In [15]:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

Question 2

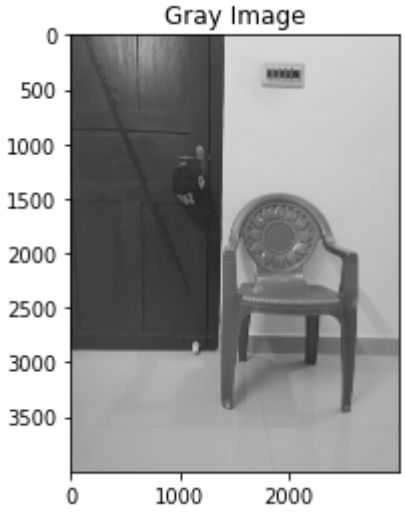
2. Each of the images considered in the first problem apply Gaussian filters of size 7*7, 9*9* and 11*11. Repeat the problem 1 and report your observation

In [16]:

```
#Sobel Filter from Q1
def sobel(gray):
    ret, binary = cv2.threshold(gray, 127, 255, cv2.THRESH_BINARY)

    #binary=gray

    sobelx = cv2.Sobel(binary,-1,1,0,ksize=5)
    sobely = cv2.Sobel(binary,-1,0,1,ksize=5)
    sobelxy = cv2.Sobel(binary,-1,1,1,ksize=5)
    print("BINARY")
    plt.imshow(cv2.cvtColor(binary, cv2.COLOR_BGR2RGB))
    plt.show()
    print("SOBEL-X")
    plt.imshow(sobelx,cmap="gray")
    plt.show()
    print("SOBEL-Y")
    plt.imshow(sobely,cmap="gray")
    plt.show()
    print("SOBEL-XY")
    plt.imshow(sobelxy,cmap="gray")
    plt.show()

    edges_sobelx=0
    for i in range(sobelx.shape[0]):
        for j in range(sobelx.shape[1]):
            if (sobelx[i,j]>0):
                edges_sobelx+=1
    print("Sobelx shape :{}".format(sobelx.shape))
    print("Edge pixels :{} out of {}".format(edges_sobelx,sobelx.size))
    edges_sobely=0
    for i in range(sobely.shape[0]):
        for j in range(sobely.shape[1]):
            if (sobely[i,j]>0):
                edges_sobely+=1
    print("Sobely shape :{}".format(sobelx.shape))
    print("Edge pixels :{} out of {}".format(edges_sobely,sobely.size))

    edges_sobelxy=0
    for i in range(sobelxy.shape[0]):
        for j in range(sobelxy.shape[1]):
            if (sobelxy[i,j]>0):
                #print(sobelx[i][j])
                edges_sobelxy+=1
    print("Sobelxy shape :{}".format(sobelxy.shape))
    print("Edge pixels :{} out of {}".format(edges_sobelxy,sobelxy.size))
```

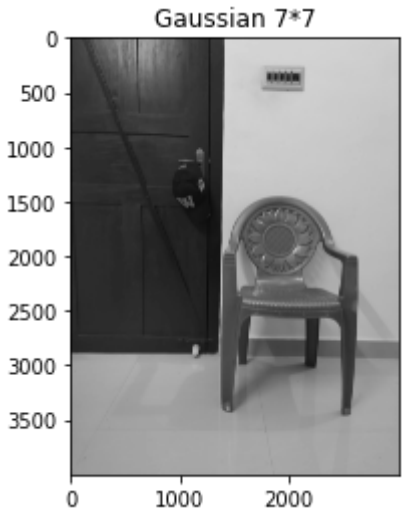
In [18]:

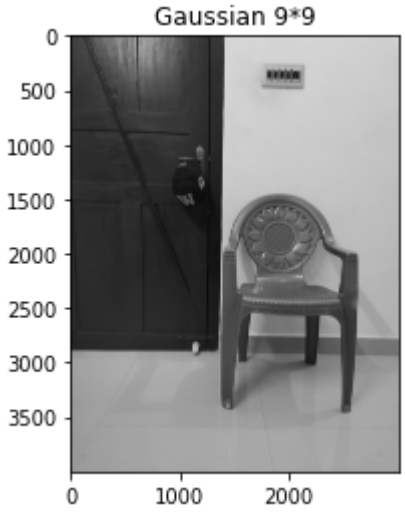
```
img = cv2.imread("room.jpg")
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.title('Original Image')
plt.show()

plt.imshow(gray, cmap='gray')
plt.title('Gray Image')
plt.show()
```

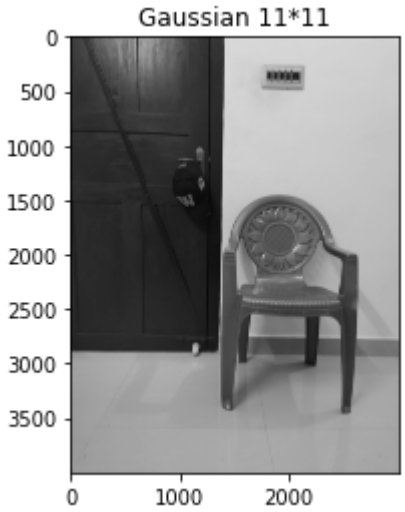


In [19]:

```
gauss7 = cv2.GaussianBlur(gray,(7,7),0)
gauss9 = cv2.GaussianBlur(gray,(9,9),0)
gauss11 = cv2.GaussianBlur(gray,(11,11),0)

plt.title('Gaussian 7*7')
plt.imshow(gauss7,cmap='gray')
plt.show()

plt.title('Gaussian 9*9')
plt.imshow(gauss9,cmap = 'gray')
plt.show()

plt.title('Gaussian 11*11')
plt.imshow(gauss11,cmap = 'gray')
plt.show()
```







