6. Edge detection using Roberts, Sobel, Prewitt, and Kenny using 2D filters as well as openCV functions.

In [45]:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
img=cv2.imread("img.jpg")
img=cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
```

In [46]:

```
#rober
def rober(img):
    k1=np.array([[1,0],[0,-1]])
    k2=np.array([[0,1],[-1,0]])
    rx=cv2.filter2D(img,-1,k1)
    ry=cv2.filter2D(img,-1,k2)
    rober1=np.sqrt(np.square(rx)+np.square(ry))
    rober1=rober1.astype('int32')
    return rober1

img_rober=rober(img)
```

In [47]:

```
#sobel
k1=np.array([[-1,0,1],[-2,0,2],[-1,0,1]])
k2=np.array([[1,2,1],[0,0,0],[-1,-2,1]])
sx=cv2.filter2D(img,-1,k1)
sy=cv2.filter2D(img,-1,k2)
sobel=np.sqrt(np.square(sx)+np.square(sy))
#function
img_sob=cv2.Sobel(img,-1,2,2)
```

In [48]:

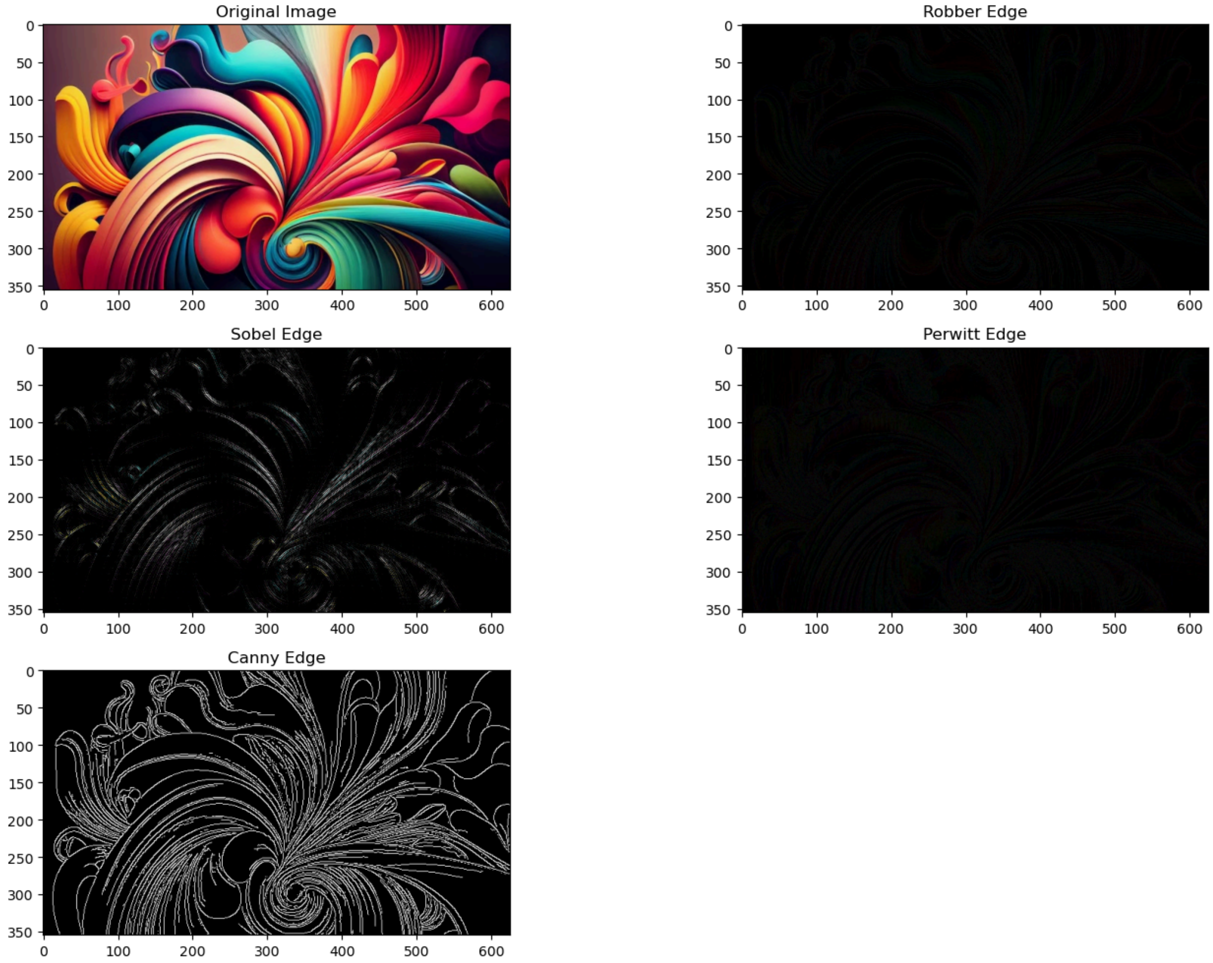
```
#perwitt
def perwitt(img):
    k1 = np.array([[-1, 0, 1],
                  [-1, 0, 1],
                  [-1, 0, 1]])
    k2 = np.array([[-1, -1, -1],
                   [0, 0, 0],
                   [1, 1, 1]])

    px=cv2.filter2D(img,-1,k1)
    py=cv2.filter2D(img,-1,k2)
    perwitt1=np.sqrt(np.square(px)+np.square(py))
    perwitt1=perwitt1.astype('int32')
    return perwitt1

img_perwitt=perwitt(img)
```

In [49]:

```
#canny
img_canny=cv2.Canny(img,50,150)
```

In [50]:

```
plt.figure(figsize=(15,10))
plt.subplot(3, 2, 1), plt.imshow(img, cmap='gray'), plt.title('Original Image')
plt.subplot(3, 2, 2), plt.imshow(img_rober, cmap='gray'), plt.title('Robber Edge')
plt.subplot(3, 2, 3), plt.imshow(img_sob, cmap='gray'), plt.title('Sobel Edge')
plt.subplot(3, 2, 4), plt.imshow(img_perwitt, cmap='gray'), plt.title('Perwitt Edge
plt.subplot(3, 2, 5), plt.imshow(img_canny, cmap='gray'), plt.title('Canny Edge')
plt.tight_layout()
plt.show()
```



Original Image

Robber Edge

Sobel Edge

Perwitt Edge

Canny Edge

In [ ]: