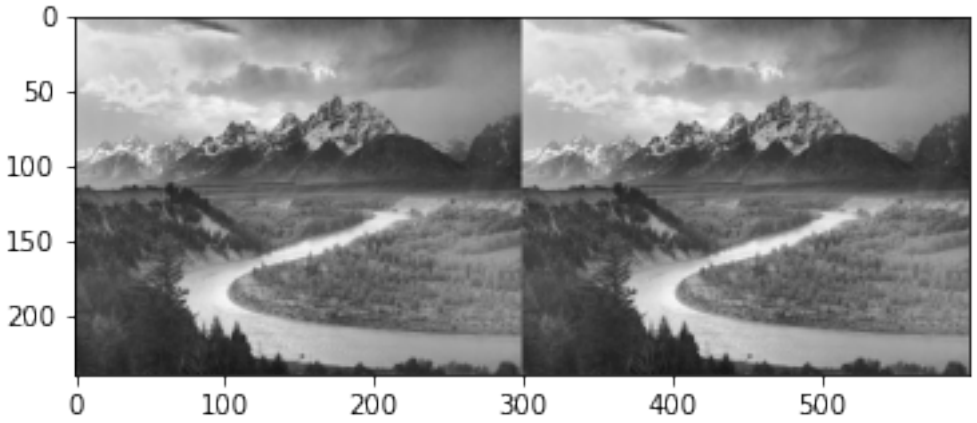
Assignment-1

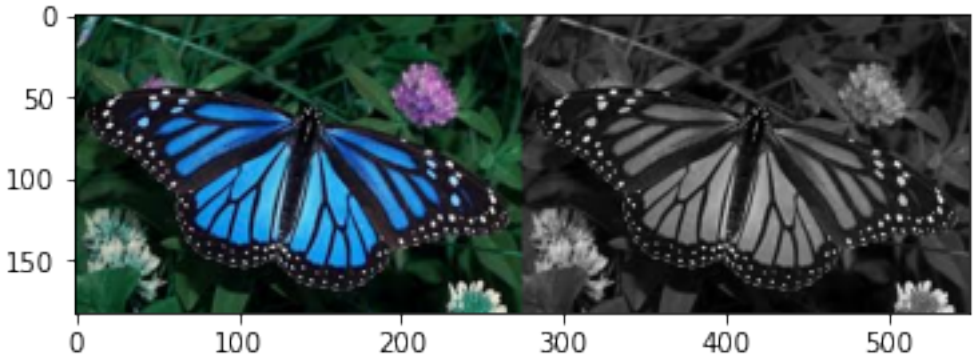
November 17, 2020

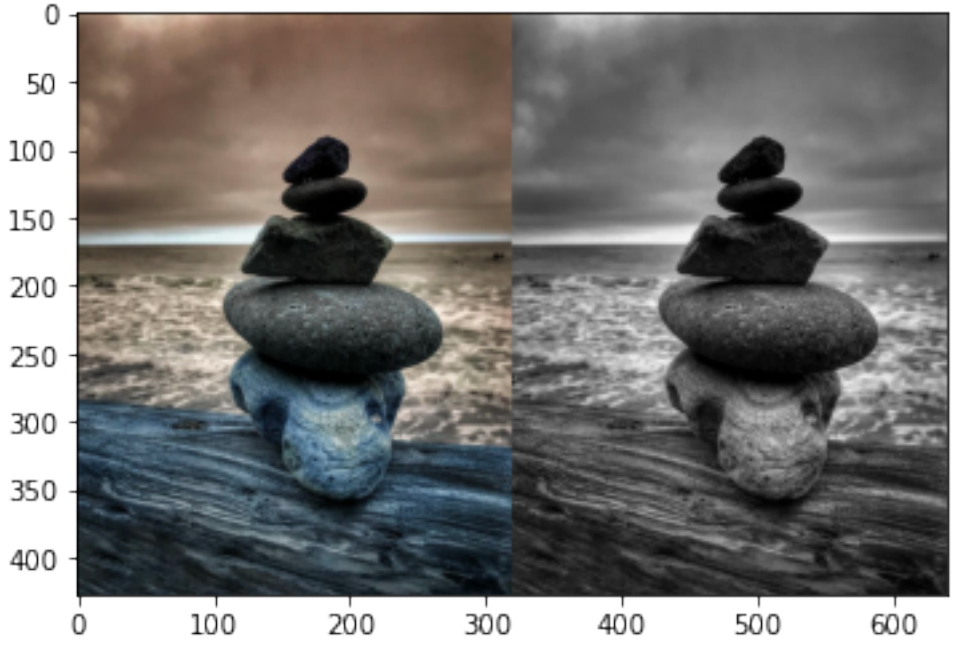
[1]:
```
import sys
sys.path.insert(0, '/home/abrar/python_ws/cv/assignment1')

import assignment_1 as a1
```

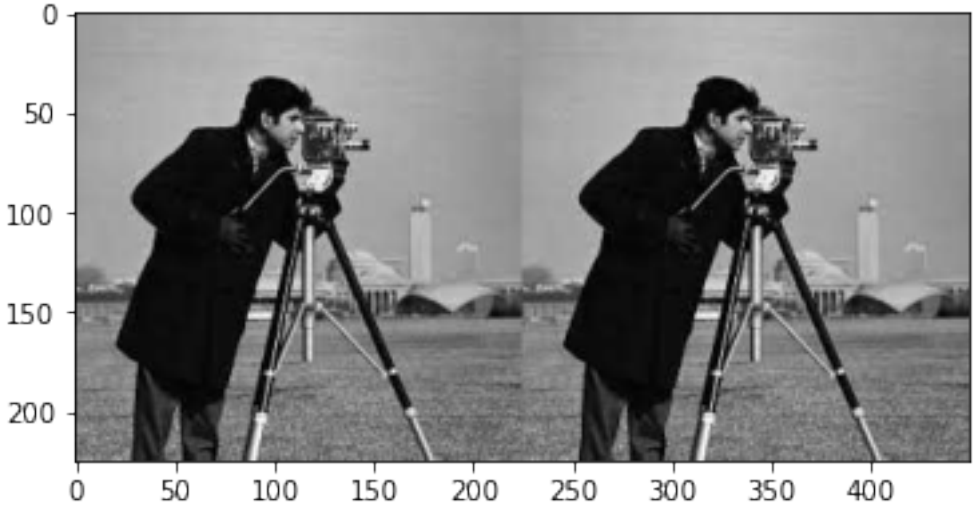
[2]:
```
a1.reading()
```

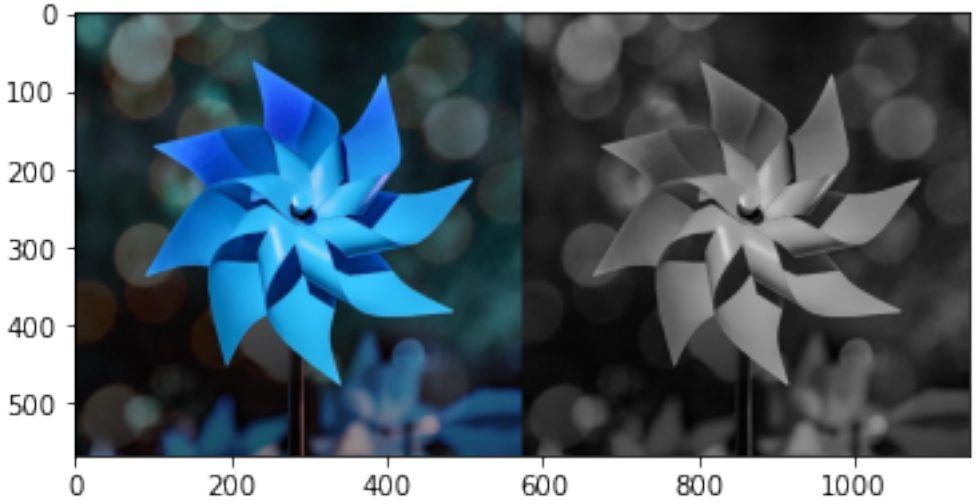


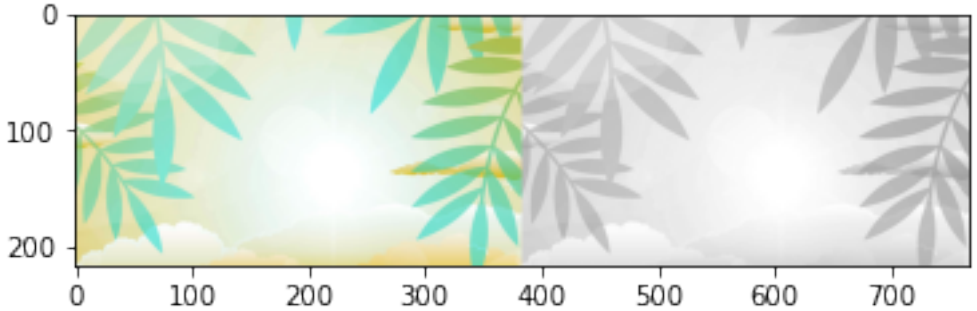


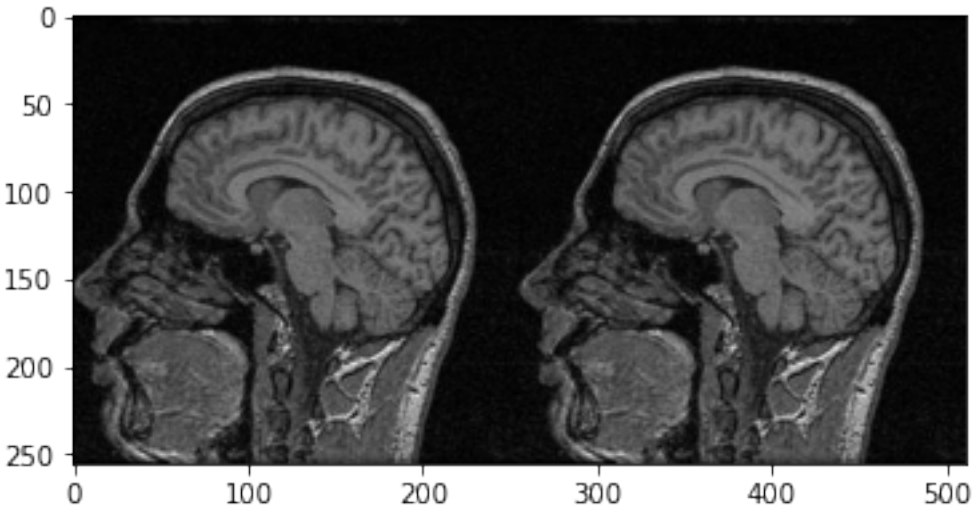

0
50
100
150
200
250
300
350
400
0
100
200
300
400
500
600

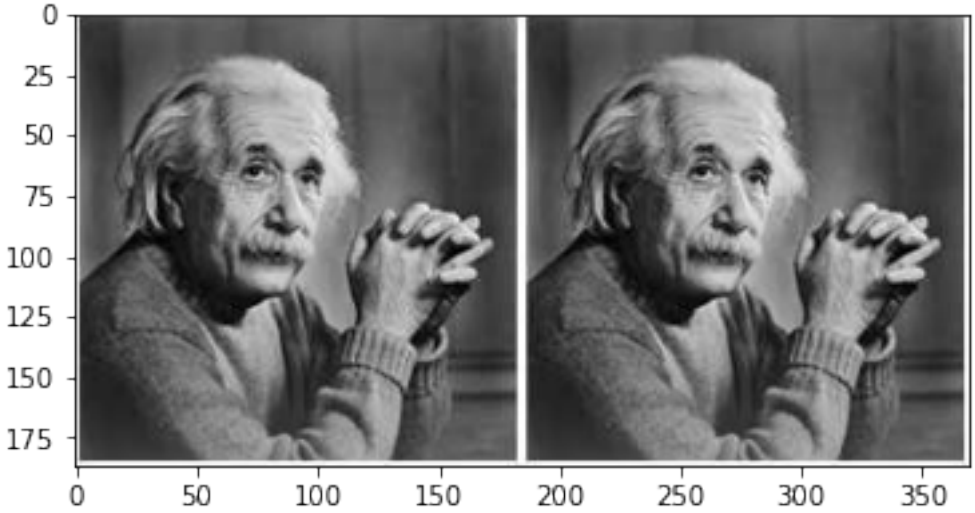

0
50
100
150
200
0
100
200
300
400
500
600


0
50
100
150
200
0
50
100
150
200
250
300
350
400


0
100
200
300
400
500
0
200
400
600
800
1000

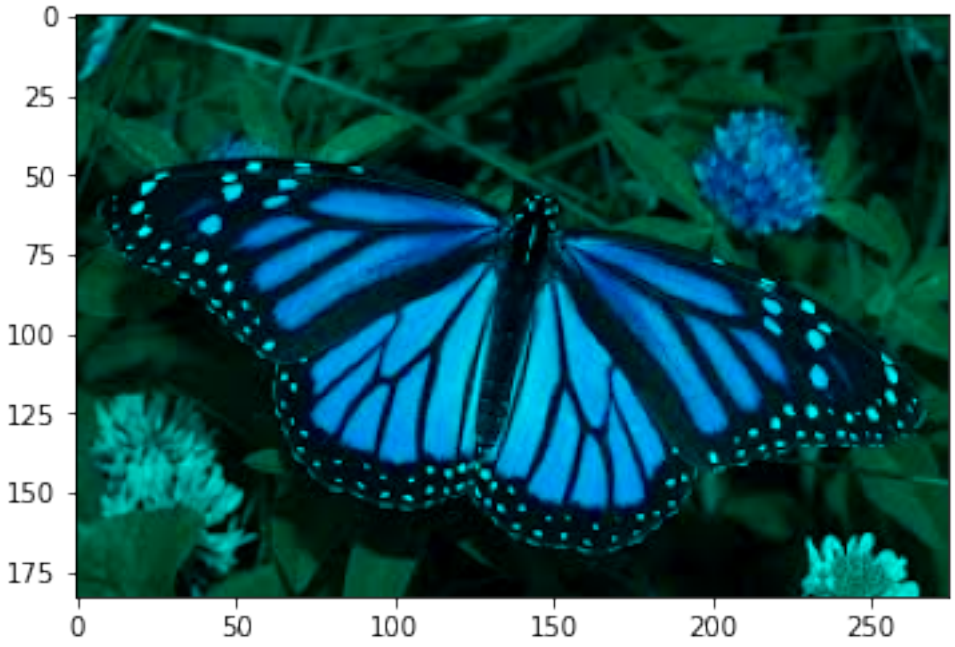

0
100
200
0
100
200
300
400
500
600
700



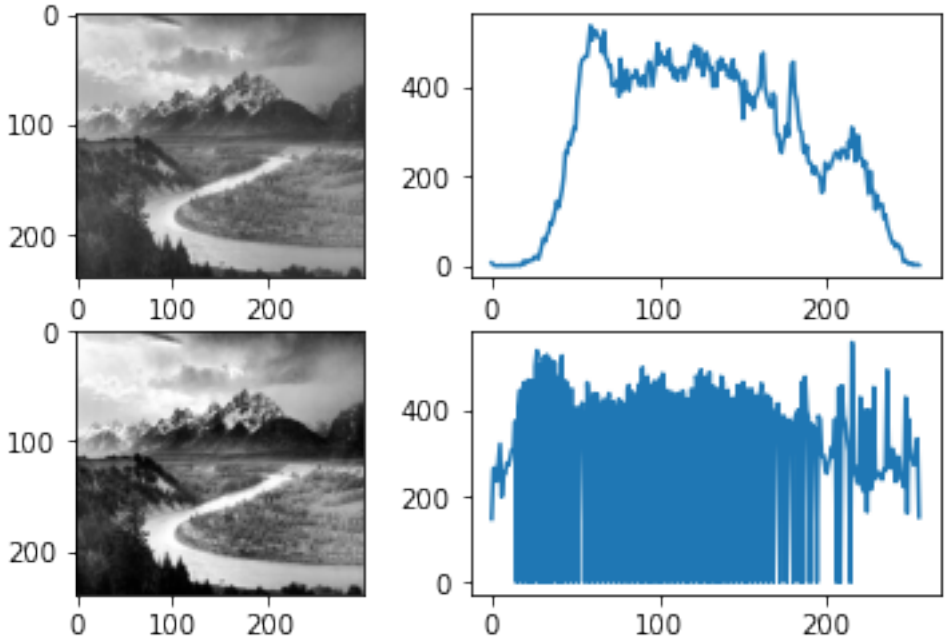


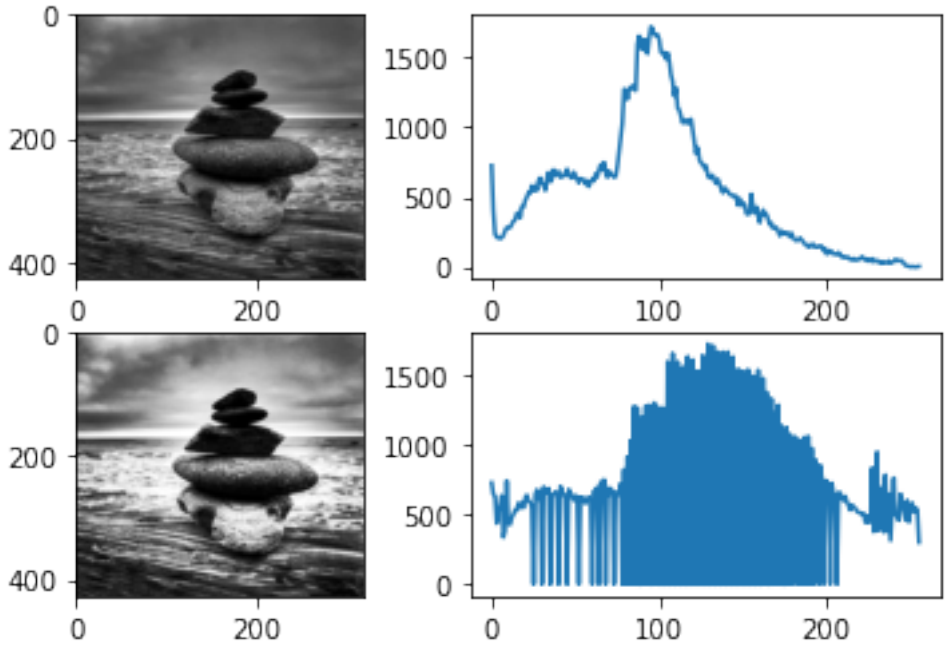
[3]:
```
a1.rgbExclusion()
```

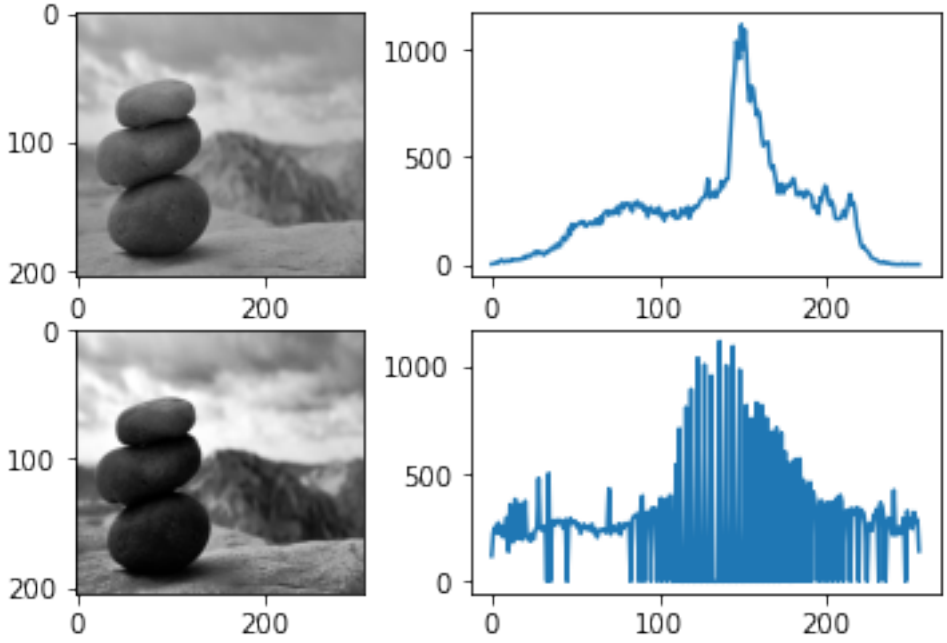
```
Enter Colour to Exclude:r
```

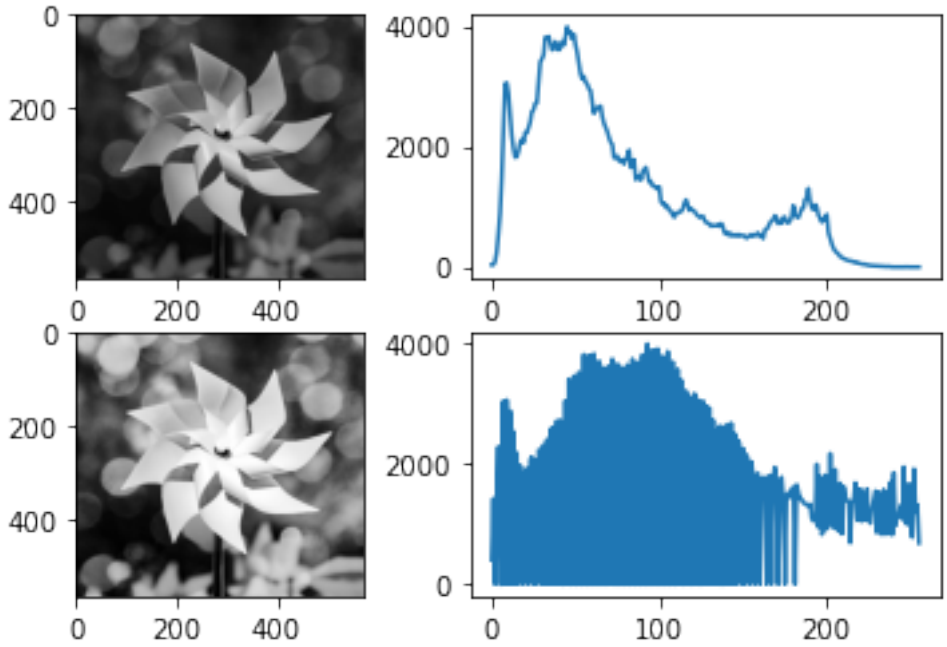


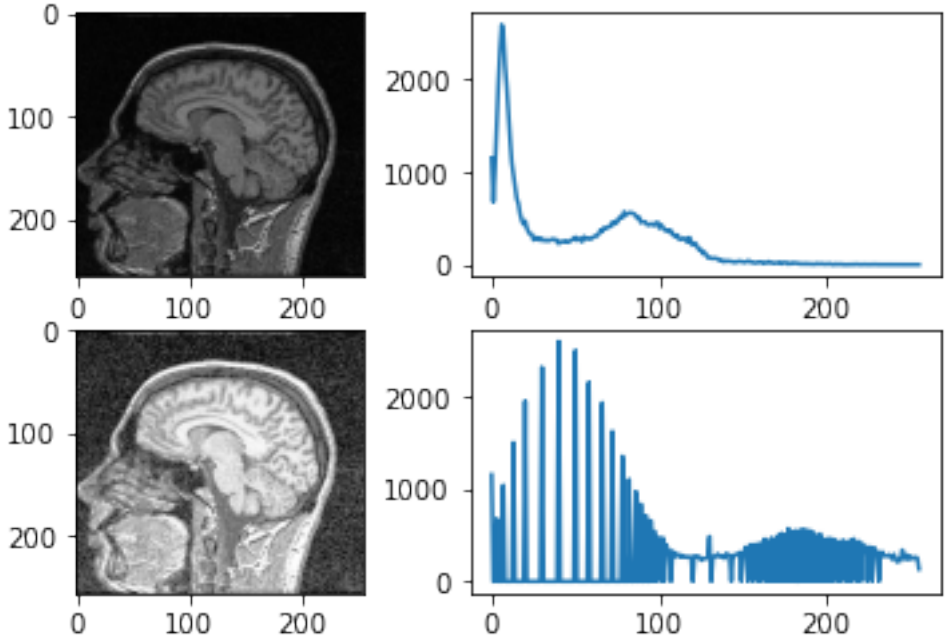
[4]:
```
a1.histogram()
```

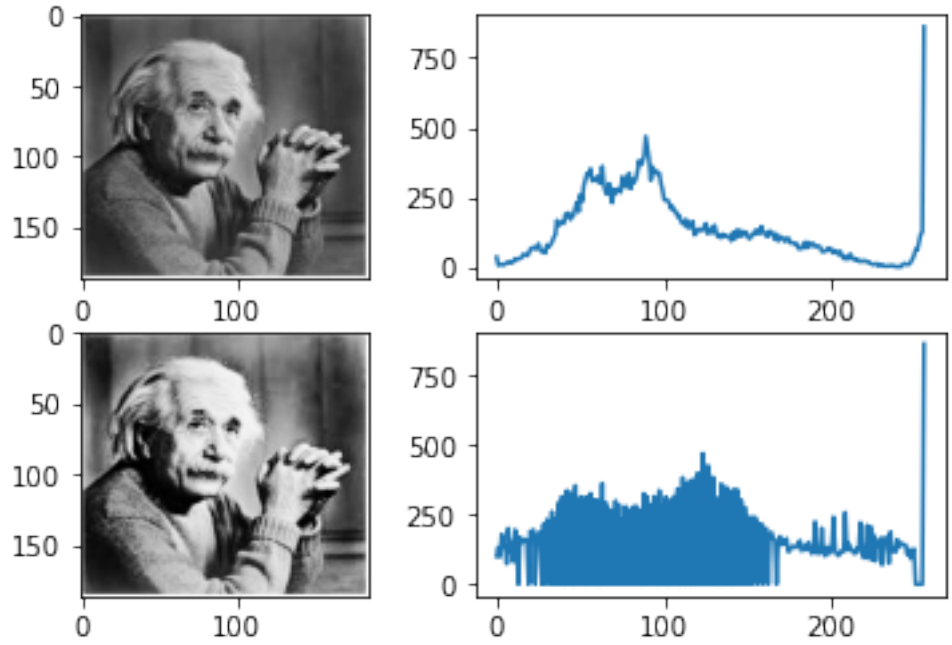



0
200
400
0
200
1500
1000
500
0
0
100
200
0
200
400
0
200
1500
1000
500
0
0
100
200


0
100
200
0
200
1000
500
0
0
100
200
0
100
200
0
200
1000
500
0
0
100
200

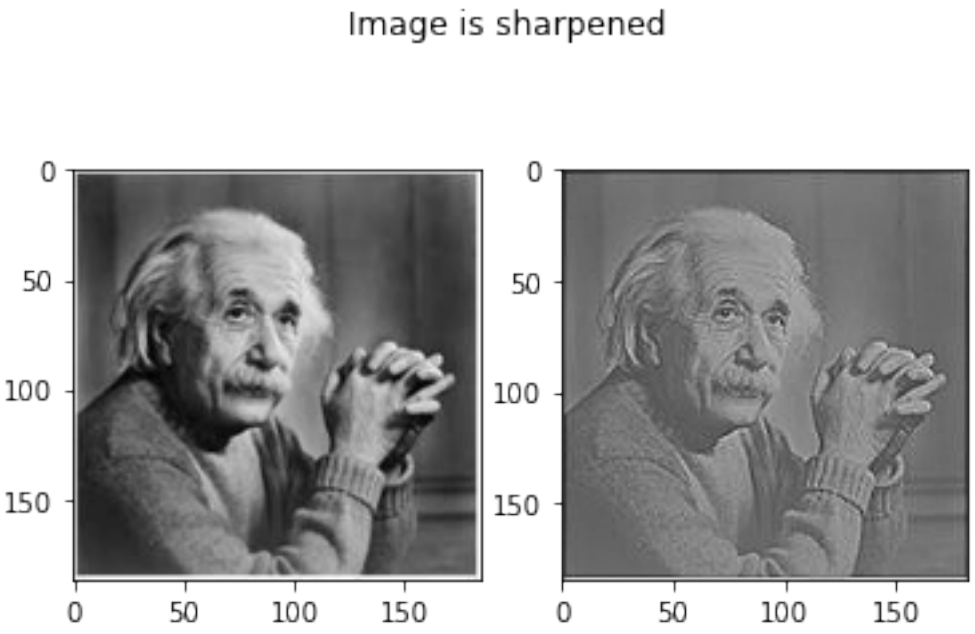

0
200
400
0
200
400
4000
2000
0
0
100
200
0
200
400
0
200
400
4000
2000
0
0
100
200


0
100
200
0
100
200
2000
1000
0
0
100
200
0
100
200
0
100
200
2000
1000
0
0
100
200

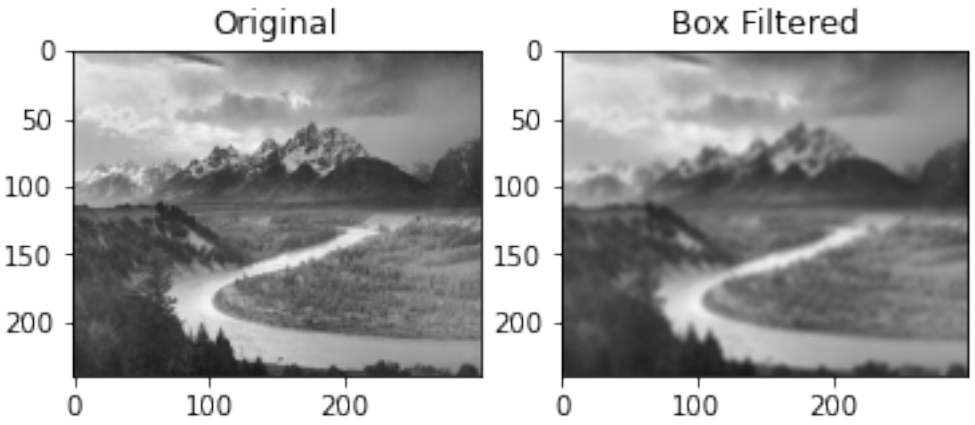


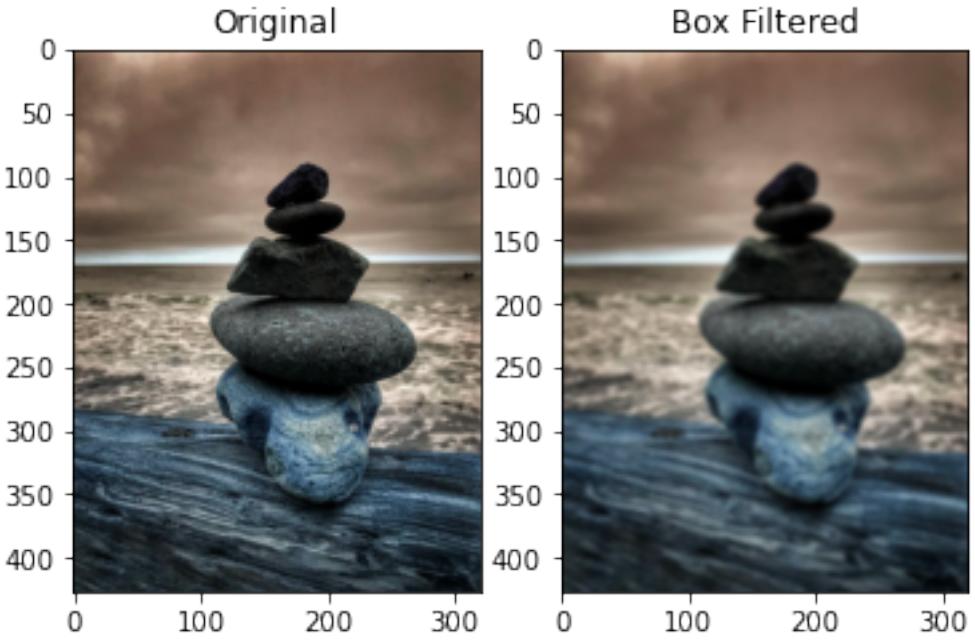
[5]:
```
a1.conv()
```

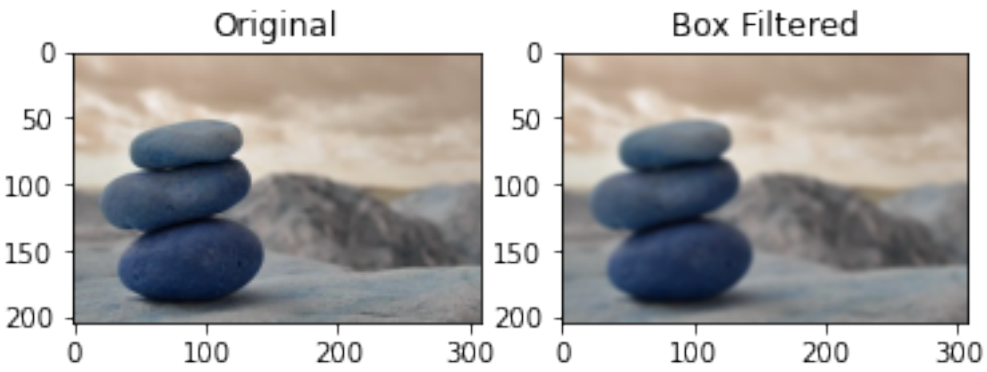
```
Enter Image Path('def' for default image): def
Enter Type of Convolution('sharpen' or 'smooth'): sharpen
```

