## Q1.

（最后需要devided by the “kernel sum“）

**The code is like this:**

def Conv2(img, H, W, kernel, n):

col = np.zeros(H)

row = np.zeros(W + 2)

img = np.insert(img, W, values=col, axis=1)

img = np.insert(img, 0, values=col, axis=1)

img = np.insert(img, H, values=row, axis=0)

img = np.insert(img, 0, values=row, axis=0)

res = np.zeros([H,W],dtype=np.float32)

for i in range(H):

for j in range(W):

temp = img[i:i + 3, j:j + 3]

temp = np.multiply(temp,kernel)

res[i][j] = temp.sum()

return res

The image padding is:

col = np.zeros(H)

row = np.zeros(W + 2)

img = np.insert(img, W, values=col, axis=1)

img = np.insert(img, 0, values=col, axis=1)

img = np.insert(img, H, values=row, axis=0)

img = np.insert(img, 0, values=row, axis=0)

and the filter is:

for i in range(H):

for j in range(W):

temp = img[i:i + 3, j:j + 3]

temp = np.multiply(temp,kernel)

res[i][j] = temp.sum()

## Q2.

**The gradient code is like this:**

def ComputeGrad(src,flag):

if flag==0:

kernel=np.zeros([3,3],dtype=np.float32);

kernel[0,0]=-1;

kernel[0,1]=-2;

kernel[0,2]=-1;

kernel[2,0]=1;

kernel[2,1]=2;

kernel[2,2]=1;

return Conv2(img,img.shape[0],img.shape[1],kernel,3)

if flag==1:

kernel=np.zeros([3,3],dtype=np.float32);

kernel[0,0]=-1;

kernel[1,0]=-2;

kernel[2,0]=-1;

kernel[0,2]=1;

kernel[1,2]=2;

kernel[2,2]=1;

return Conv2(img,img.shape[0],img.shape[1],kernel,3)

The edge strength image is fig 1.（**should be greyscale**）



Fig.1. The strength edge image.

## Q3.

**The function which finds edge is like that:**

def FindEdge(img):

gradx=ComputeGrad(img,1);

grady=ComputeGrad(img,0);

t=np.square(gradx)+np.square(grady)

t=np.sqrt(t)

ret,m=cv2.threshold(t,200,255,cv2.THRESH\_BINARY)

return m

We hist the gradient image, the hisgram is fig.2.

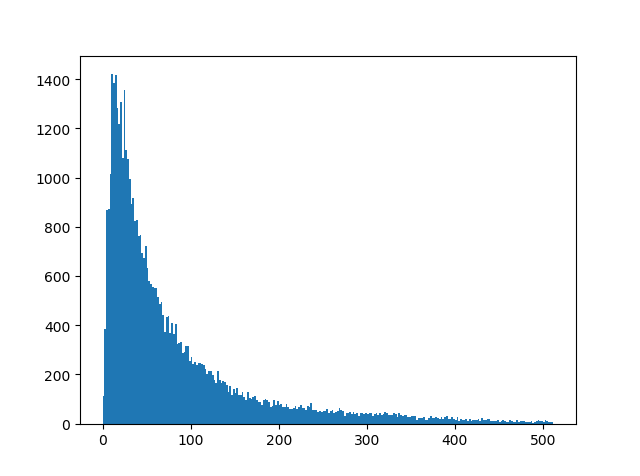


Fig.1. The histogram of the gradient image.

Through the histogram we select 160 （might 130）as the threshold. The result is the fig.3.



Fig.3. The best strength edge image.

## Q4.（comparing the image smoothing and smooth before）

(explaining the differences and says the smoothing one is better)

The smooth kernel has been used on image.And the histogram is like fig.4.

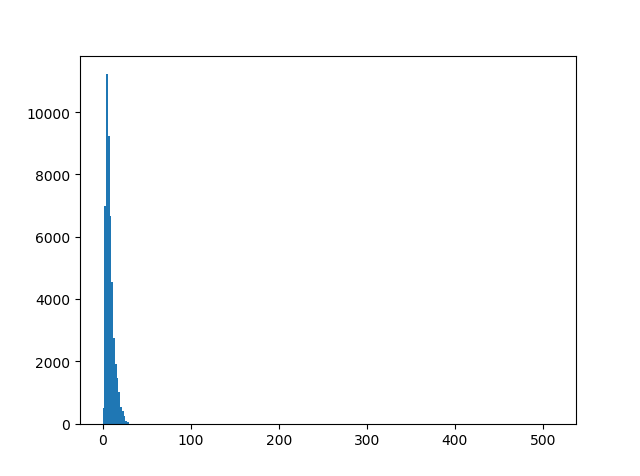


Fig.4. The histogram of the smoothed gradient image.

According to histogram, we select the threshold 15 as the divide threshold. And the image is fig.5. 

Fig.5. The best strength edge-smoothed image.

It is obvious that smoothing can erase the small edges and keep the big ones. If we find big edges on images, we can use smoothing before doing gradient.

## conclusions.

1. image padding should be called when you want to keep whole information of the image.
2. gradient image can be calculated by convolution.
3. Smoothing is a good idea when finding the main edge of the image.
4. 在q4的地方做更多的experiments 比如 5\*5 kernal， 7\* 7 kernal ，9\*9， 然后比较几者的效果（可能是77的最好，之后的实验开始下降）
5. Use a different way（比如全是1的kernel，或者其他的smoothing 方法）