Experiment-5

Image Segmentation based on Edge based approach

Name : N U Praneeth Reddy   **Reg.No: 21BAI1500**

Aim: To create programs in Python to complete the given image segmentation tasks which include horizontal edges, vertical edges, diagonal edges, kernels and noisy image too.

Resources Used: Anaconda Python Environment

                Google Colab Jupyter Notebook

Theory :

      OpenCV stands as an open-source library designed for computer vision and machine learning applications. Its primary goal is to offer a unified foundation for computer vision projects and to facilitate the integration of machine perception into various commercial products.

      On the other hand, NumPy serves as a Python library, enabling support for large, multi-dimensional arrays and matrices, accompanied by an extensive array of high-level mathematical functions for manipulating these arrays.

      Additionally, Matplotlib functions as a Python plotting library, directly connected to the numerical mathematics capabilities of NumPy. It delivers an object-oriented API for seamlessly embedding plots within applications.
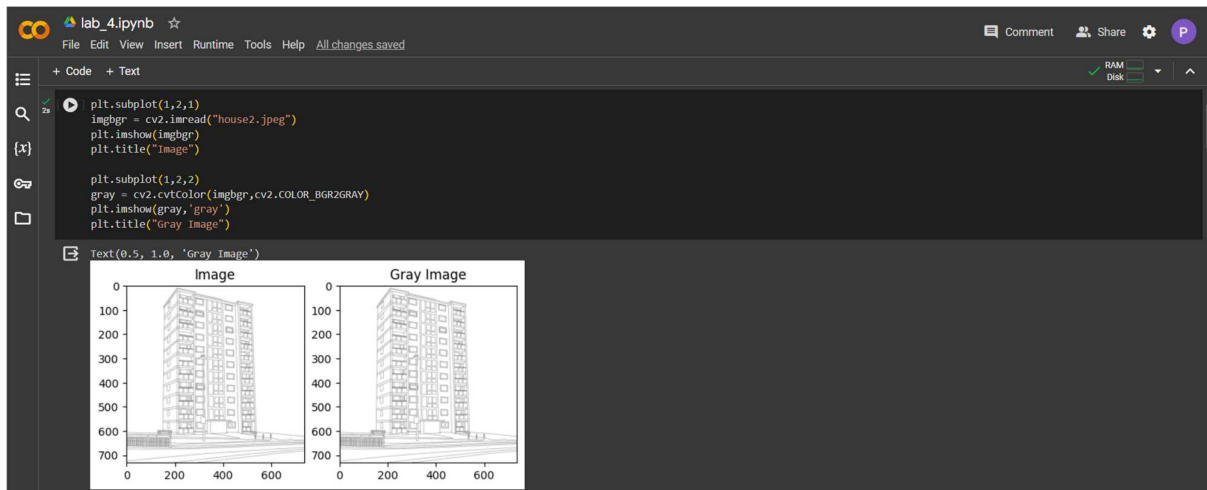
Tasks:

1) Extract the horizontal edges of an input image using a suitable edge mask.

2) Extract the vertical edges of an input image using a suitable edge mask.

3) Extract the diagonal edges of an input image using a suitable edge mask.

4) Perform Edge Detection based on the kernels: Roberts/ Sobel / Perwitt

5) Perform edge detection for the noise image given in the link below ,
   https://bit.ly/3SLYCmq

Procedure :

- Open Google Colab and create a new Jupyter Notebook.
- Import important libraries namely OpenCV, Numpy and Matplotlib.
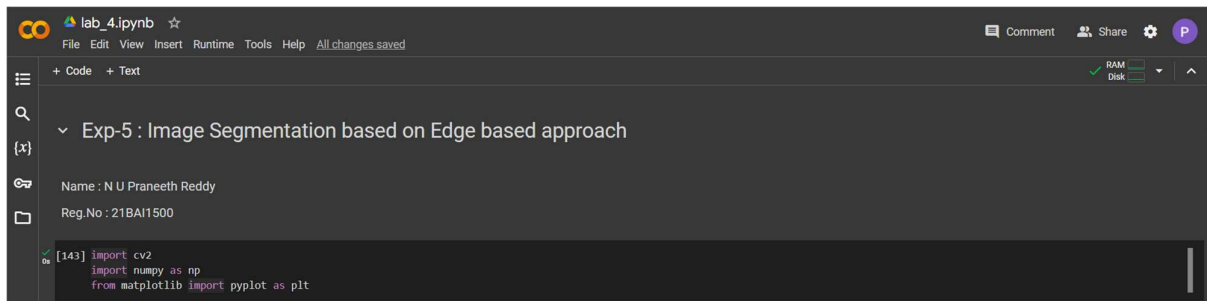
- Read the image using imread in the OpenCV library in BGR (Blue-Green-Red) format and convert it to