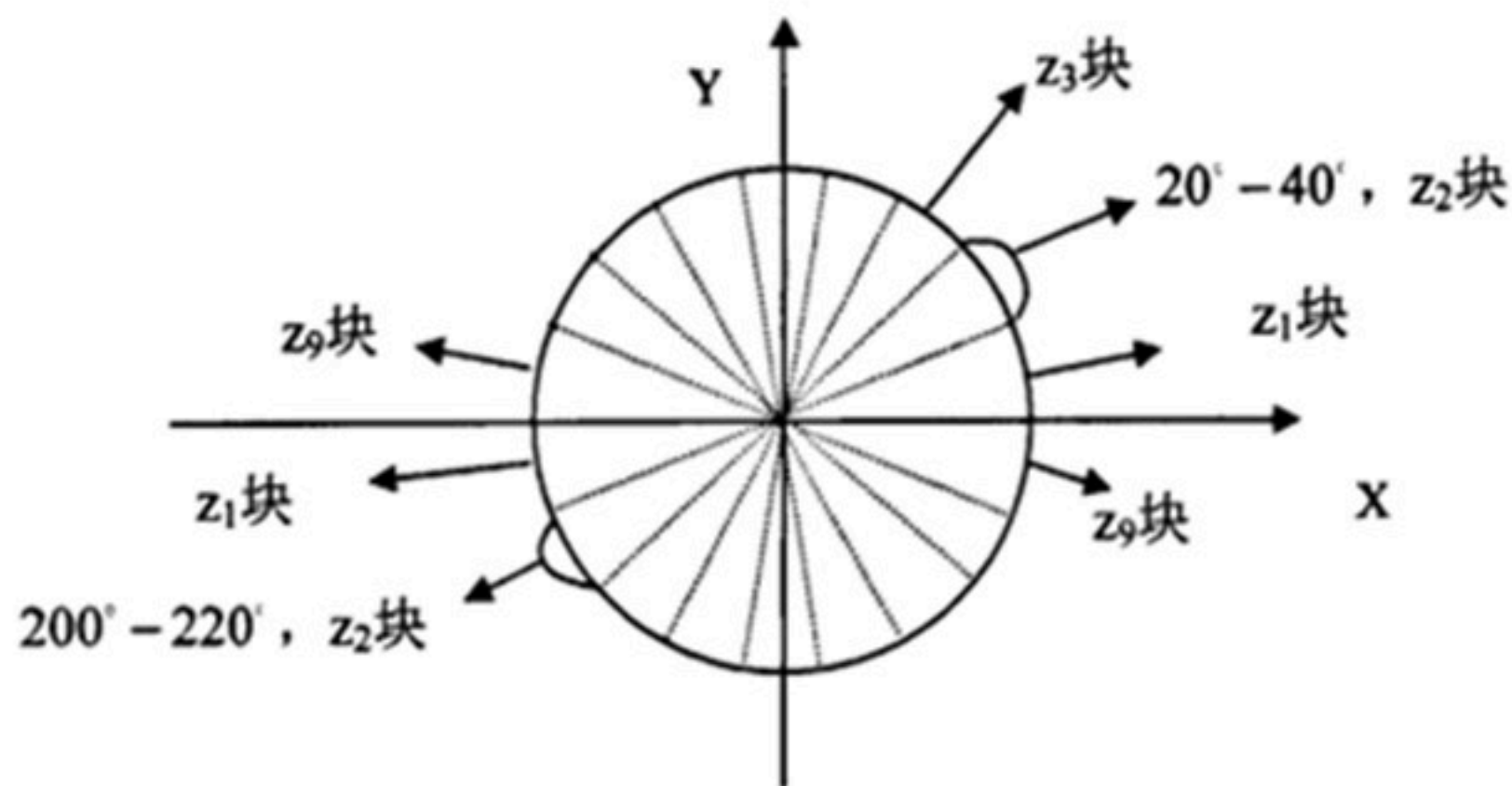
```
d_mag = np.sqrt(gx ** 2 + gy ** 2)
d_ori = np.arctan(gy / (gx + eps)) * 2 / np.pi
```