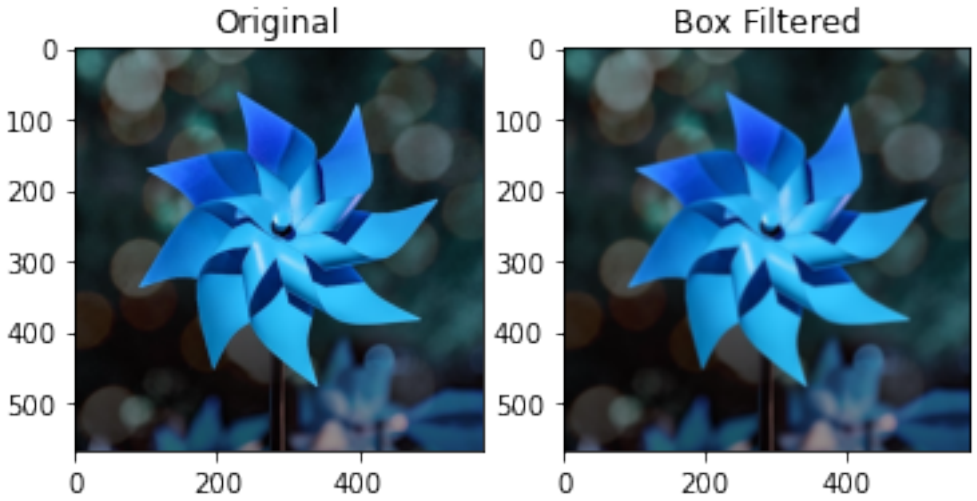


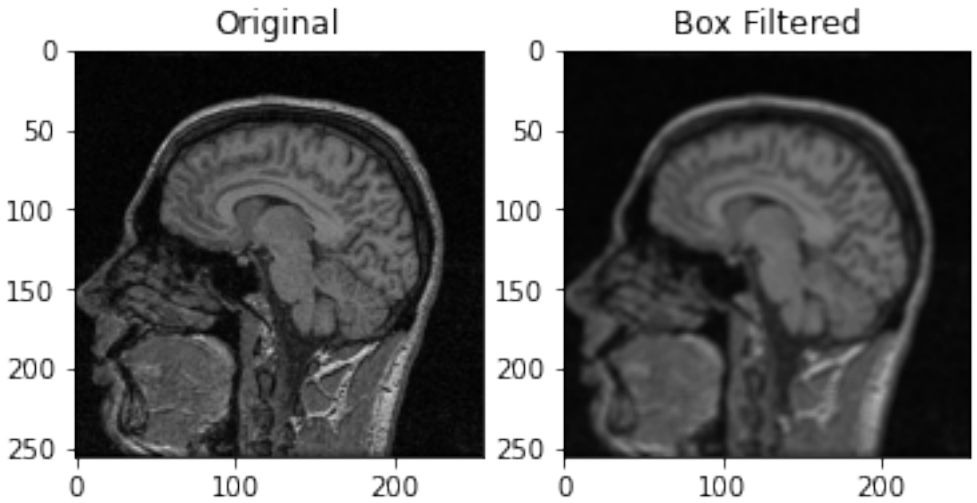
[6]:
```
a1.box()
```

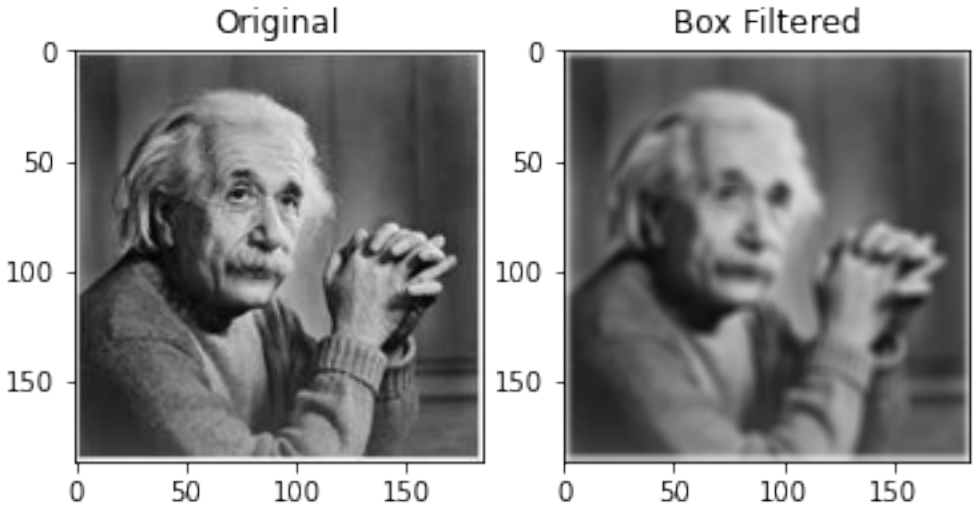

Original
Box Filtered
0
50
100
150
200
0
100
200


Original
Box Filtered
0
50
100
150
200
250
300
350
400
0
100
200
300


Original
Box Filtered
0
50
100
150
200
0
100
200
300

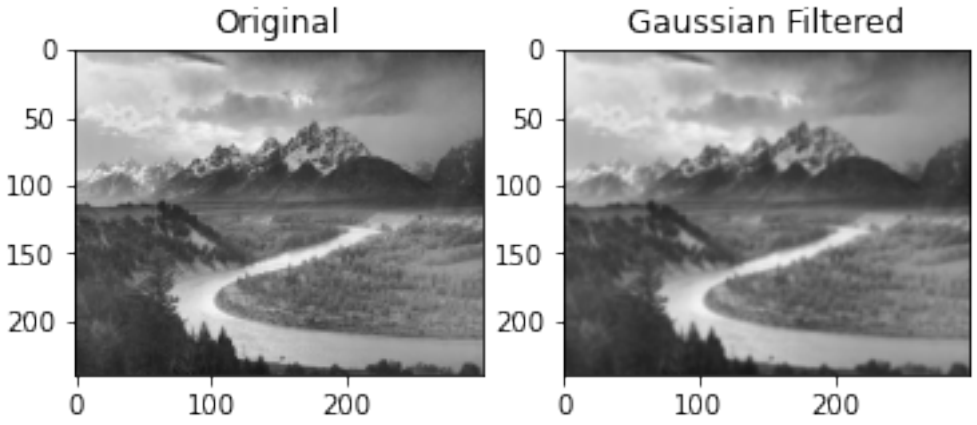

Original
Box Filtered
0
100
200
300
400
500
0
200
400

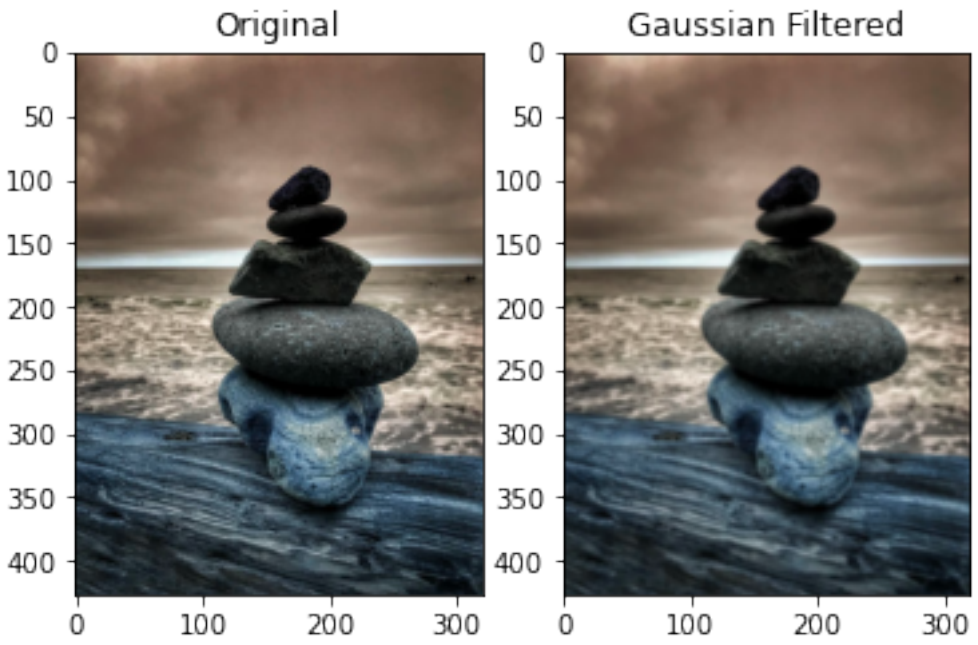

Original
Box Filtered
0
50
100
150
200
250
0
100
200

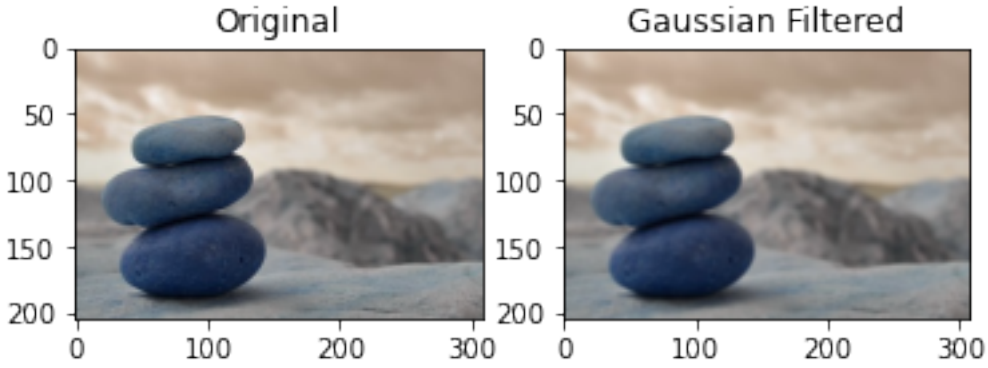


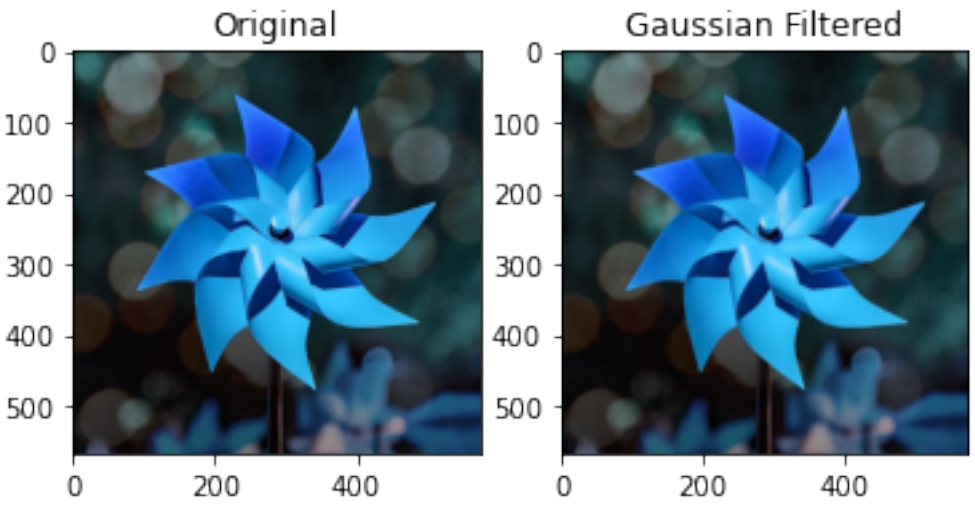
[7]:
```
a1.gaus()
```

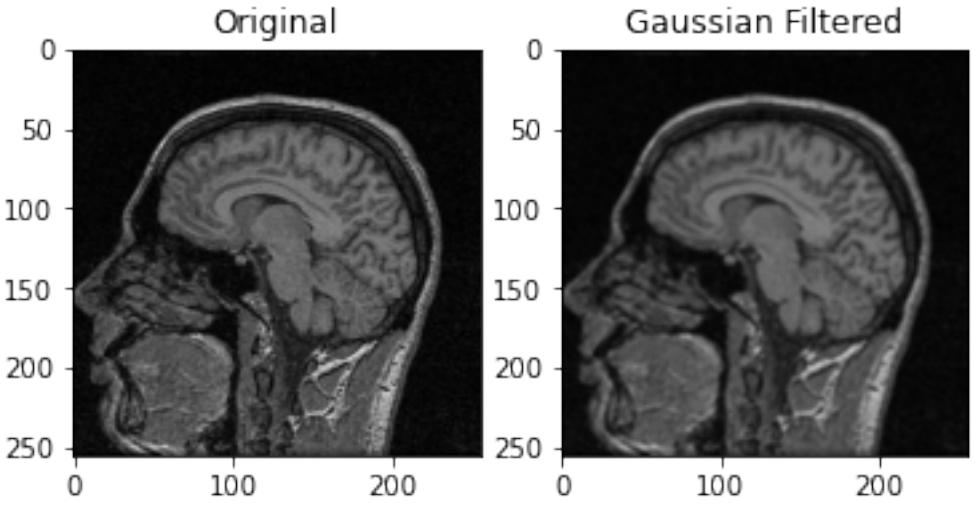
```
Enter sigma value for filter= 10
```

