الجامعة الإسلامية العالمية شيتاغونغ
International Islamic University Chittagong

Department of Computer Science and Engineering

Submitted By:

Student Id:	C181208
Name:	Sameha Hasan
Section:	8AF
Course Code:	CSE-4875
Course Title:	Pattern Recognition and Image Processing sessional
Email:	samehasan25@gmail.com

Submitted To:

Mr. Mohammad Mahadi Hassan

Associate Professor,

Dept. of CSE , IIUC.

Lab 08

Sharpening Filtering

- **Highpass**
- **Highboost**
- **Derivative filtering**
  - **Prewitt**
  - **Sobel**
  - **Roberts**
  - **Laplacian**
  - **Canny**

upyter C181208 Lab8 IP Last Checkpoint: Last Wednesday at 08:54

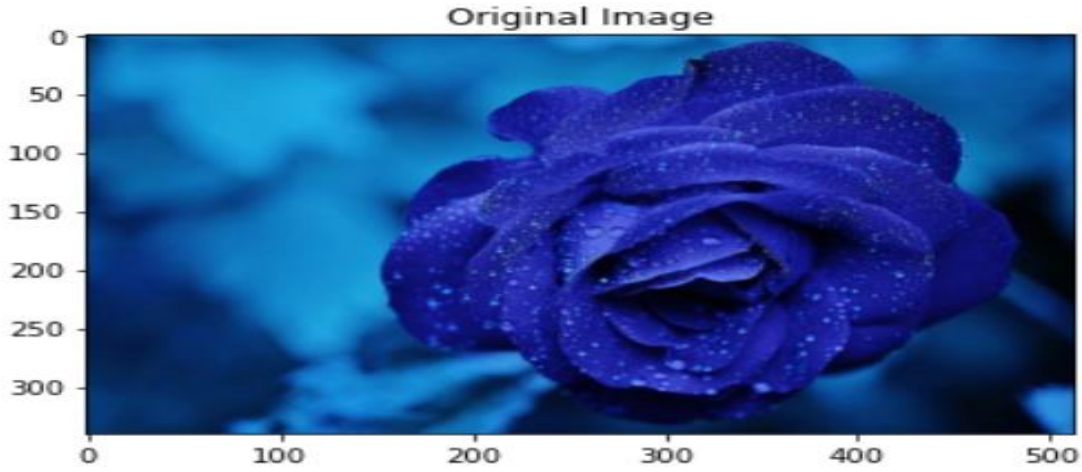
Edit View Insert Cell Kernel Widgets Help

Run Code

Sharpening Filtering

In [2]:
```
from PIL import Image, ImageFilter
import cv2
import numpy as np
import matplotlib.pyplot as plt
from scipy import ndimage
%matplotlib inline
```

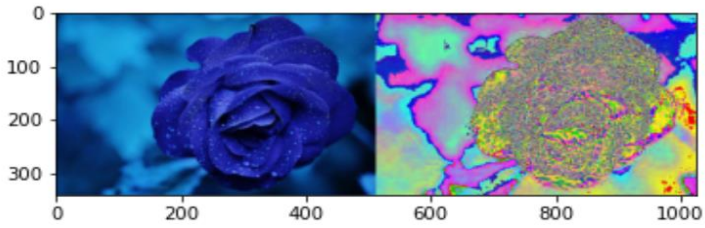
In [3]:
```
img = cv2.imread('blue.png')
img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
plt.imshow(img)
plt.title('Original Image')
img_data=np.array(img)
```



High Pass Filtering

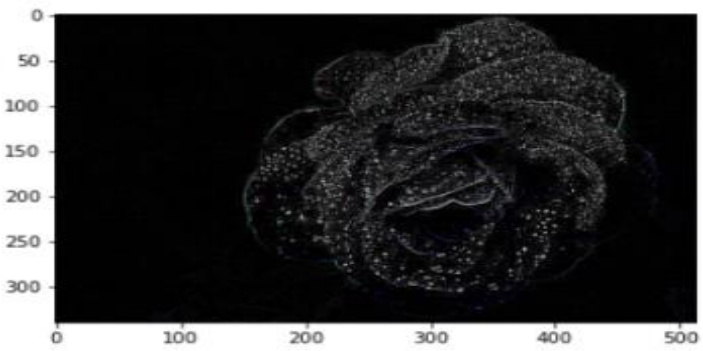
In [4]:
```
kernel= np.array([[[-1,-1,-1],
[-1,8,-1],
[-1,-1,-1]]])
highpass_3x3= ndimage.convolve(img_data,kernel)
fil_img= np.hstack((img_data,highpass_3x3))
plt.imshow(fil_img)
```