gray scale and print both the images side by side using subplot.
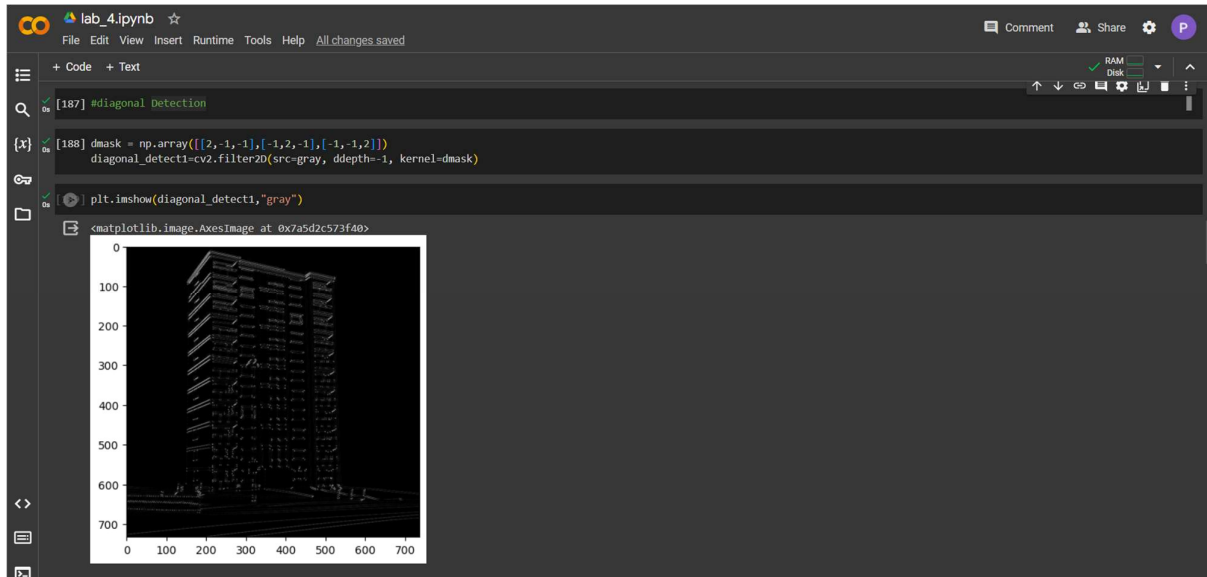
Task – 1 : Extract the horizontal edges of an input image using a suitable edge mask.
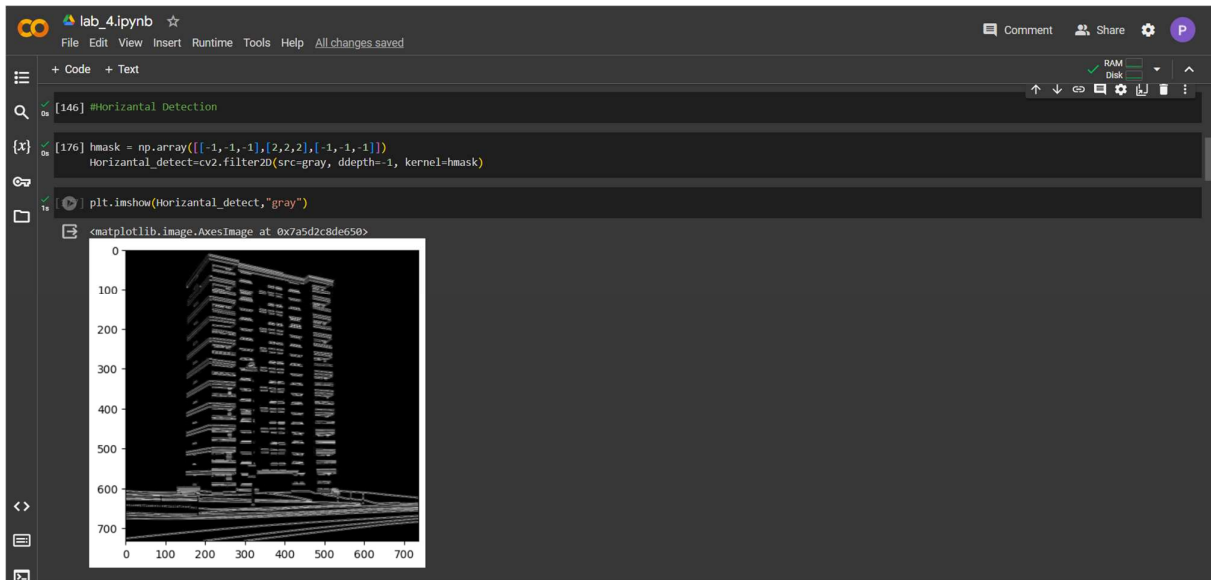
Task -2 : Extract the vertical edges of an input image using a suitable edge mask.
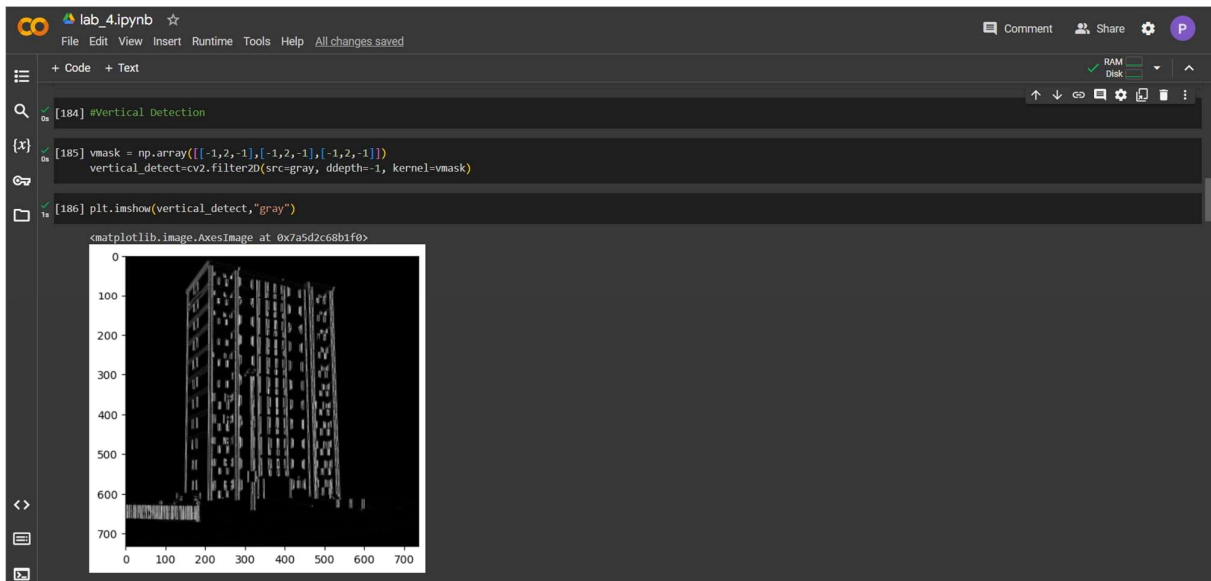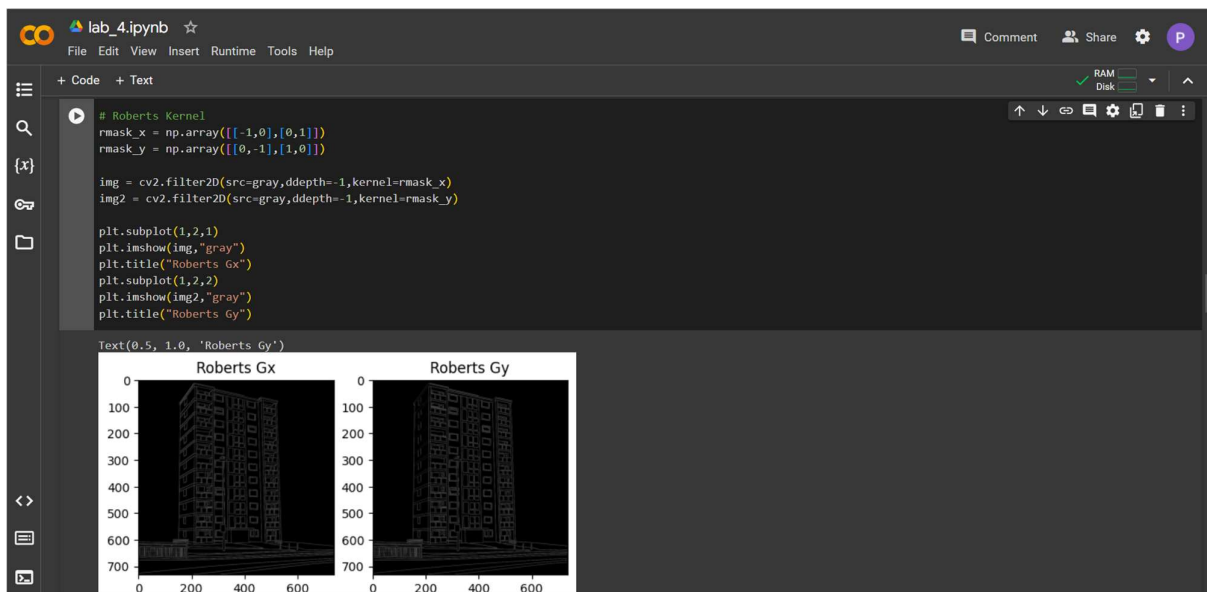
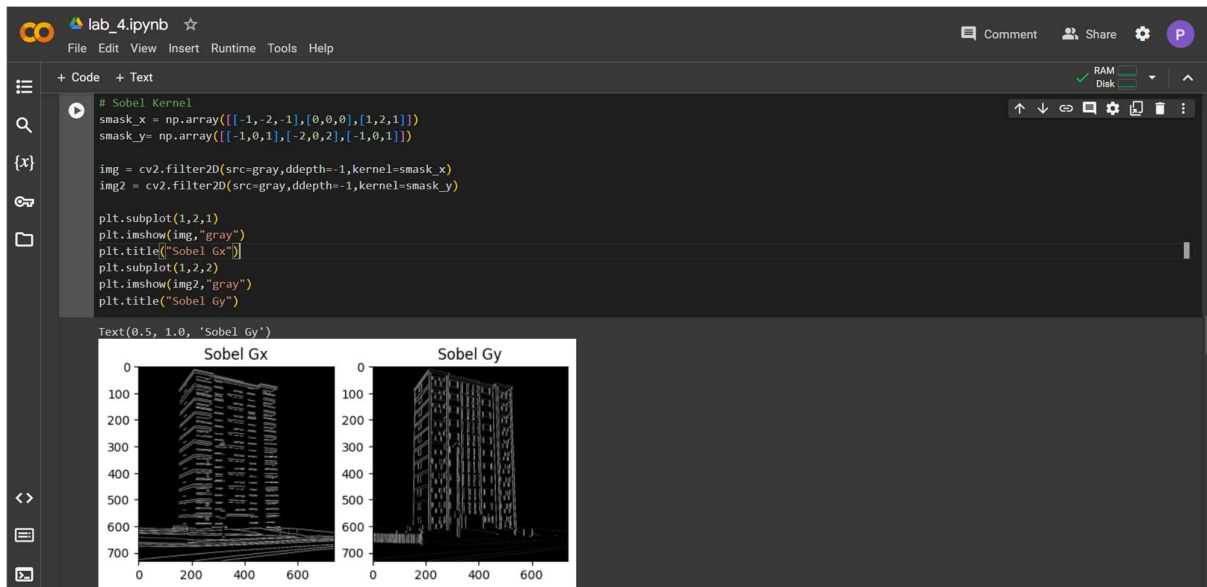**Task -3 :** Extract the diagonal edges of an input image using a suitable edge mask.

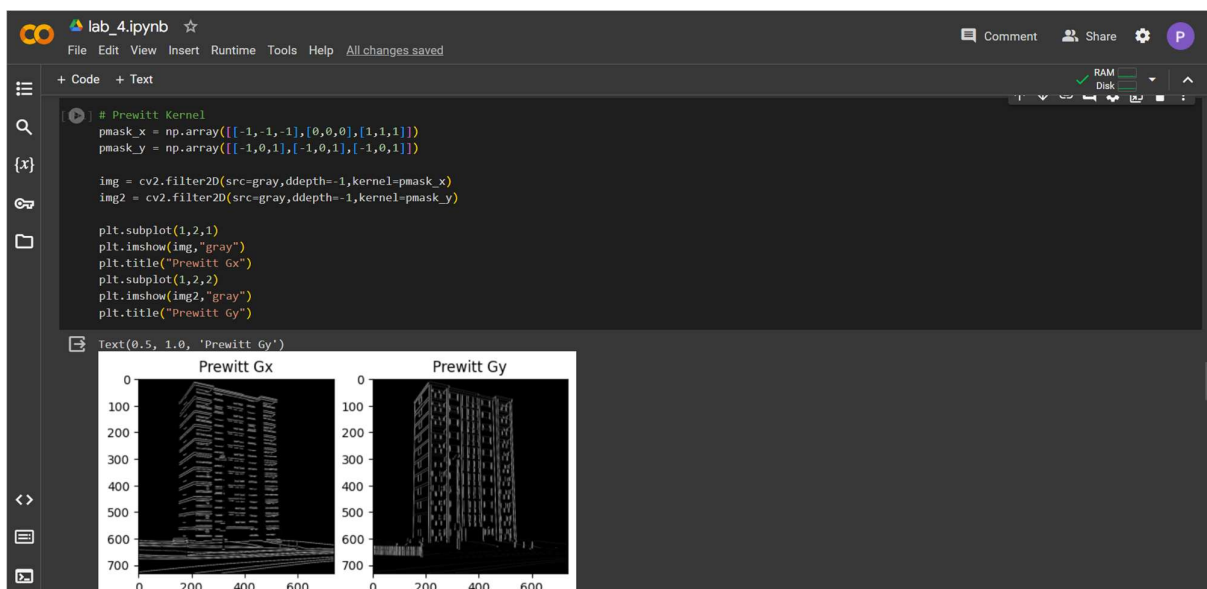**Task-4:** Perform Edge Detection based on the kernels: Roberts/ Sobel / Prewitt

- **Robert**

- ## **Sobel**


```python
# Sobel Kernel
smask_x = np.array([[-1,-2,-1],[0,0,0],[1,2,1]])
smask_y= np.array([[-1,0,1],[-2,0,2],[-1,0,1]])

img = cv2.filter2D(src=gray,ddepth=-1,kernel=smask_x)
img2 = cv2.filter2D(src=gray,ddepth=-1,kernel=smask_y)

plt.subplot(1,2,1)
plt.imshow(img,"gray")
plt.title("Sobel Gx")
plt.subplot(1,2,2)
plt.imshow(img2,"gray")
plt.title("Sobel Gy")
```

Text(0.5, 1.0, 'Sobel Gy')

- ## **Prewitt**


```python
# Prewitt Kernel
pmask_x = np.array([[-1,-1,-1],[0,0,0],[1,1,1]])
pmask_y = np.array([[-1,0,1],[-1,0,1],[-1,0,1]])

img = cv2.filter2D(src=gray,ddepth=-1,kernel=pmask_x)
img2 = cv2.filter2D(src=gray,ddepth=-1,kernel=pmask_y)

plt.subplot(1,2,1)
plt.imshow(img,"gray")
plt.title("Prewitt Gx")
plt.subplot(1,2,2)
plt.imshow(img2,"gray")
plt.title("Prewitt Gy")
```
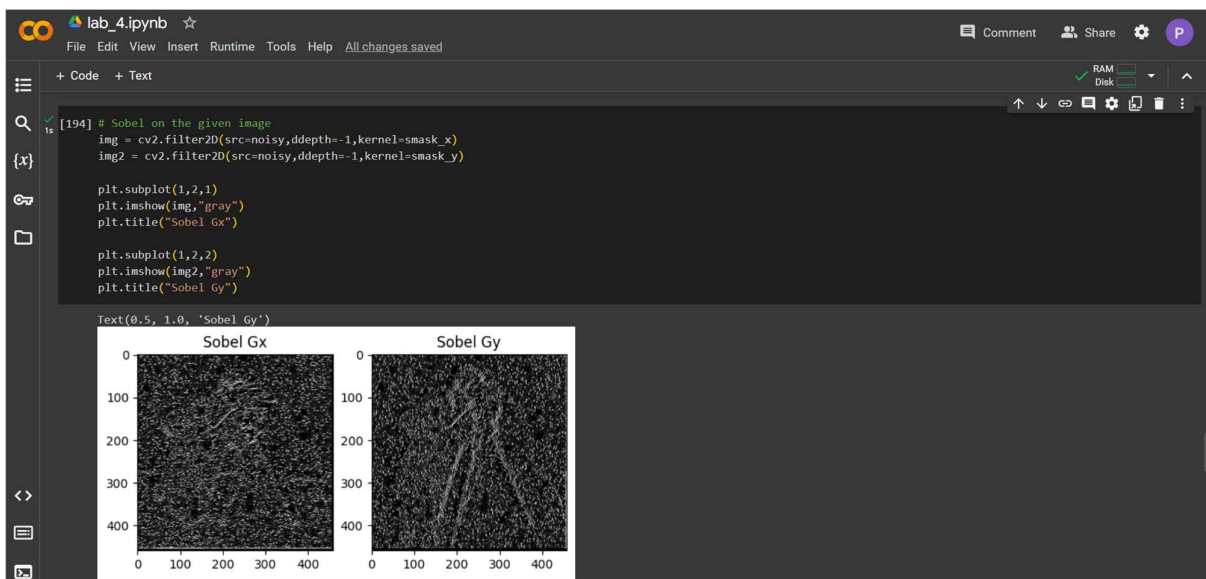
Text(0.5, 1.0, 'Prewitt Gy')

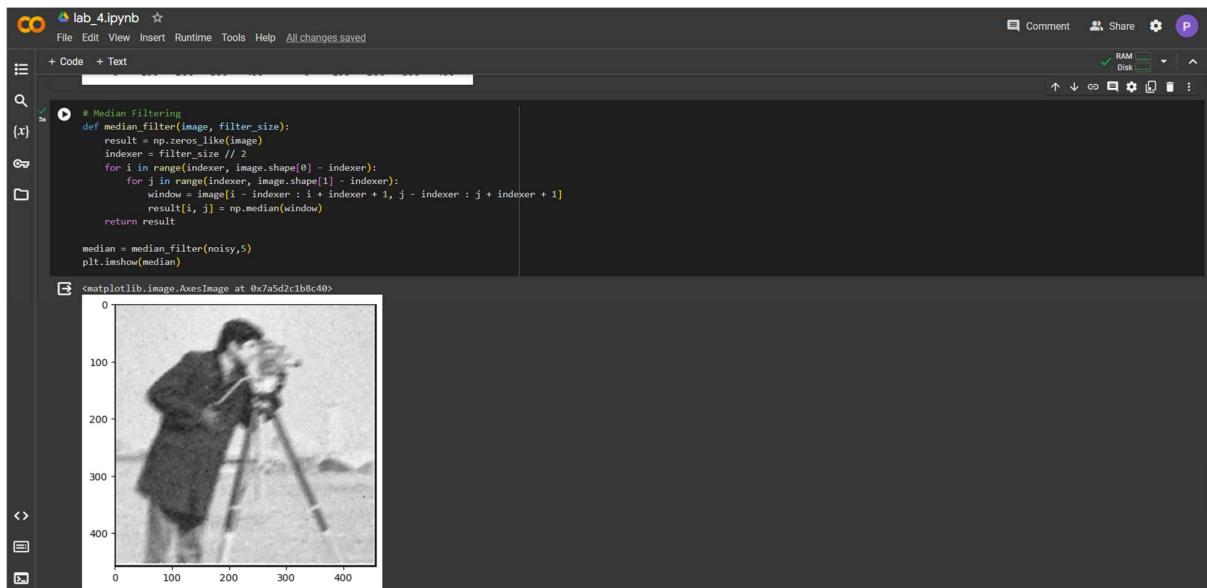**Task-5:** Perform edge detection for noise image given in the link below

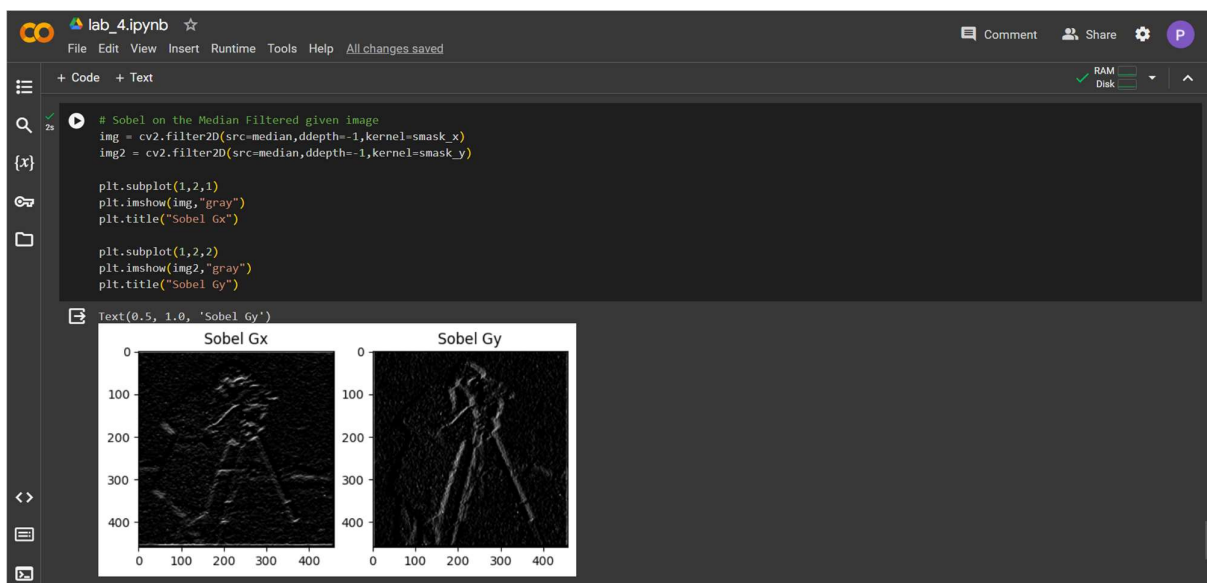- Sobel kernel on the given image

The edges are not detected properly due to the salt and pepper noise in the given image.

- Performing median filtering on the given image to remove the salt and pepper noise.

- Sobel Kernel on the median filtered image

We observe that the edges are still not detected properly due to the blur left behind after the noise removal using median filtering.

Results: The given tasks have been done using programs in Python using CV2, Matplotlib and numpy libraries.

Conclusion: Python programs has been created to perform edge segmentation for horizontal edges, vertical edges and diagonal edges using the suitable kernels. We have applied Roberts, Prewitt and Sobel kernel for edge segmentation. For the given image we observed salt and pepper noise, removed it using Median filtering and applied Sobel kernel for edge segmentation.

Google Collab Link :

https://colab.research.google.com/drive/1gdlvWeMnF0XiNmHLY5ATXB44xc6tJb8w?usp=sharing