

# Code for Gaussian Kernel -

```
def gaussian_kernel(sigma_x,sigma_y):  
    k_size_x = int (5*sigma_x)  
    k_size_y = int (5*sigma_y)  
    if(k_size_x % 2 == 0):  
        k_size_x+=1  
    if(k_size_y % 2 == 0):  
        k_size_y+=1  
    norm = 1/( 2*3.141592*sigma_x*sigma_y  
    gaussian = np.zeros((k_size_x,k_size_y),np.float32  
    for x in range(k_size_x):  
        for y in range(k_size_y):  
            px = (x**2)/(sigma_x**2)  
            py = (y**2)/(sigma_y**2)  
            p = (px + py )/2  
            p = math.exp(-p)  
            gaussian[x,y]= p*norm  
    print(gaussian)  
    return gaussian
```

## Gaussian Filter (Grayscale)

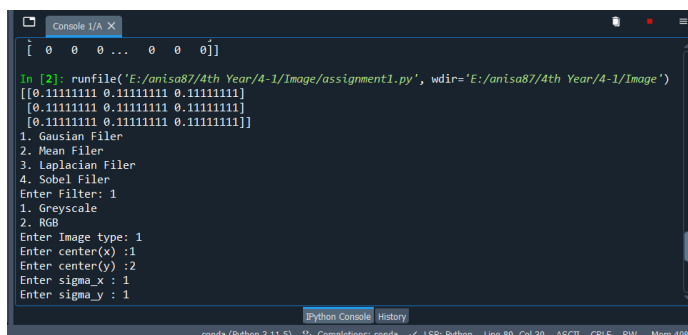


Fig 1.1 : Selection of Gaussian Filter and Grayscale image

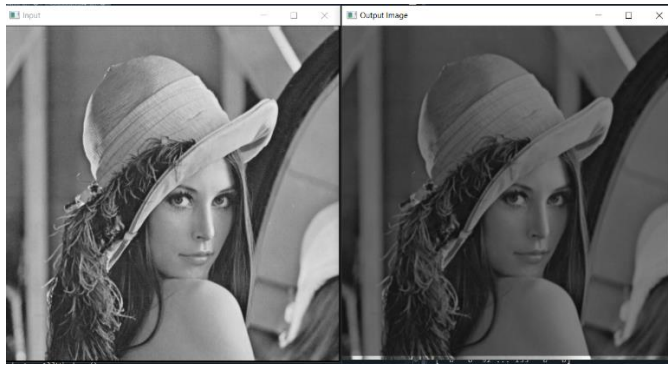


Fig 1.2 : Grayscale image before and after convolution

## Gaussian Filter (RGB)

```

IPython 8.15.0 -- An enhanced Interactive Python.

In [1]: runfile('E:/anisa87/4th Year/4-1/Image/assignment1.py', wdir='E:/anisa87/4th Year/4-1/Image')
[[0.11111111 0.11111111 0.11111111]
 [0.11111111 0.11111111 0.11111111]
 [0.11111111 0.11111111 0.11111111]]
1. Gaussian Filer
2. Mean Filer
3. Laplacian Filer
4. Sobel Filer
Enter Filter: 1
1. Greyscale
2. RGB
Enter Image type: 2
Enter center(x) :1
Enter center(y) :2
Enter sigma_x : 1
Enter sigma_y : 1
  
```

Fig 1.3 : Selection of Gaussian Filter and RGB image

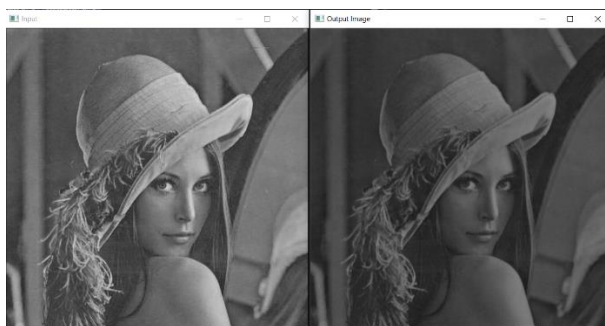


Fig 1.4: Blue Image for RGB

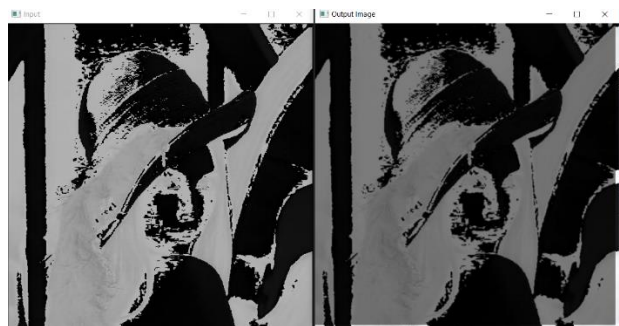


Fig 1.5: Blue Image for HSV

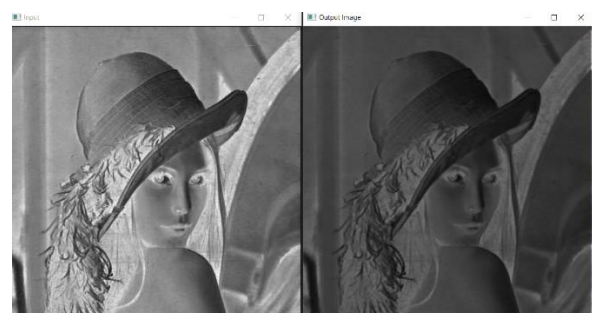
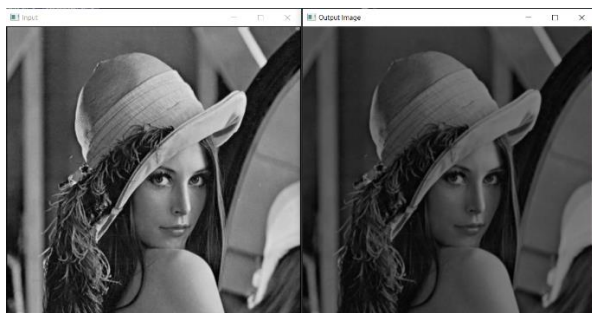