Out[4]: <matplotlib.image.AxesImage at 0x24bdbcff5b0>



In [5]:
```
highkernel=np.array([[-1,-1,-1],
[-1,8,-1],
[-1,-1,-1]])
highpass = cv2.filter2D(src=img,kernel=highkernel,ddepth=-1)
plt.imshow(highpass)
```

Out[5]: <matplotlib.image.AxesImage at 0x24bdc004d90>



High Boost Filtering

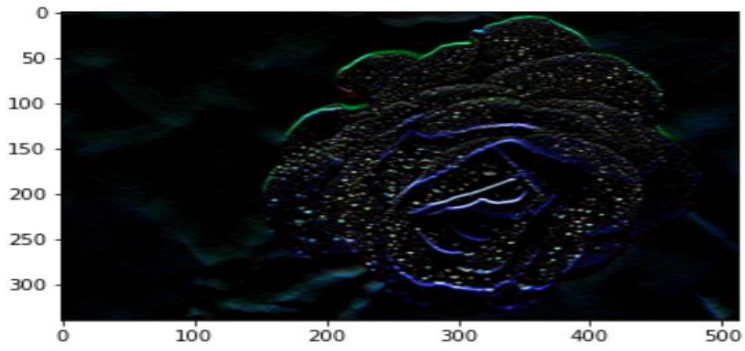
In [6]:
```
image = cv2.imread('blue.png',1)
image = cv2.cvtColor(image,cv2.COLOR_BGR2RGB)
#Blur the image
gauss = cv2.GaussianBlur(image,(7,7),0)
#Apply unsharp masking
unsharp_image= cv2.addWeighted(image,2,gauss,-1,0)
fil_img= np.hstack((image,unsharp_image))
plt.imshow(fil_img)
```

Out[6]: <matplotlib.image.AxesImage at 0x24bdbdd2910>

Derivative Filtering , Prewitt

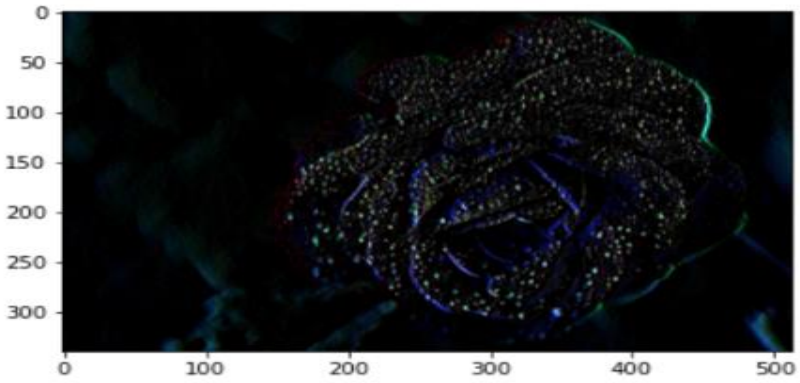
In [7]:
```
prewittkernelH=np.array([[1,1,1],
[0,0,0],
[-1,-1,-1]])
prewittH = cv2.filter2D(src=image,kernel=prewittkernelH,ddepth=-1)
plt.imshow(prewittH)
```

Out[7]: <matplotlib.image.AxesImage at 0x24bdbe33b20>



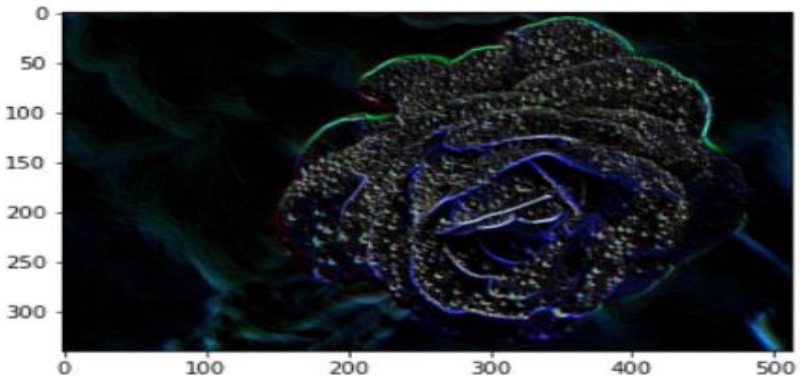
In [8]:
```
prewittkernelV=np.array([[-1,0,1],
[-1,0,1],
[-1,0,1]])
prewittV = cv2.filter2D(src=image,kernel=prewittkernelV,ddepth=-1)
plt.imshow(prewittV)
```

