Name: Jay Goyal

Roll no.: C017

Semester: VI

Program: B.Tech

Branch: EXTC

Date of performance: 19th March

Date of Submission: 26th March

Experiment Number: 7

Aim:

To write a program in PYTHON to highlight horizontal, vertical and diagonal edges of an image

Theory:

Sobel Operator

-1 -2 -1

0 0 0

1 2 1

Horizontal edge Fx

-1 0 1

-2 0 2

-1 0 1

Vertical Edge Fy

- Convolve Fx mask to the original image to obtain the x gradient of the image
- Convolve Fy mask to the original image to obtain the y gradient of the image
- Add the results of the above two steps

0 1 2

-1 0 1

-2 -1 0

Diagonal Edge

Laplacian operator

0 1 0

1 -4 1

0 1 0

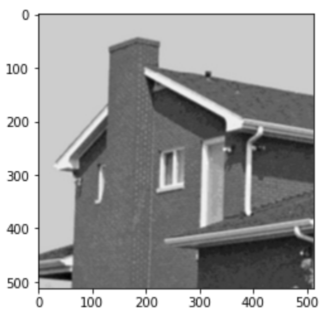
Conclusion:

We learnt about the significance of filter masks for edge enhancement

and Implemented Sobel and Laplacian operators

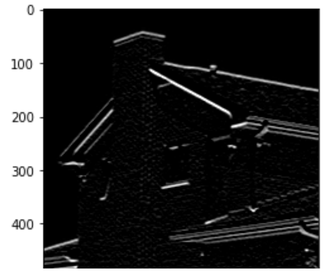
```
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal
import cv2
```

```
img= cv2.imread("/content/house.tif",0) #Read the image
plt.imshow(img, cmap="gray", vmin=0, vmax=255)
m,n= img.shape
```



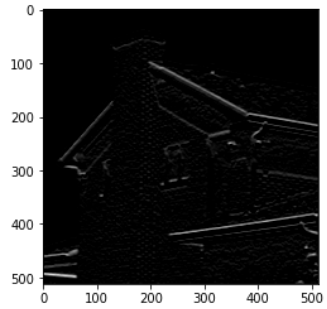
```
#Detect horizontal edges using in built convolution function
sobel_h= np.array([[-1,-2,-1], [0,0,0], [1,2,1]])
img_h= signal.convolve(img,sobel_h,mode='same')
plt.imshow(img_h,cmap="gray", vmin=0, vmax=255)
```

<matplotlib.image.AxesImage at 0x7f54df9c2b50>



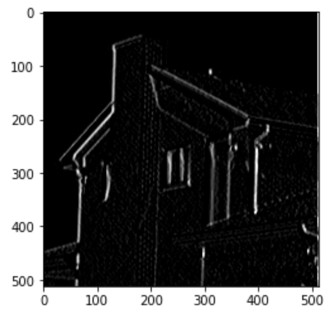
```
#Detect horizontal edges without using in built convolution function
mask= np.array([[-1,-2,-1], [0,0,0], [1,2,1]])
img_new_h=np.zeros([m,n])
for i in range(1,m-1):
    for j in range(1,n-1):
        temp= img[i-1,j-1]*mask[0,0]+img[i-1,j]*mask[0,1]+img[i-1,j+1]*mask[0,2]+img[i,j-1]*mask[1,0]+img[i,j]*mask[1,1]+img[i,j+1]*mask[1,2]+img[i+1,j-1]*mask[2

        img_new_h[i,j]=temp

plt.imshow(img_new_h, cmap="gray", vmin=0)
```

<matplotlib.image.AxesImage at 0x7f54df949310>



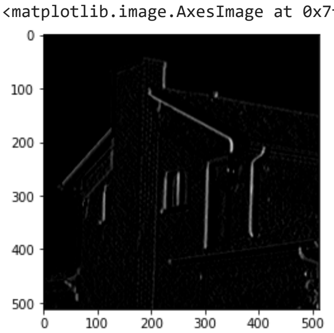
```
#Detect vertical edges using inbuilt function for convolution
sobel_v= np.array([[-1,0,1], [-2,0,2], [-1,0,1]])
img_v= signal.convolve(img,sobel_v,mode='same')
plt.imshow(img_v,cmap="gray", vmin=0, vmax=255)
```