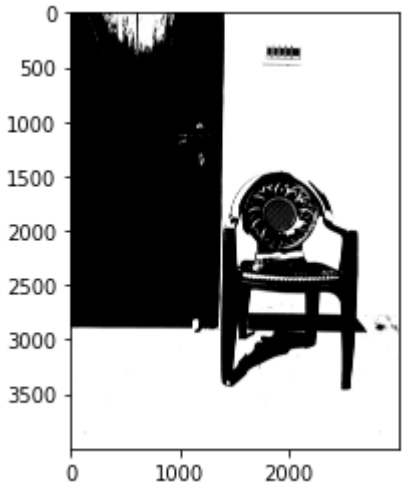
In [ ]:

```
1
2
```

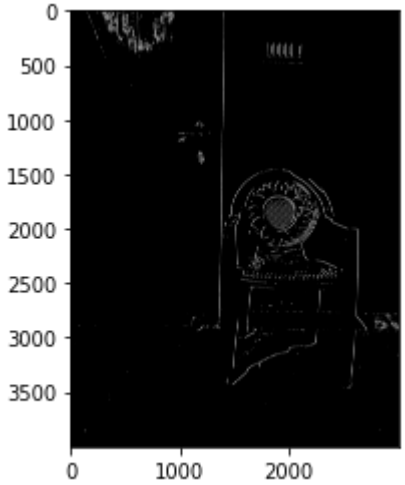
In [20]:

```
1 sobel(gauss7)
```

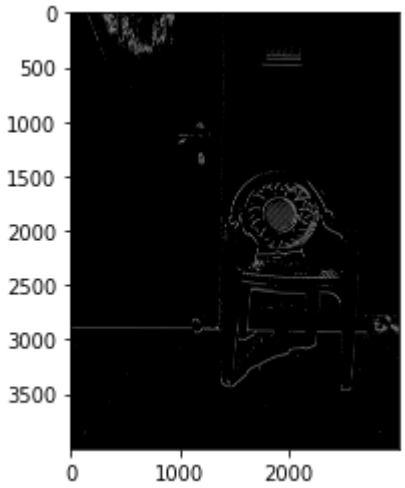
BINARY



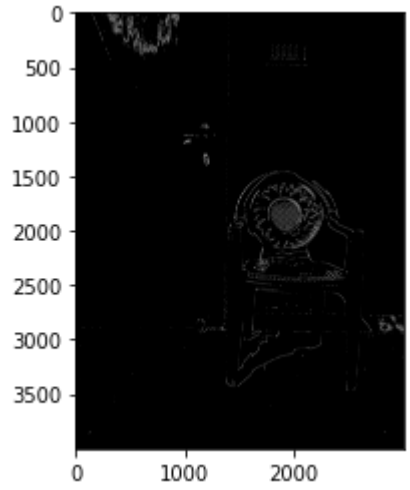
SOBEL-X



SOBEL-Y



SOBEL-XY

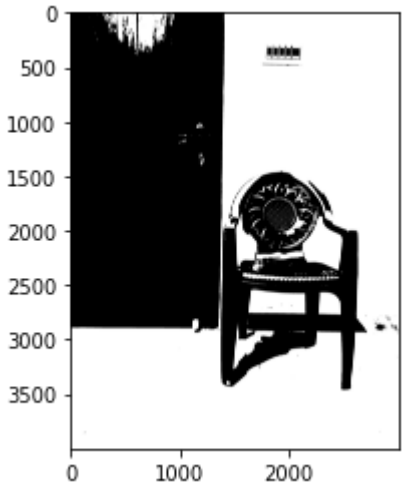



```
Sobelx shape :(4000, 3000)
Edge pixels :143443 out of 12000000
Sobely shape :(4000, 3000)
Edge pixels :131114 out of 12000000
Sobelxy shape :(4000, 3000)
Edge pixels :107271 out of 12000000
```

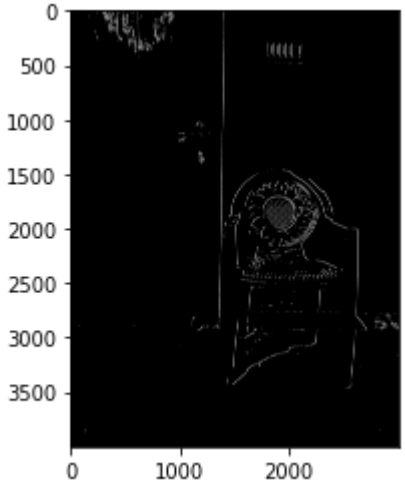
In [21]:

```
1 sobel(gauss9)
```

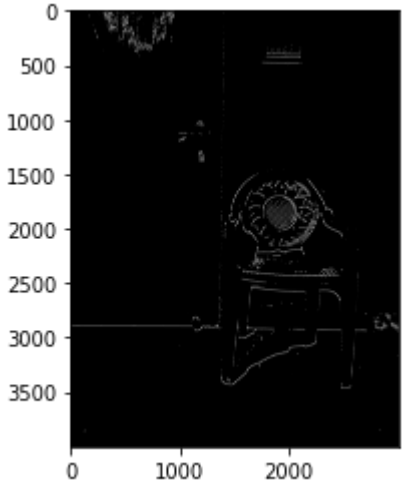
BINARY



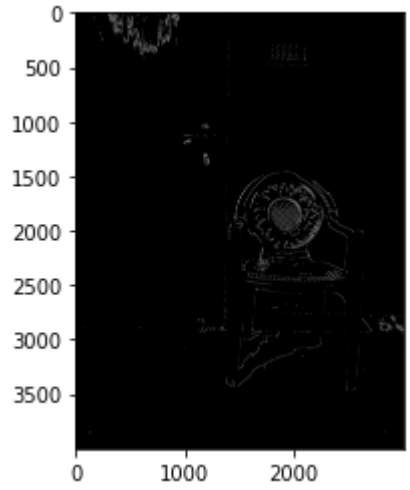
SOBEL-X



SOBEL-Y



SOBEL-XY

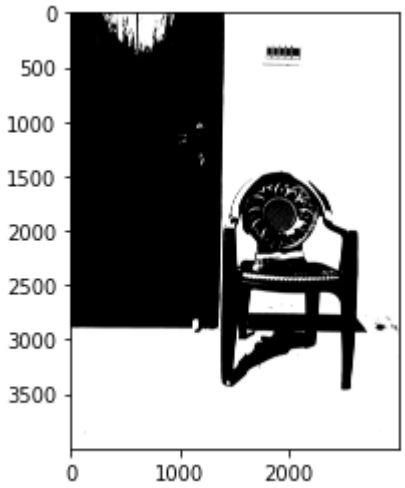



```
Sobelx shape :(4000, 3000)
Edge pixels :130286 out of 12000000
Sobely shape :(4000, 3000)
Edge pixels :117077 out of 12000000
Sobelxy shape :(4000, 3000)
Edge pixels :94707 out of 12000000
```

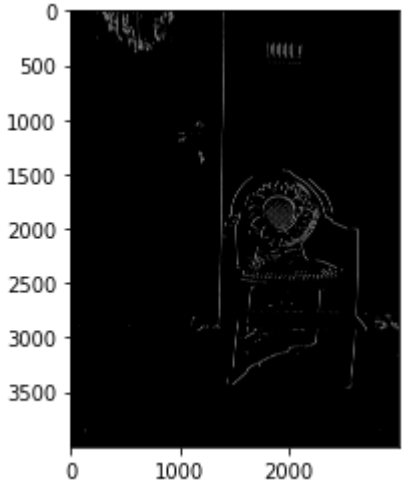
In [22]:

```
1 sobel(gauss11)
```

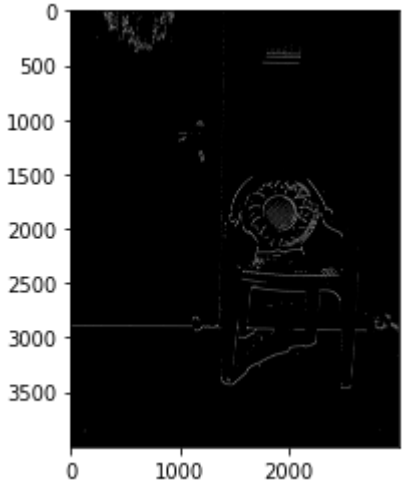
BINARY



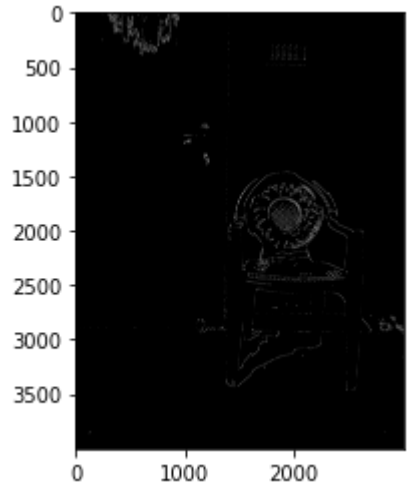
SOBEL-X



SOBEL-Y



SOBEL-XY

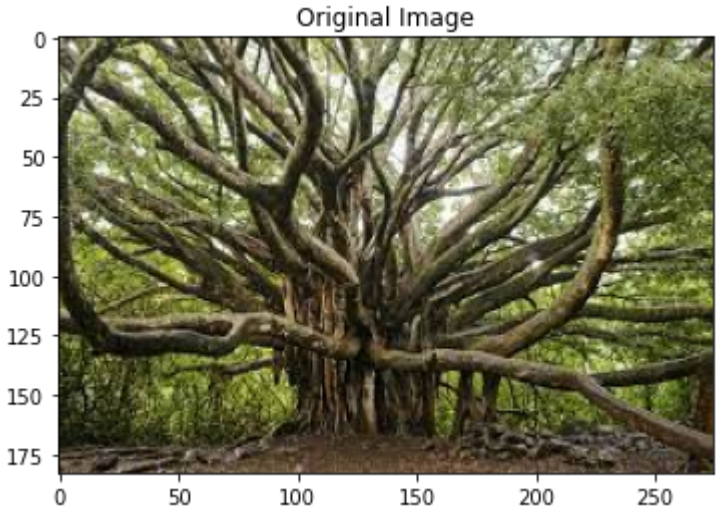


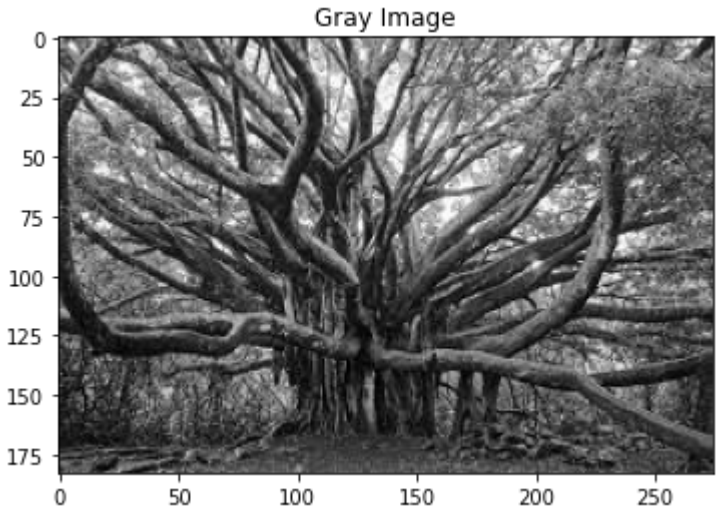

```
Sobelx shape :(4000, 3000)
Edge pixels :119442 out of 12000000
Sobely shape :(4000, 3000)
Edge pixels :105982 out of 12000000
Sobelxy shape :(4000, 3000)
Edge pixels :84267 out of 12000000
```