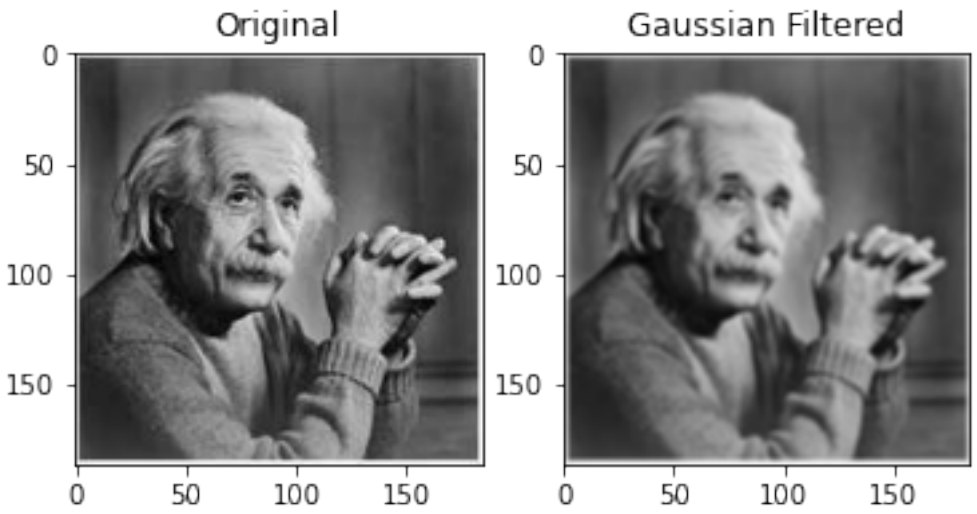



Original
Gaussian Filtered
0
50
100
150
200
250
300
350
400
0
100
200
300


Original
Gaussian Filtered
0
50
100
150
200
0
100
200
300

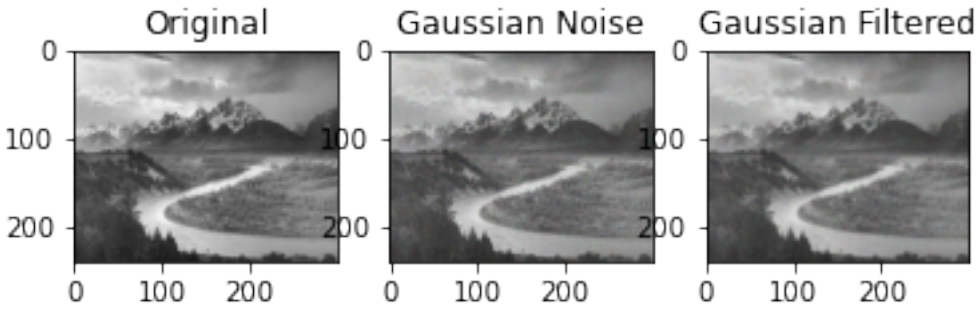

Original
Gaussian Filtered
0
100
200
300
400
500
0
200
400

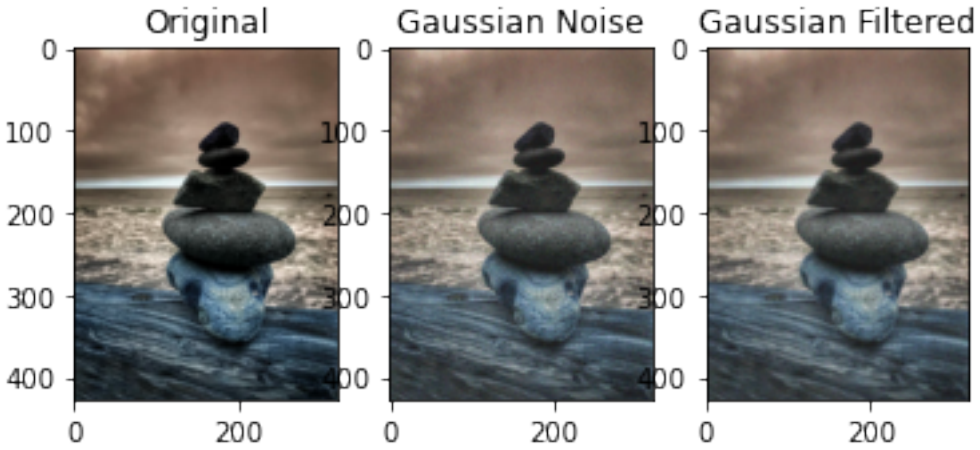

Original
Gaussian Filtered
0
50
100
150
200
250
0
100
200

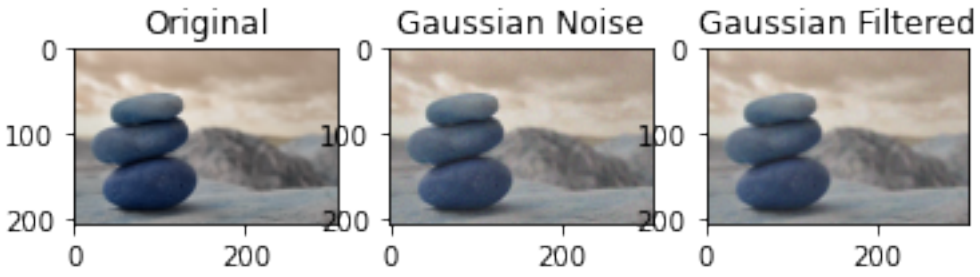


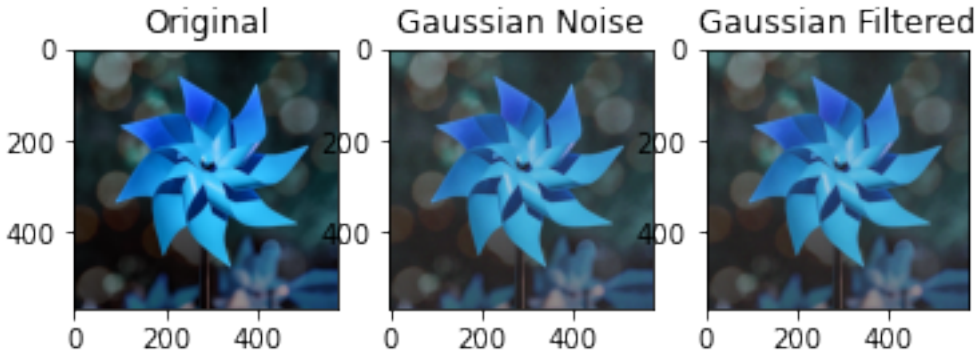
[8]:
```
a1.gausnoise()
```

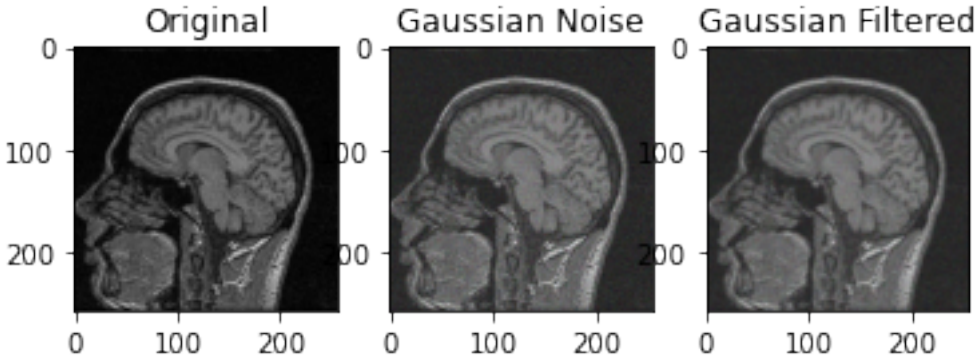
```
Enter sigma value for noise= 10
```

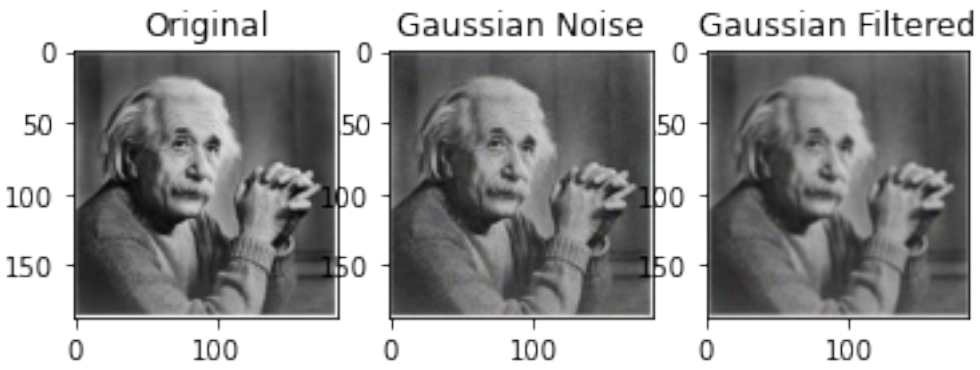





Original
Gaussian Noise
Gaussian Filtered
0
100
200
0
200

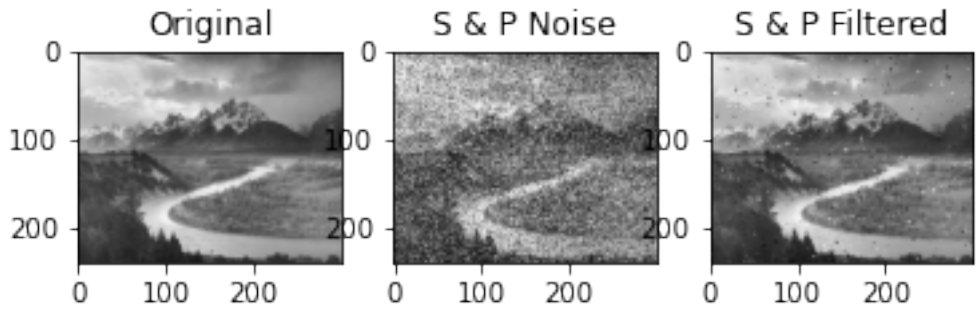

Original
Gaussian Noise
Gaussian Filtered
0
200
400
0
200
400

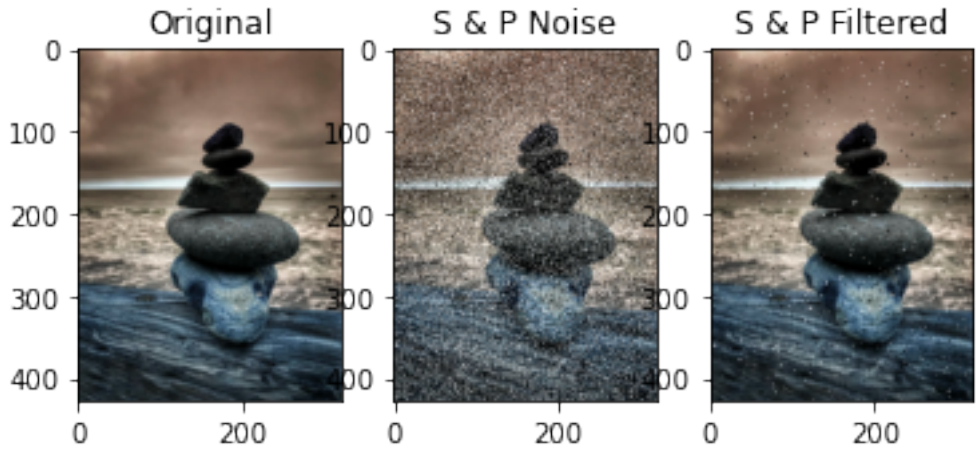

Original
Gaussian Noise
Gaussian Filtered
0
100
200
0
100
200

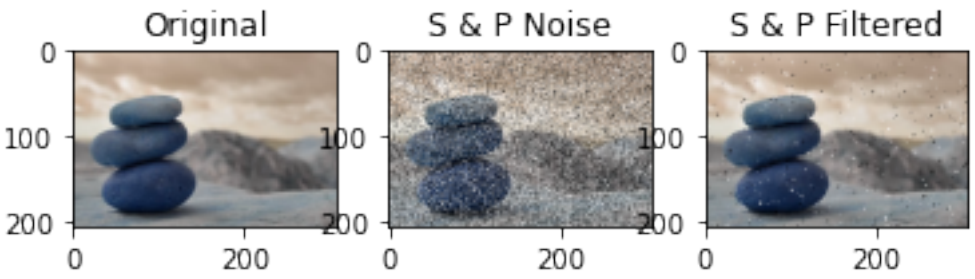


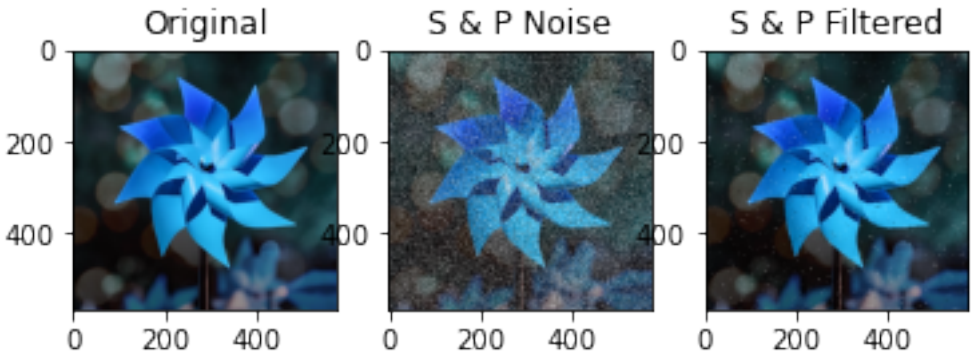
[9]:
```
a1.saltandpepper()
```

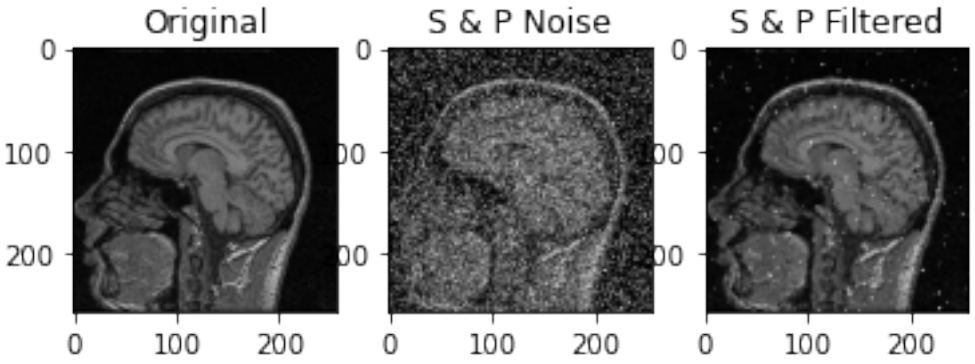
```
Enter probability of S&P Noise= 0.3
```

