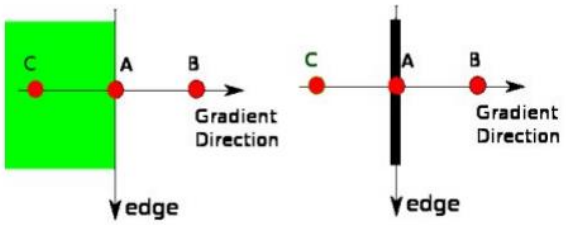
安装 opencv

OpenCV 中是按 BGR，matplotlib 中是按 RGB 排列):

pip install opencv-python

边缘检测

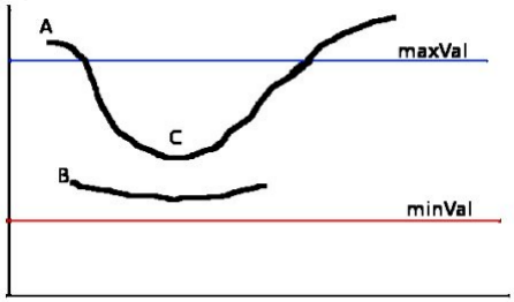
在获得梯度的方向和大小之后，应该对整幅图像做一个扫描，去除那些非边界上的点。对每一个像素进行检查，看这个点的梯度是不是周围具有相同梯度方向的点中最大的。如下图所示：



现在你得到的是一个包含“窄边界”的二值图像。

滞后阈值

现在要确定那些边界才是真正的边界。这时我们需要设置两个阈值：minVal 和 maxVal。当图像的灰度梯度高于 maxVal 时被认为是真的边界，那些低于 minVal 的边界会被抛弃。如果介于两者之间的话，就要看这个点是否与某个被确定为真正的边界点相连，如果是就认为它也是边界点，如果不是就抛弃。如下图：



A 高于阈值 maxVal 所以是真正的边界点，C 虽然低于 maxVal 但高于 minVal 并且与 A 相连，所以也被认为是真正的边界点。而 B 就会被抛弃，因为他不仅低于 maxVal 而且不与真正的边界点相连。所以选择合适的 maxVal 和 minVal 对于能否得到好的结果非常重要。

在这一步一些小的噪声点也会被除去，因为我们假设边界都是一些长的线段。

示例代码

在 OpenCV 中只需要一个函数：**cv2.Canny()**，就可以完成以上几步。让我们看如何使用这个函数。这个函数的第一个参数是输入图像。第二和第三个分别是 **minVal** 和 **maxVal**。第三个参数设置用来计算图像梯度的 Sobel 卷积核的大小，默认值为 3。最后一个参数是 **L2gradient**，它可以用来设定求梯度大小的方程。如果设为 **True**，就会使用我们上面提到过的方程，否则使用方程：$Edge\_Gradient\,(G) = |G_x^2| + |G_y^2|$ 代替，默认值为 **False**。

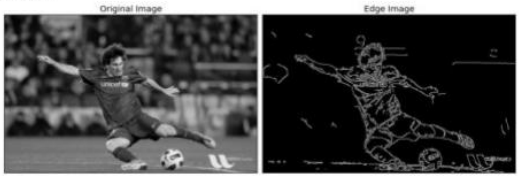
```
import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('messi5.jpg',0)
edges = cv2.Canny(img,100,200)

plt.subplot(121),plt.imshow(img,cmap = 'gray')
plt.title('Original Image'), plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(edges,cmap = 'gray')
plt.title('Edge Image'), plt.xticks([]), plt.yticks([])

plt.show()
```

结果：



通过均衡直方提升亮度

直方图均衡化的作用是图像增强。有两个问题比较难懂，一是为什么要选用累积分布函数，二是为什么使用累积分布函数处理后像素值会均匀分布。

均衡化过程中，必须要保证两个条件：①像素无论怎么映射，一定要保证原来的大小关系不变，较亮的区域，依旧是较亮的，较暗依旧暗，只是对比度增大，绝对不能明暗颠倒；②如果是八位图像，那么像素映射函数的值域应在 0 和 255 之间的，不能越界。综合以上两个条件，累积分布函数是个好的选择，因为累积分布函数是单调增函数（控制大小关系），并且值域是 0 到 1（控制越界问题），所以直方图均衡化中使用的是累积分布函数。

来看看通过上述公式怎样实现的拉伸。假设有如下图像：

255	128	200	50
50	200	255	50
255	200	128	128
200	200	255	50

得图像的统计信息如下图所示，并根据统计信息完成灰度值映射：

灰度值	像素个数	概率	累积概率	根据函数映射后灰度值	取整
50	4	0.25	0.25	0.25*（255-0）=63.75	64
128	3	0.1875	0.4375	111.5625	112
200	5	0.3125	0.75	191.25	191
255	4	0.25	1	255	255

映射后的图像如下所示：

255	112	191	64
64	191	255	64
255	191	112	112
191	191	255	64

以上就是直方图映射均衡化的步骤，当然还有一些基于此的更优算法，比如Photoshop中的方法，在此就不一一列举了，大同小异。

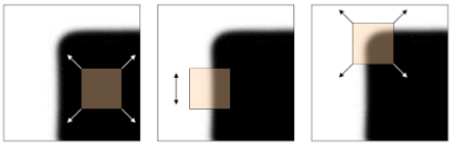
角点检测

在图像中定义一个局部小窗口，然后沿各个方向移动这个窗口，则会出现 a) b) c) 三种情况，分别对应平坦区、边缘和角点

a) 窗口内的图像强度，在窗口向各个方向移动时，都没有发生变化，则窗口内都是 “平坦区”，不存在角点

b) 窗口内的图像强度，在窗口向某一个 (些) 方向移动时，发生较大变化；而在另一些方向不发生变化，那么，窗口内可能存在 “边缘”

c) 窗口内的图像强度，在窗口向各个方向移动时，都发生了较大的变化，则认为窗口内存在 “角点”



a) flat region　　b) edge　　c) corner

参数

- img：数据类型为 float32 的输入图像。
- blockSize：角点检测中要考虑的邻域大小。
- ksize：sobel求导使用的核大小
- k： 角点检测方程中的自由参数，取值参数为 [0.04，0.06].

star 特征检测

有些特征用轮廓并无法来检测，比如脸，在内容上显示的特征就不能用边缘检测了，要想把这些关键点检测出来就需要用 star 检测器。

sift 特征检测

图像中的角具有旋转不变性，即旋转图像时角不会发生变化，但在放大或缩小图像时，角可能发生变化。SIFT 是指尺度不变特征变换，SIFT 算法用于查找图像中的尺度不变特征，返回图像中的关键点。

图像识别

见代码