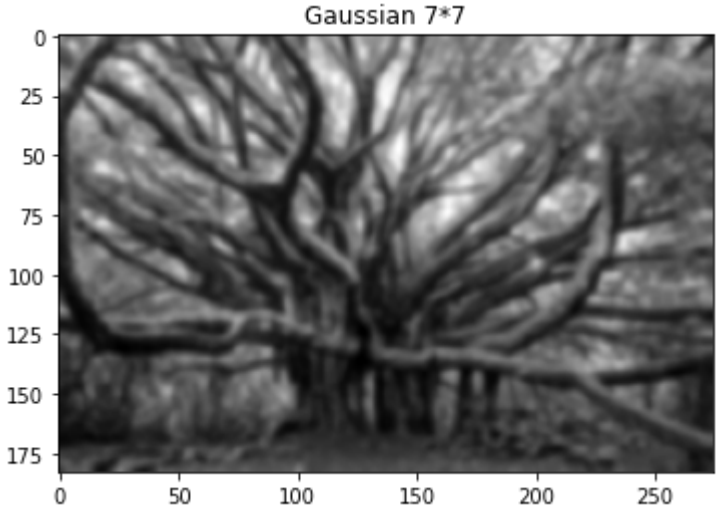
TREE

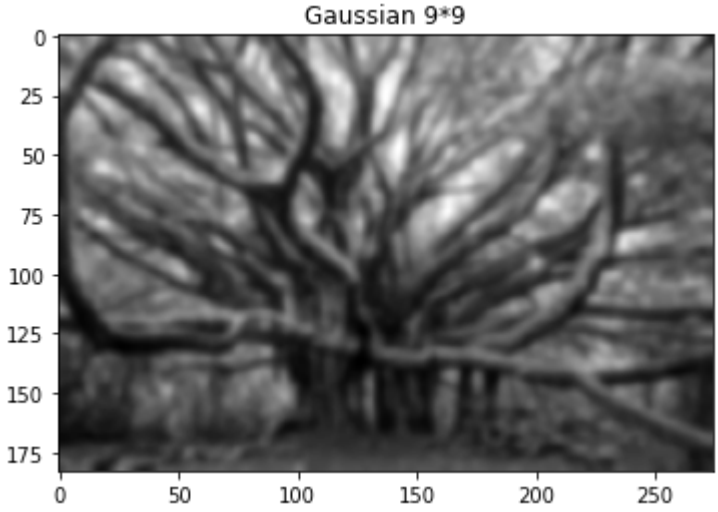
In [23]:

```
img = cv2.imread("tree.jpg")
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.title('Original Image')
plt.show()

plt.imshow(gray, cmap='gray')
plt.title('Gray Image')
plt.show()


gauss7 = cv2.GaussianBlur(gray,(7,7),0)
gauss9 = cv2.GaussianBlur(gray,(9,9),0)
gauss11 = cv2.GaussianBlur(gray,(11,11),0)

plt.title('Gaussian 7*7')
plt.imshow(gauss7,cmap='gray')
plt.show()

plt.title('Gaussian 9*9')
plt.imshow(gauss9,cmap = 'gray')
plt.show()

plt.title('Gaussian 11*11')
plt.imshow(gauss11,cmap = 'gray')
plt.show()
```