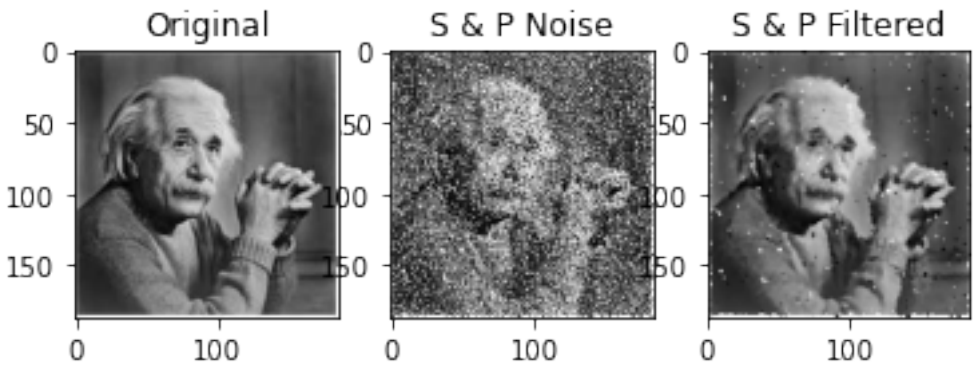





Original
S & P Noise
S & P Filtered
0
100
200
0
200

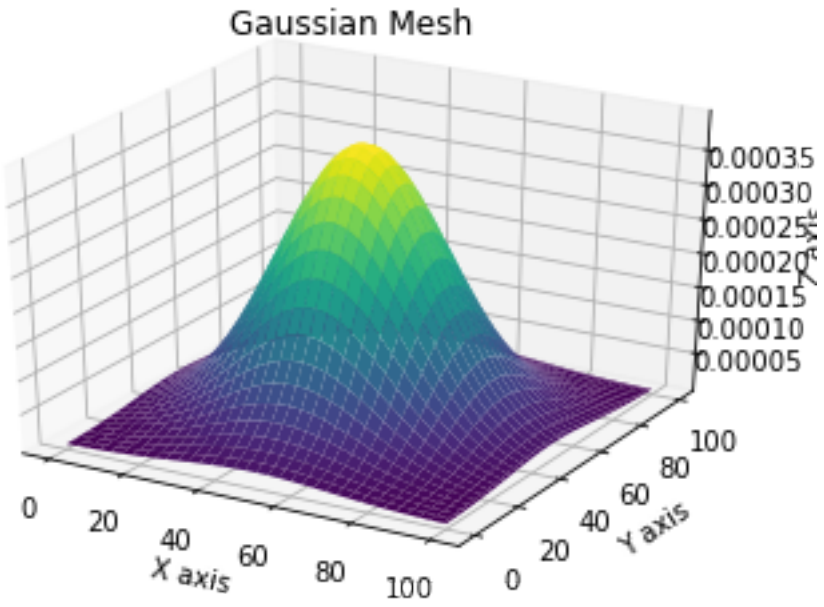

Original
S & P Noise
S & P Filtered
0
200
400
0
200
400

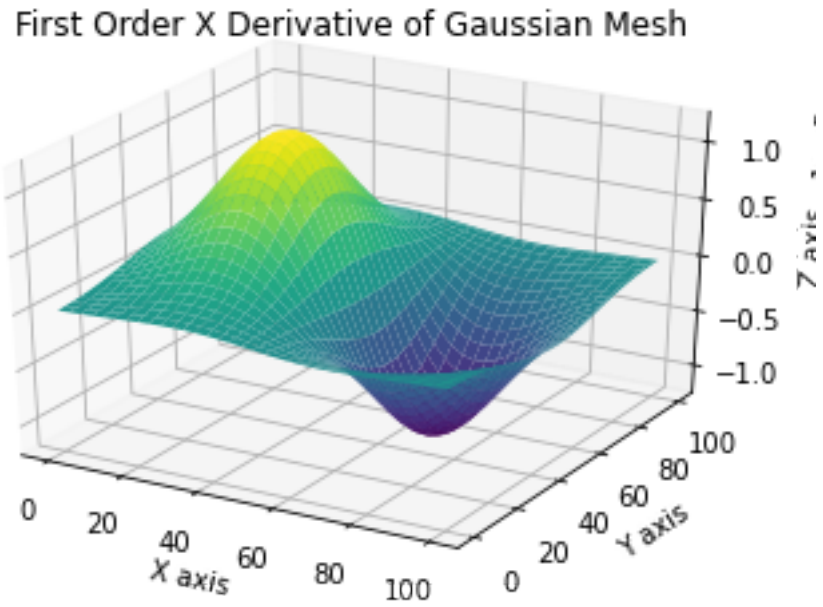

Original
S & P Noise
S & P Filtered
0
100
200
0
100
200

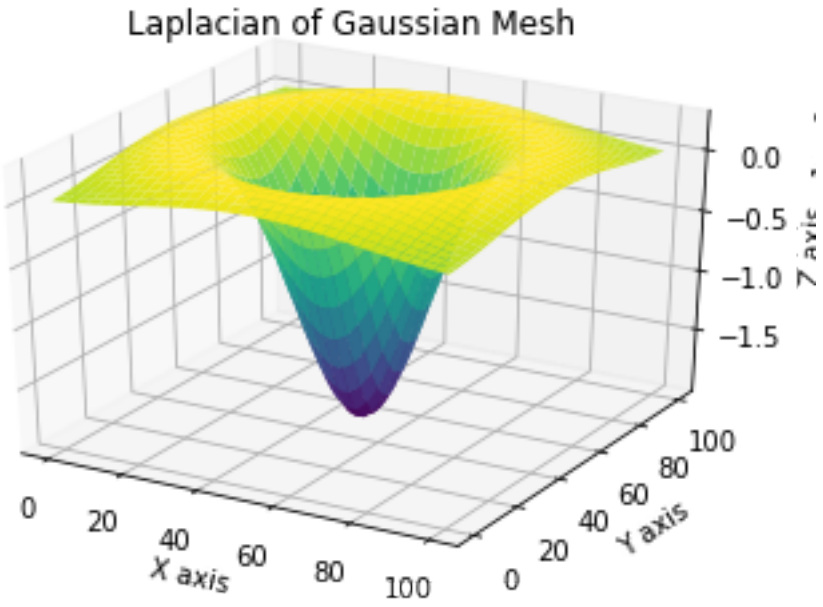


[2]:
```
a1.mesh()
```

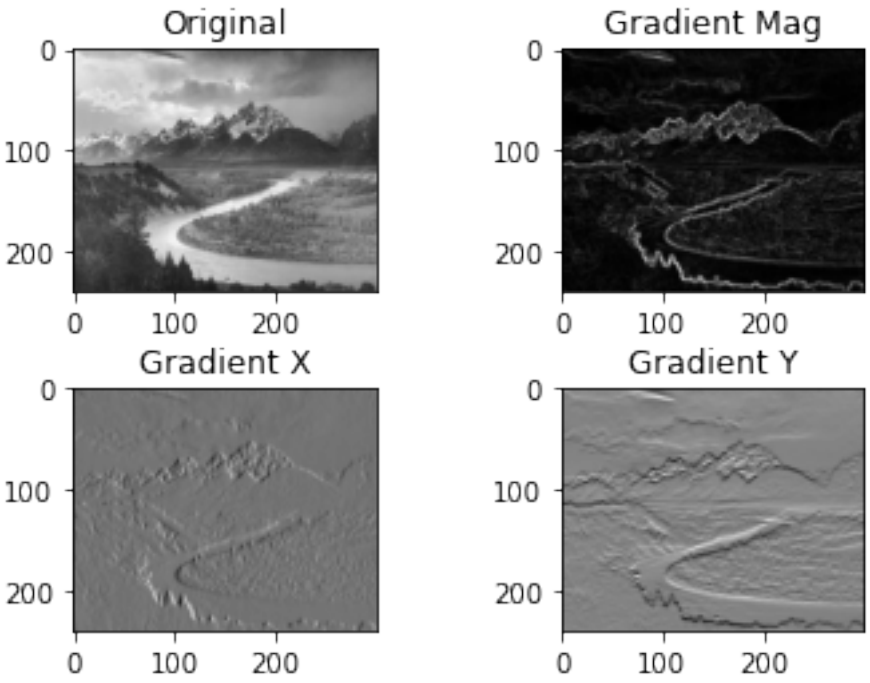
```
Enter sigma value for mesh plots= 20
```

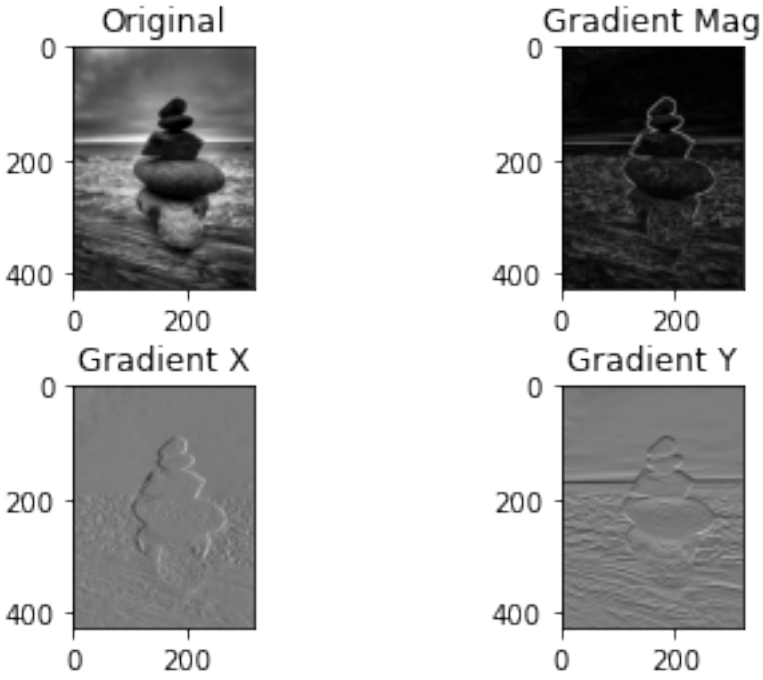


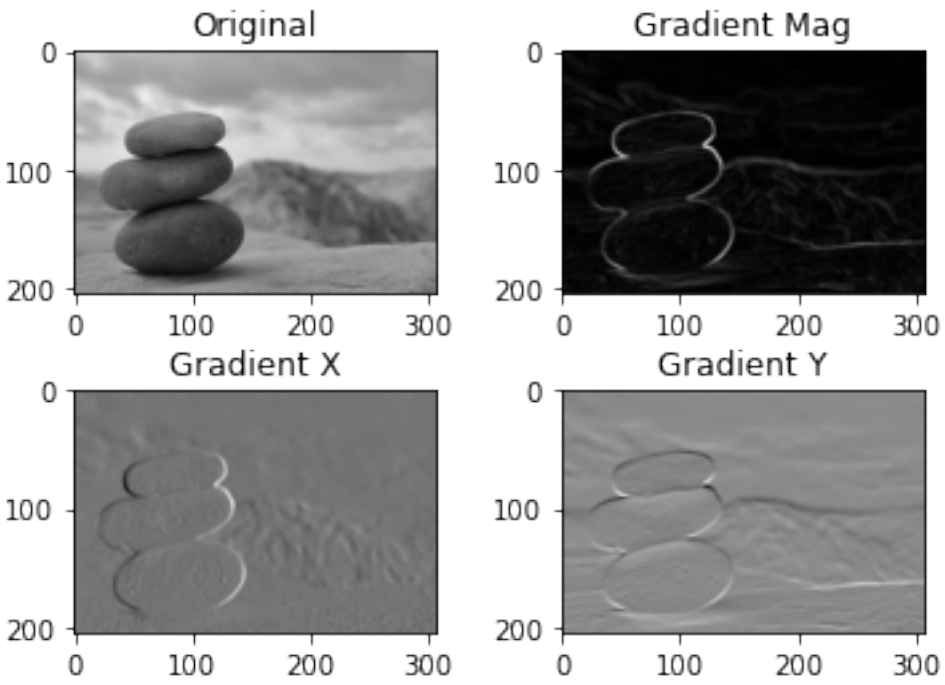

First Order X Derivative of Gaussian Mesh
1.0
0.5
0.0
−0.5
−1.0
0
20
40
60
80
100
X axis
0
20
40
60
80
100
Y axis

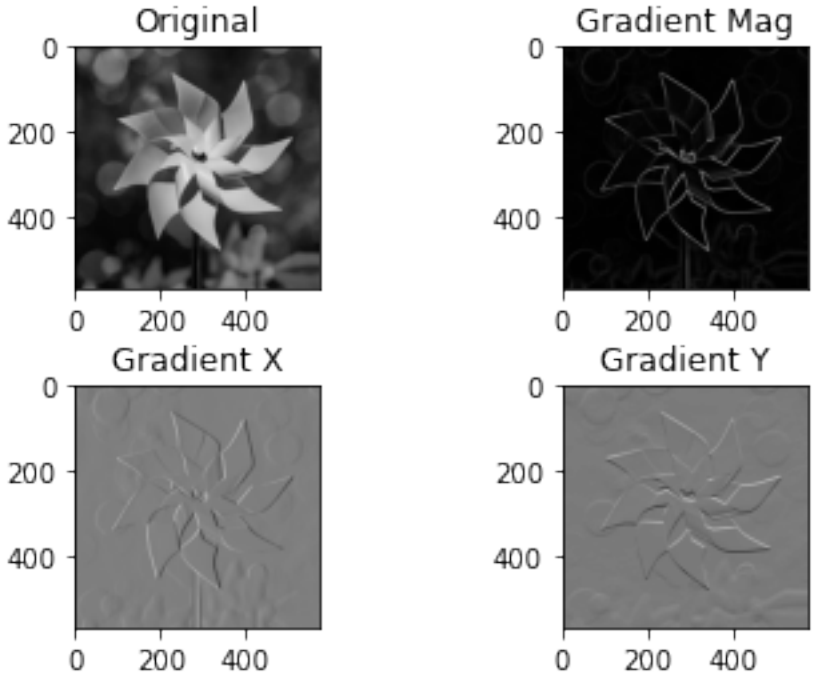

Laplacian of Gaussian Mesh
0.0
−0.5
−1.0
−1.5
0
20
40
60
80
100
X axis
0
20
40
60
80
100
Y axis

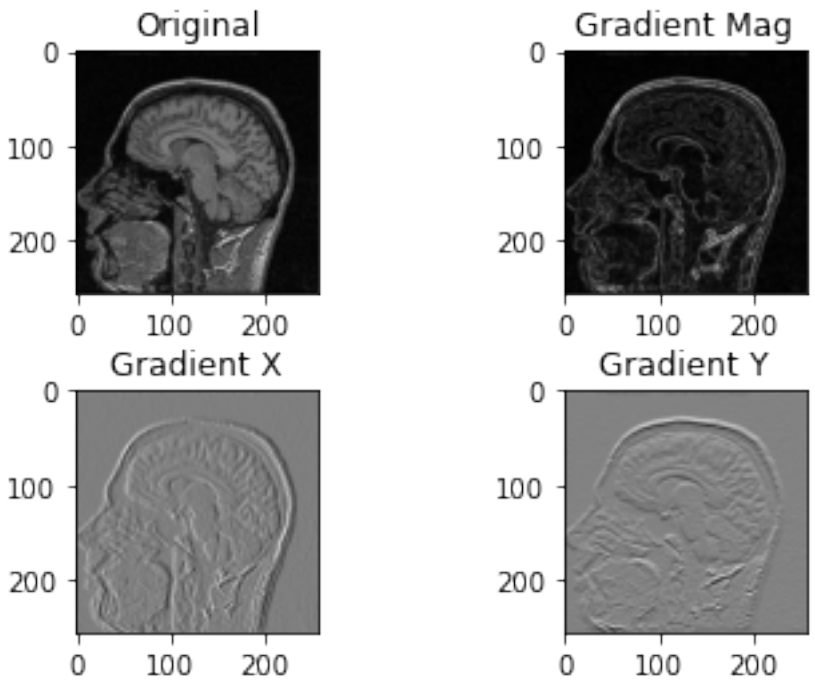
[11]:
```
a1.sobel()
```

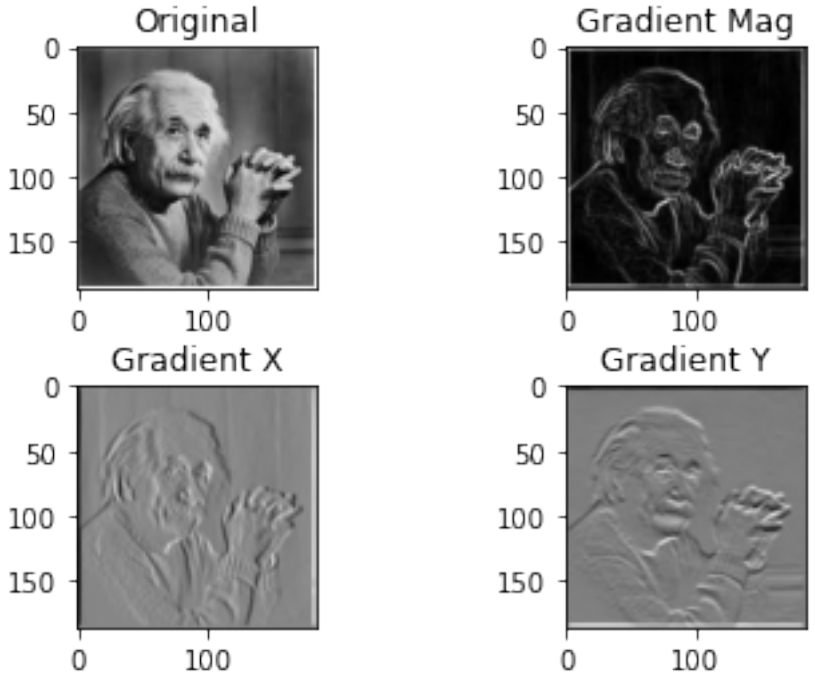

Original
Gradient Mag
Gradient X
Gradient Y
0
100
200


Original
Gradient Mag
Gradient X
Gradient Y
0
200
400

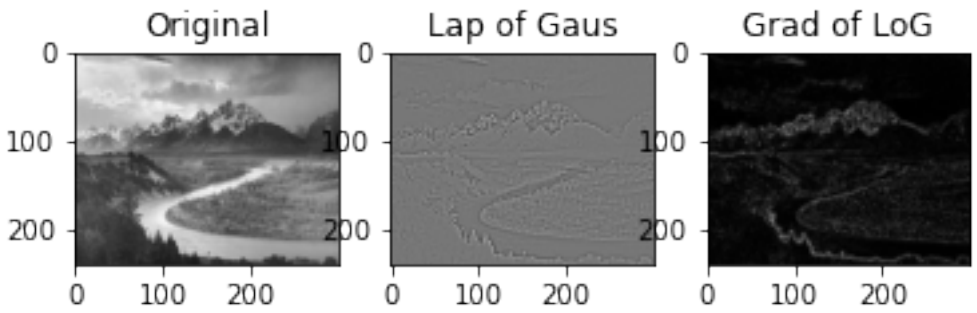

Original
Gradient Mag
Gradient X
Gradient Y
0
100
200
300

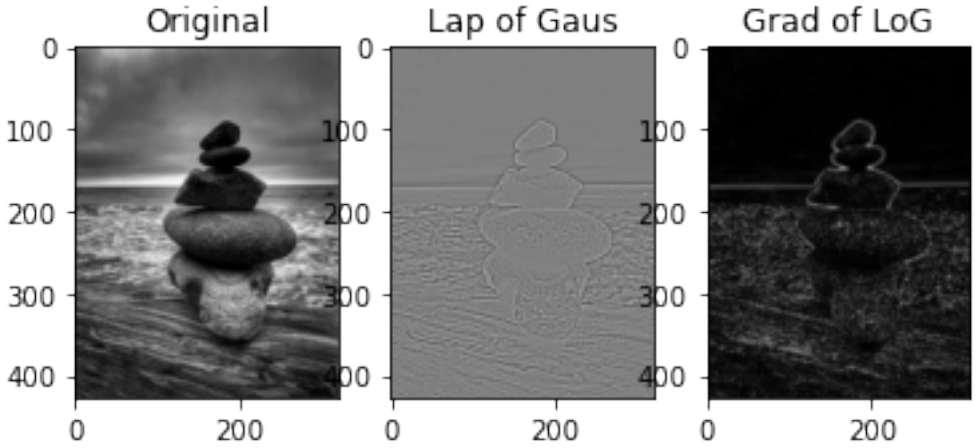

Original
Gradient Mag
Gradient X
Gradient Y
0
200
400

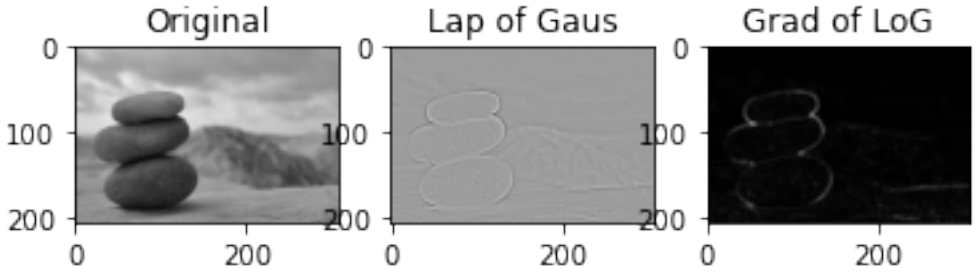

Original
Gradient Mag
Gradient X
Gradient Y
0
100
200

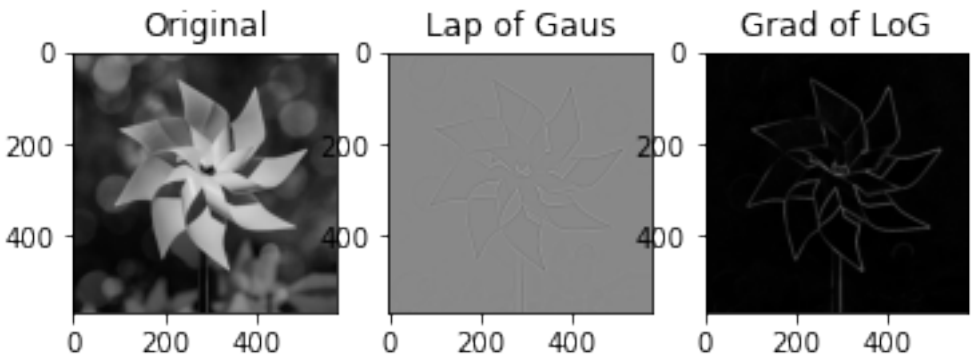

Original
Gradient Mag
Gradient X
Gradient Y
0
50
100
150

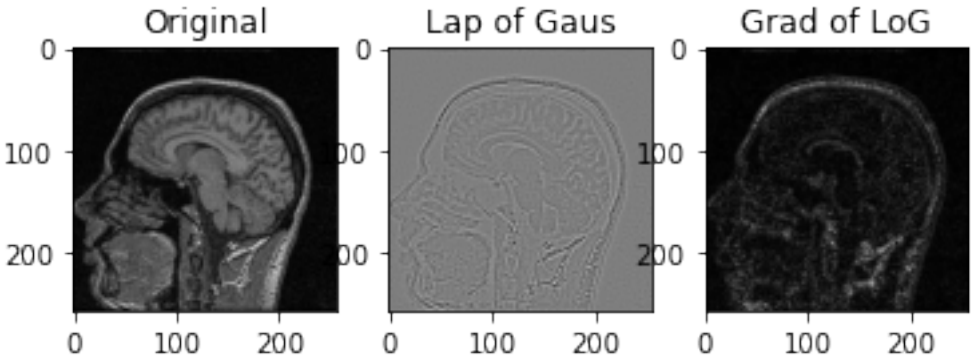
[12]:
```
a1.LoG()
```

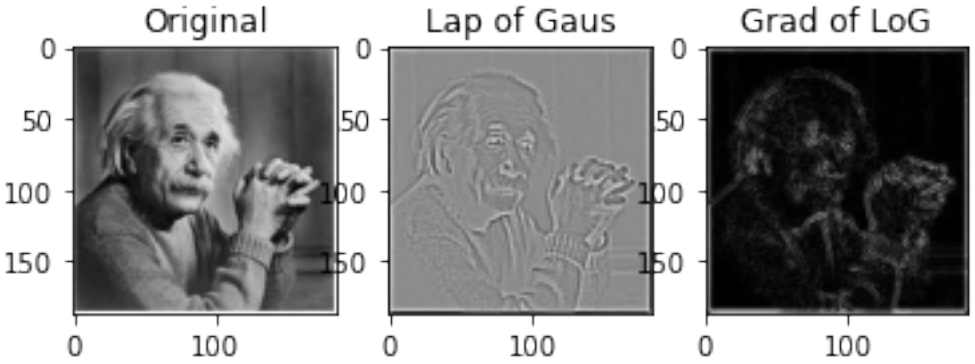

Original
Lap of Gaus
Grad of LoG
0
100
200
0
100
200


Original
Lap of Gaus
Grad of LoG
0
100
200
300
400
0
200

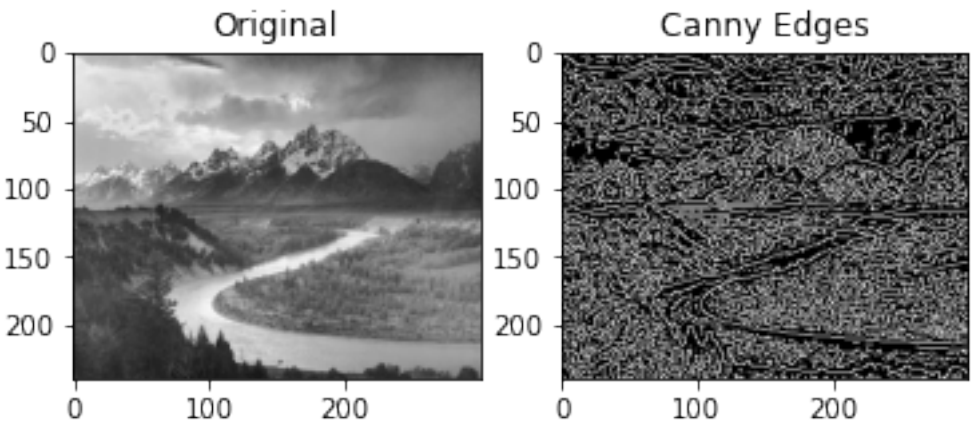

Original
Lap of Gaus
Grad of LoG
0
100
200
0
200

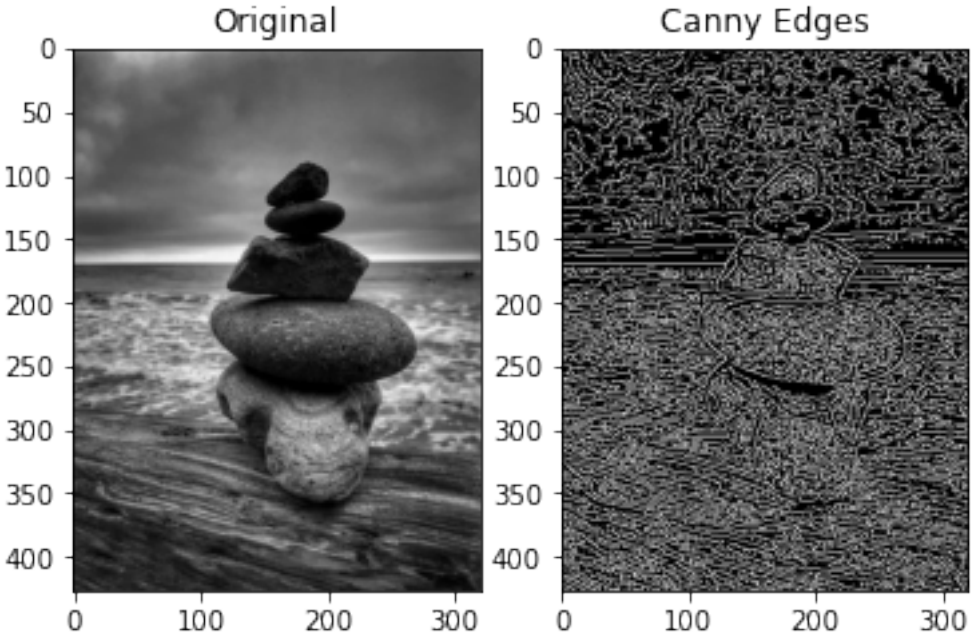

Original
Lap of Gaus