(3) 为每个细胞单元构建梯度方向直方图

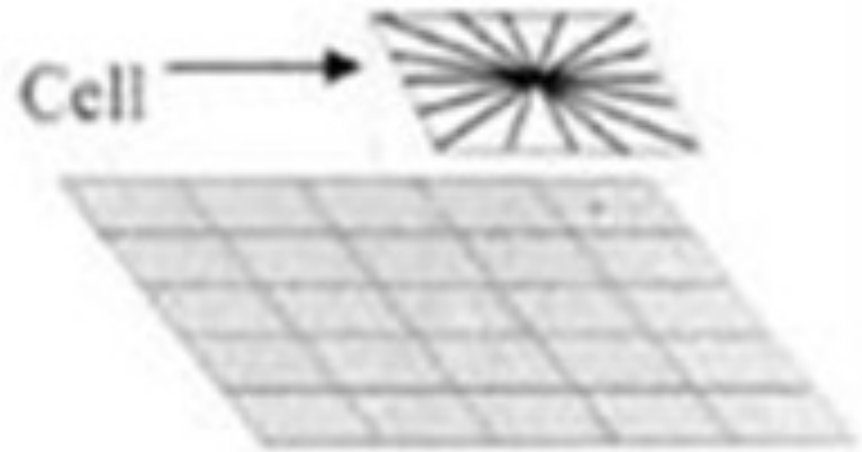
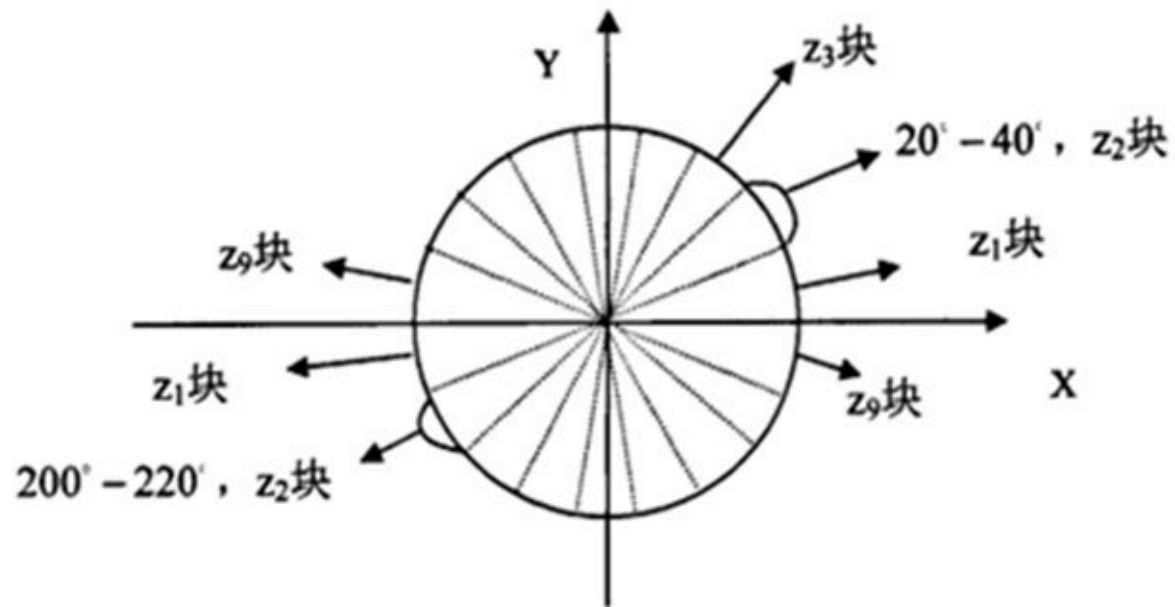
第三步的目的是为局部图像区域提供一个编码，同时能够保持对图像中人体对象的姿势和外观的弱敏感性。