Out[8]: <matplotlib.image.AxesImage at 0x24bdbe9e4f0>



In [9]:
```
prewitt= prewittH + prewittV
plt.imshow(prewitt)
```

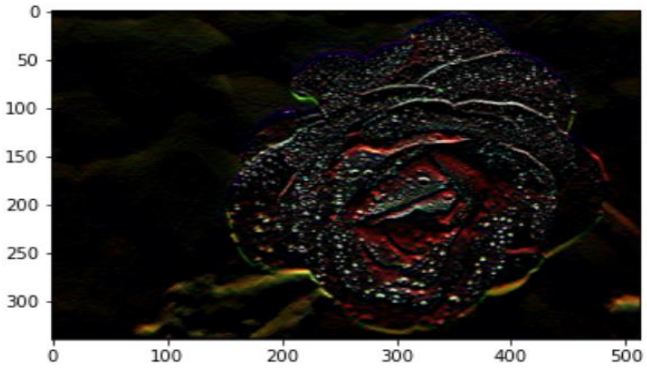
Out[9]: <matplotlib.image.AxesImage at 0x24bdd293730>



Sobel

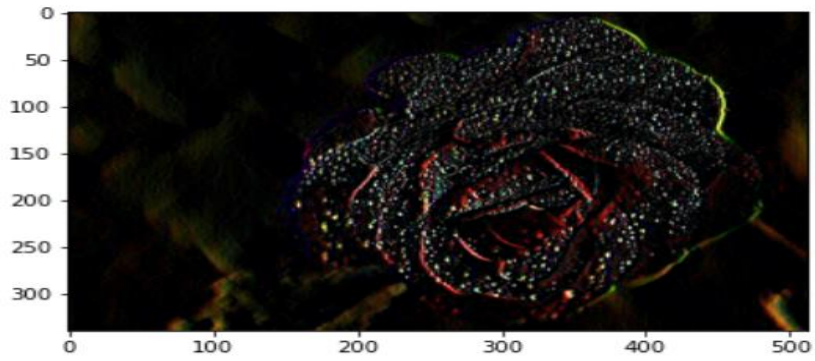
In [10]:
```
image = cv2.imread('blue.png',1)
sobelkernelH=np.array([[-1,-2,-1],
[0,0,0],
[1,2,1]])
sobelH = cv2.filter2D(src=image,kernel=sobelkernelH,ddepth=-1)
plt.imshow(sobelH)
```

Out[10]: <matplotlib.image.AxesImage at 0x24bdd2fb640>



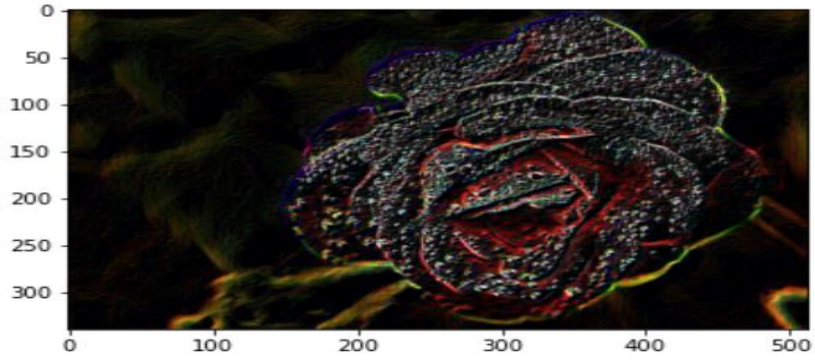
In [11]:
```
sobelkernelV=np.array([[-1,0,1],
[-2,0,2],
[-1,0,1]])
sobelV = cv2.filter2D(src=image,kernel=sobelkernelV,ddepth=-1)
plt.imshow(sobelV)
```

Out[11]: <matplotlib.image.AxesImage at 0x24bdd36c070>



In [12]:
```
sobel=sobelH + sobelV
plt.imshow(sobel)
```

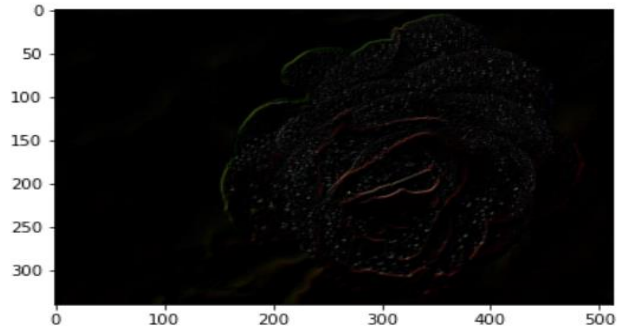
Out[12]: <matplotlib.image.AxesImage at 0x24bdd3ce3a0>



Robert

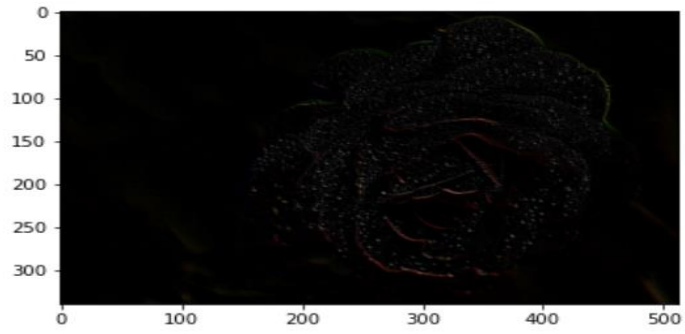
In [13]:
```
image = cv2.imread('blue.png',1)
robertkernelH=np.array([[1,0,0],
[0,-1,0],
[0,0,0]])
robertH = cv2.filter2D(src=image,kernel=robertkernelH,ddepth=-1)
plt.imshow(robertH)
```

Out[13]: <matplotlib.image.AxesImage at 0x24bdd43a0d0>



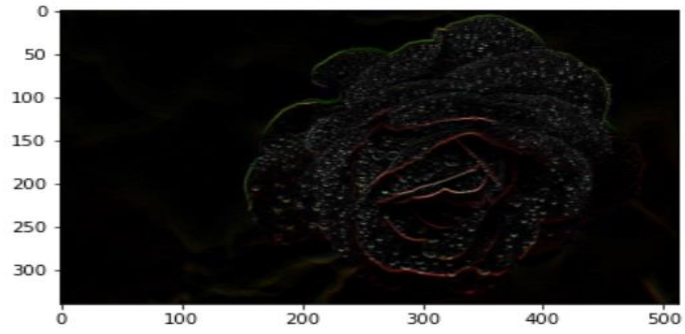
In [14]:
```
robertkernelV=np.array([[0,1,0],
[-1,0,0],
[0,0,0]])
robertV = cv2.filter2D(src=image,kernel=robertkernelV,ddepth=-1)
plt.imshow(robertV)
```

Out[14]: <matplotlib.image.AxesImage at 0x24bdd492af0>



In [15]:
```
robert = robertH + robertV
plt.imshow(robert)
```

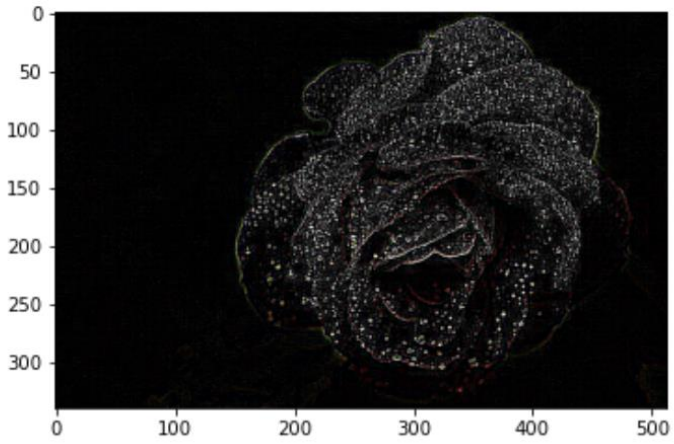
Out[15]: <matplotlib.image.AxesImage at 0x24bde4c2e80>



Laplacian

In [16]:
```
laplaciankernel=np.array([[-1,-1,-1],
[-1,8,-1],
[-1,-1,-1]])
laplacian = cv2.filter2D(src=image,kernel=laplaciankernel,ddepth=-1)
plt.imshow(laplacian)
```

Out[16]: <matplotlib.image.AxesImage at 0x24bde526d30>



Canny

In [17]:
```
image = cv2.imread('blue.png',1)
image = cv2.cvtColor(image,cv2.COLOR_BGR2RGB)
img_blur = cv2.GaussianBlur(image, (3,3), 0)
# Canny Edge Detection
edges = cv2.Canny(image=img_blur, threshold1=100, threshold2=200)
plt.imshow(edges)
```

Out[17]: <matplotlib.image.AxesImage at 0x24bde58e880>