Grad of LoG
0
200
400
0
200
400

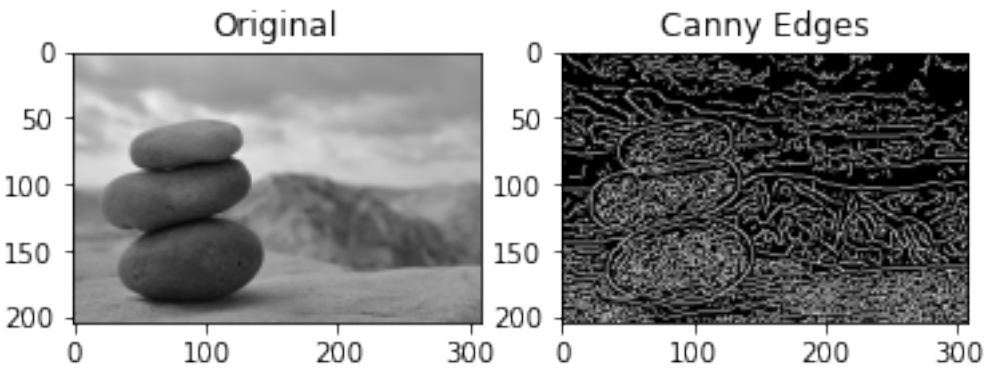

Original
Lap of Gaus
Grad of LoG
0
100
200
0
100
200

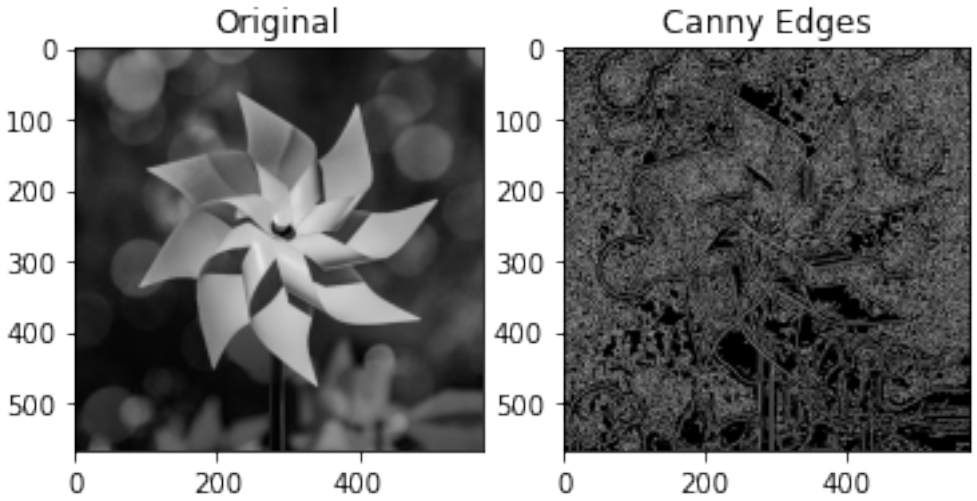

Original
Lap of Gaus
Grad of LoG
0
50
100
150
0
100

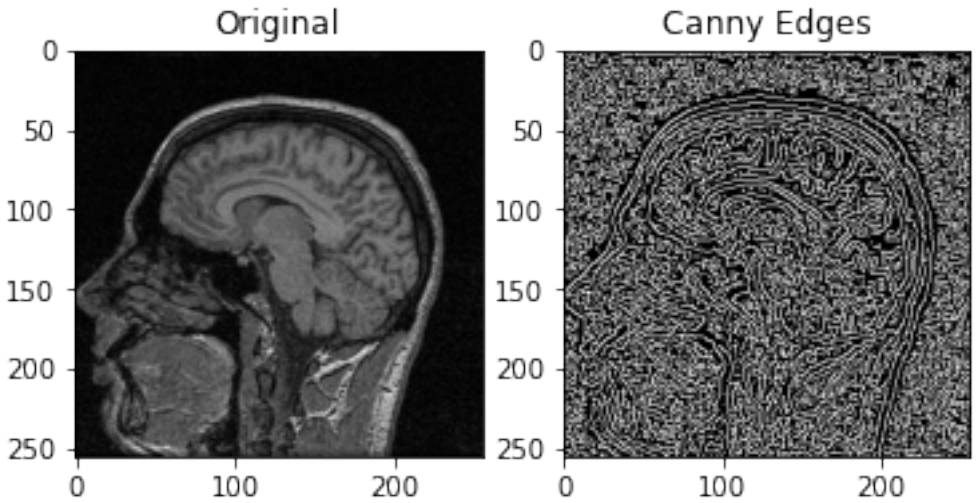
[2]:
```
a1.canny()
```

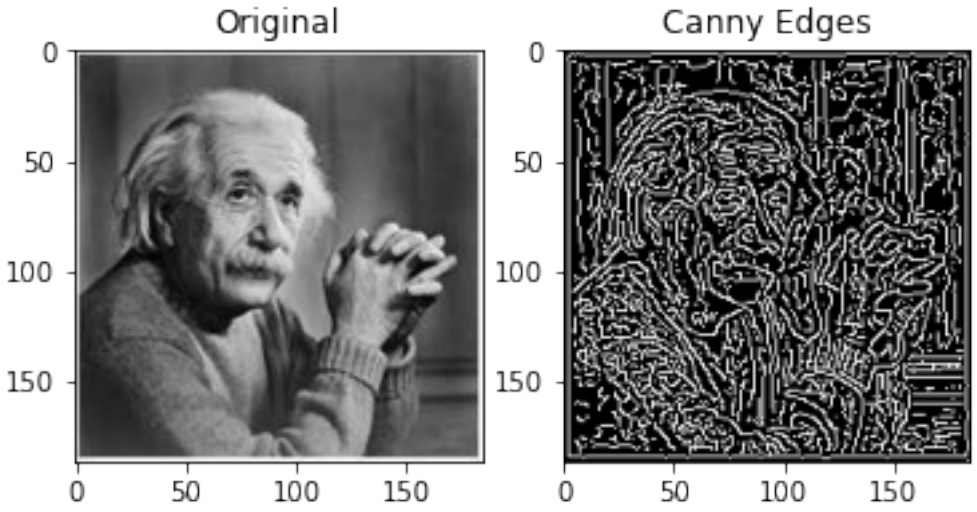

Original
Canny Edges
0
50
100
150
200
0
100
200


Original
Canny Edges
0
50
100
150
200
250
300
350
400
0
100
200
300

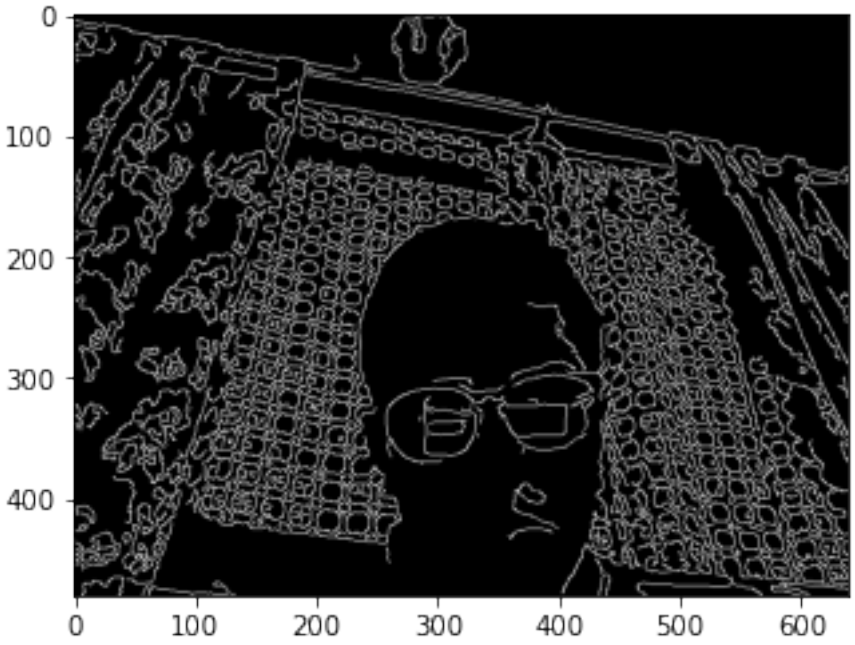

Original
Canny Edges
0
50
100
150
200
0
100
200
300

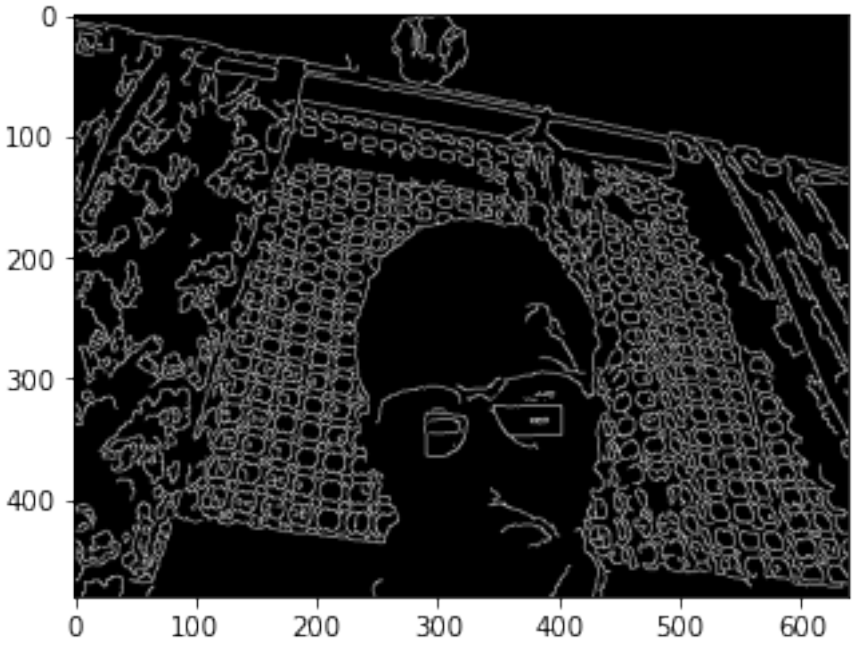

Original
Canny Edges
0
100
200
300
400
500
0
200
400

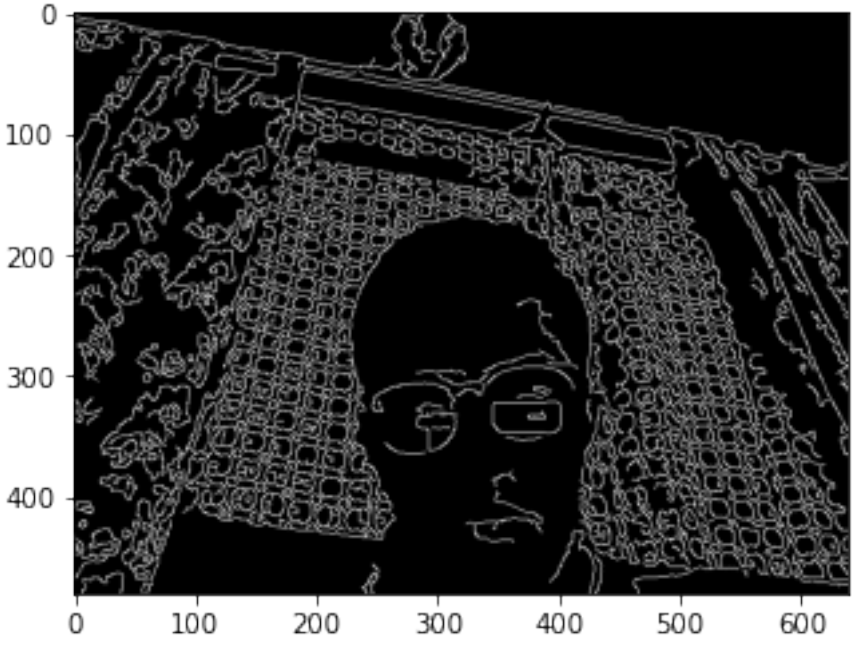

Original
Canny Edges
0
50
100
150
200
250
0
100
200

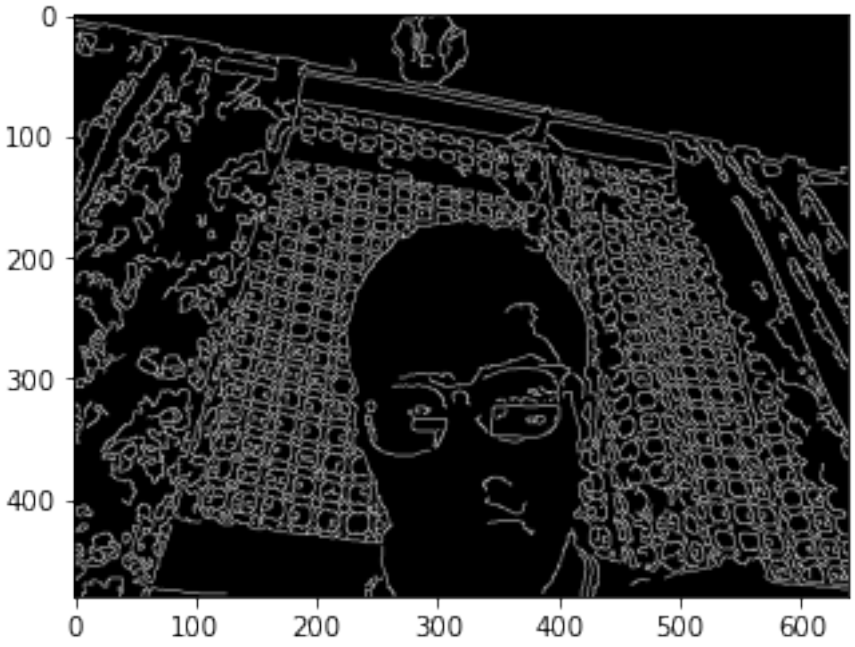


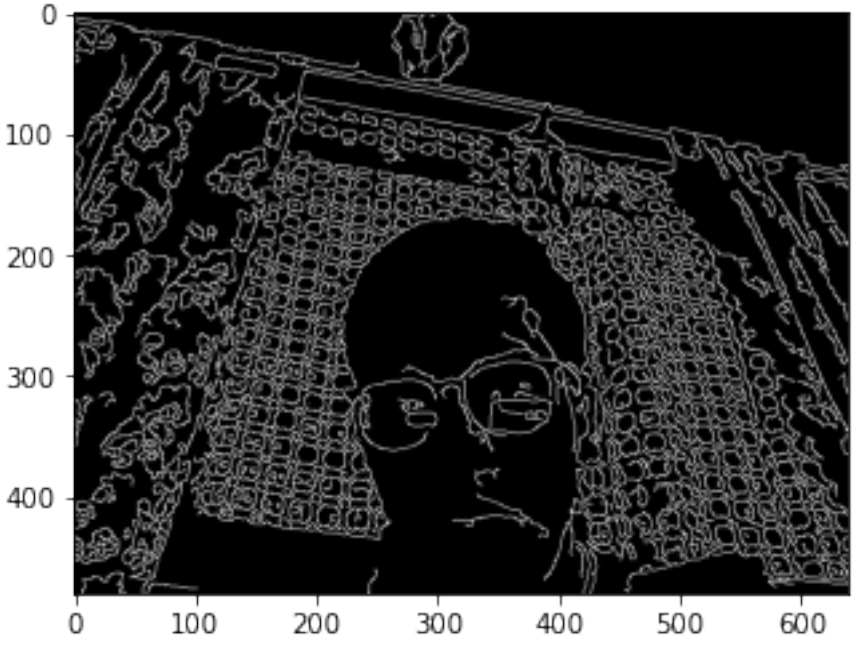
[2]:
```
a1.cannyvid()
```

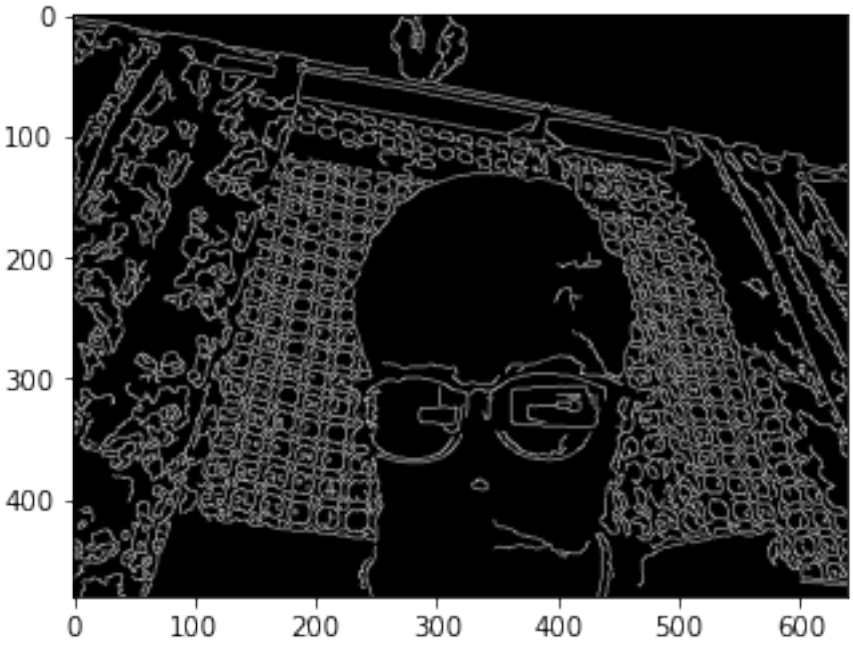


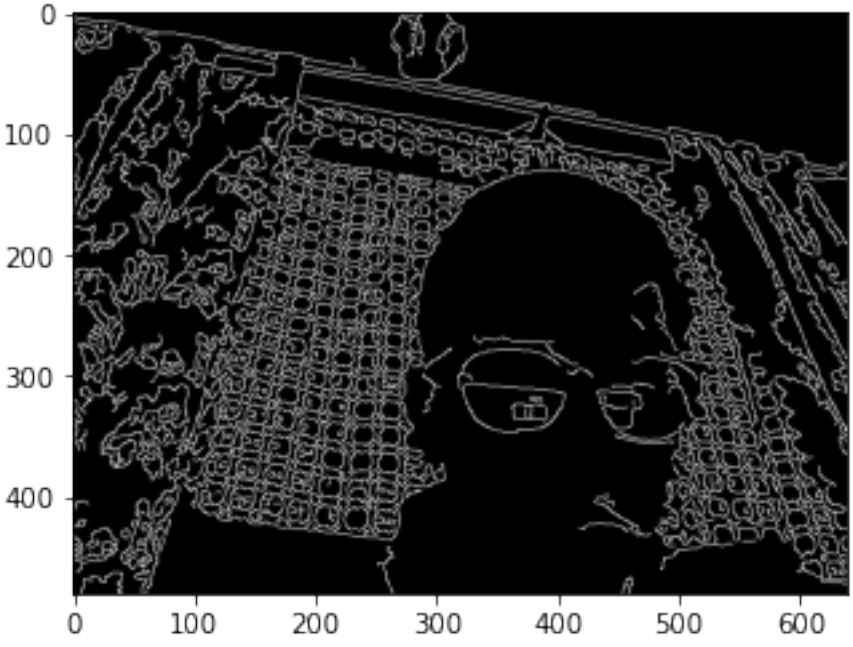

0
100
200
300
400
0
100
200
300
400
500
600

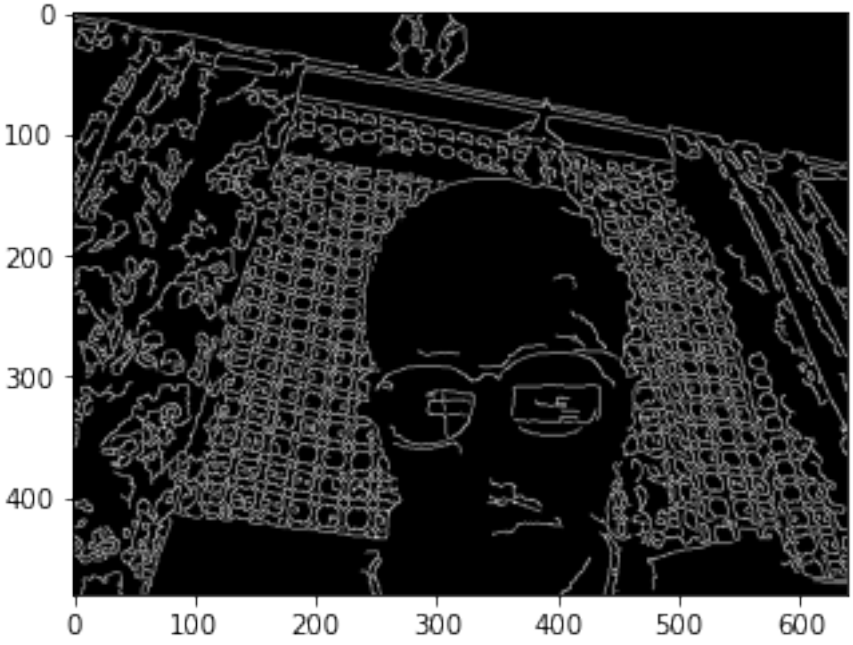