我们将图像分成若干个“单元格cell”，例如每个cell为6*6个像素。假设我们采用9个bin的直方图来统计这6*6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）。



构建梯度方向直方图



```
n_cellsx = sx / cx
```

```
n_cellsy = sy / cy
```

```
orientation_histogram = np.zeros((n_cellsx, n_cellsy, orientations))
```

```
unit = 360.0 / orientations
```

```
grad_ori = np.array(grad_ori / unit, dtype=np.int)
```

```
for i in xrange(sx):
```

```
    for j in xrange(sy):
```

```
        orientation_histogram[(i-1)/cx+1, (j-1)/cy+1, grad_ori[i,j]] += grad_mag[i,j]
```


(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。

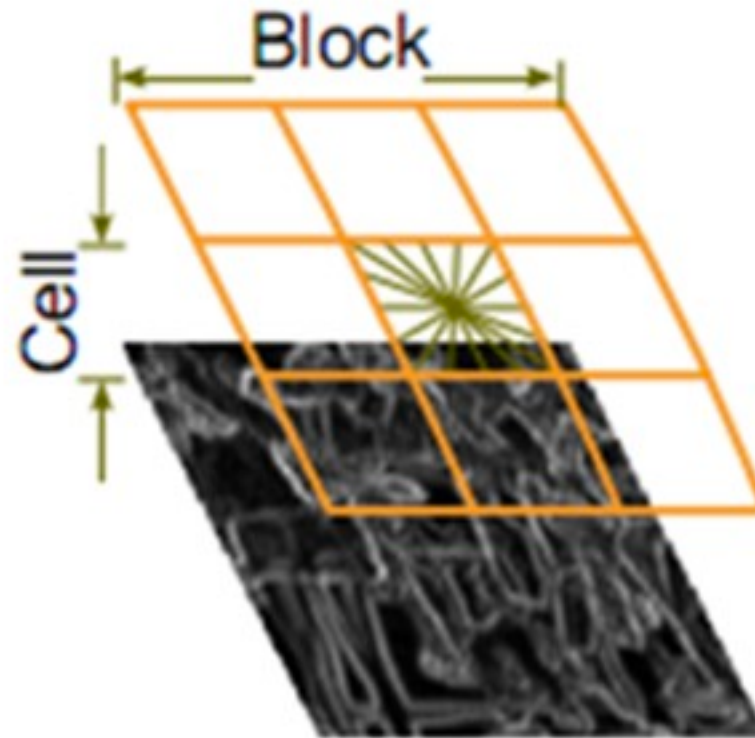
作者采取的办法是：把各个细胞单元组合成大的、空间上连通的区间 (blocks)。这样，一个block内所有cell的特征向量串联起来便得到该block的HOG特征。这些区间是互有重叠的，这就意味着：每一个单元格的特征会以不同的结果多次出现在最后的特征向量中。我们将归一化之后的块描述符 (向量) 就称之为HOG描述符。

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：3*3*9；

(5) 收集HOG特征

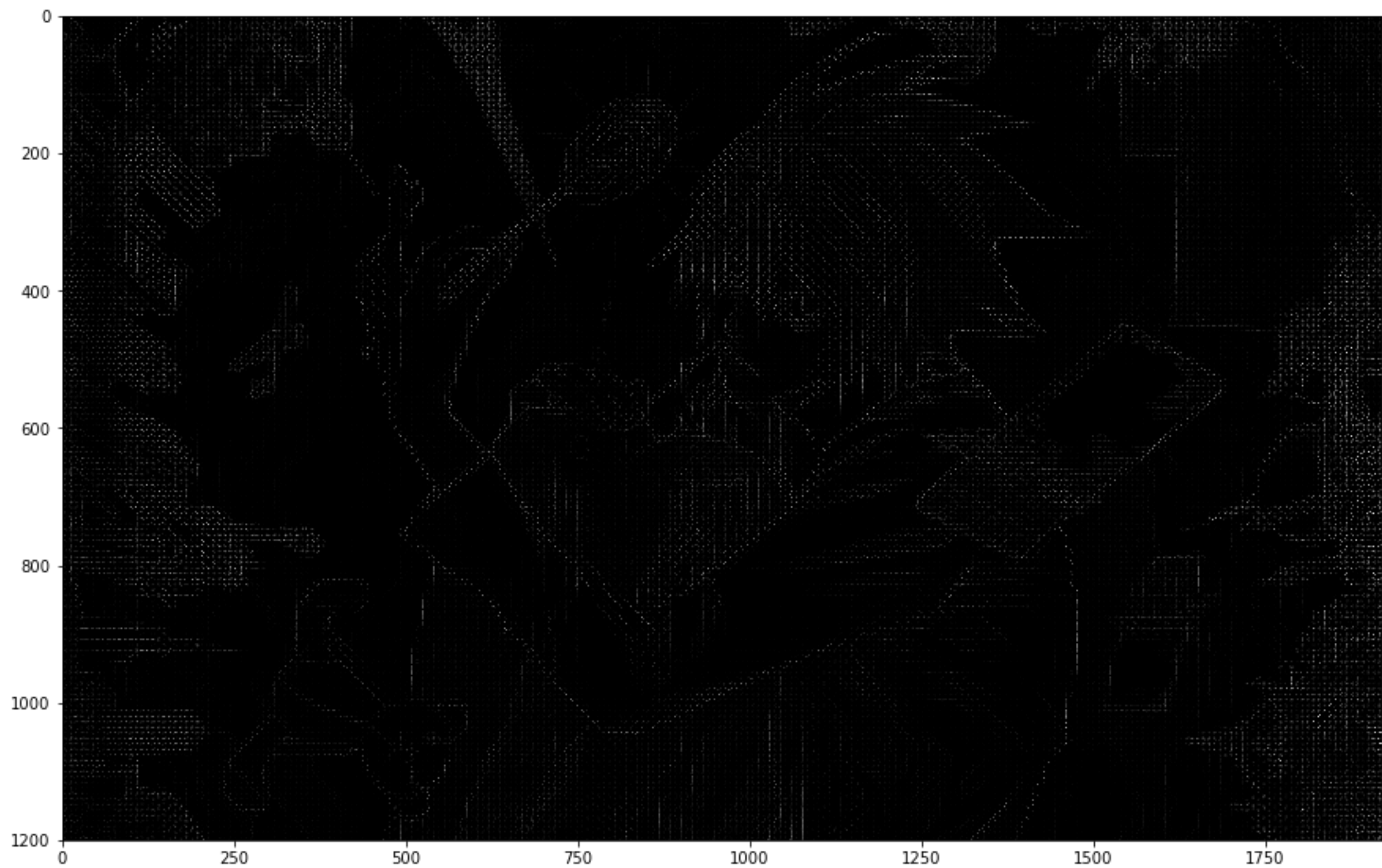
最后一步就是将检测窗口中所有重叠的块进行HOG特征的收集，并将它们结合成最终的特征向量供分类使用。

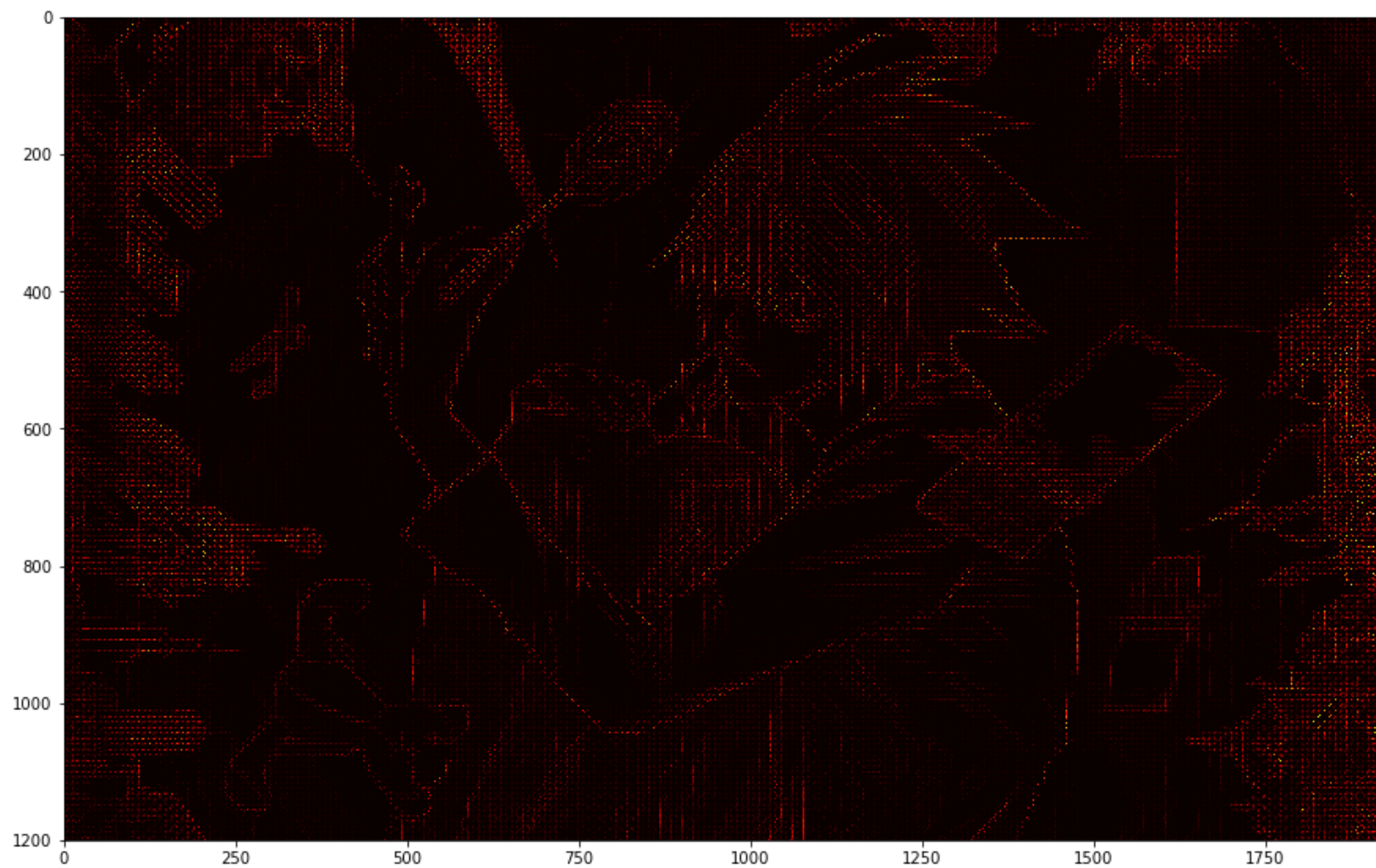


```
n_blocksx = (n_cellsx - bx) + 1
n_blocksy = (n_cellsy - by) + 1
normalised_blocks = np.zeros((n_blocksy, n_blocksx,
                              by, bx, orientations))

for x in range(n_blocksx):
    for y in range(n_blocksy):
        block = orientation_histogram[y:y + by, x:x + bx, :]
        normalised_blocks[y, x, :] = block - block.mean() / np.sqrt(block.sum() ** 2 + eps)
```

```
return normalised_blocks.ravel()
```





QA

plane



car



bird



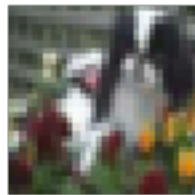
cat



deer



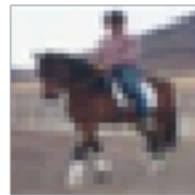
dog



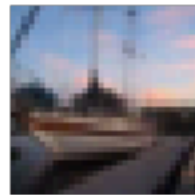
frog



horse



ship



truck