Gray Image
0
25
50
75
100
125
150
175
0
50
100
150
200
250


Gaussian 7*7
0
25
50
75
100
125
150
175
0
50
100
150
200
250

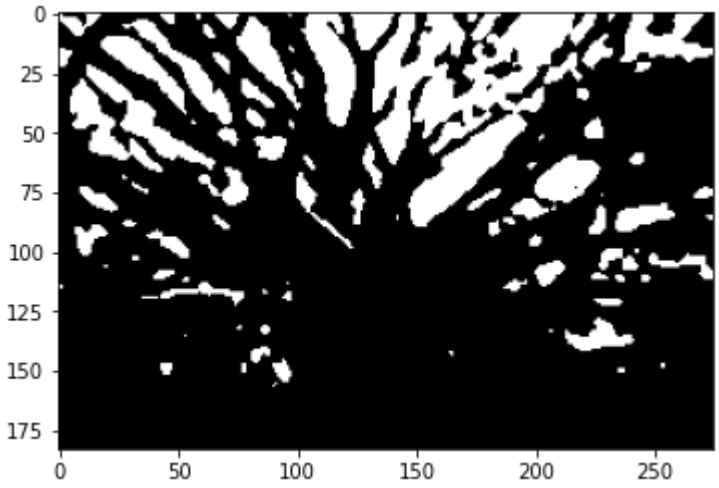

Gaussian 9*9
0
25
50
75
100
125
150
175
0
50
100
150
200
250


Gaussian 11*11
0

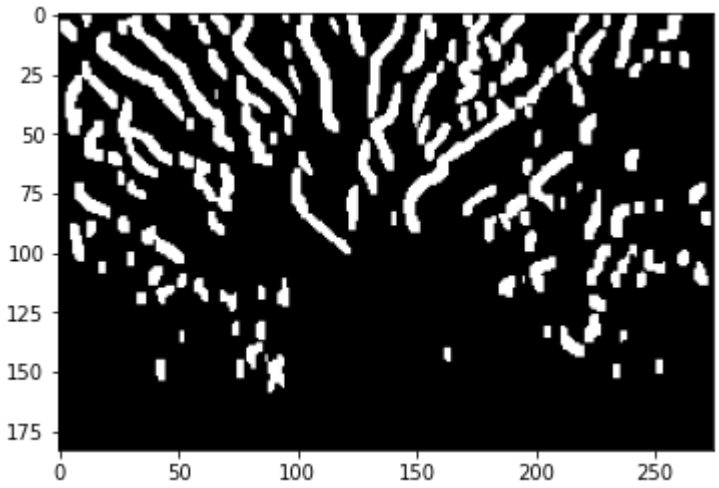
In [24]:

```
1 sobel(gauss7)
```

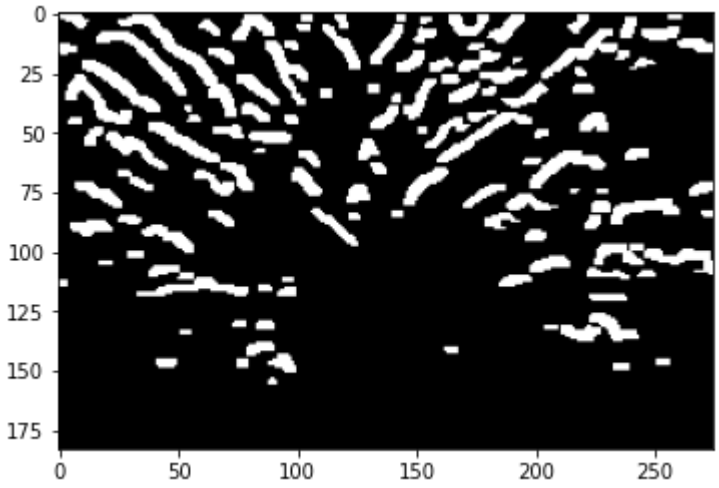
BINARY



SOBEL-X



SOBEL-Y



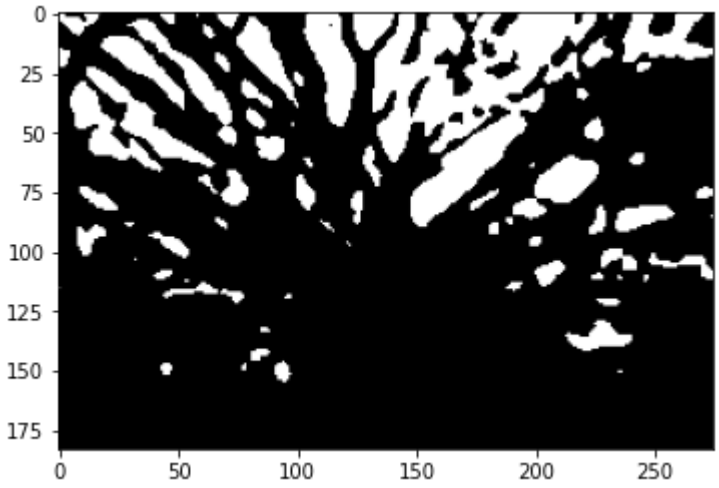
SOBEL-XY


```
Sobelx shape :(183, 275)
Edge pixels :7974 out of 50325
Sobely shape :(183, 275)
Edge pixels :7531 out of 50325
Sobelxy shape :(183, 275)
Edge pixels :7213 out of 50325
```

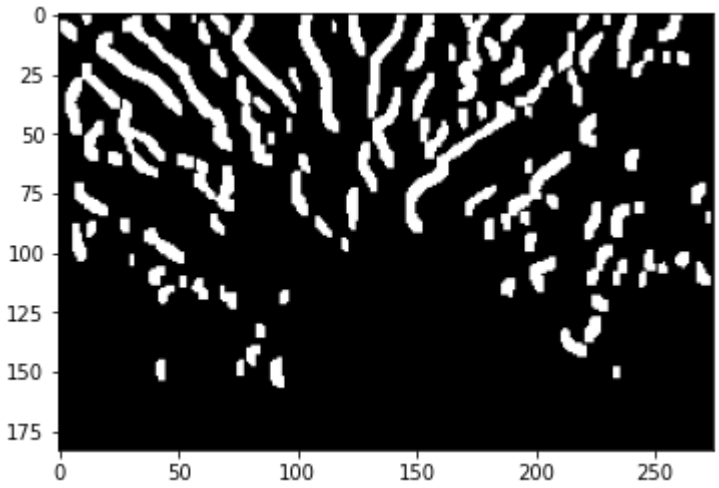
In [26]:

```
1 sobel(gauss9)
```

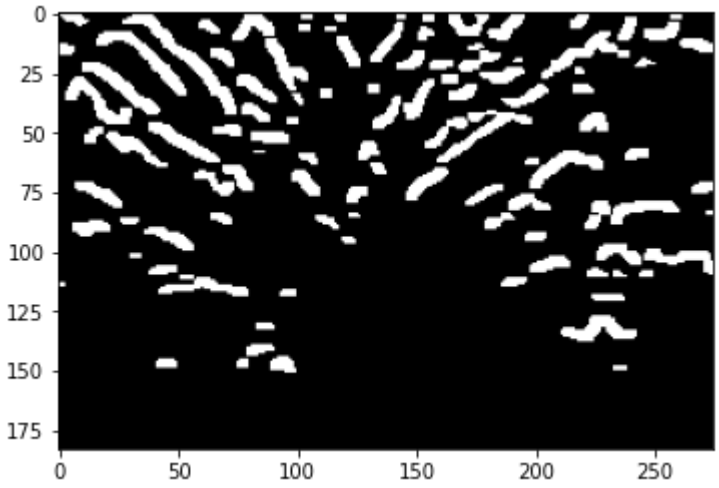
BINARY



SOBEL-X



SOBEL-Y



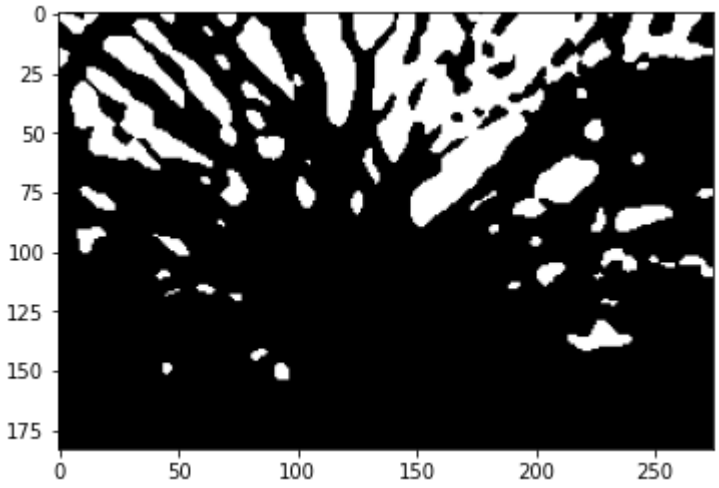
SOBEL-XY


```
Sobelx shape :(183, 275)
Edge pixels :7125 out of 50325
Sobely shape :(183, 275)
Edge pixels :6660 out of 50325
Sobelxy shape :(183, 275)
Edge pixels :6429 out of 50325
```

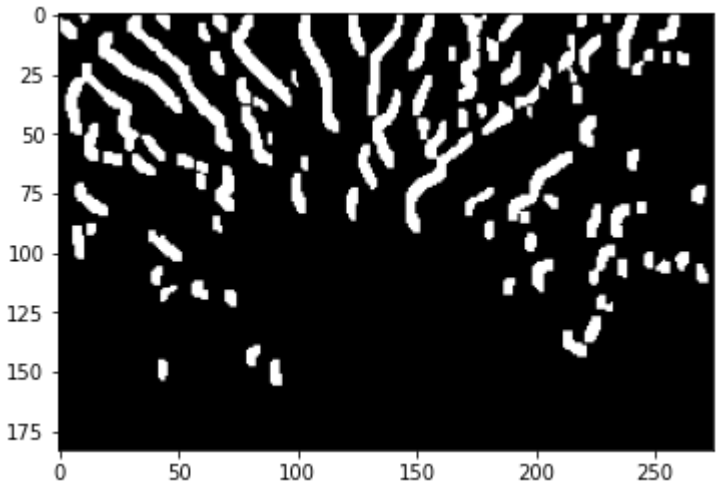
In [27]:

```
1 sobel(gauss11)
```

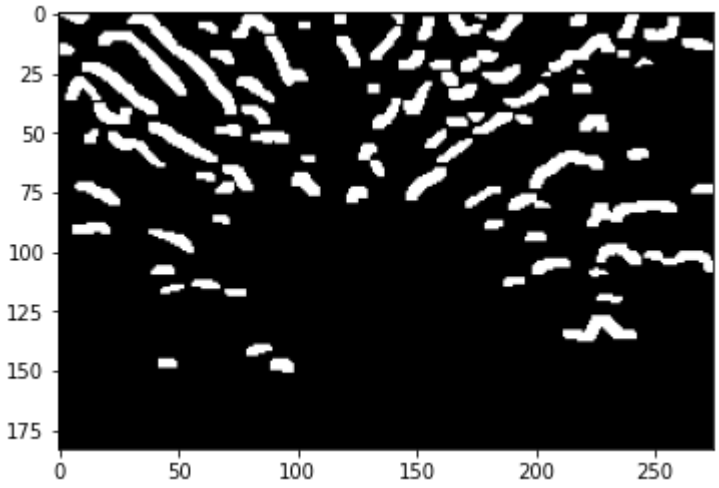
BINARY



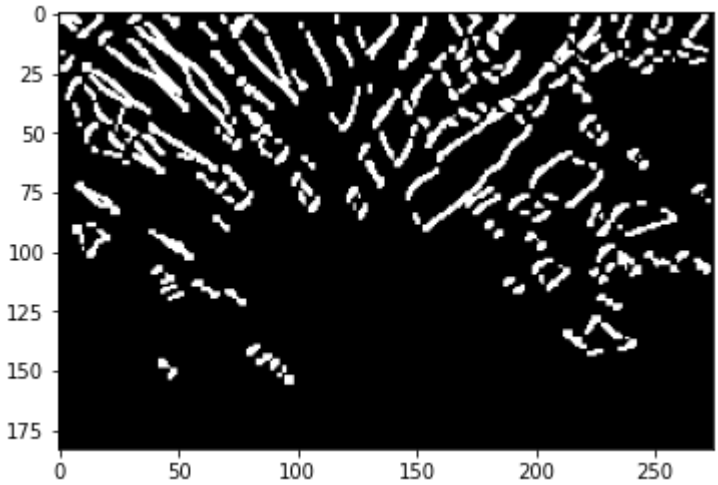
SOBEL-X



SOBEL-Y



SOBEL-XY




```
Sobelx shape :(183, 275)
Edge pixels :6304 out of 50325
Sobely shape :(183, 275)
Edge pixels :5840 out of 50325
Sobelxy shape :(183, 275)
Edge pixels :5688 out of 50325
```

INFERENCE

Gaussian Filter results in reduction of noise and details which makes it blurred. As we increase the size of the gaussian kernel, the resulting image becomes more blurred and there is a decrease in the number of edge pixels detected accordingly.

In [ ]:

```
1
```

In [ ]:

```
1
```