0
100
200
300
400
0
100
200
300
400
500
600

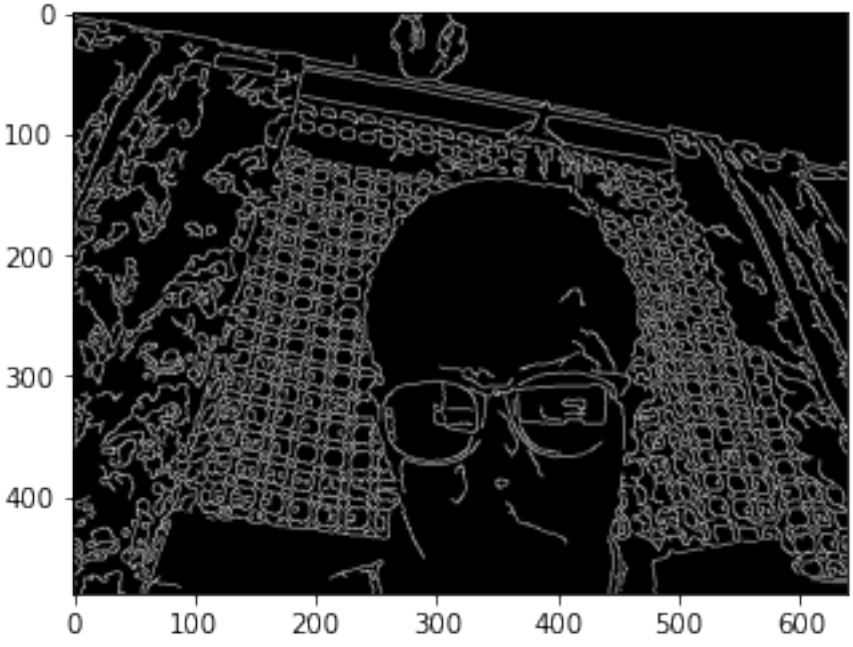

0
100
200
300
400
0
100
200
300
400
500
600

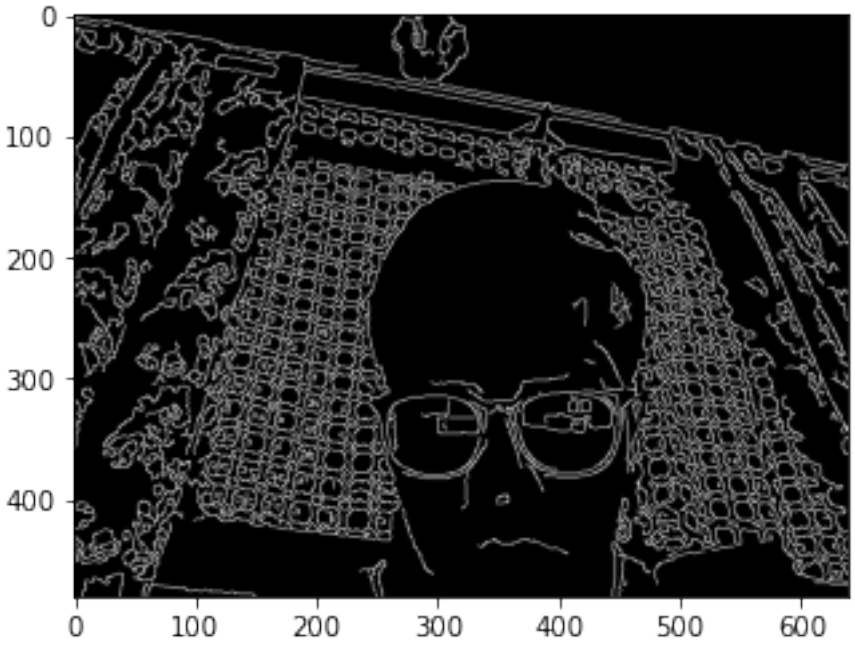

0
100
200
300
400
0
100
200
300
400
500
600

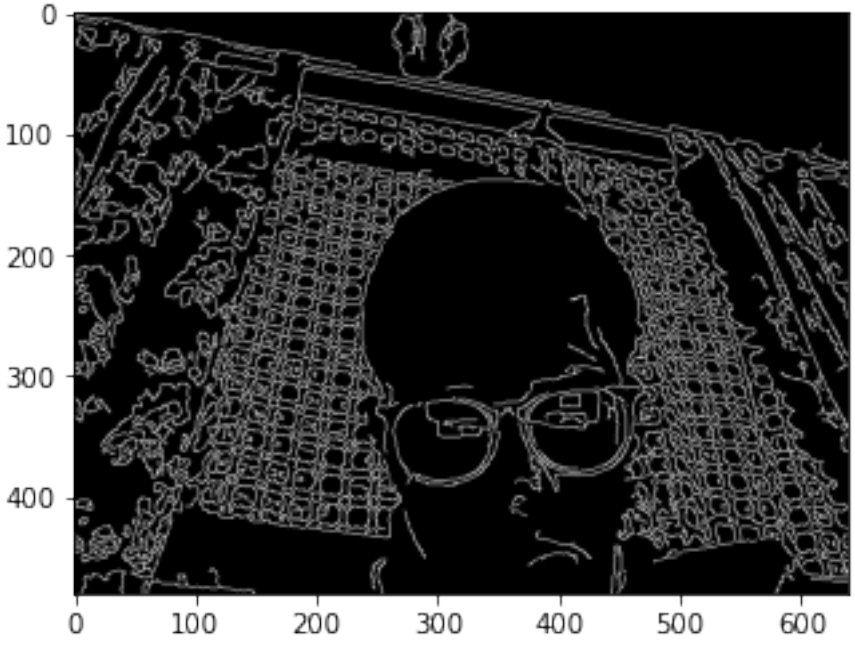

0
100
200
300
400
0
100
200
300
400
500
600

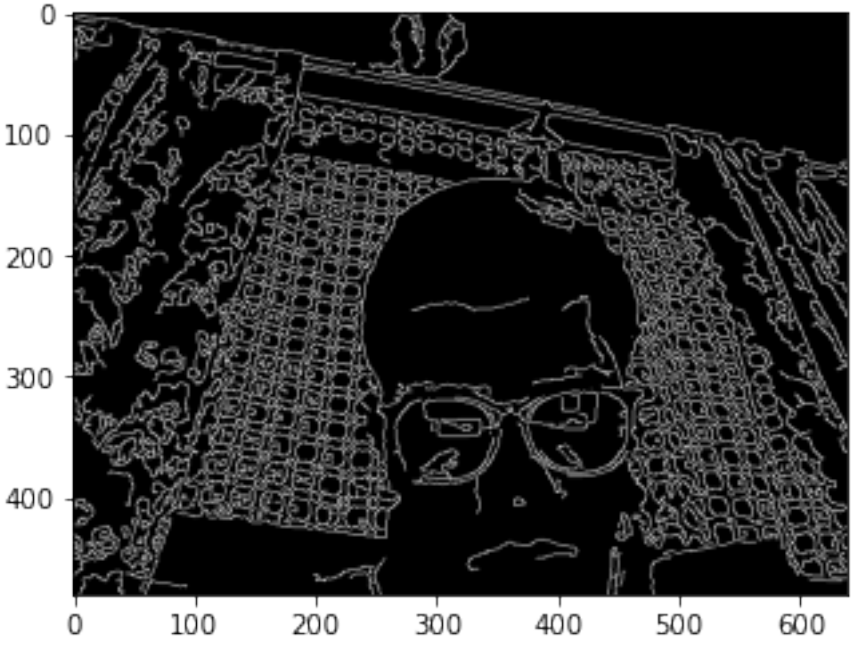

0
100
200
300
400
0
100
200
300
400
500
600

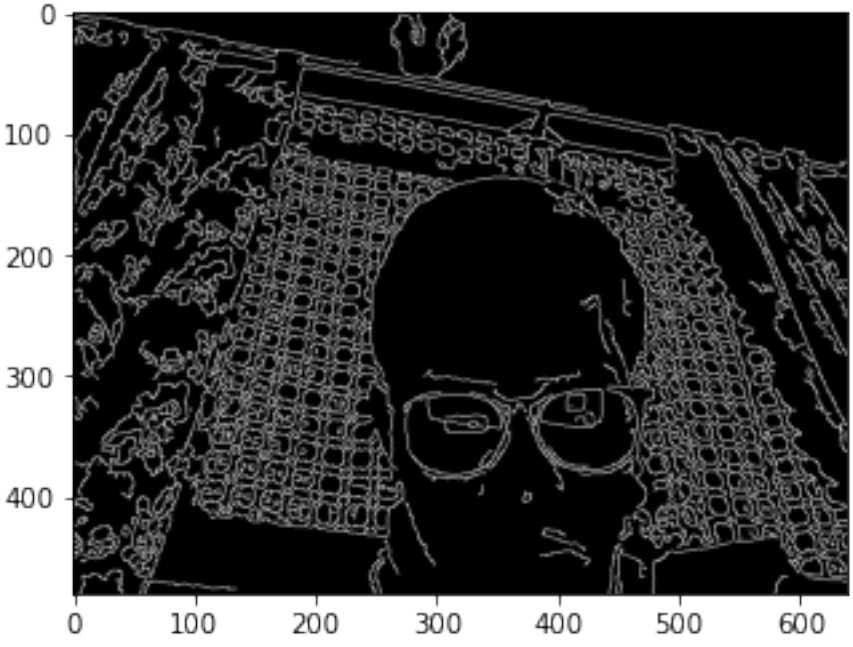

0
100
200
300
400
0
100
200
300
400
500
600

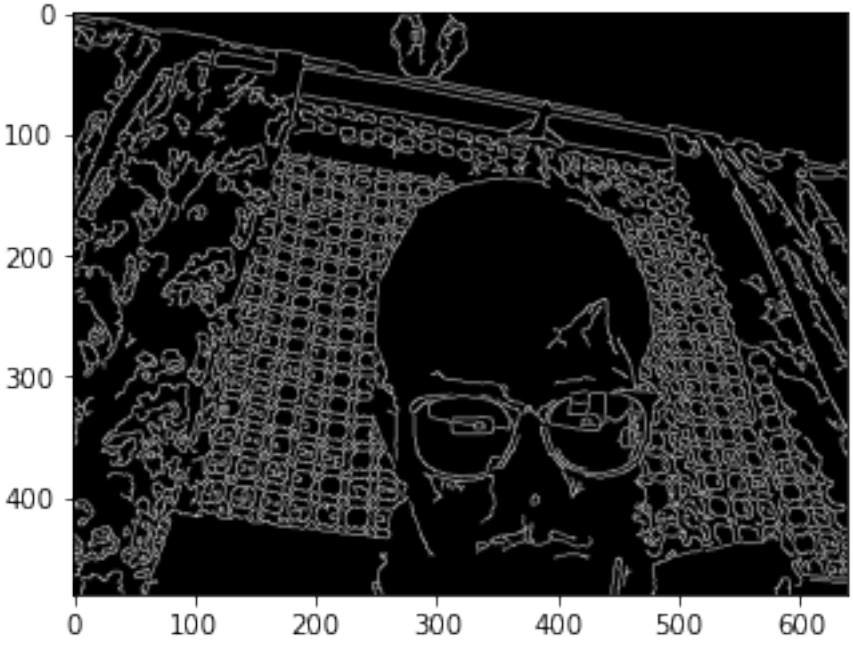

0
100
200
300
400
0
100
200
300
400
500
600

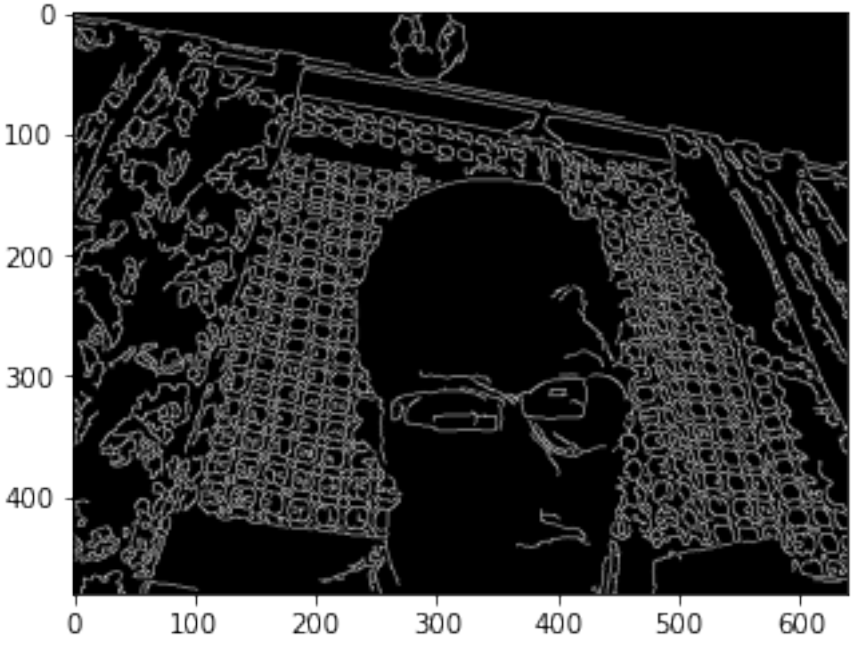

0
100
200
300
400
0
100
200
300
400
500
600

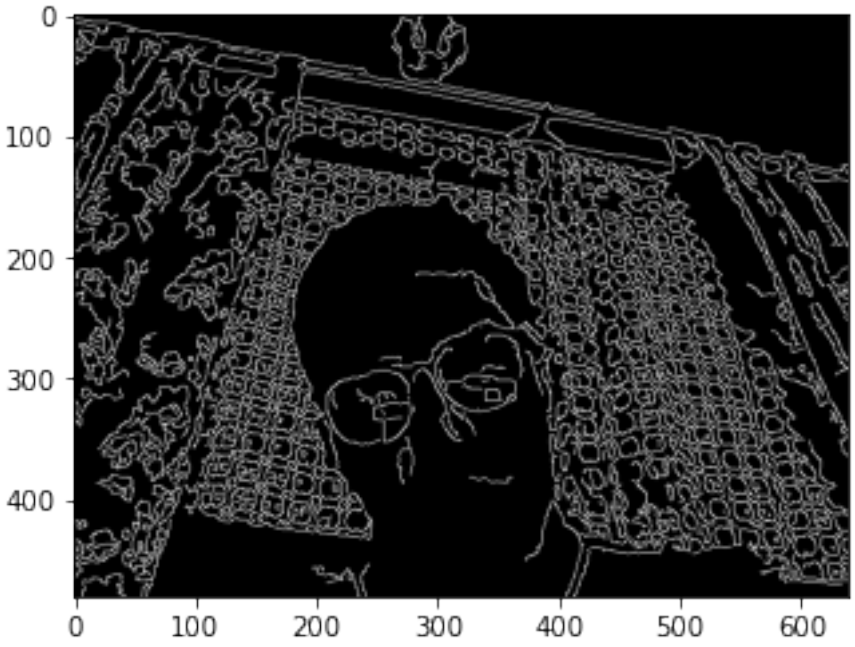

0
100
200
300
400
0
100
200
300
400
500
600


0
100
200
300
400
0
100
200
300
400
500
600


0
100
200
300
400
0
100
200
300
400
500
600


0
100
200
300
400
0
100
200
300
400
500
600


0
100
200
300
400
0
100
200
300
400
500
600


0
100
200
300
400
0
100
200
300
400
500
600

[ ]: