Assignment 05

October 28, 2018

1 Syed Farhan Alam Zaidi

2 2018210031

3 Assignment 05

3.1 Image Convolution and Gradient

Github Link: https://github.com/farhan-93/assignment05.git

Import required libraries for execution of program.

```
In [2]: import matplotlib.pyplot as plt
        import numpy as np
        from scipy import signal
        from skimage import io, color
```

It this code section, the file is upload. In this portion, the kernel is defined for taking gradient of the image. A 3 × 3 kernel is used with weight of Sobel Kernel.

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

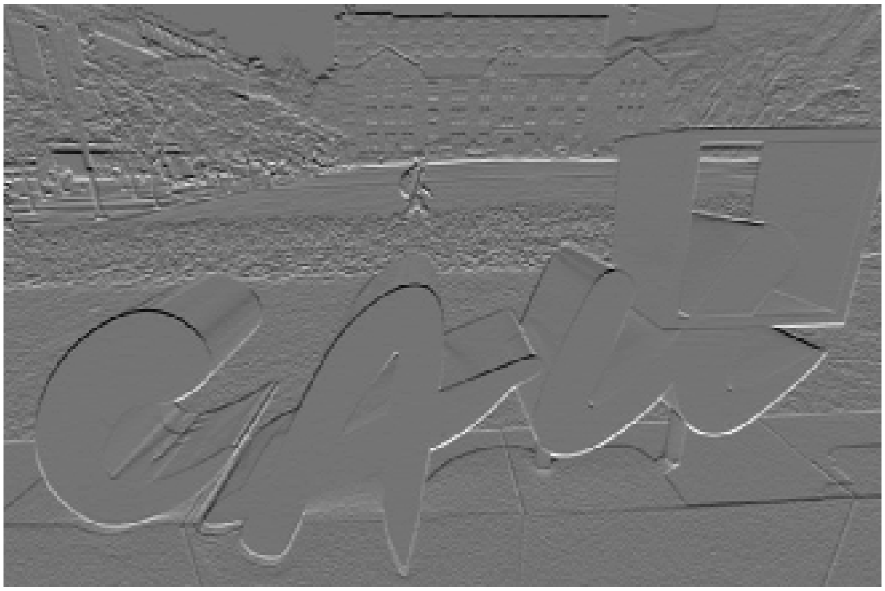
$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

```
In [26]: ############## Image Path ##############
         file_image         = 'cau1.jpg'

         ############## Reading File
         im_color= io.imread(file_image)

         ############# Converting RGB image to Gray Scale Image
         im_gray = color.rgb2gray(im_color)

         ############# Kernel for taking Gradient with respect to x-axis
         gx=np.array([[-1, 0, 1],[-2, 0, 2],[-1, 0, 1]])
```

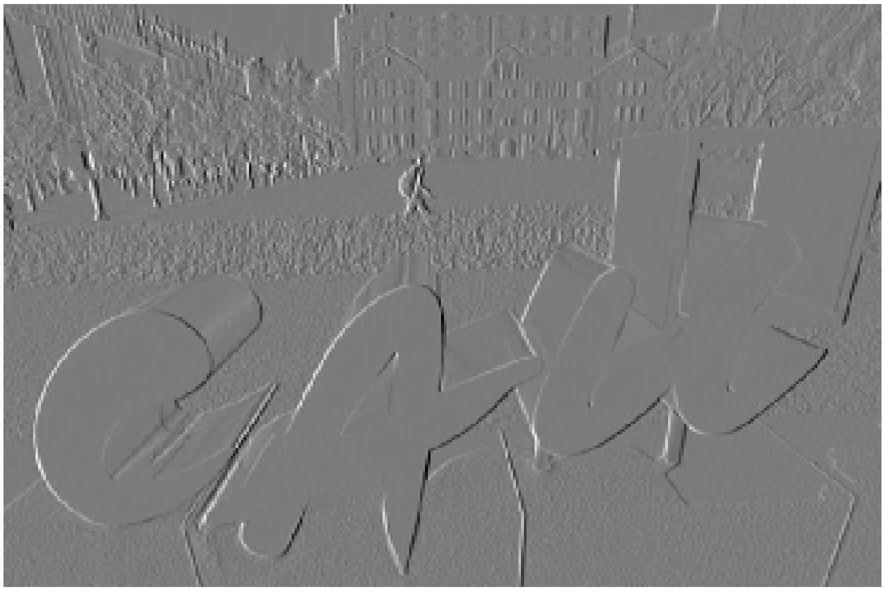
```
        ############# Kernel for taking Gradient with respect to y-axis
        gy=np.array([[1, 2, 1],[0, 0, 0],[-1, -2, -1]])

        ############# Plot Colored Image
        plt.imshow(im_color)
        plt.axis('off')

Out[26]: (-0.5, 739.5, 489.5, -0.5)
```

```
In [14]: ############# Plot Gray Scale Image
         plt.imshow(im_gray,cmap='gray')
         plt.axis('off')

Out[14]: (-0.5, 739.5, 489.5, -0.5)
```

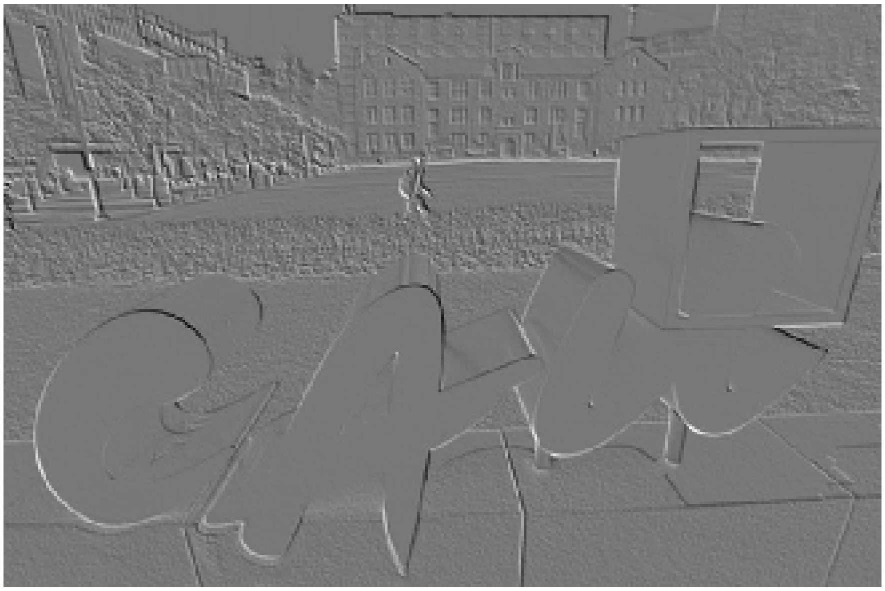
```
In [15]: ############# Convolve the image with "gy" to get gradient of the image with respect
         gradienty= signal.convolve2d(im_gray, gy, boundary='symm', mode='same')

         ############# Plot Gradient image w.r.t y-axis
         plt.imshow(gradienty, cmap='gray')
         plt.axis('off')

Out[15]: (-0.5, 739.5, 489.5, -0.5)
```



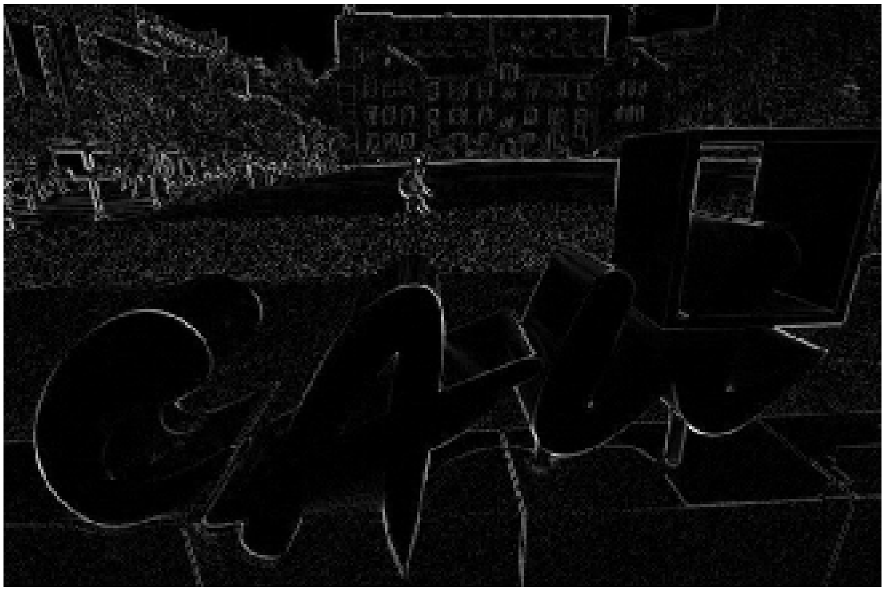
```
In [16]: ############# Convolve the image with "gx" to get gradient of the image with respect
         gradientx= signal.convolve2d(im_gray, gx, boundary='symm', mode='same')

         ############# Plot Gradient image w.r.t x-axis
         plt.imshow(gradientx, cmap='gray')
         plt.axis('off')

Out[16]: (-0.5, 739.5, 489.5, -0.5)
```



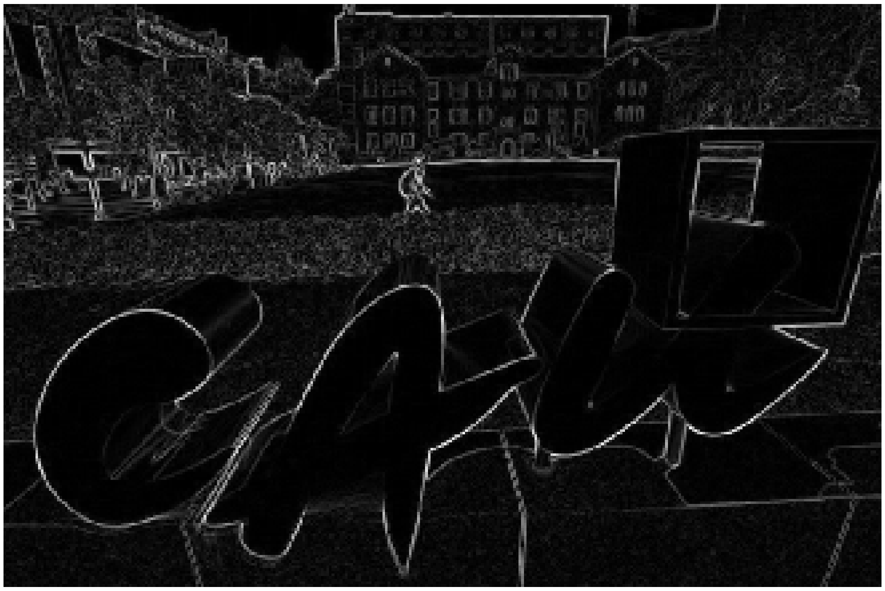
```
In [17]: ############# adding both numpy arrays gradientx and gradienty to get both gradient w
         grad=gradientx+gradienty

         ############# Plot the image with both gradient x and gradient y
         plt.imshow(grad, cmap='gray')
         plt.axis('off')

Out[17]: (-0.5, 739.5, 489.5, -0.5)
```



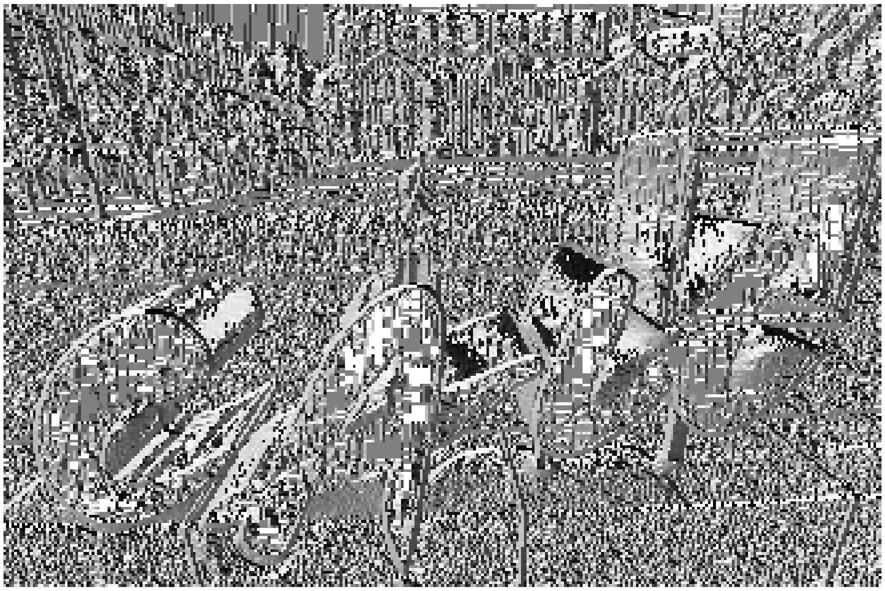
```
In [18]: ############# Find the absolute value of gradient image's numpy array
         absolute=np.absolute(grad)

         ############# Plot absolute gradient image
         plt.imshow(absolute, cmap='gray')
         plt.axis('off')

Out[18]: (-0.5, 739.5, 489.5, -0.5)
```



```
In [19]: ############# Calculate magnitude of gradient image.
         magnitude= np.sqrt(gradientx**2 + gradienty**2)

         ############# Plot magnitude of gradient image
         plt.imshow(magnitude, cmap='gray')
         plt.axis('off')

Out[19]: (-0.5, 739.5, 489.5, -0.5)
```



```
In [20]: ############# find the the direction of the gradient image
         direction = np.arctan2(gradientx, gradienty)

         ############# Plot directional image
         plt.imshow(direction, cmap='gray')
         plt.axis('off')

Out[20]: (-0.5, 739.5, 489.5, -0.5)
```



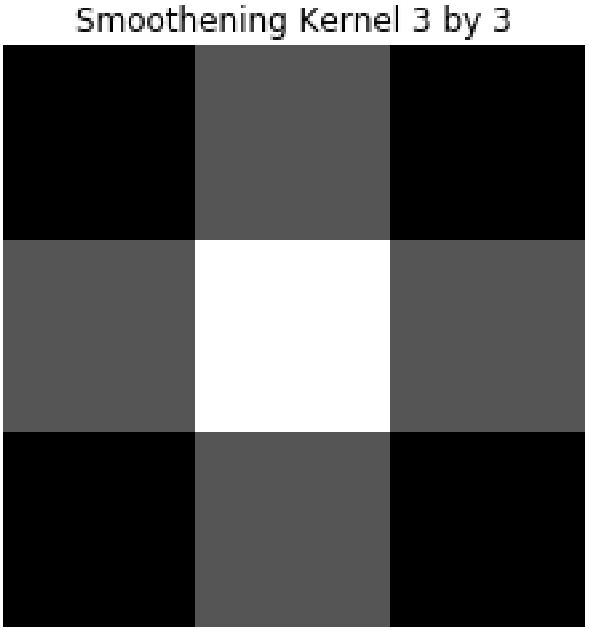
The Image will smooth using the Gaussian kernel.

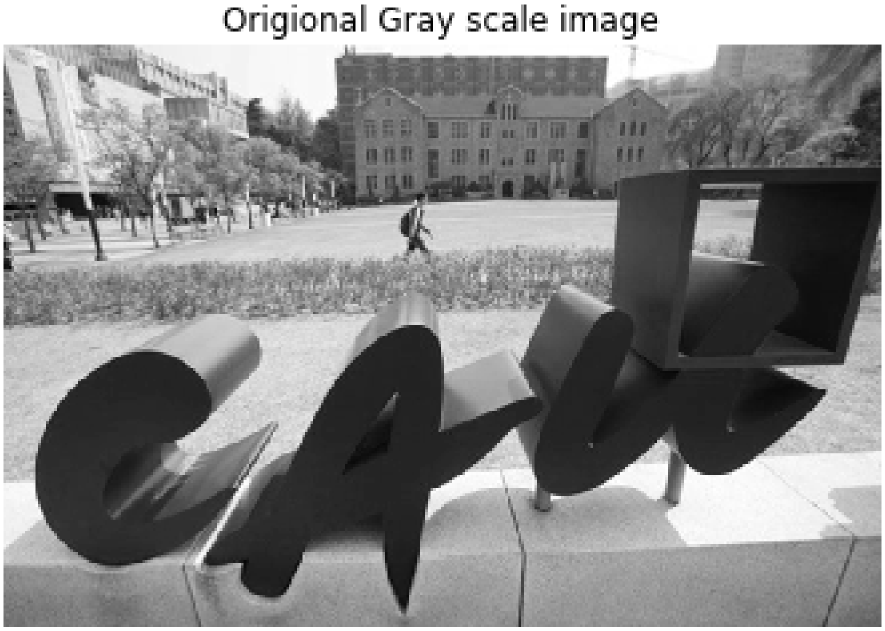
$$Gaussian = \frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix}$$

```
In [24]: ############# Define Kernel for smoothening the image, Inspired by Gaussian Kernel
         smooth_ker=[[1/16, 1/8, 1/16],[1/8, 1/4, 1/8],[1/16, 1/8, 1/16]]

         ############# Plot Smoothing Kernel
         plt.figure(2)
         plt.title("Smoothening Kernel 3 by 3")
         plt.imshow(smooth_ker, cmap='gray')
         plt.axis('off')

         ############# Convolve Gaussian Kernel on gray image for smoothening the image
         smoothimage= signal.convolve2d(im_gray, smooth_ker, boundary='symm', mode='same')

         ############# Plot Origional Gray scale image
         plt.figure(3)
         plt.title("Origional Gray scale image")
         plt.imshow(im_gray, cmap='gray')
         plt.axis('off')

         ############# Plot Smooth image
         plt.figure(4)
         plt.title("Image after Convolution with Soothening Kernel")
```

```
        plt.imshow(smoothimage, cmap='gray')
        plt.axis('off')
```

```
Out[24]: (-0.5, 739.5, 489.5, -0.5)
```

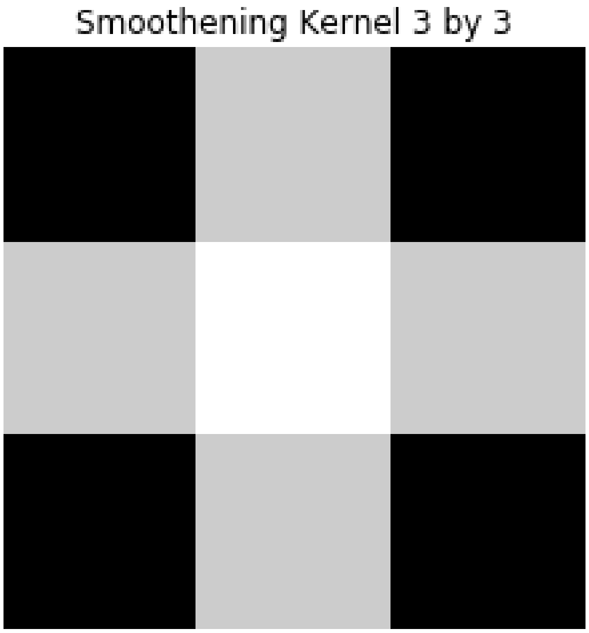




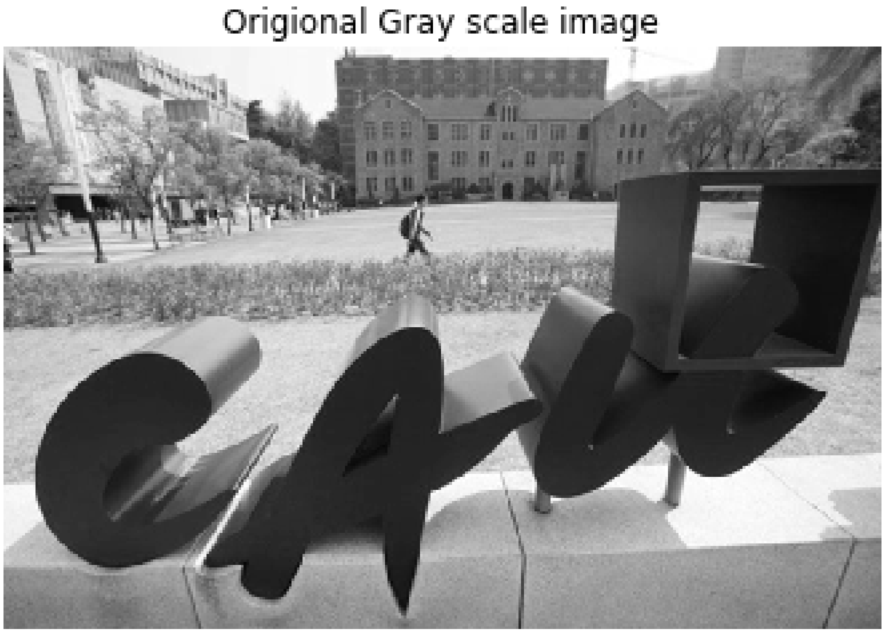
In [25]:
```
############# Define Own Kernel
own_kernel=[[-3,1,-3],[1, 2, 1],[-3, 1, -3]]


############# Plot Own Kernel
plt.figure(2)
plt.title("Smoothening Kernel 3 by 3")
plt.imshow(own_kernel, cmap='gray')
plt.axis('off')

############# Convolve Own Kernel on gray image
ownimage= signal.convolve2d(im_gray, own_kernel, boundary='symm', mode='same')

############# Plot Origional Gray scale image
plt.figure(3)
plt.title("Origional Gray scale image")
plt.imshow(im_gray, cmap='gray')
plt.axis('off')

############# Plot image after convolution
plt.figure(4)
plt.title("Image after Convolution")
```

```
        plt.imshow(ownimage, cmap='gray')
        plt.axis('off')
```

```
Out[25]: (-0.5, 739.5, 489.5, -0.5)
```



Smoothening Kernel 3 by 3



Origional Gray scale image

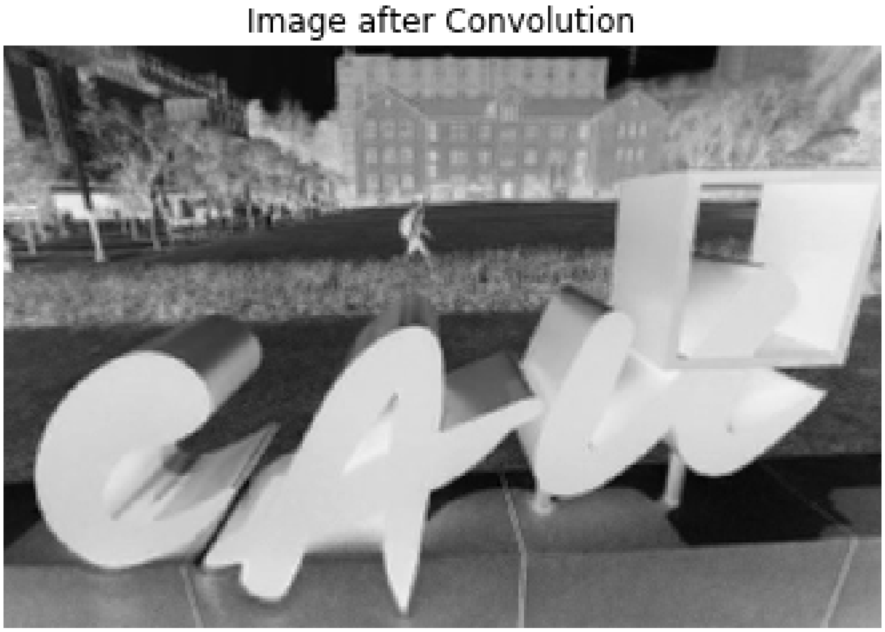


Image after Convolution