*'''  
2021.1.30FromIvicxDS:openCV;E4图像形态处理：Canny边缘检测算法  
'''***'''  
流程  
1.高斯滤波平滑图像滤除噪音  
2.计算图像中每个点的梯度强度和方向  
3.应用非极大值抑制【Non-Maximum Suppression】，消除边缘检测带来的杂散效应  
4.应用双阈值【Double-Threshold】检测确定真实和潜在的边缘  
5.通过抑制孤立弱边缘最终完成检测  
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  
step1.————————————————————————————————————————————————————  
a b c h1 h2 h3  
d e f \* h4 h5 h6 ->结果  
g h i h7 h8 h9  
step2.————————————————————————————————————————————————————  
ex:  
<Sx=  
-1 0 1  
-2 0 2  
-1 0 1>  
<Sy=  
1 2 1  
0 0 0  
-1 -2 -1>  
G=sqr(Gx^2+Gy^2)  
sta=arctan(Gy/Gx)  
<Gx=Sx\*A=  
-1 0 1 a b c  
-2 0 2 \* d e f  
-1 0 1 g h i  
<Gy同理  
step3.————————————————————————————————————————————————————  
F1:  
线性差值法:  
g1 Q g2 g  
| \ |  
g C g  
| \ |  
g g3 Z g4  
\代表梯度方向  
g1step2的结果M(g1),g2...M(g2)  
则:**

**M(Q)=w\*M(g2)+(1-w)\*M(g1),w=distance(Q,g2)/(g1,g2)  
#按照权重计算假想点Q的梯度浮值  
#如果M(C)>M(Q) and M(C)>M(Z) then C为边界点，保留  
F2:  
直接用周围的8个点，【拟合为45度】  
A B C ↓  
 |  
D E F ↓  
 |   
G H I ↓  
↓:梯度方向  
if E>D and E>F then E 是边界  
step4.————————————————————————————————————————————————————  
指定一个maxValue,一个minValue  
if A>maxValue then A->边界  
elseif A<maxValue and A>minValue then   
 if A与边界相连 then A->边界 else A->X  
elseif A<minValue then A->X  
'''**import cv2*#读取格式为BGR*def showPicture(name,picture):  
 *#图像的显示,也可以显示多窗口* cv2.imshow(name,picture)  
 *#在键盘中按任意键退出显示并向后执行语句  
 #cv2.waitKey(1000)表示只显示1秒* cv2.waitKey(0)  
 cv2.destroyAllWindows()  
  
img=cv2.imread(**"image/x2.jpg"**)  
  
v1=cv2.Canny(img,180,450)*#img,minValue,maxValue*showPicture(**"Canny"**,v1)