<matplotlib.image.AxesImage at 0x7f54df927e50>



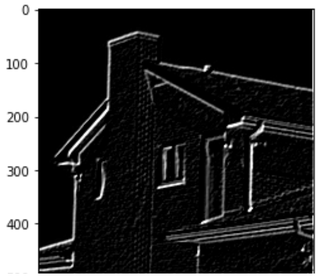
```
#Detect vertical edges without using in built convolution function
mask_v= np.array([[-1,0,1], [-2,0,2], [-1,0,1]])
img_new_v=np.zeros([m,n])
for i in range(1,m-1):
    for j in range(1,n-1):
        temp= img[i-1,j-1]*mask_v[0,0]+img[i-1,j]*mask_v[0,1]+img[i-1,j+1]*mask_v[0,2]+img[i,j-1]*mask_v[1,0]+img[i,j]*mask_v[1,1]+img[i,j+1]*mask_v[1,2]+img[i+1

        img_new_v[i,j]=temp

plt.imshow(img_new_v, cmap="gray", vmin=0)
```

<matplotlib.image.AxesImage at 0x7f54df89f110>



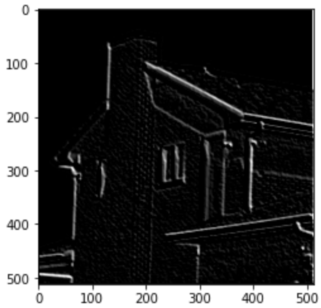
```
#Detecting vertical and horizontal edges
img_hv= img_h+img_v
plt.imshow(img_hv, cmap="gray", vmin=0, vmax=255)
```

```
<matplotlib.image.AxesImage at 0x7f54df7ff590>
```



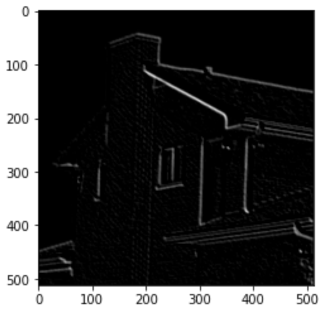
```
# Detecting Diagonal with inbuilt convolve
sobel_d= np.array([[0,1,2], [-1,0,1], [-2,-1,0]])
img_d= signal.convolve(img,sobel_d,mode='same')
plt.imshow(img_d, cmap="gray", vmin=0, vmax=255)
```

```
<matplotlib.image.AxesImage at 0x7f54df7e4c50>
```



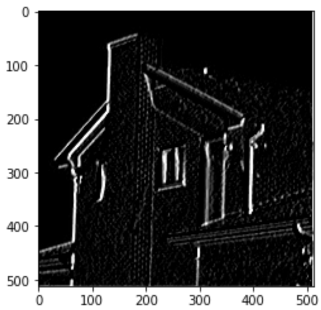
```
# Detecting Diagonal without inbuilt convolve function
mask_d= np.array([[0,1,2], [-1,0,1], [-2,-1,0]])
img_new_d=np.zeros([m,n])
for i in range(1,m-1):
    for j in range(1,n-1):
        temp= img[i-1,j-1]*mask_d[0,0]+img[i-1,j]*mask_d[0,1]+img[i-1,j+1]*mask_d[0,2]+img[i,j-1]*mask_d[1,0]+img[i,j]*mask_d[1,1]+img[i,j+1]*mask_d[1,2]+img[i+1

        img_new_d[i,j]=temp

plt.imshow(img_new_d, cmap="gray", vmin=0)
```

```
<matplotlib.image.AxesImage at 0x7f54df87ca10>
```



```
#Horizontal, vertical and diagonal edges
img_hvd= img_h+img_v+img_d
plt.imshow(img_hvd, cmap="gray", vmin=0, vmax=255)
```

```
<matplotlib.image.AxesImage at 0x7f54dfe806d0>
```

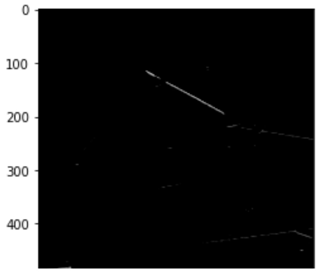


```
#centre 0 row in horizontal and centre column 0 for vertical
```

Horizontal edges variation

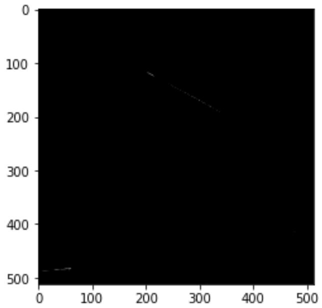
```
#Detect horizontal edges using in built convolution function
sobel_h_1= np.array([[-1,-4,-1], [0,0,0], [1,2,1]])
img_h_1= signal.convolve(img,sobel_h_1,mode='same')
plt.imshow(img_h_1,cmap="gray", vmin=0, vmax=255)
```

```
<matplotlib.image.AxesImage at 0x7f54ddf17a90>
```



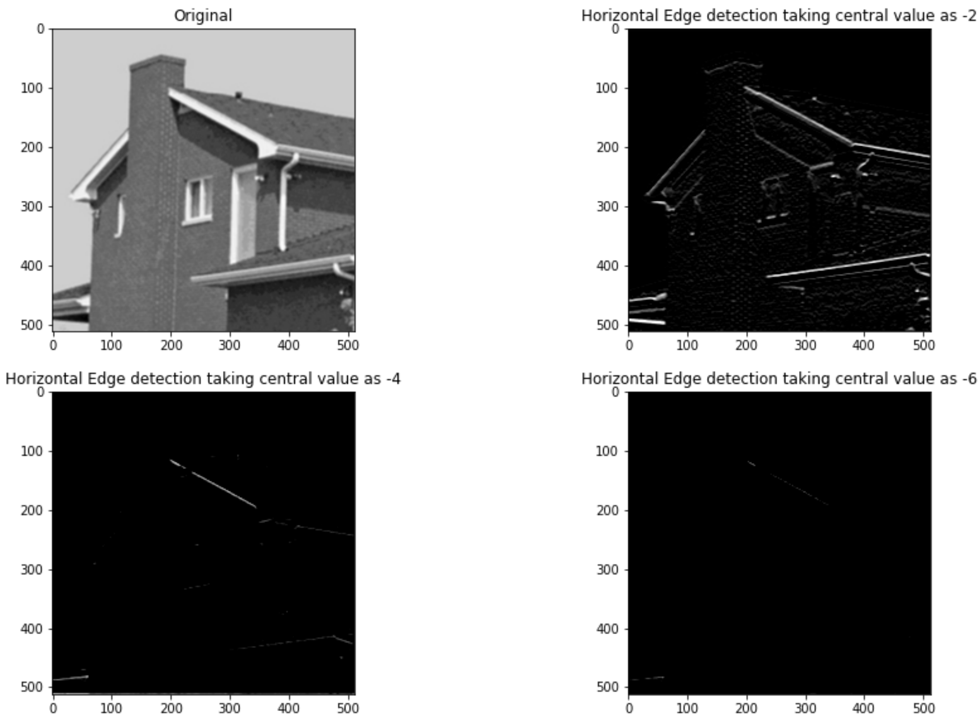
```
#Detect horizontal edges using in built convolution function
sobel_h_2= np.array([[-1,-6,-1], [0,0,0], [1,2,1]])
img_h_2= signal.convolve(img,sobel_h_2,mode='same')
plt.imshow(img_h_2,cmap="gray", vmin=0, vmax=255)
```

```
<matplotlib.image.AxesImage at 0x7f54ddd678d0>
```



```
plt.figure(figsize=(15,15))
plt.subplot(3,2,1)
plt.imshow(img, cmap="gray")
plt.title("Original")
plt.subplot(3,2,2)
plt.title("Horizontal Edge detection taking central value as -2")
plt.imshow(img_new_h, cmap="gray",vmin=0, vmax=255)
plt.subplot(3,2,3)
plt.title("Horizontal Edge detection taking central value as -4")
plt.imshow(img_h_1, cmap="gray",vmin=0, vmax=255)
plt.subplot(3,2,4)
plt.imshow(img_h_2, cmap="gray",vmin=0, vmax=255)
plt.title("Horizontal Edge detection taking central value as -6")
```

```
Text(0.5, 1.0, 'Horizontal Edge detection taking central value as -6')
```

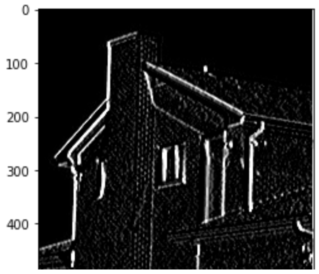



```
#Horizontal Detection as we increase the values the horizontal values becomes more and more clear
```

Vertical edges variation

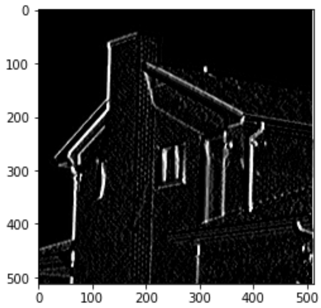
```
#Detect vertical edges using inbuilt function for convolution
sobel_v_1= np.array([[-1,0,1], [-8,0,8], [-1,0,1]])
img_v_1= signal.convolve(img,sobel_v_1,mode='same')
plt.imshow(img_v_1,cmap="gray", vmin=0, vmax=255)
```

<matplotlib.image.AxesImage at 0x7f54dde538d0>



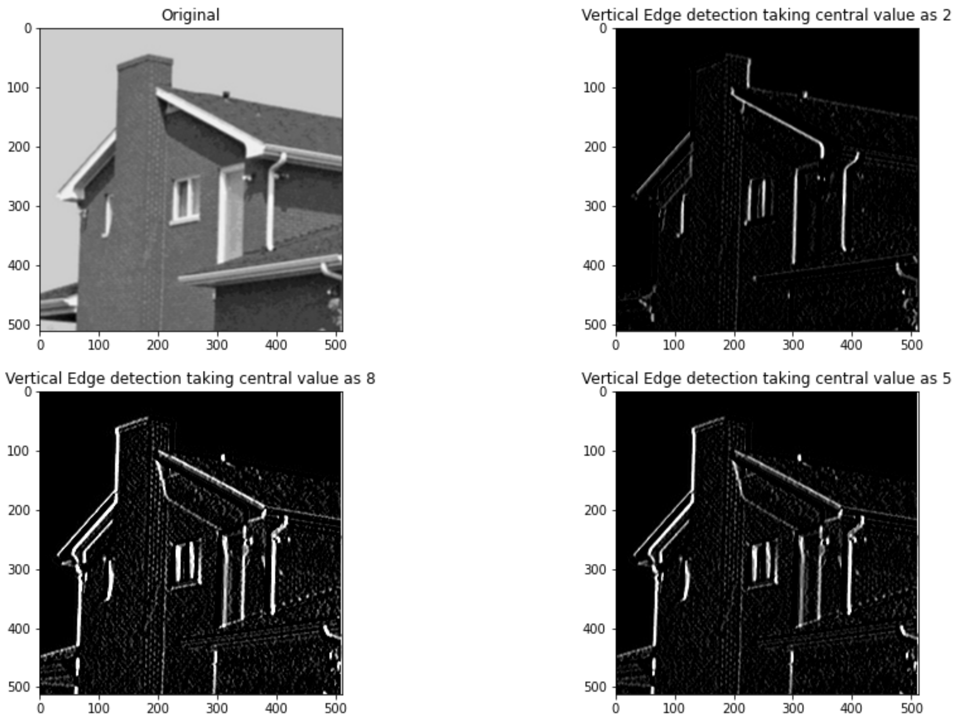
```
#Detect vertical edges using inbuilt function for convolution
sobel_v_2= np.array([[-1,0,1], [-5,0,5], [-1,0,1]])
img_v_2= signal.convolve(img,sobel_v_2,mode='same')
plt.imshow(img_v_2,cmap="gray", vmin=0, vmax=255)
```

<matplotlib.image.AxesImage at 0x7f54dd8abf50>



```
plt.figure(figsize=(15,15))
plt.subplot(3,2,1)
plt.imshow(img, cmap="gray")
plt.title("Original")
plt.subplot(3,2,2)
plt.title("Vertical Edge detection taking central value as 2")
plt.imshow(img_new_v, cmap="gray",vmin=0, vmax=255)
plt.subplot(3,2,3)
plt.title("Vertical Edge detection taking central value as 8")
plt.imshow(img_v_1, cmap="gray",vmin=0, vmax=255)
plt.subplot(3,2,4)
plt.imshow(img_v_2, cmap="gray",vmin=0, vmax=255)
plt.title("Vertical Edge detection taking central value as 5")
```

Text(0.5, 1.0, 'Vertical Edge detection taking central value as 5')



#Vertical Detection as we increase the values the vertical values becomes less and less clear