Fig 1.6 : Green Image for RGB

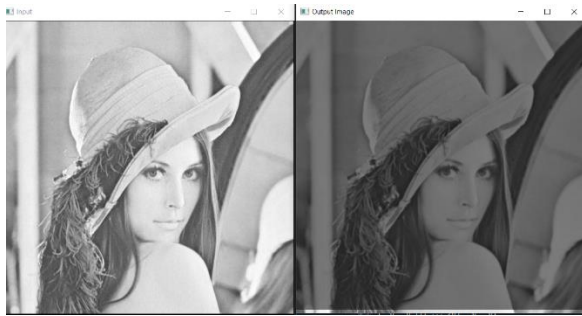


Fig 1.7 : Green Image for HSV

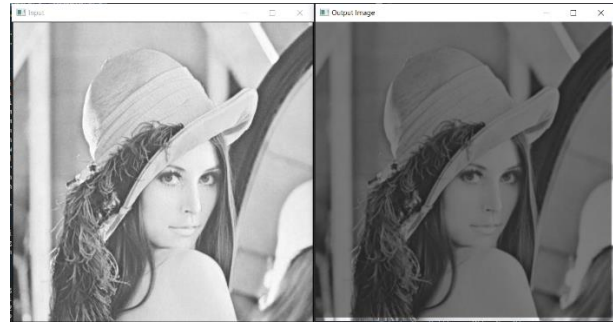


Fig 1.8 : Red Image for RGB

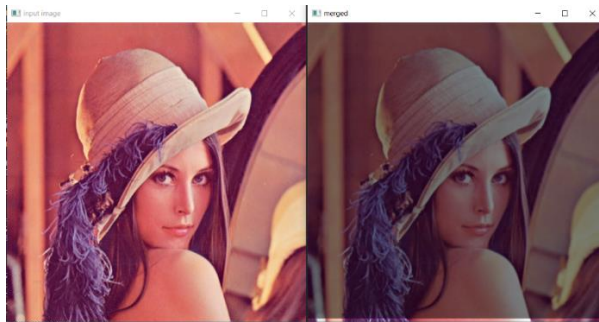


Fig 1.9 : Red Image for HSV

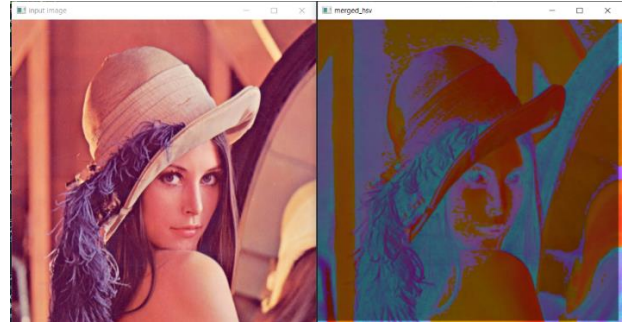


Fig 1.10 : Merge after convolution

Fig 1.11 : Merge after Concolution HSV

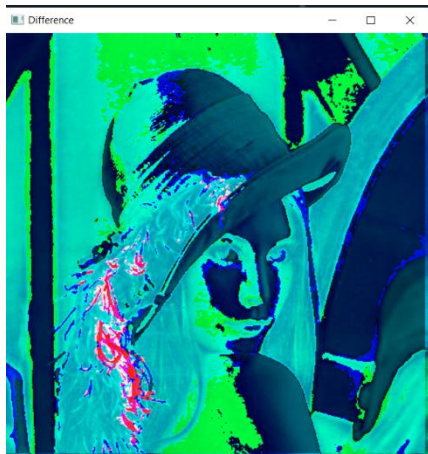
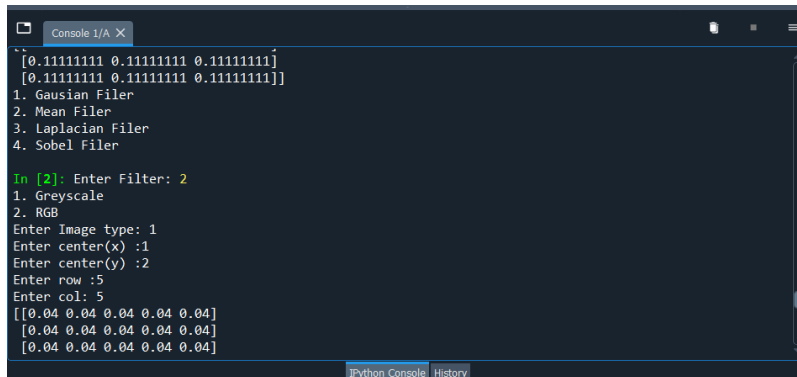


Fig 1.12: Difference of RGB and HSV convoluted image

## Code for Mean Kernel -

```
def mean_kernel(row,col):  
    meann = (1 / (row * col)) * np.ones((row,col), dtype=np.uint8)  
    print(meann)  
    return meann
```

## Mean Filter (Grayscale)



```
Console 1/A X  
[[0.11111111 0.11111111 0.11111111]  
 [0.11111111 0.11111111 0.11111111]]  
1. Gaussian Filer  
2. Mean Filer  
3. Laplacian Filer  
4. Sobel Filer  
  
In [2]: Enter Filter: 2  
1. Greyscale  
2. RGB  
Enter Image type: 1  
Enter center(x) :1  
Enter center(y) :2  
Enter row :5  
Enter col: 5  
[[0.04 0.04 0.04 0.04 0.04]  
 [0.04 0.04 0.04 0.04 0.04]  
 [0.04 0.04 0.04 0.04 0.04]]  
Python Console History
```

Fig 1.12: Selection of Mean filter in Grayscale image

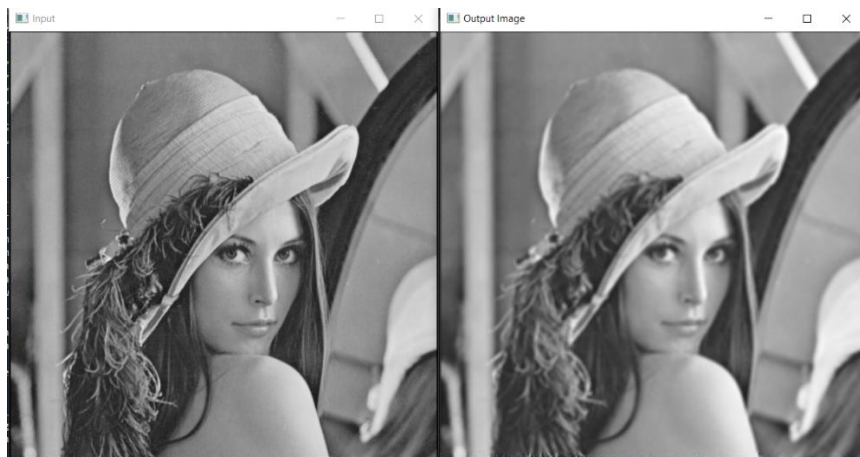


Fig 1.13: Grayscale image

# Mean Filter (RGB)

```
In [3]: runfile('E:/anisa87/4th Year/4-1/Image/assignment1.py', wdir='E:/anisa87/4th Year/4-1/Image')
[[0.11111111 0.11111111 0.11111111]
 [0.11111111 0.11111111 0.11111111]
 [0.11111111 0.11111111 0.11111111]]
1. Gaussian Filer
2. Mean Filer
3. Laplacian Filer
4. Sobel Filer
Enter Filter: 2
1. Greyscale
2. RGB
Enter Image type: 2
Enter center(x) :1
Enter center(y) :2
Enter row :5
Enter col: 5
[[0.04 0.04 0.04 0.04 0.04]]
```

Fig 1.14: Selection of Mean filter in RGB image

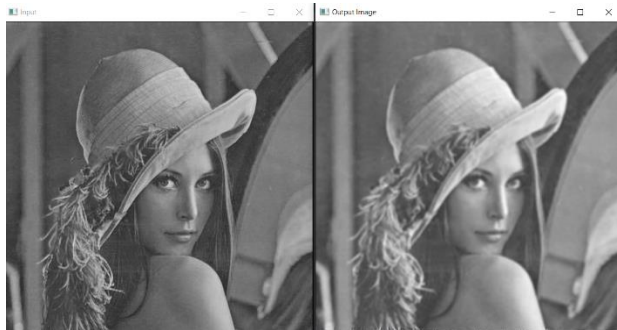


Fig 1.15: Blue Image



Fig 1.16: Blue Image HSV

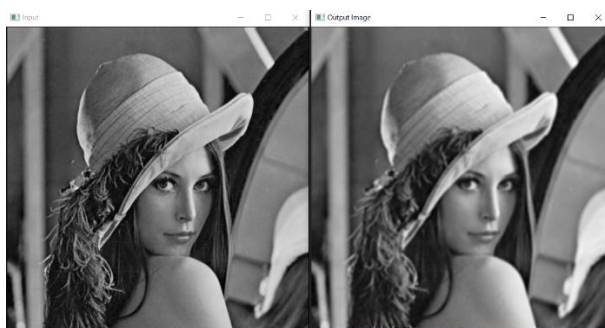


Fig 1.17: Green Image

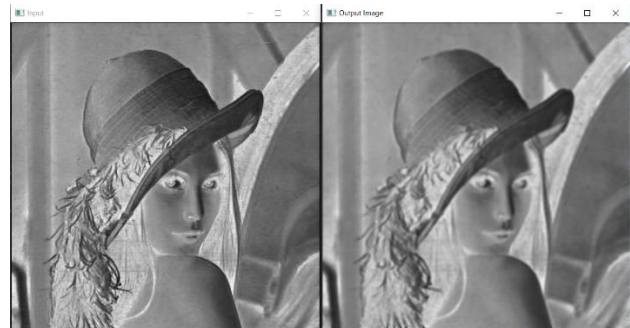


Fig 1.18 : Green Image HSV



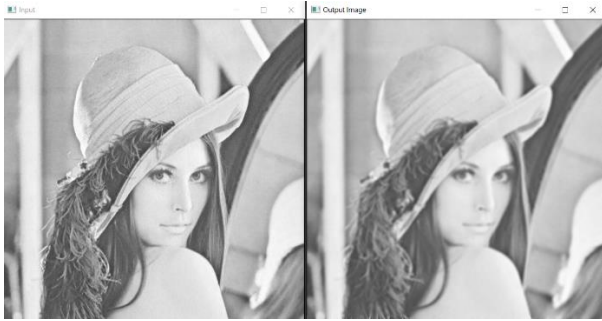


Fig 1.19: Red Image

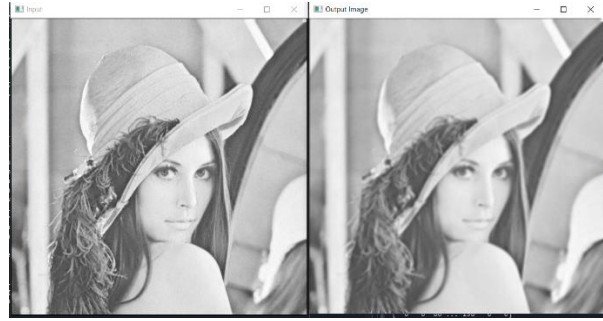


Fig 1.20 :Red Image HSV

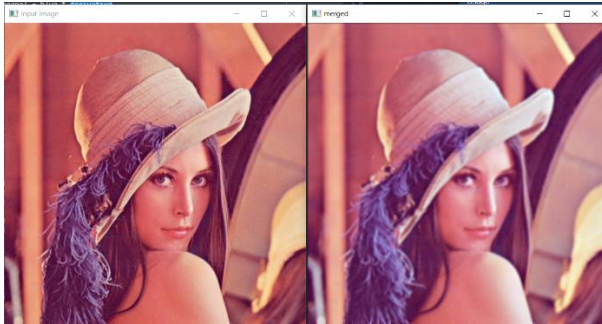


Fig 1.21: Merged Image after Convolution

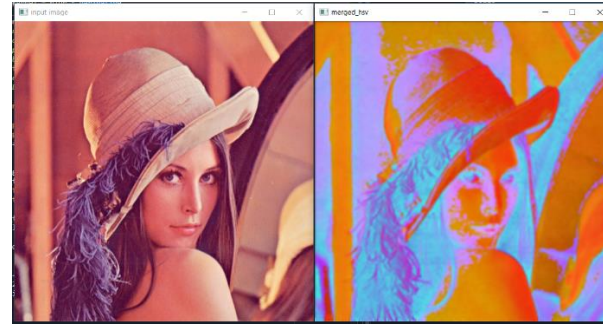


Fig 1.22: Merged Image after Convolution HSV

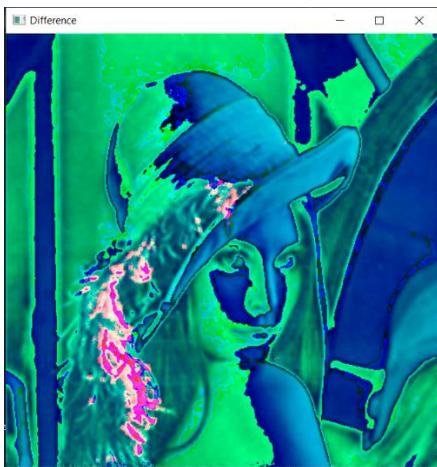
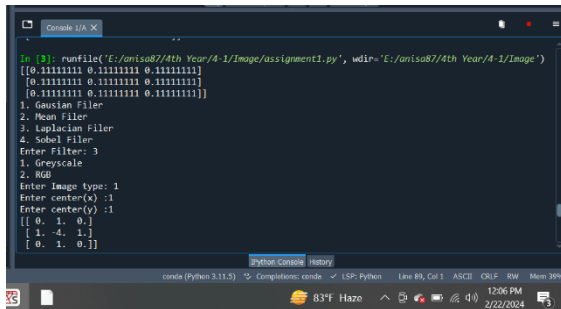


Fig 1.23 :Difference of RGB and HSV convoluted image

# Code for Laplacian Kernel -

```
def laplacian_kernel():  
    laplacian = np.array([[0,1,0],  
                          [1,-4,1],  
                          [0,1,0]],np.float32)  
    print(laplacian)  
    return laplacian
```

## Laplacian Filter(Grayscale)



```
In [3]: runfile('E:/uniso87/4th Year/4-1/Image/assignment1.py', wdir='E:/uniso87/4th Year/4-1/Image')  
[[0.11111111 0.11111111 0.11111111]  
 [0.11111111 0.11111111 0.11111111]  
 [0.11111111 0.11111111 0.11111111]]  
1. Gaussian Filter  
2. Mean Filter  
3. Laplacian Filter  
4. Sobel Filter  
Enter Filter:- 3  
1. Greyscale  
2. RGB  
Enter Image type: 1  
Enter center(x) :1  
Enter center(y) :1  
[[ 0.  1.  0.]  
 [ 1. -4.  1.]  
 [ 0.  1.  0.]]
```

Fig 1.24: Selection of Mean filter in Grayscale image

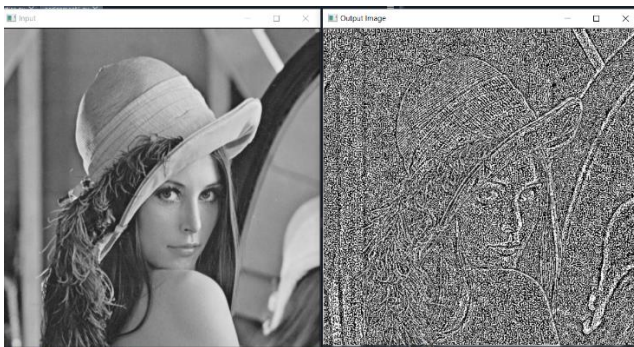


Fig 1.25: Grayscale image

## Laplacian Filter (RGB)



```

In [4]: runfile('E:/anisa87/4th Year/4-1/Image/assignment1.py', wdir='E:/anisa87/4th Year/4-1/Image')
[[0.11111111 0.11111111 0.11111111]
 [0.11111111 0.11111111 0.11111111]
 [0.11111111 0.11111111 0.11111111]]
1. Gaussian Filter
2. Mean Filter
3. Laplacian Filter
4. Sobel Filter
Enter Filter: 3
1. Greyscale
2. RGB
Enter Image type: 2
Enter center(x) :1
Enter center(y) :1
[[ 0.  1.  0.]
 [ 1. -4.  1.]
 [ 0.  1.  0.]]

```

Fig 1.26: Selection of Laplacian filter in RGB image

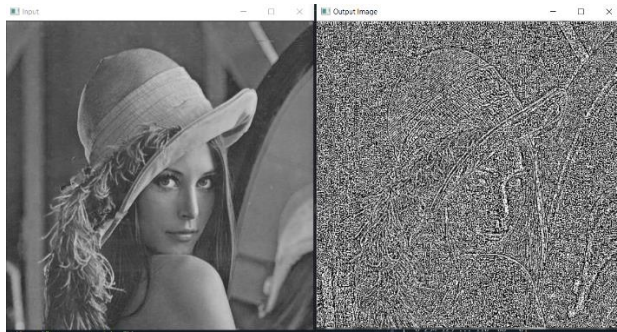


Fig 1.27: Blue Image



Fig 1.28: Blue Image HSV

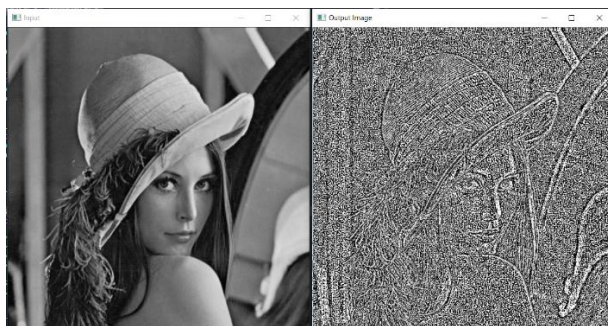


Fig 1.29: Green Image

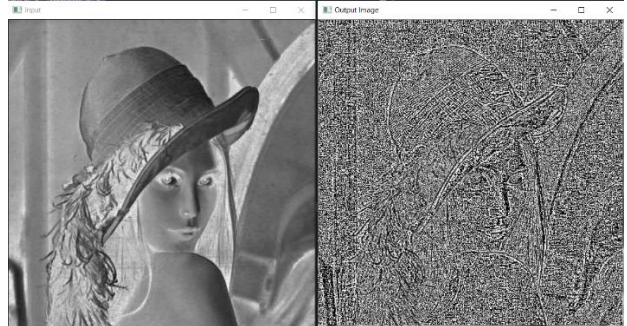


Fig 1.30: Green Image HSV

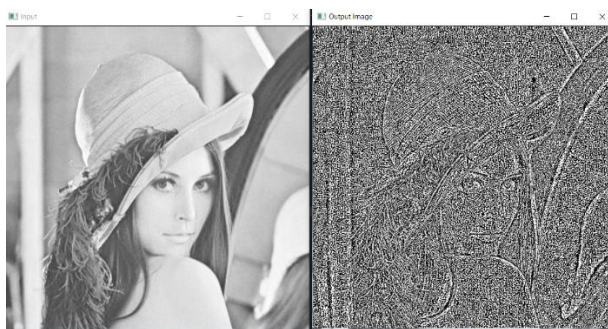


Fig 1.31: Red Image

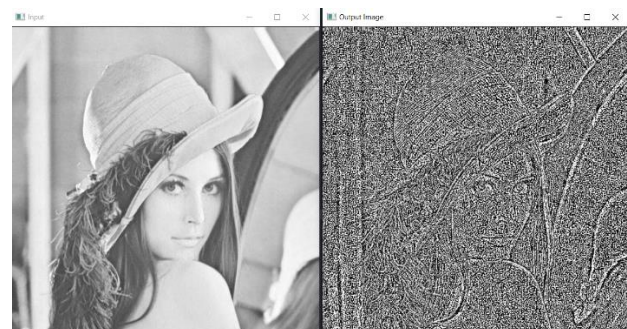


Fig 1.32: Red Image HSV



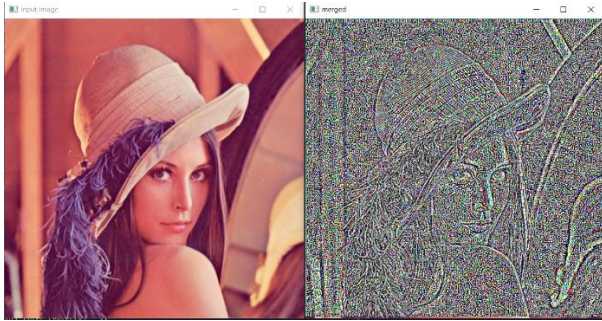


Fig 1.33: Merged Image after Convolution

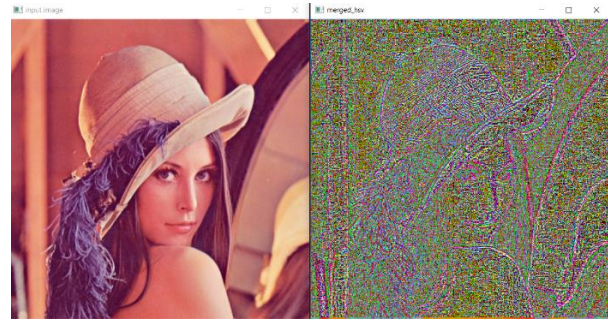


Fig 1.34: Merged Image after Convolution HSV

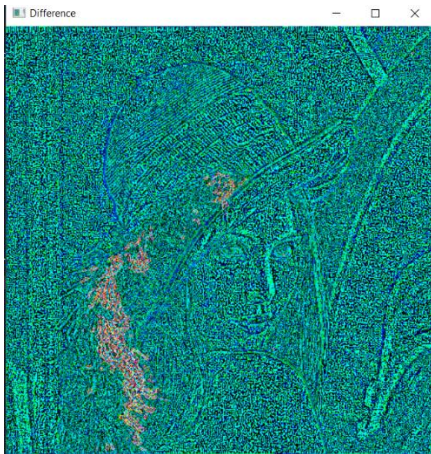


Fig 1.35 :Difference of RGB and HSV convoluted image

## Code for Sobel Kernel -

```
def sobel_kernel():
    blur = np.array([[1],
                     [2],
                     [1]], dtype=np.float32)

    derivative = np.array([1, 0, -1], dtype=np.float32)
    h_sobel_kernel = blur * derivative
    return h_sobel_kernel
```

## Sobel Filter (Grayscale)

```

Console I/A X
In [5]: runfile('E:/anisa87/4th Year/4-1/Image/assignment1.py', wdir='E:/anisa87/4th Year/4-1/Image')
[[0.11111111 0.11111111 0.11111111]
 [0.11111111 0.11111111 0.11111111]
 [0.11111111 0.11111111 0.11111111]]
1. Gaussian Filter
2. Mean Filter
3. Laplacian Filter
4. Sobel Filter
Enter Filter: 4
1. Grayscale
2. RGB
Enter Image type: 1
Enter center(x) :1
Enter center(y) :1
[[ 0  0  0 ...  0  0  0]
 [ 0 230 252 ... 154 130  0]
 [ 0 136 255 ... 155 126  0]
 ...
Python Console History

```

Fig 1.36: Selection of Sobel filter in Grayscale image

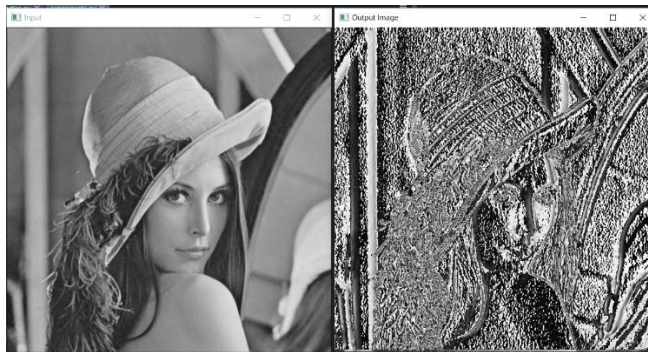


Fig 1.37: Grayscale image

## Sobel Filter (RGB)

```

Console I/A X
[ 0  42  44 ... 106 109  0]
[ 0  0  0 ...  0  0  0]]
In [6]: runfile('E:/anisa87/4th Year/4-1/Image/assignment1.py', wdir='E:/anisa87/4th Year/4-1/Image')
[[0.11111111 0.11111111 0.11111111]
 [0.11111111 0.11111111 0.11111111]
 [0.11111111 0.11111111 0.11111111]]
1. Gaussian Filter
2. Mean Filter
3. Laplacian Filter
4. Sobel Filter
Enter Filter: 4
1. Grayscale
2. RGB
Enter Image type: 2
Enter center(x) :1
Enter center(y) :1
Python Console History

```

Fig 1.38: Fig 1.26: Selection of Sobel filter in RGB image

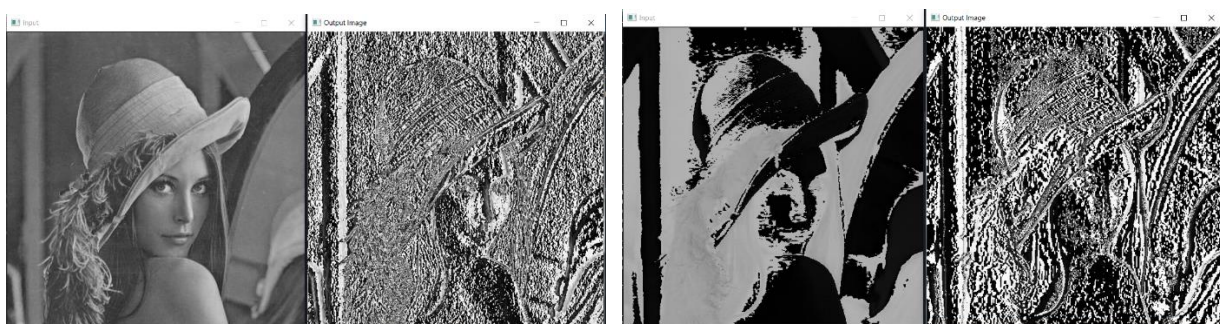




Fig 1.39: Blue Image

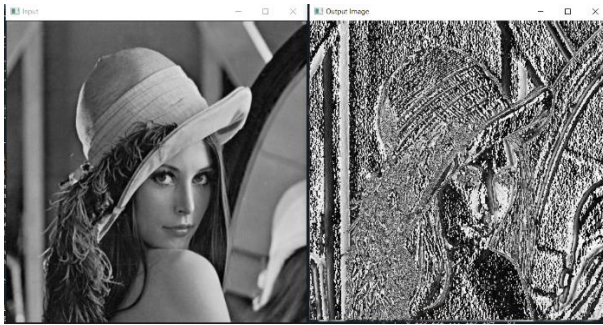


Fig 1.40: Blue Image HSV

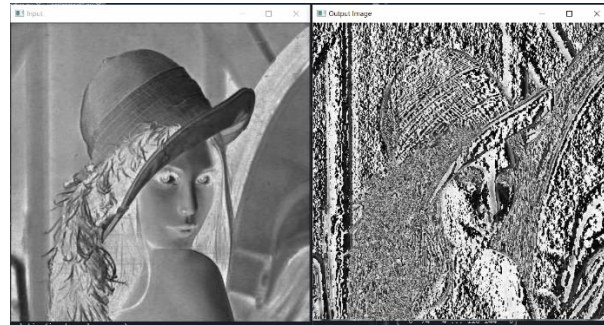


Fig 1.41: Green Image

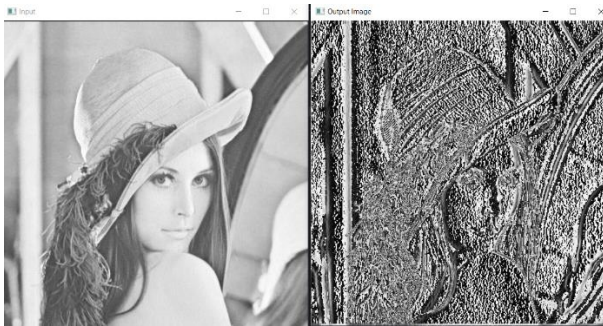


Fig 1.42: Green Image HSV

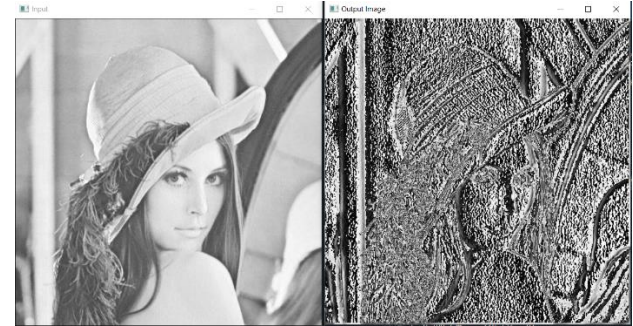


Fig 1.43: Red Image

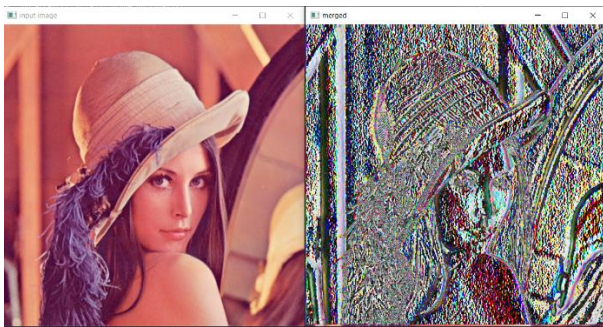


Fig 1.44: Red Image HSV

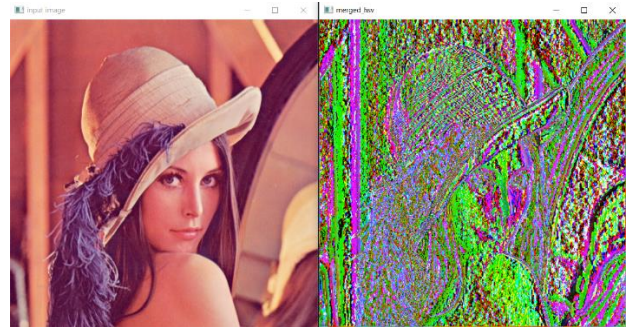


Fig 1.45: Merged Image after Convolution

Fig 1.46: Merged Image after Convolution HSV

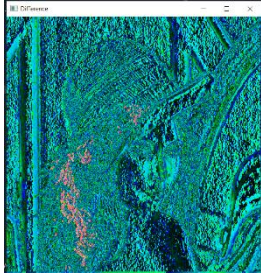


Fig 1.47: Difference of RGB and HSV convoluted image