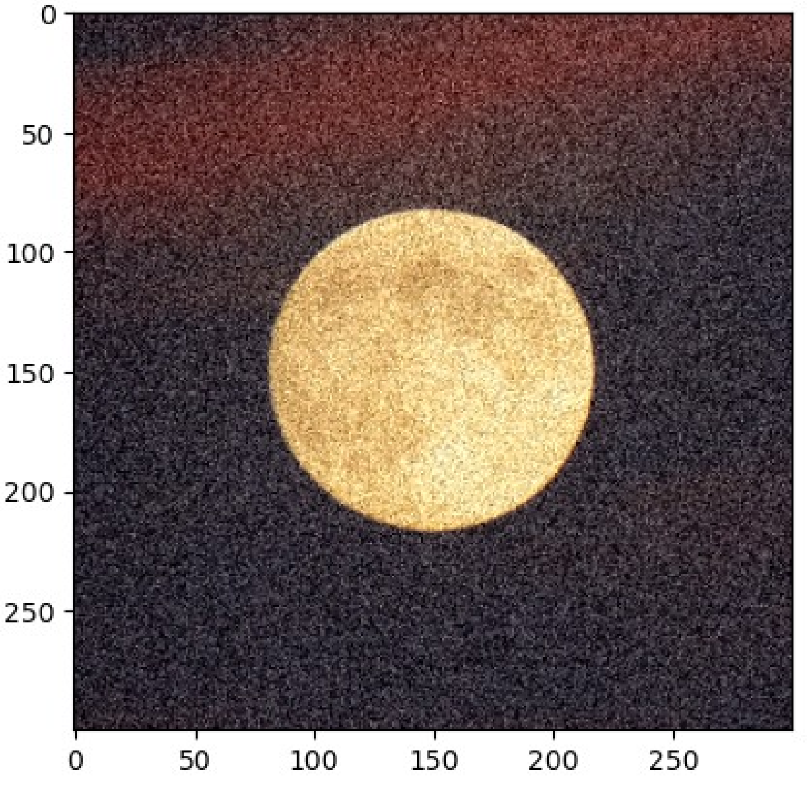
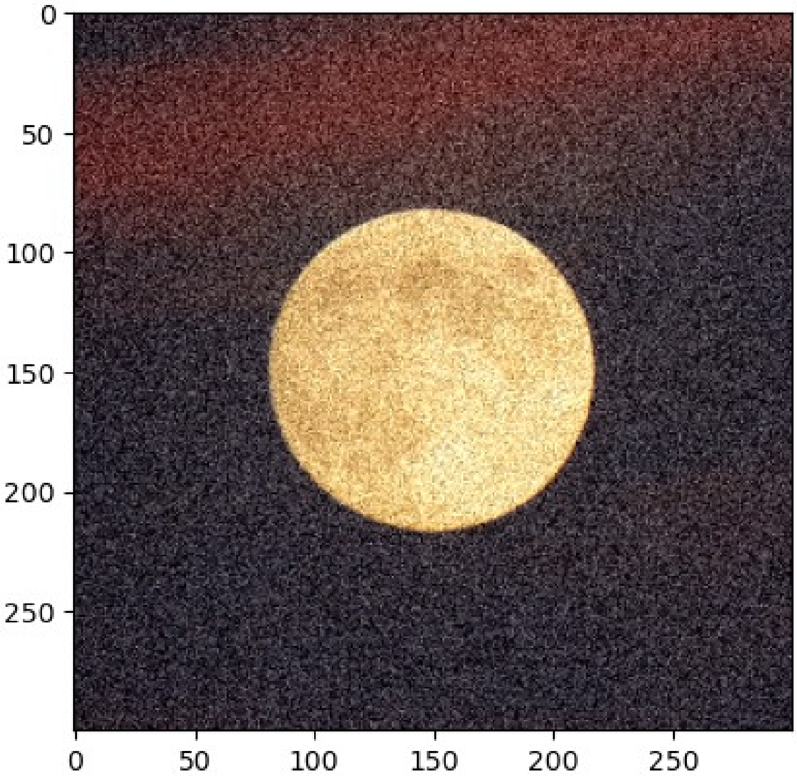
NAME : BHASKAR KAROL

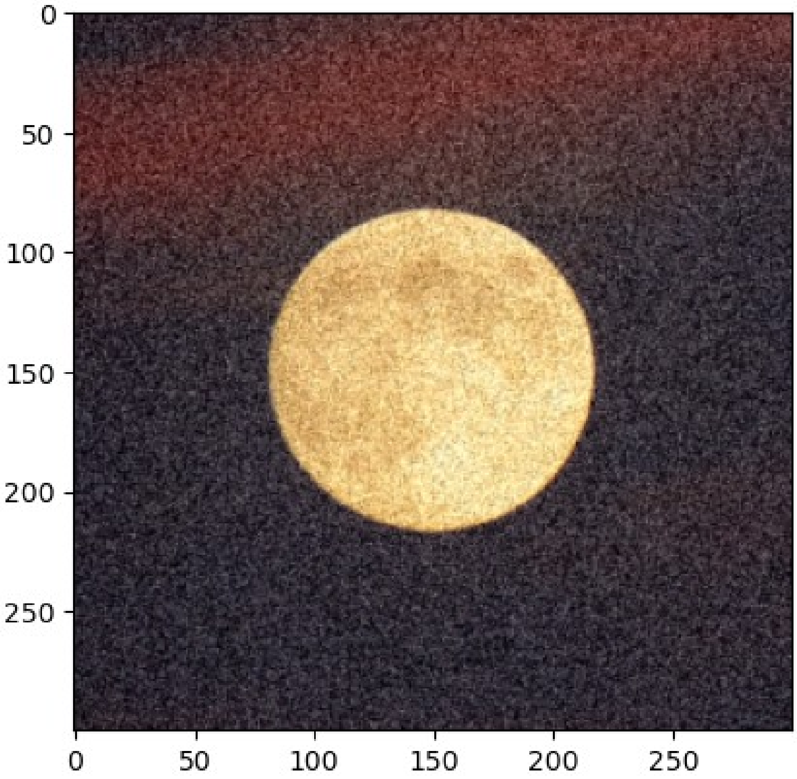
SR.NO : 24076

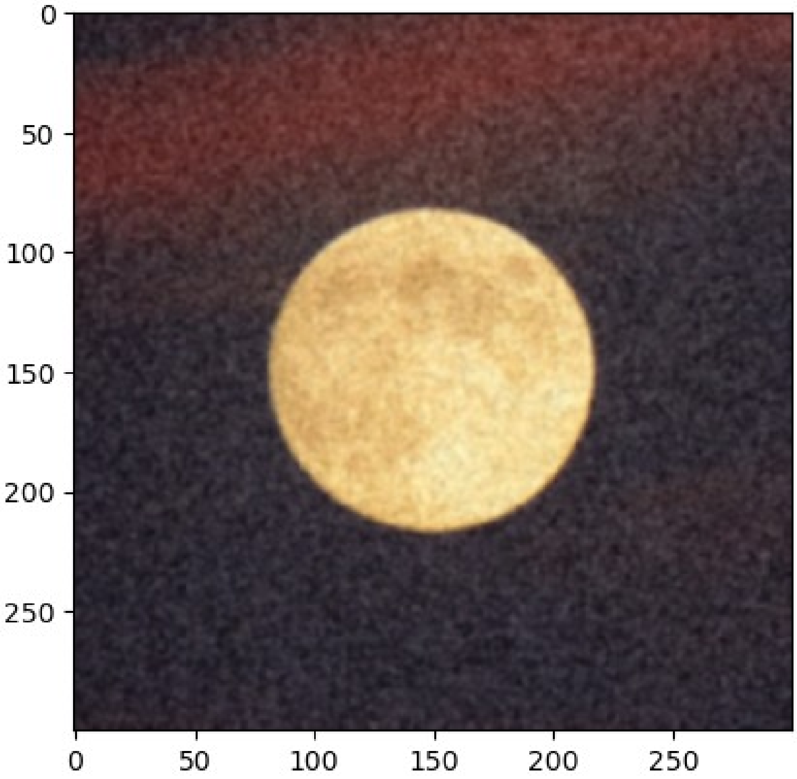
SUBJECT : DIGITAL IMAGE PROCESSING

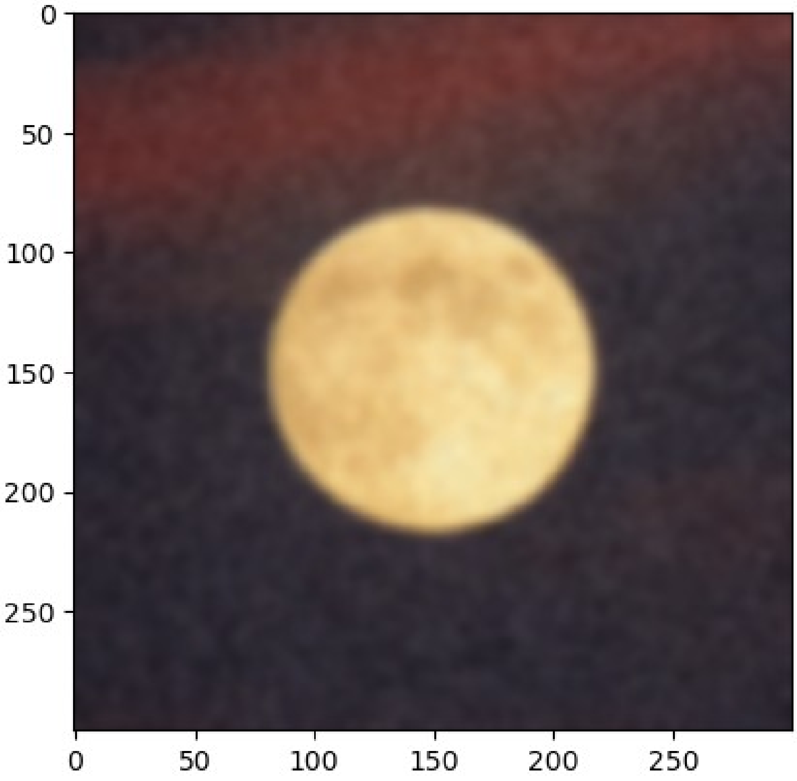
DEPARTMENT : ECE

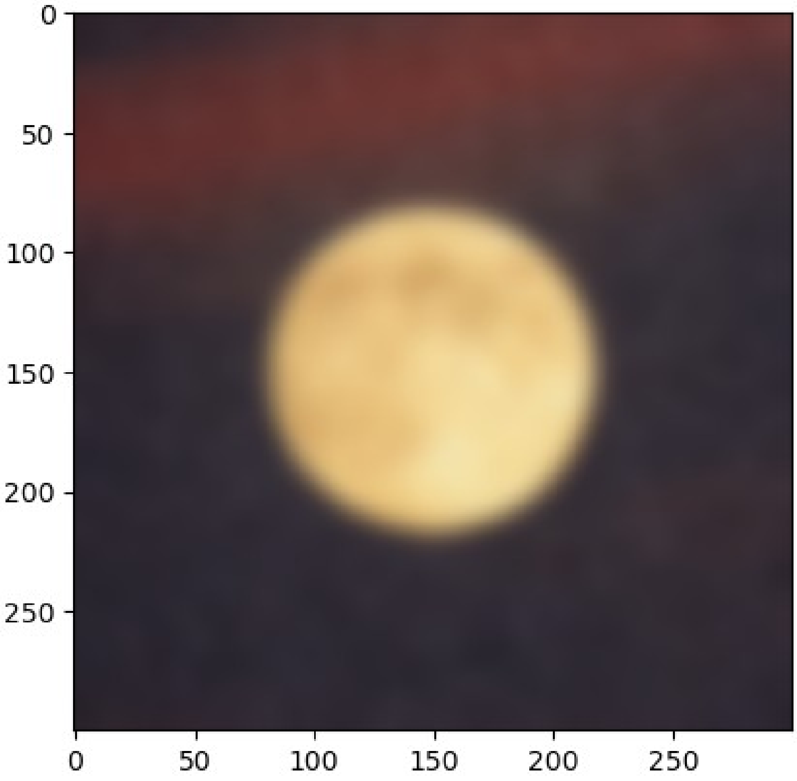


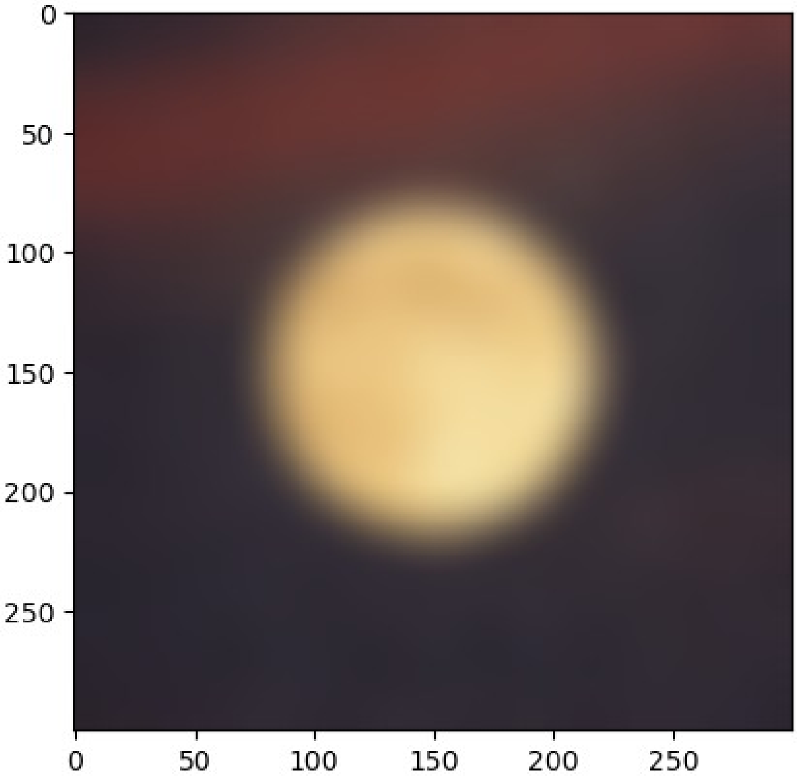

0
50
100
150
200
250
0
50
100
150
200
250

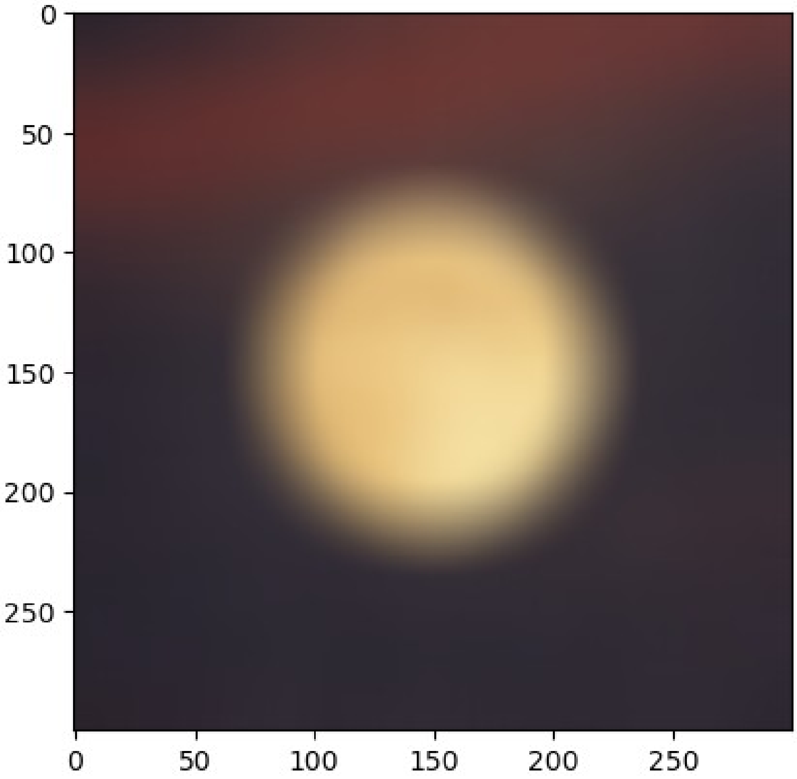

0
50
100
150
200
250
0
50
100
150
200
250

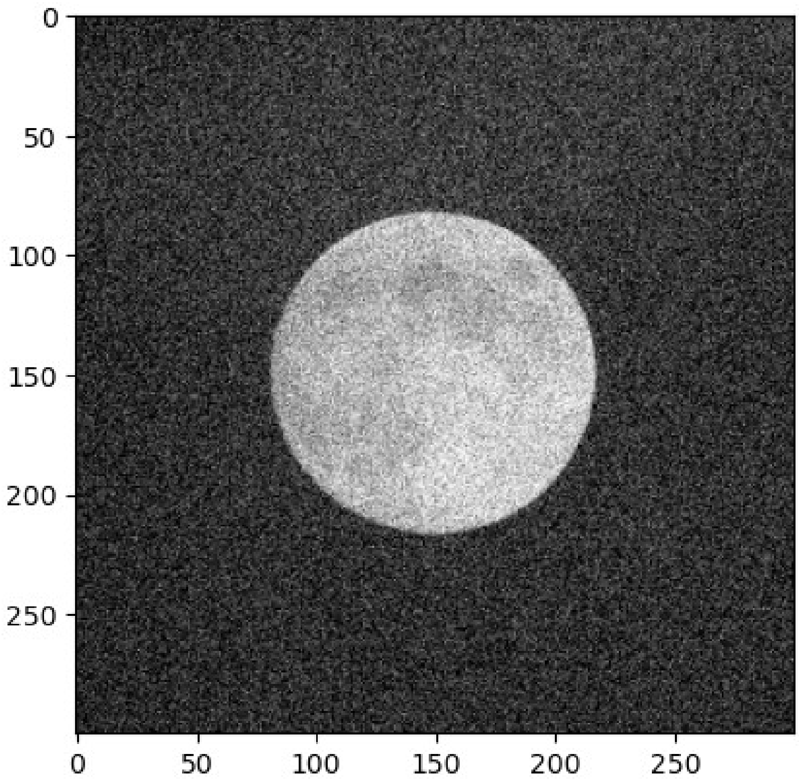

0
50
100
150
200
250
0
50
100
150
200
250


0
50
100
150
200
250
0
50
100
150
200
250

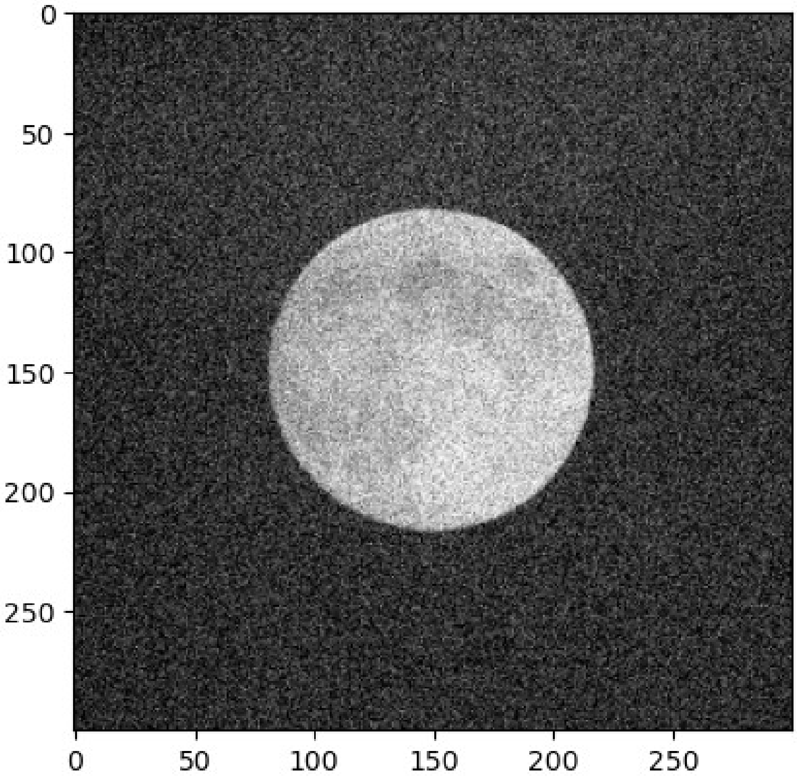

0
50
100
150
200
250
0
50
100
150
200
250


0
50
100
150
200
250
0
50
100
150
200
250

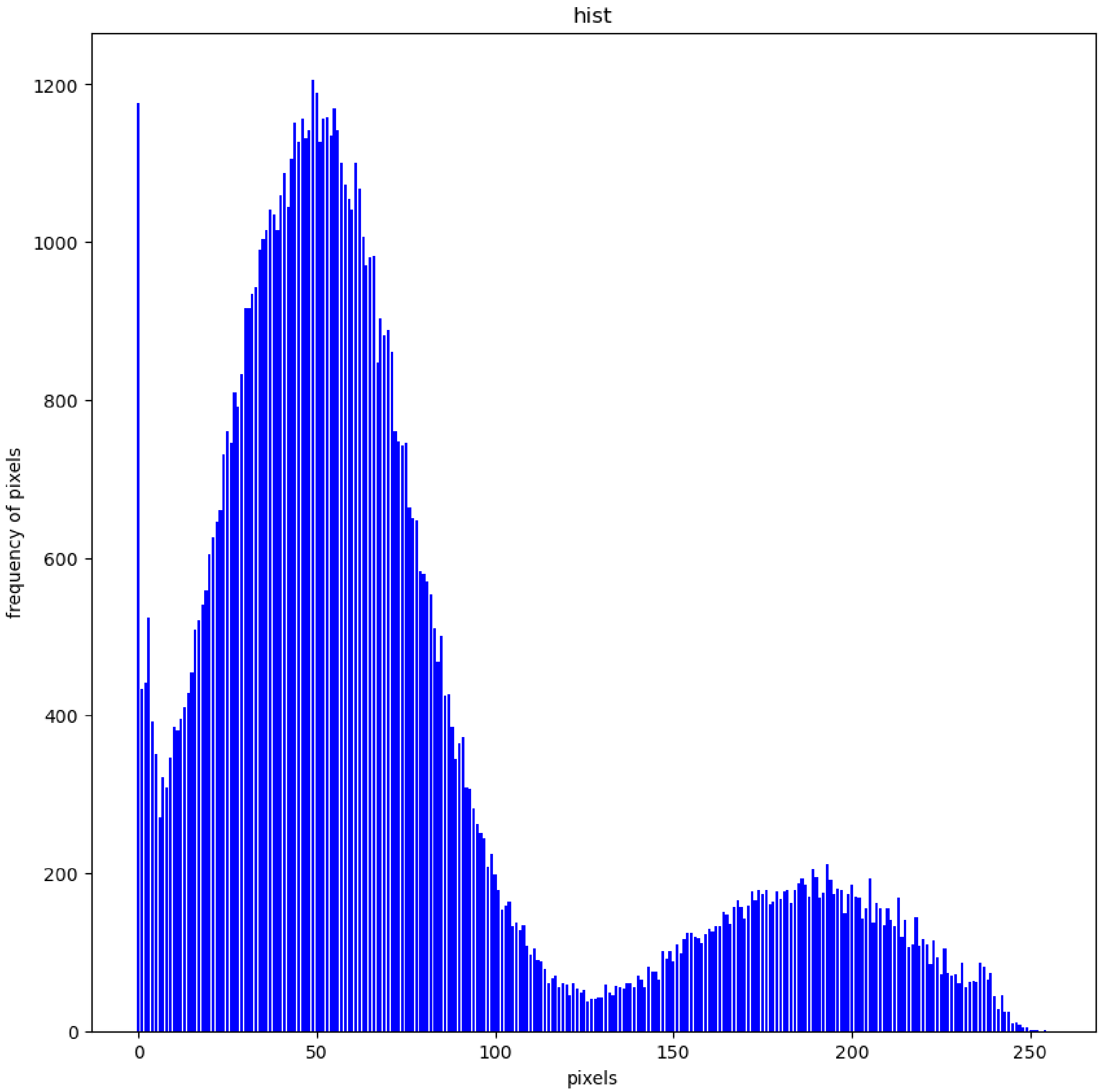

0
50
100
150
200
250
0
50
100
150
200
250



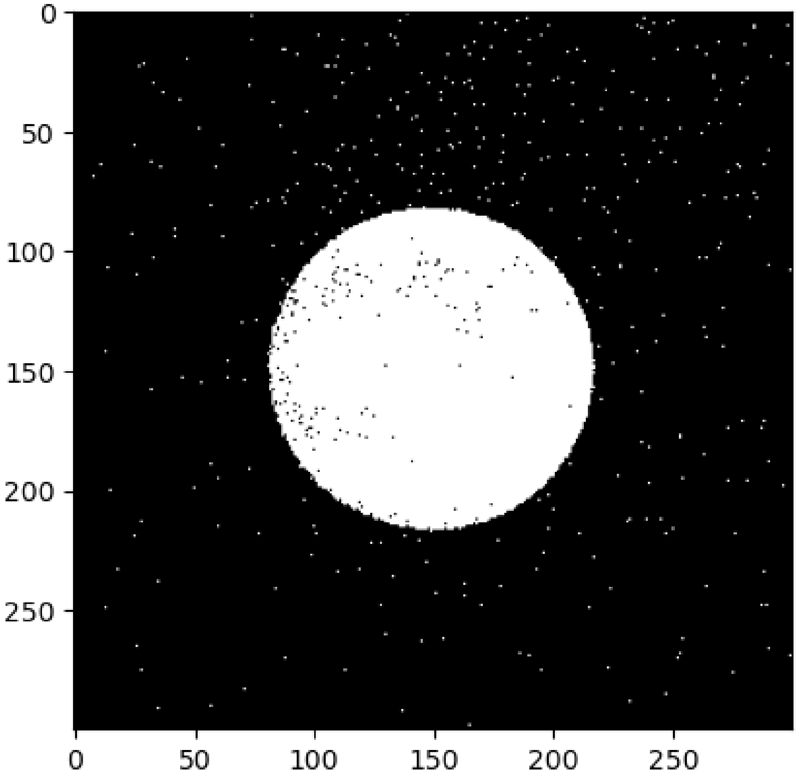
his, grayscaleimg , binarized img



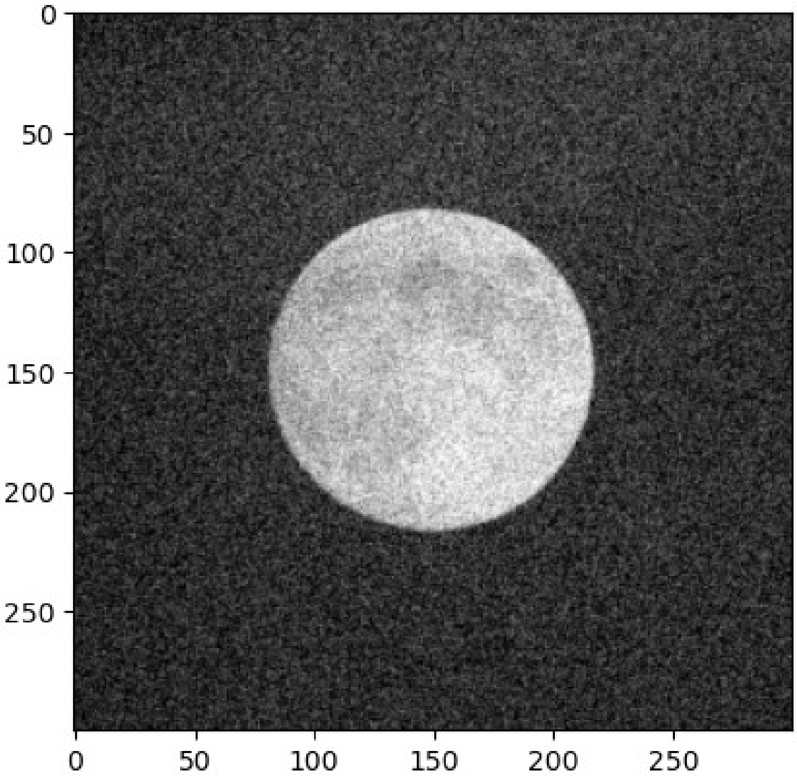
```
None
117 670.8329451085989
```


hist
frequency of pixels
1200
1000
800
600
400
200
0
0
50
100
150
200
250
pixels

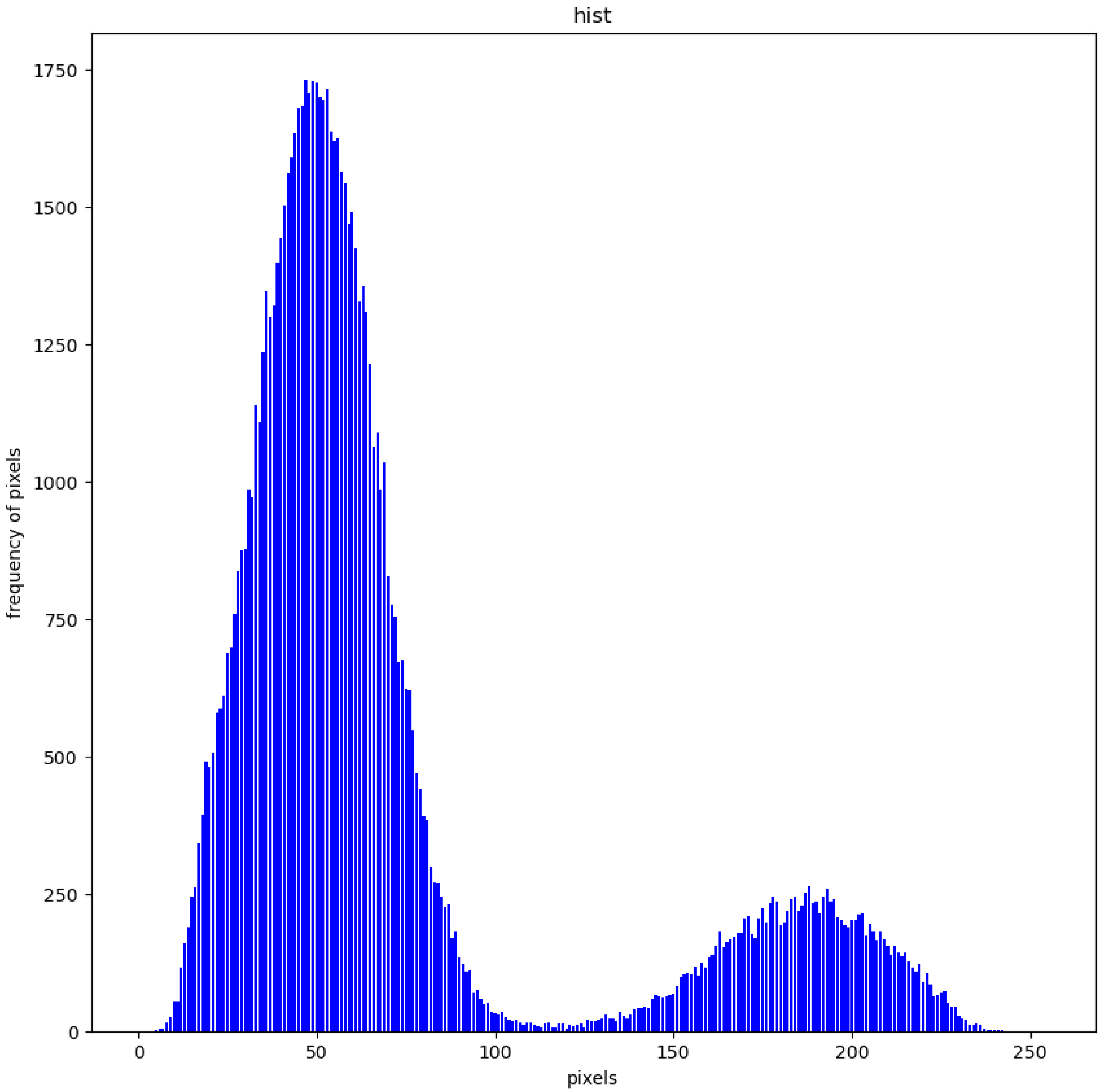
```
<matplotlib.image.AxesImage at 0x7fac2d9bd670>
```



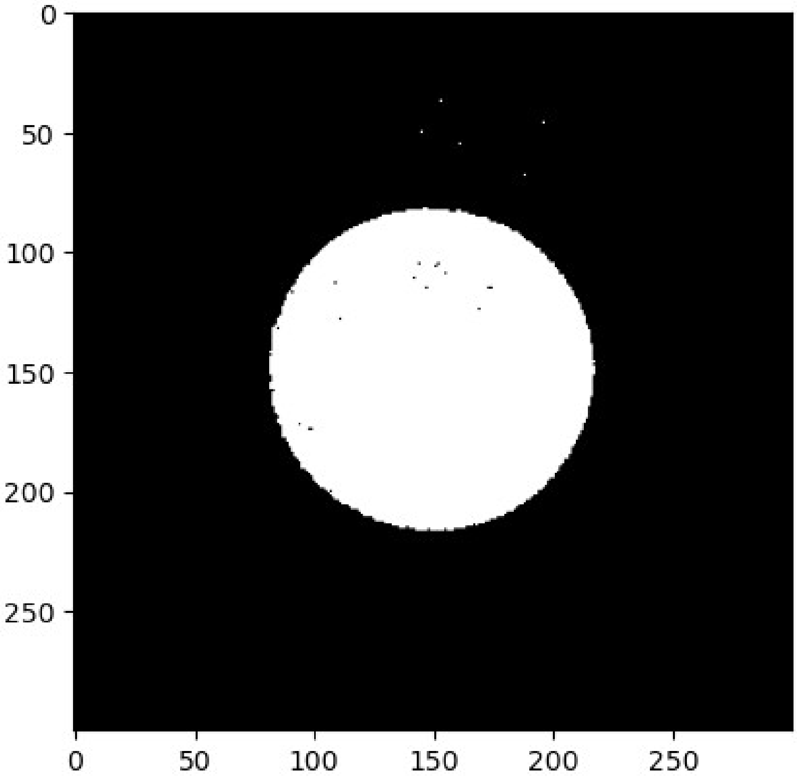
sigma = 0.5



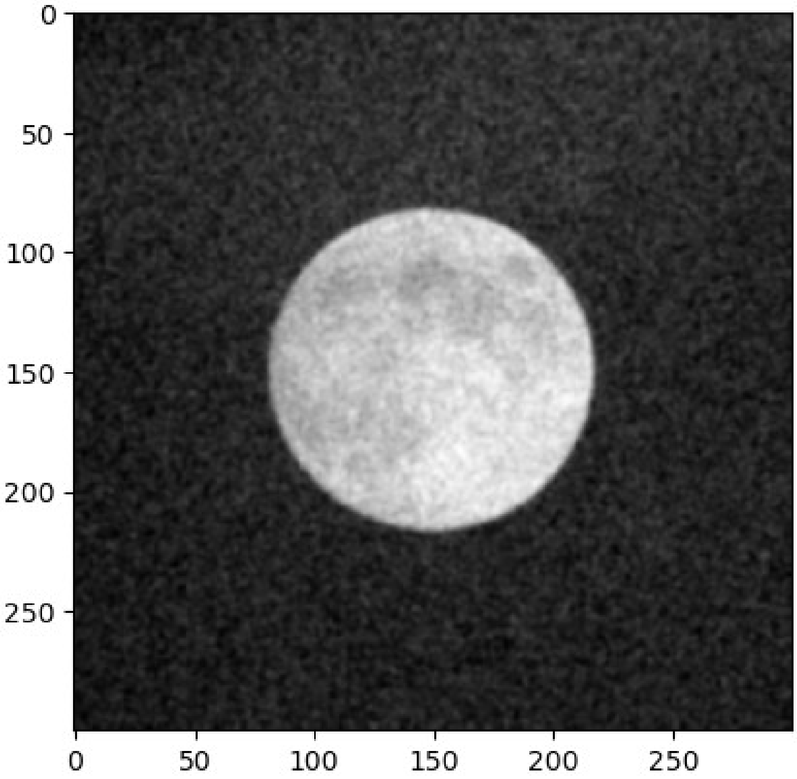
```
None
117 333.3382853965876
```


hist
frequency of pixels
1750
1500
1250
1000
750
500
250
0
0
50
100
150
200
250
pixels

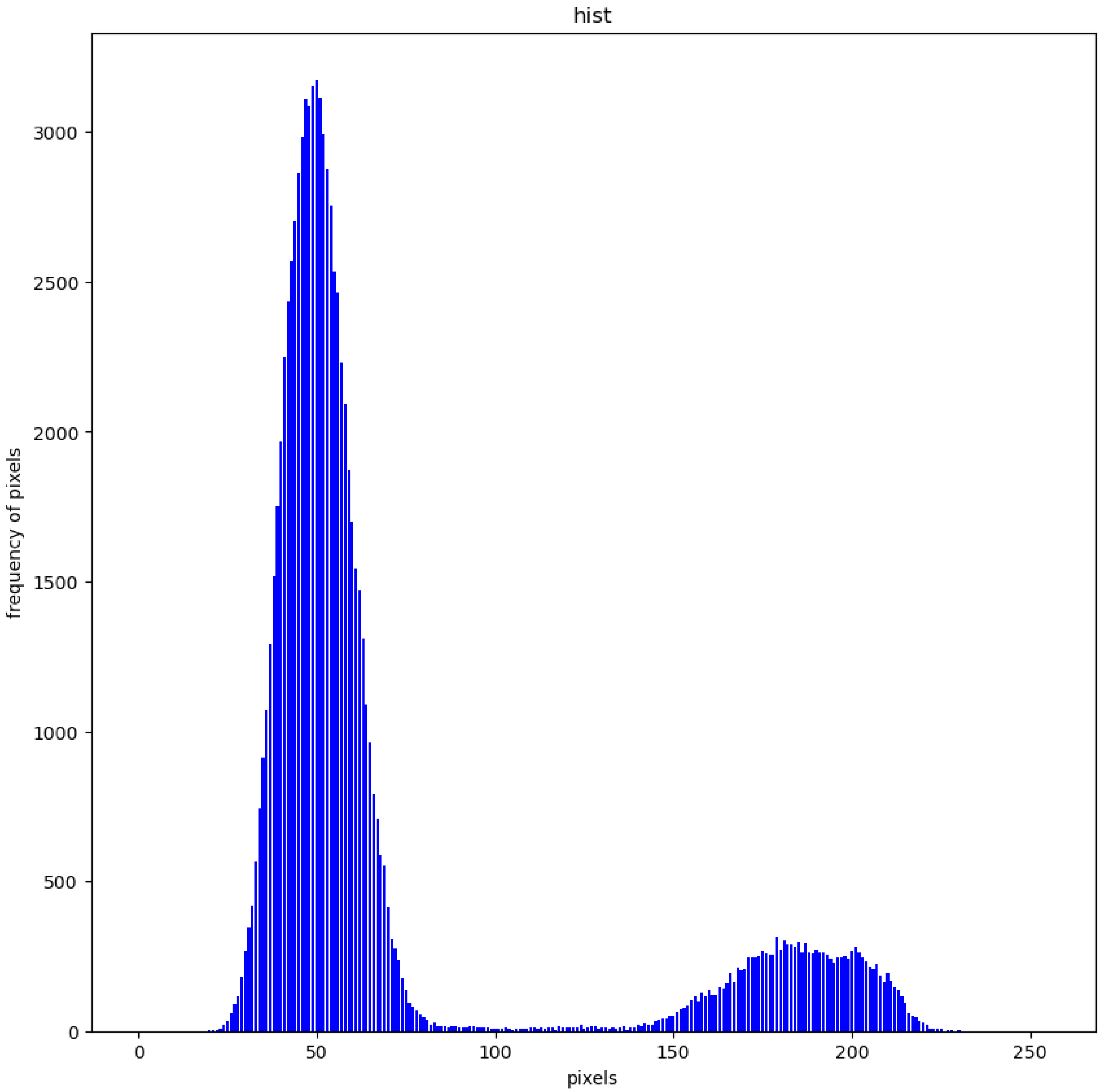
```
<matplotlib.image.AxesImage at 0x7fac2e321340>
```



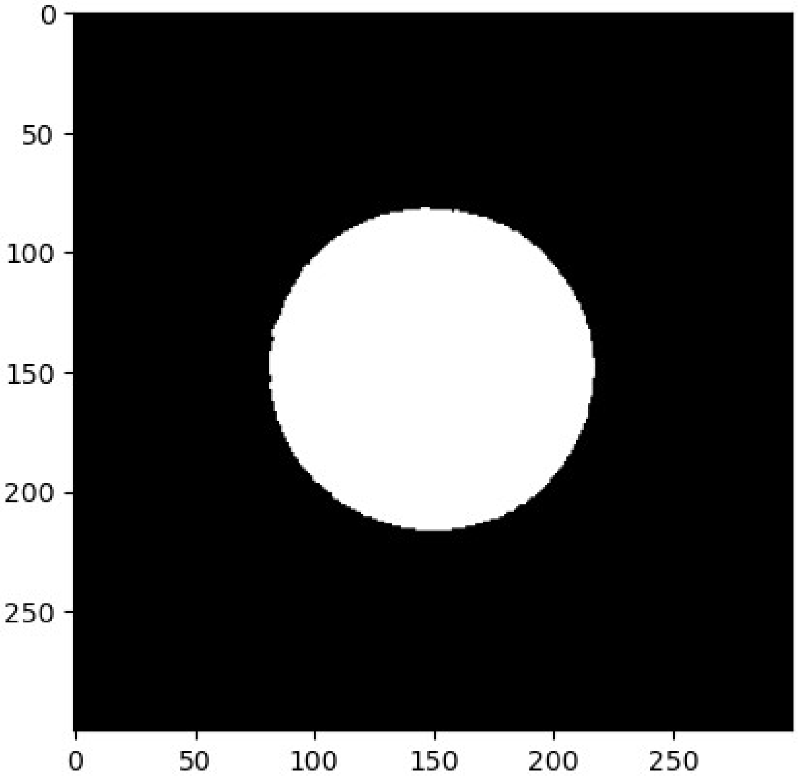
simga = 1



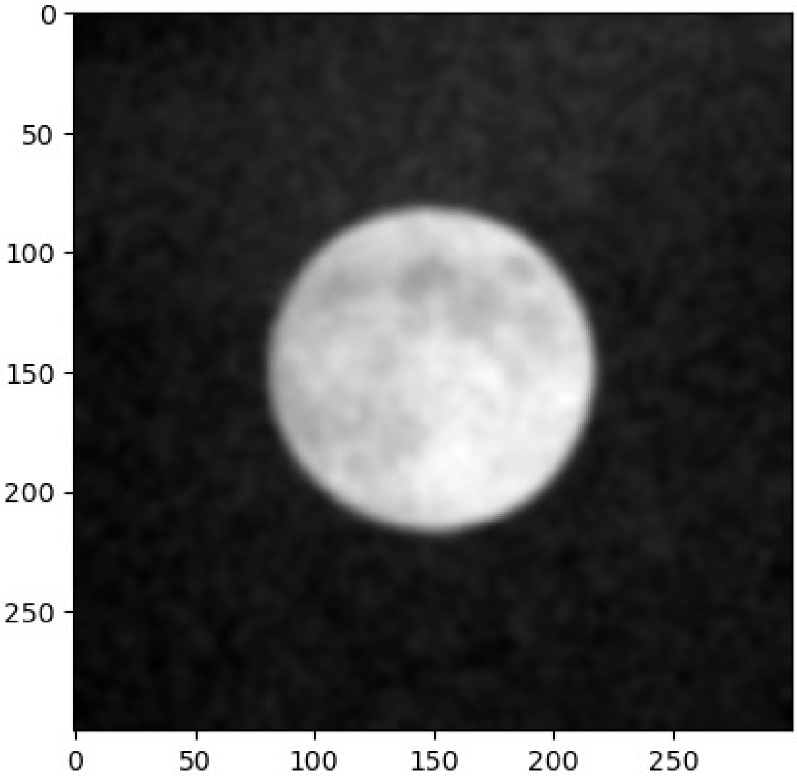
```
None
117 141.36212216598216
```


hist
frequency of pixels
0
500
1000
1500
2000
2500
3000
0
50
100
150
200
250
pixels

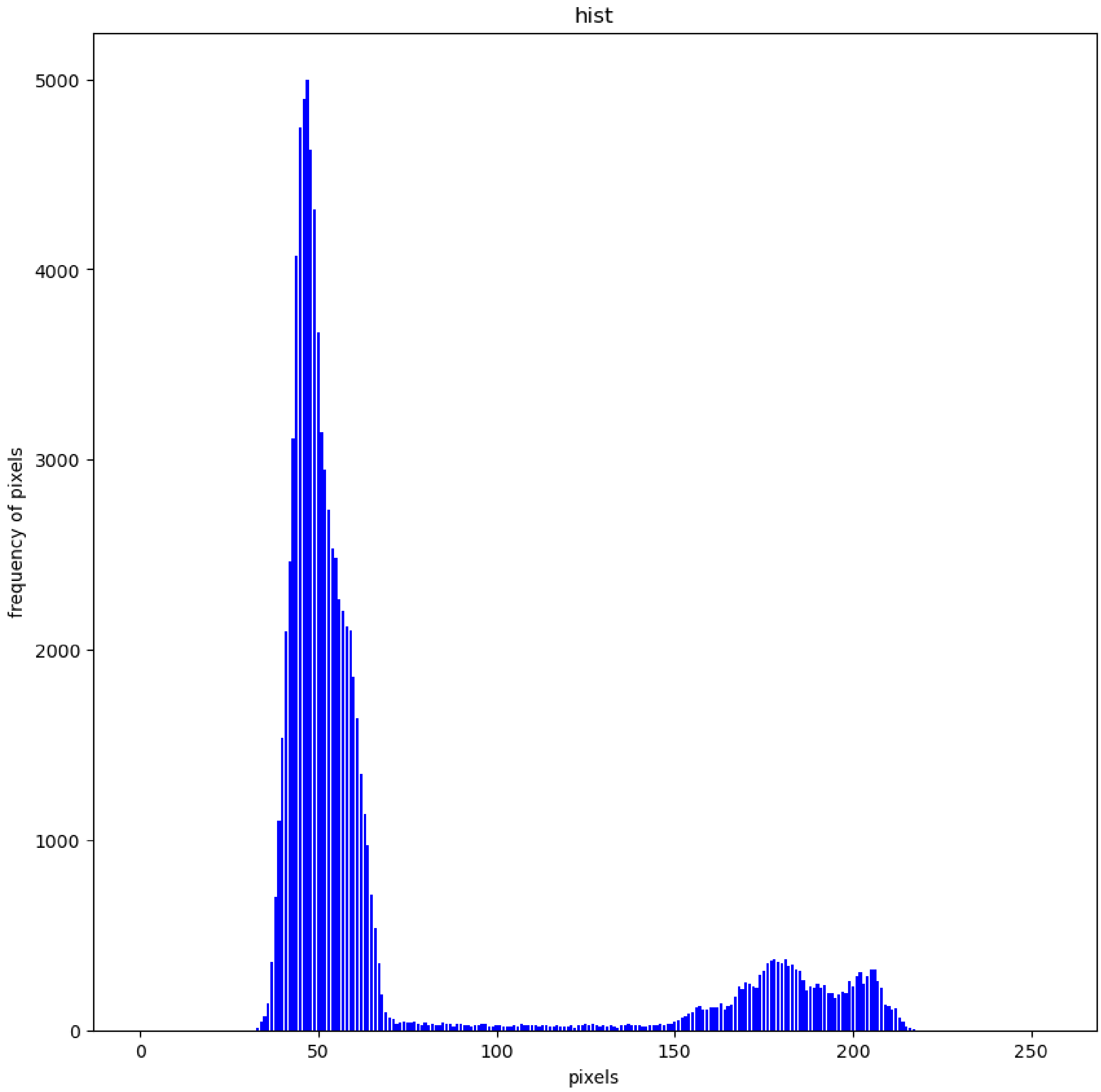
```
<matplotlib.image.AxesImage at 0x7fac2dafe4c0>
```



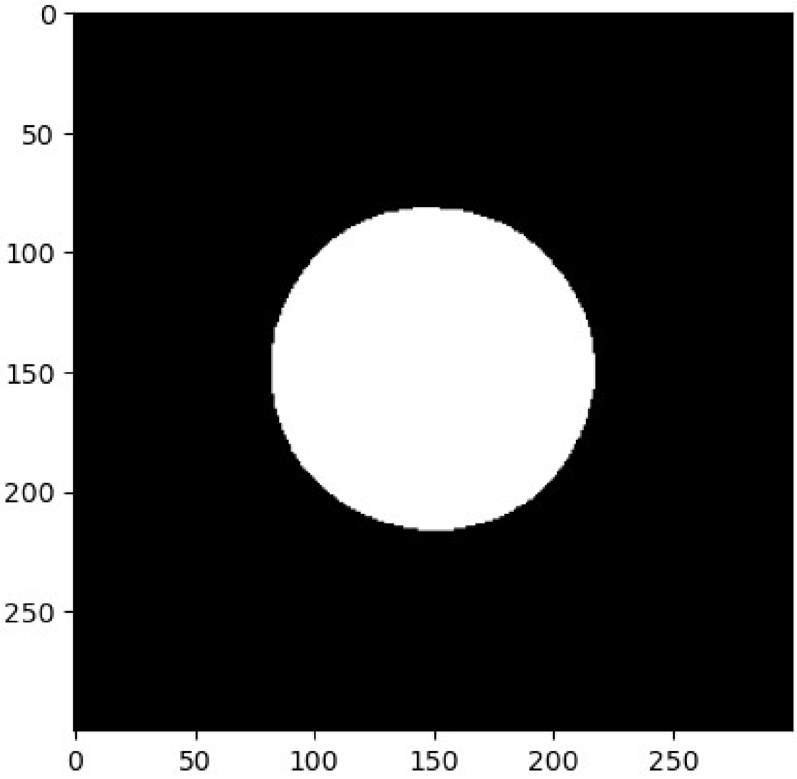
sigma = 2.5



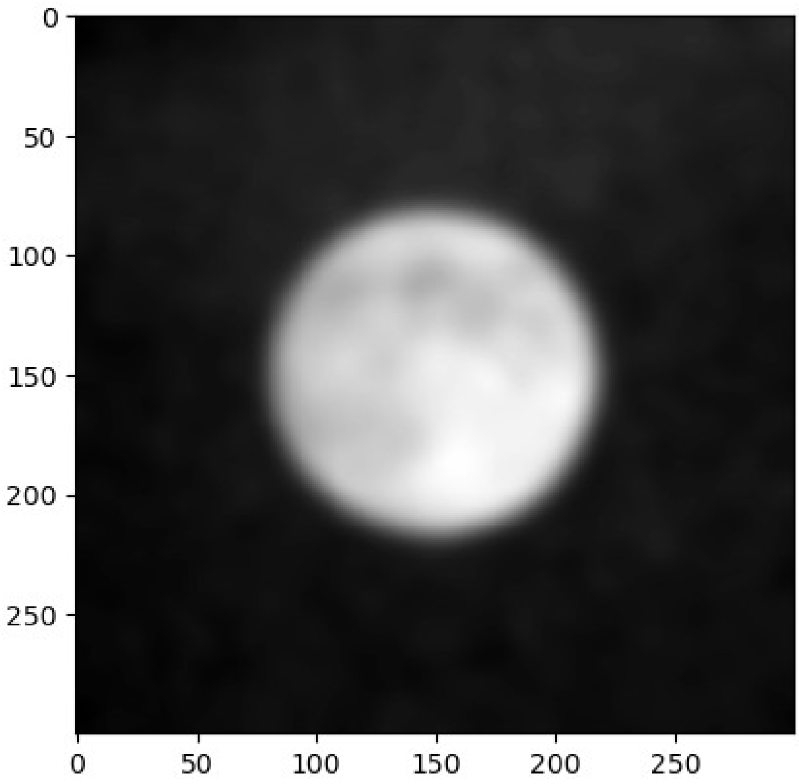
```
None
116 122.99550838650401
```


hist
frequency of pixels
5000
4000
3000
2000
1000
0
0
50
100
150
200
250
pixels

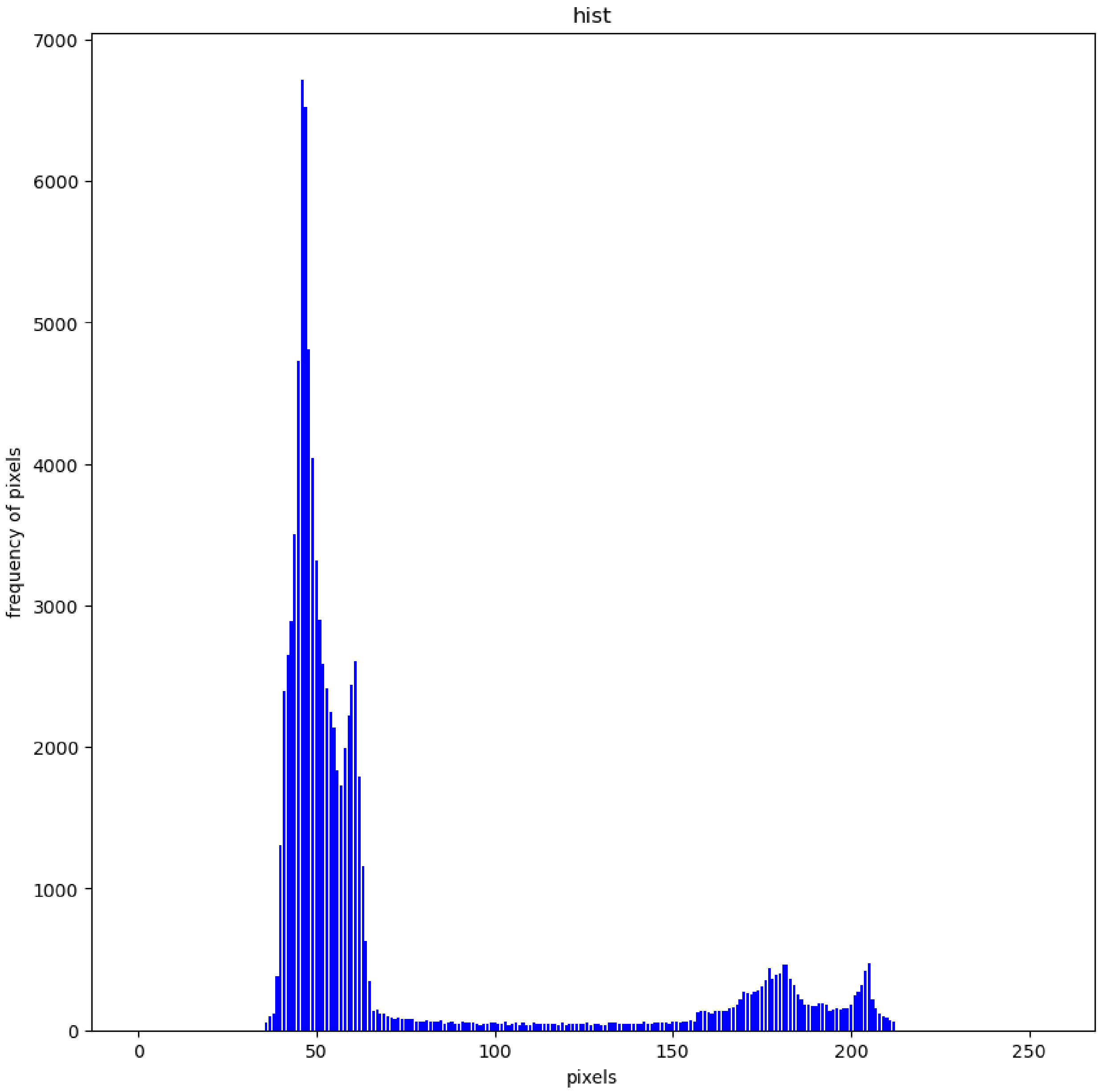
```
<matplotlib.image.AxesImage at 0x7fac2e488730>
```



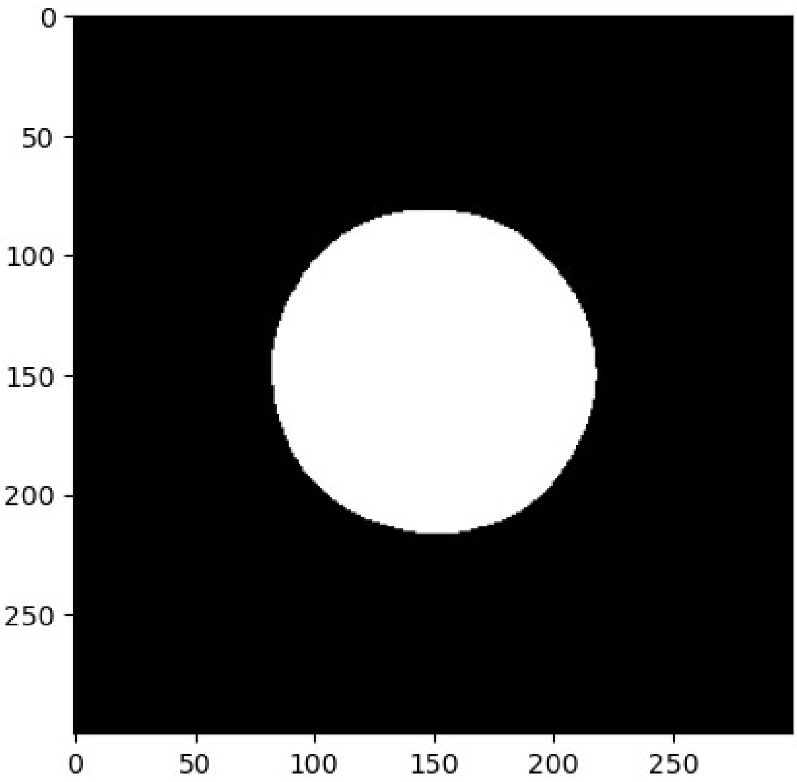
sigma = 5



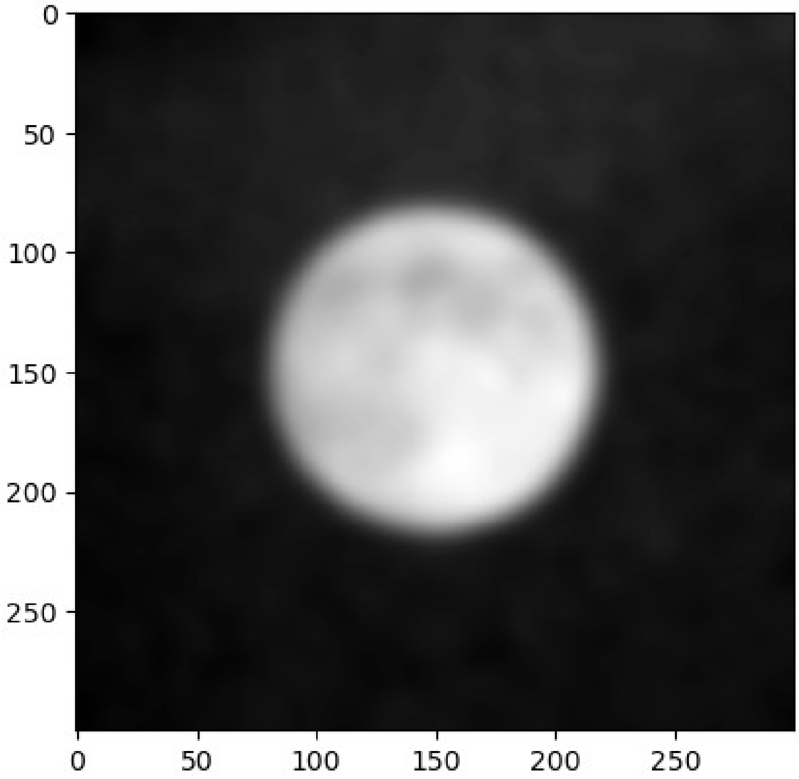
```
None
114 153.64105068928217
```


hist
frequency of pixels
7000
6000
5000
4000
3000
2000
1000
0
0
50
100
150
200
250
pixels

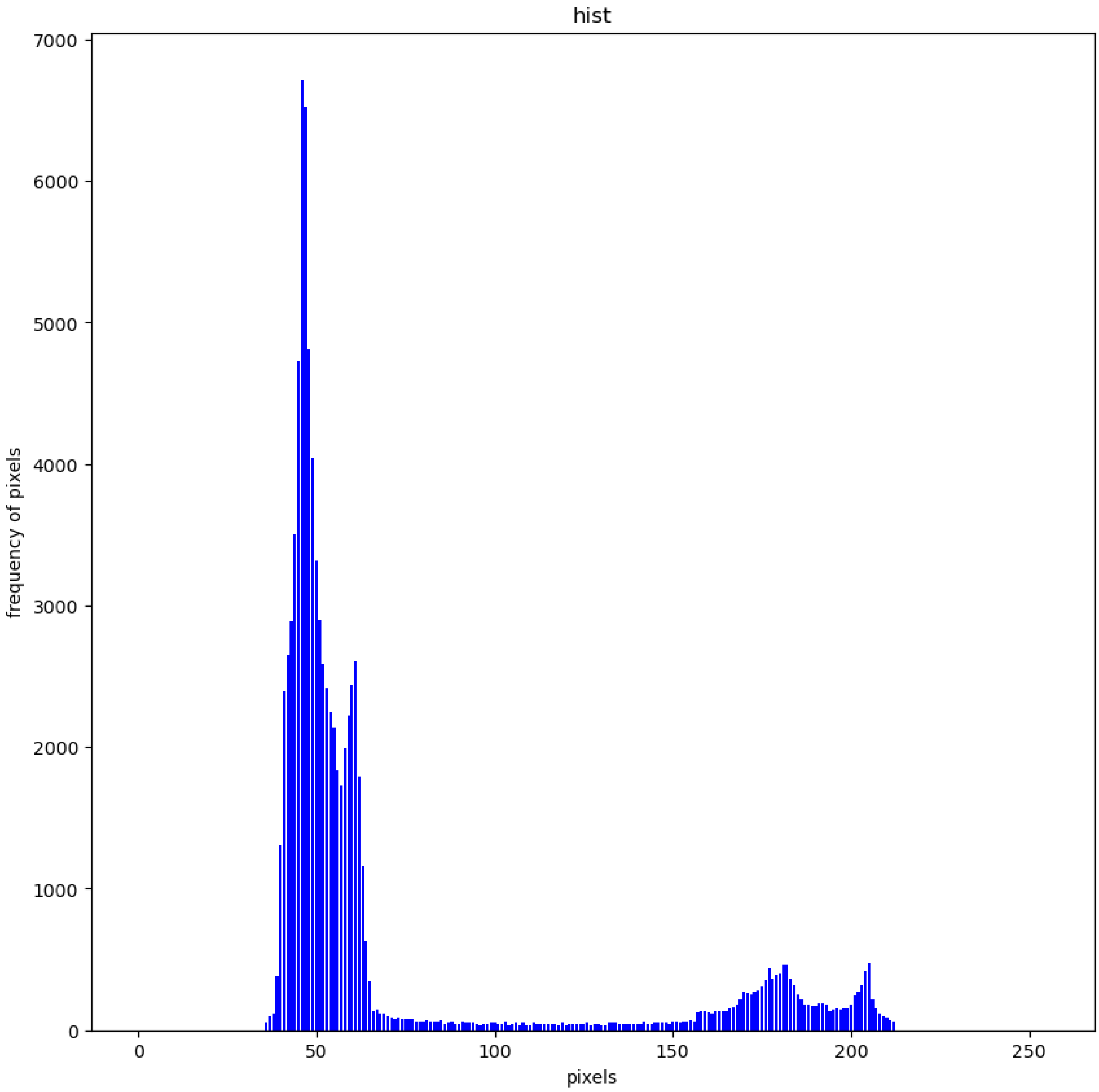
```
<matplotlib.image.AxesImage at 0x7fac2df9ab20>
```



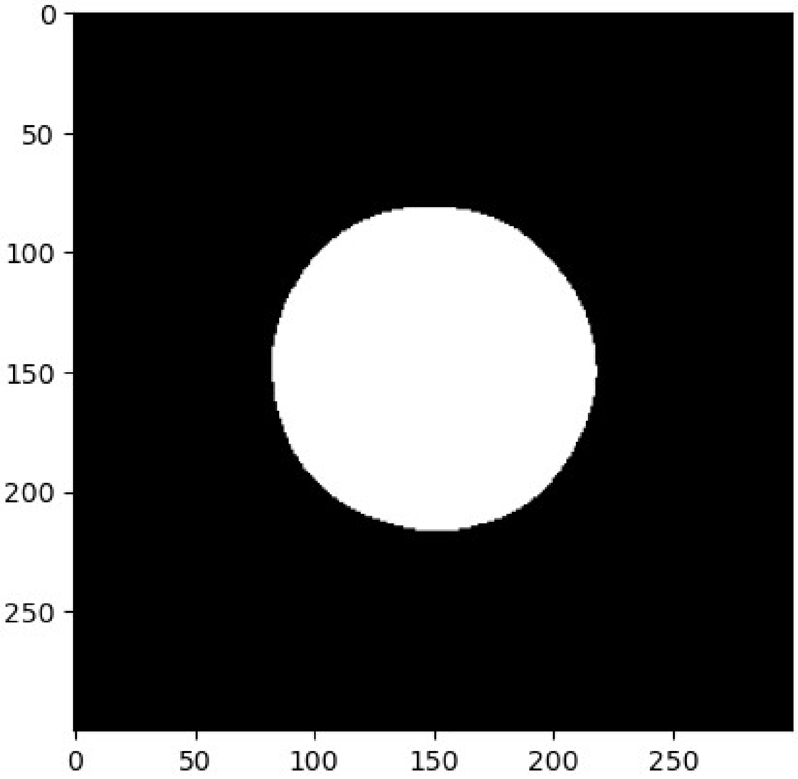
sigma = 10



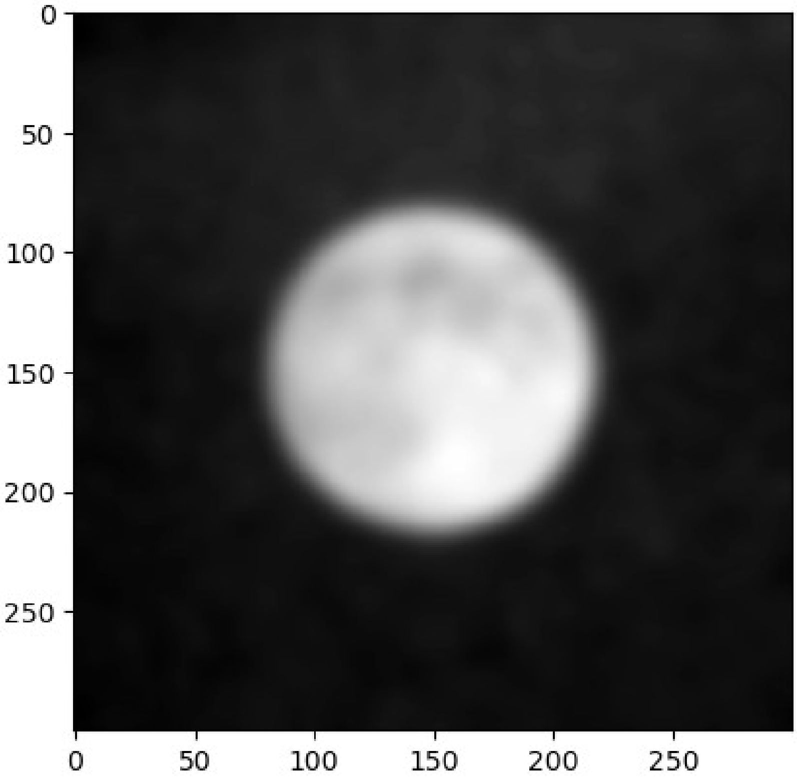
```
None
114 153.64105068928217
```


hist
frequency of pixels
7000
6000
5000
4000
3000
2000
1000
0
0
50
100
150
200
250
pixels

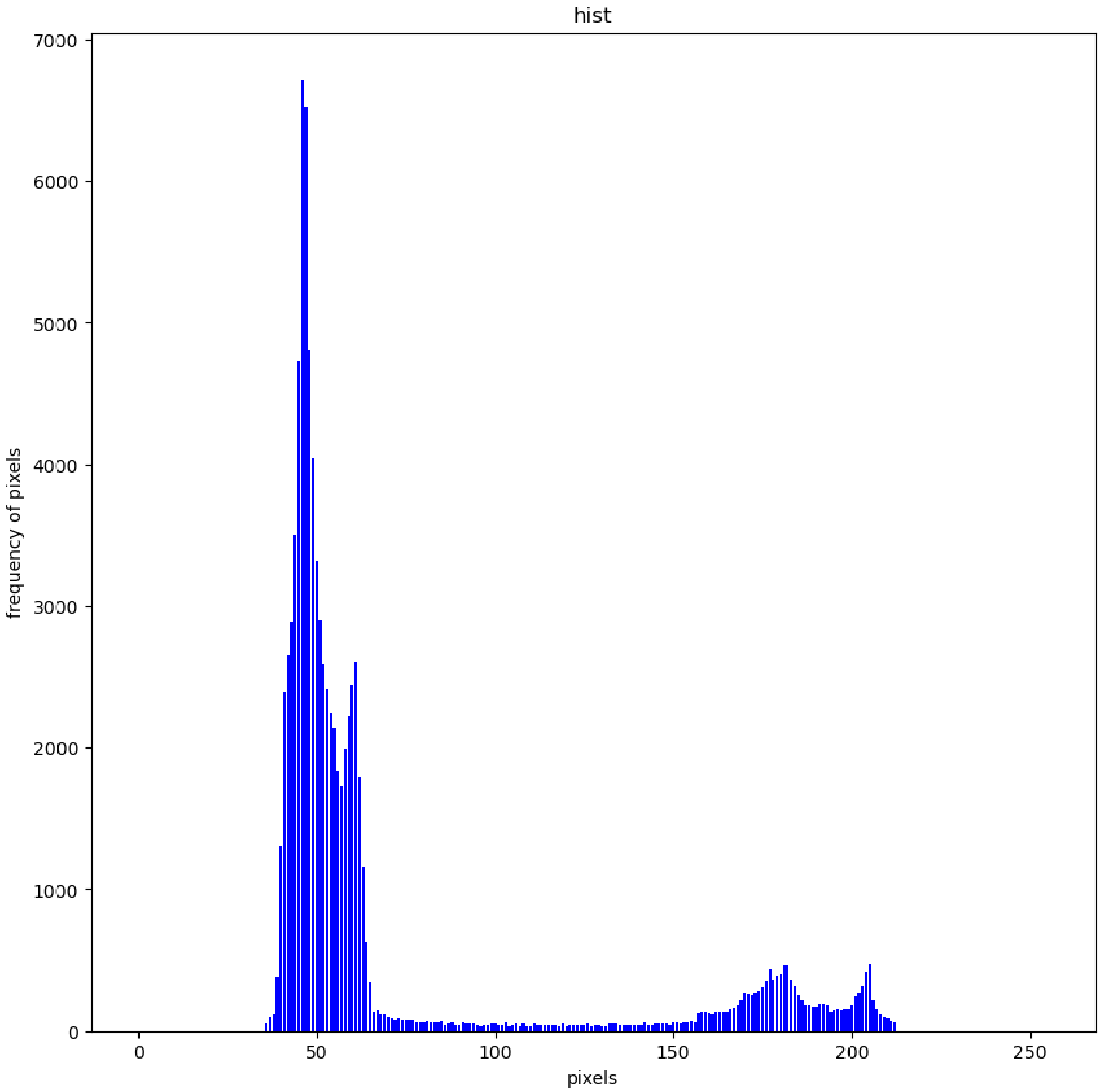
```
<matplotlib.image.AxesImage at 0x7fac2deeaf40>
```



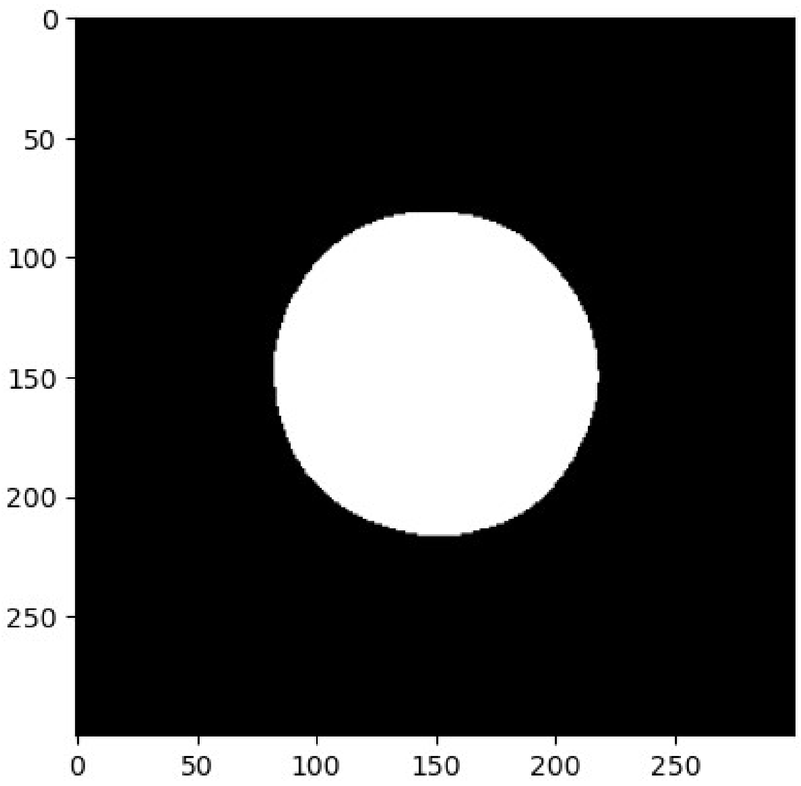
sigma = 20



```
None
114 153.64105068928217
```


hist
frequency of pixels
0
1000
2000
3000
4000
5000
6000
7000
pixels
0
50
100
150
200
250

```
<matplotlib.image.AxesImage at 0x7fac2dea9be0>
```



comment:

As we are increasing the sigma_g value the degree of blurness is also increasing. This result in smoothing of the image but it leads of loss of details in picture. white dots are getting disappeared in background and black region on the moon is getting blurred or white

For lower sigma_g values= 0.1, binarized image has more feature from original image (it has black dots on moon and white on background)but as bluring is increasing or sigma_g value is increasing filter assigns incorrect values to the region where there is high change in intensity values but the separation between the background and foreground is getting good.

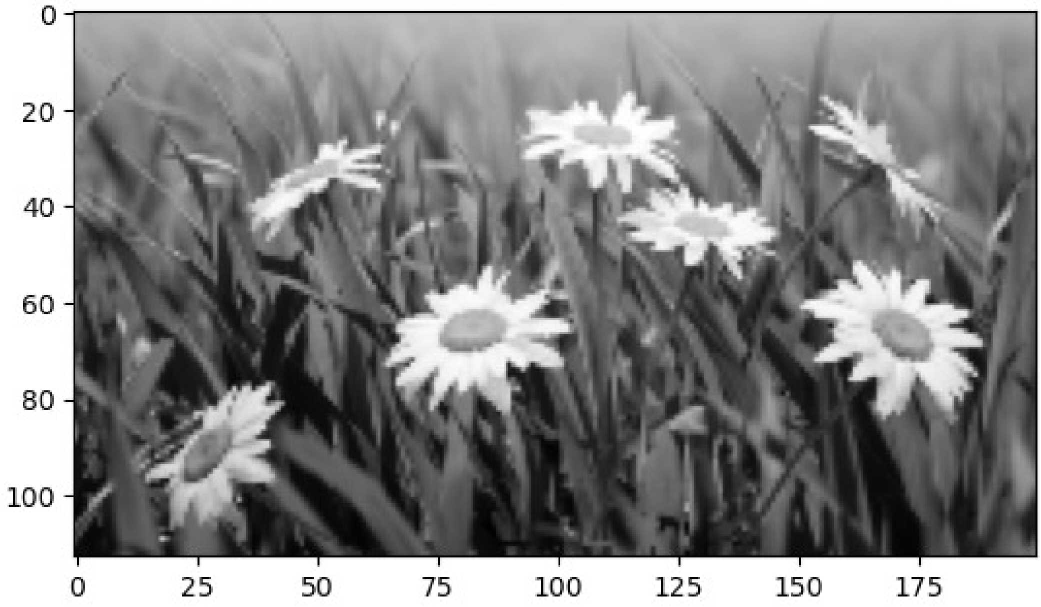
optimal within class variance for different sigma_g decreases upto a certain point then starts increasing again after sigma_g > 2.5. It decreases due to improved separation of foreground and background image in the grayscale image . After 122.9 it starts increasing again because image is getting more blurred. But image doesnt change much because threshold value is not changing.

At optimal sigma_g= 2.5 we have best case. dots in background of image are there but they appear very blurred as it reatained features of original image.

Question 2

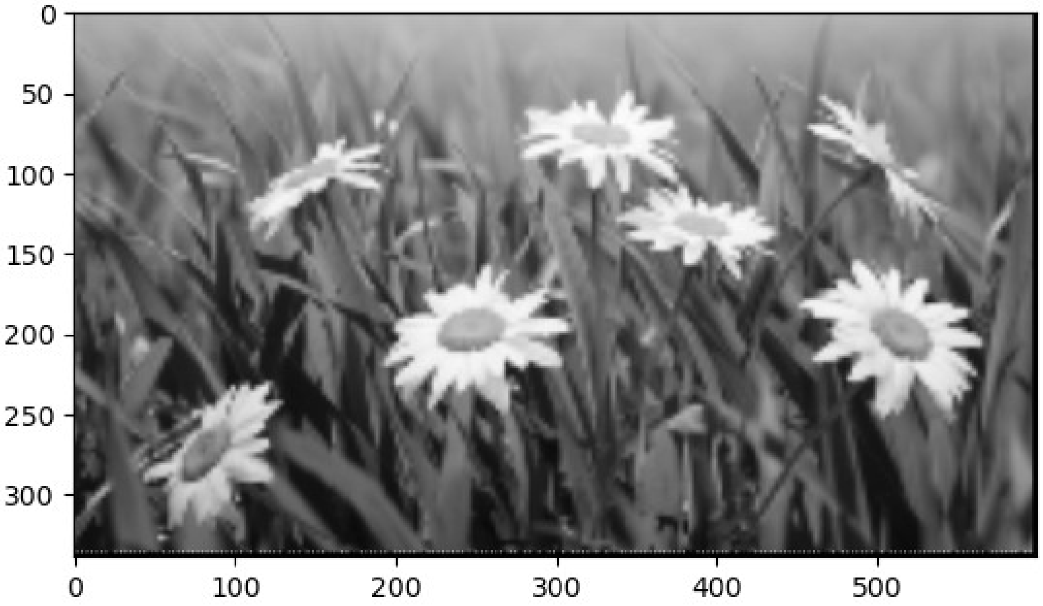
downsampling by factor of 2

```
(113, 200)
```



```
array([[179., 177., 175., ..., 174., 174., 173.],
       [177., 175., 172., ..., 171., 172., 171.],
       [176., 172., 170., ..., 169., 169., 169.],
       ...,
       [ 49.,  21.,  15., ...,  78.,  63.,  89.],
       [ 19.,  20.,  20., ...,  96.,  90., 106.],
       [ 19.,  19.,  21., ..., 110., 103., 117.]])
```

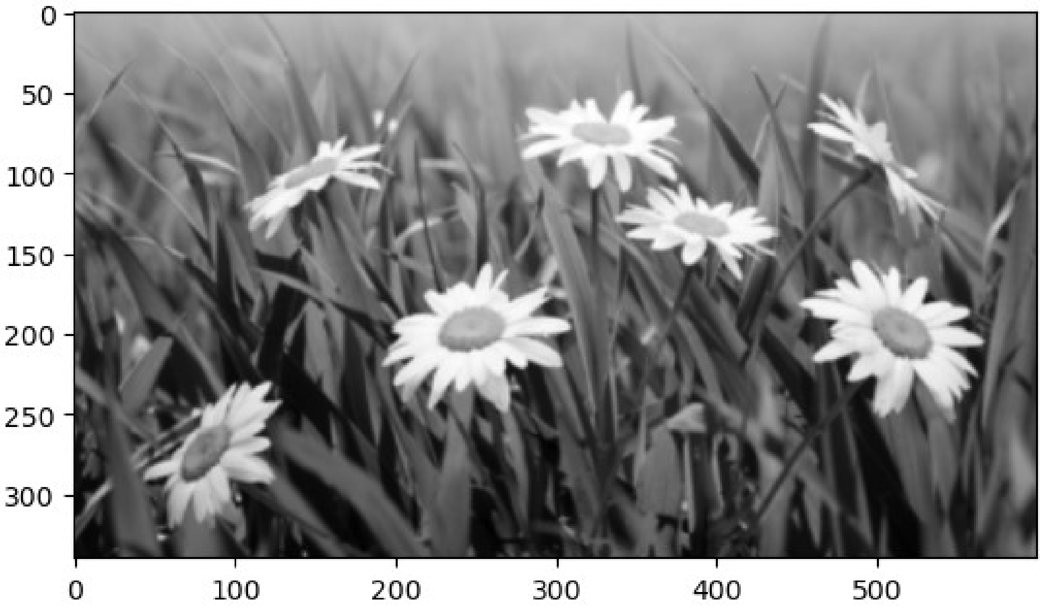
upsampling the downsampled image by 3



```
(339, 600)
```

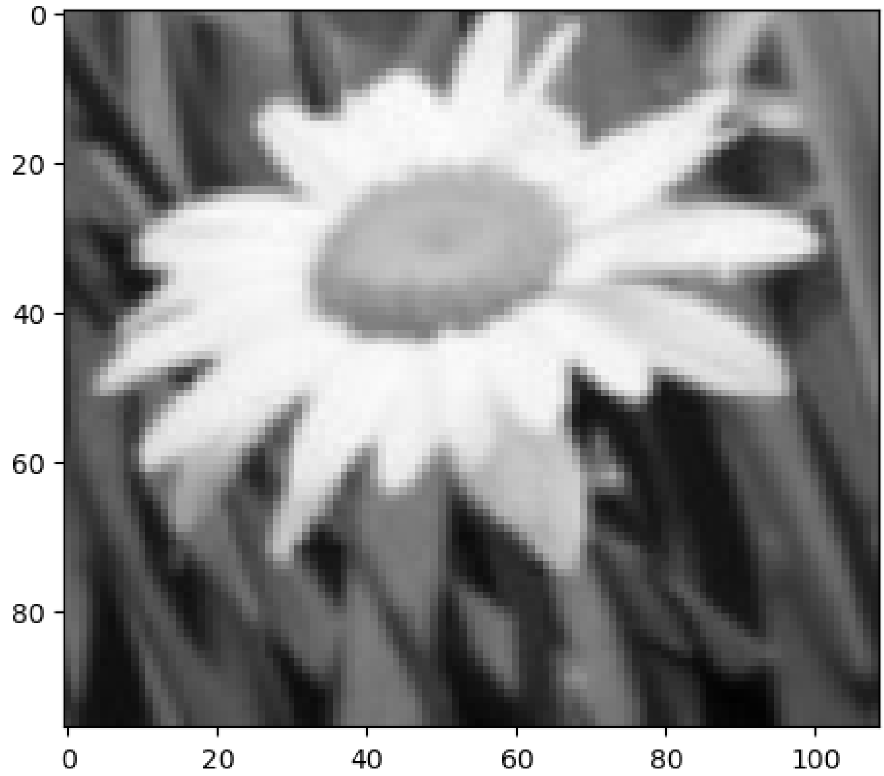
part b

```
(340, 600)
```



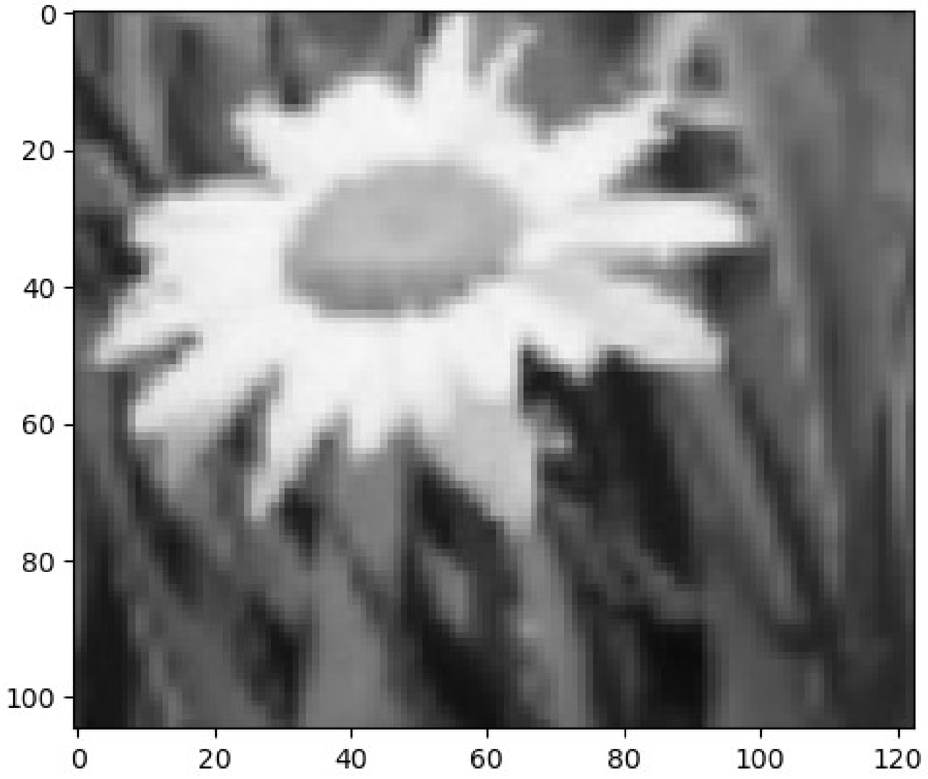
```
## image of upsampled by 3 of downsampled by 2
```

```
<matplotlib.image.AxesImage at 0x7fab2881fbb0>
```



```
# image of upsampled by 1.5
```

```
<matplotlib.image.AxesImage at 0x7fab268aaa60>
```



Comment:

downsampling :The quality diminishes as some pixel information is lost during the downsampling process.

Upsampling by 3: The image downsampled by a factor of 2 and then upsampled by a factor of 3, exhibits noticeable artifacts and a loss of detail, resulting in a blurred appearance.

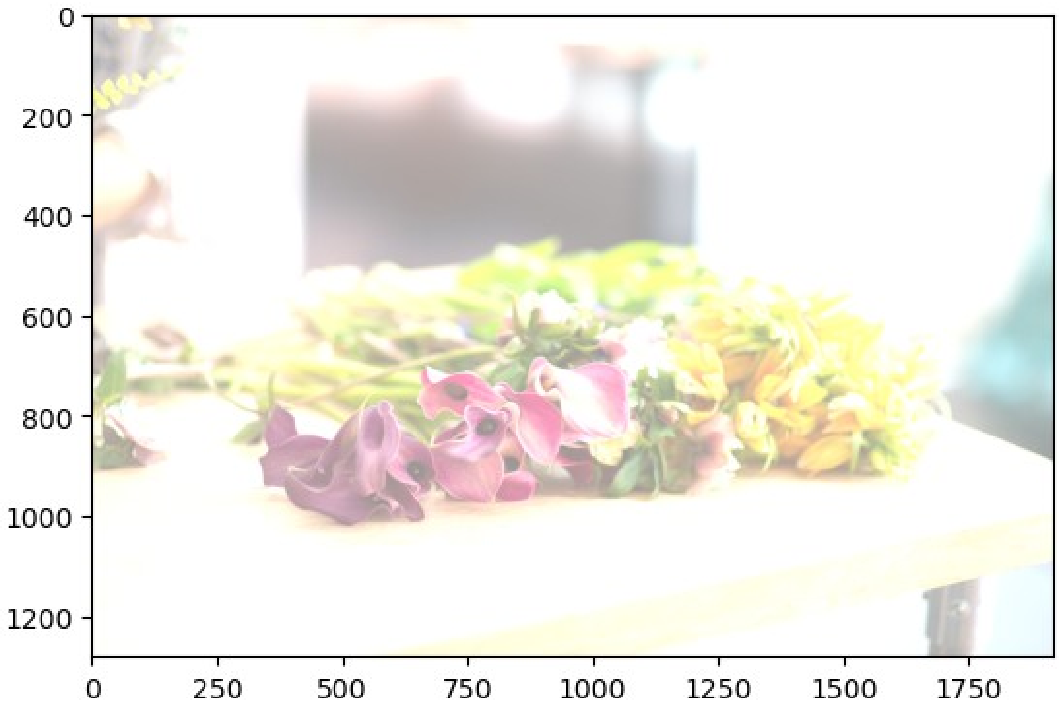
upsampled by 1.5: The image upsampled by a factor of 1.5 retains more detail and produces a smoother, more coherent output.

Question 3

Brightness

```
#brightness
from skimage import img_as_float
def brightness(imag,p):
    norm_img = img_as_float(imag)
    n = norm_img +(2*p-1)
    nn = np.clip(n,0,1)
    return plt.imshow(nn)
brightness(q3flowers_array,0.8)
```

```
<matplotlib.image.AxesImage at 0x7fab6a70da00>
```

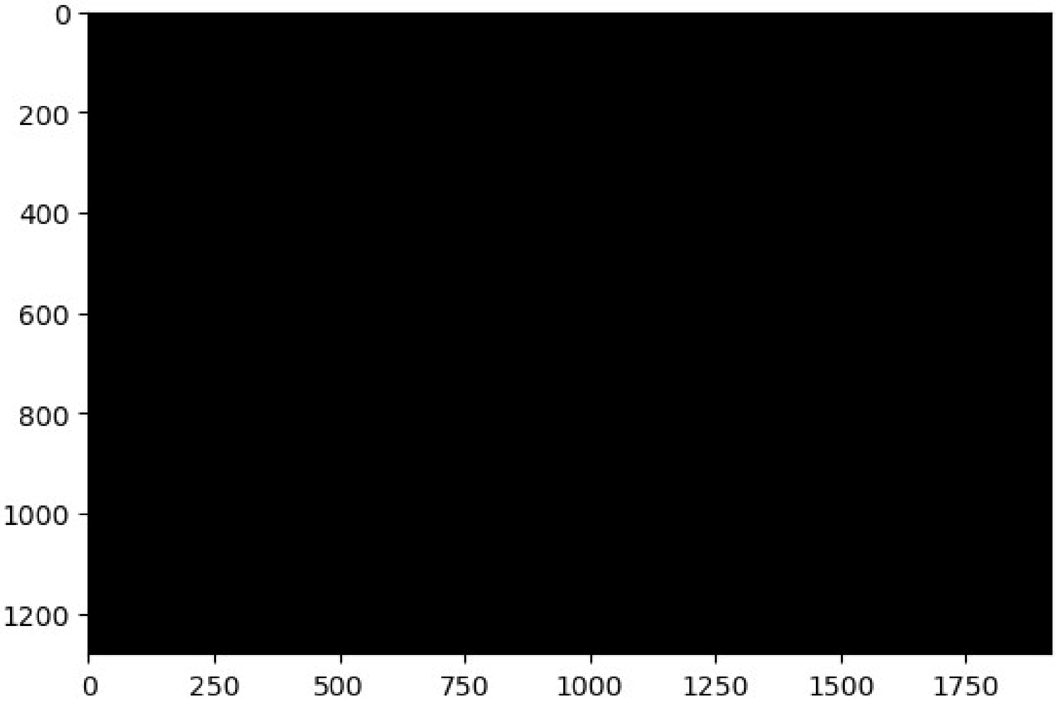


```
brightness(q3flowers_array,0.5)
```

```
<matplotlib.image.AxesImage at 0x7fab4da55640>
```

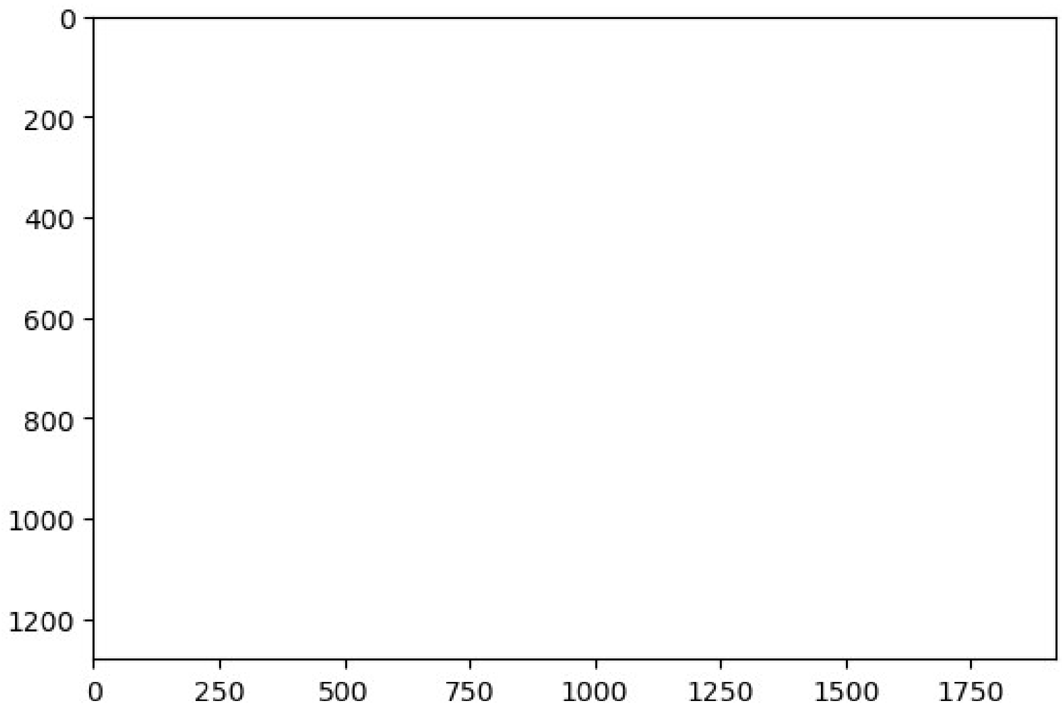
```
brightness(q3flowers_array,0)
```

```
<matplotlib.image.AxesImage at 0x7fab3ff3c6a0>
```



```
brightness(q3flowers_array,1)
```

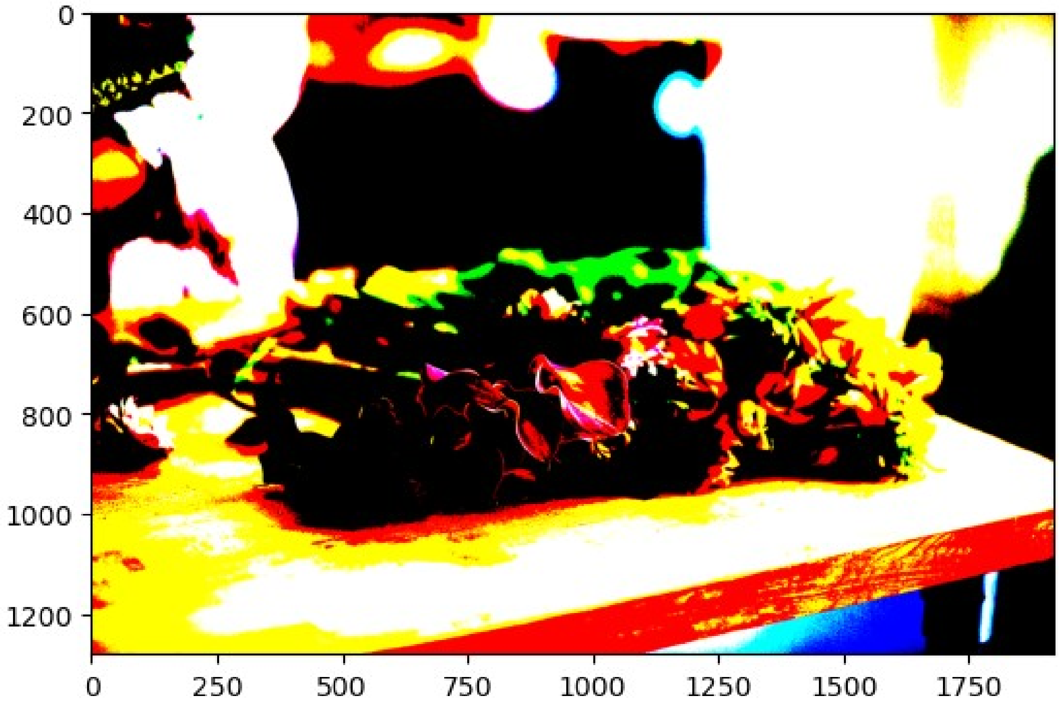
```
<matplotlib.image.AxesImage at 0x7fab39b7b820>
```



Contrast

```
ps = q3flowers_array
b, g, r = cv2.split(q3flowers_array)
b_zero = np.zeros(b.shape)
g_zero = np.zeros(g.shape)
r_zero = np.zeros(r.shape)
image_zeros = cv2.merge([b_zero, g_zero, r_zero])
b_ones = np.ones(b.shape)
g_ones = np.ones(g.shape)
r_ones = np.ones(r.shape)
image_ones = cv2.merge([b_ones, g_ones, r_ones])

def contrast_adjust(image, p):
    img_normalized = img_as_float(image)

    if p == 0.5:
        #original image
        return image

    elif p == 0:
        #grey_image
        ps = image_ones * 0.5
    elif p == 1:
        ps = np.where(img_normalized > 0.5, 1, 0)

    else:

        ps = (1 / (1 - p)) * img_normalized + 0.5
        ps = np.clip(ps, 0, 1)
    ps = (ps * 255).astype(np.uint8)

    return ps
cont = contrast_adjust(q3flowers_array, 1)
plt.imshow(cont)
```

```
<matplotlib.image.AxesImage at 0x7fab25c1d460>
```

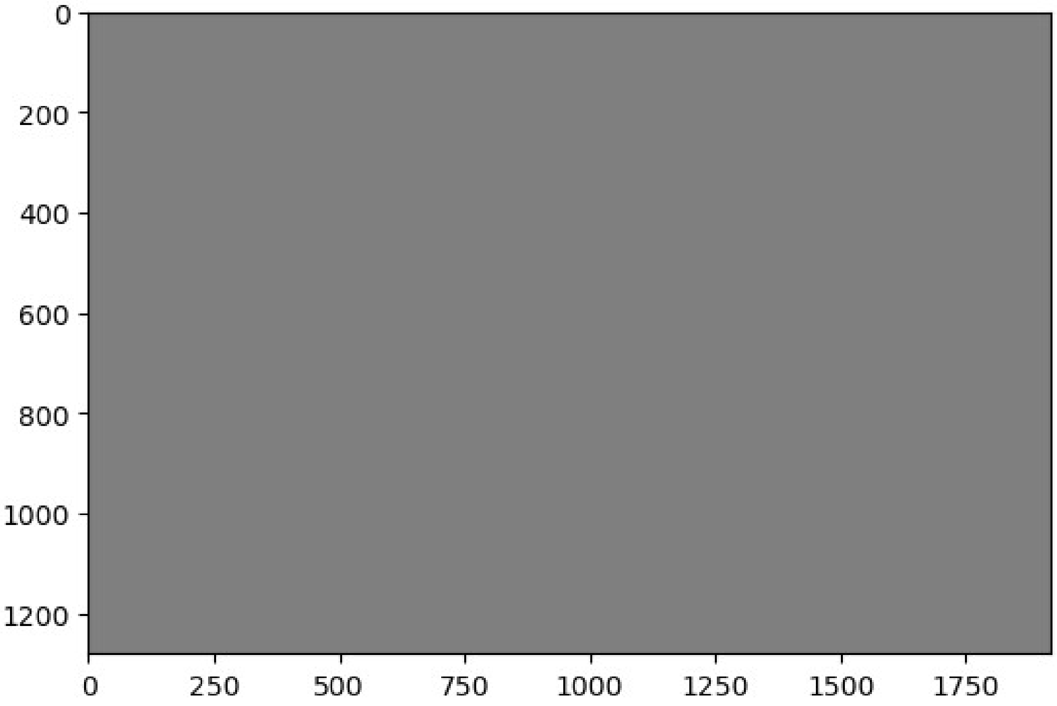


```
plt.imshow(contrast_adjust(q3flowers_array,0.5))
```

```
<matplotlib.image.AxesImage at 0x7fab5ad3e070>
```

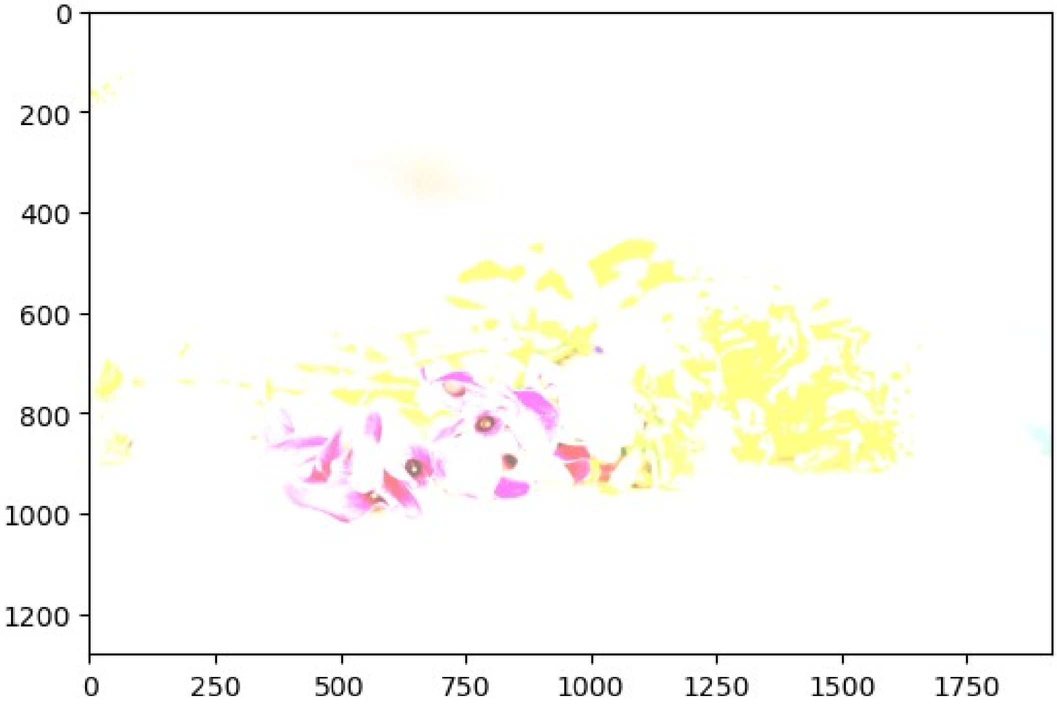
```
plt.imshow(contrast_adjust(q3flowers_array,0))
```

```
<matplotlib.image.AxesImage at 0x7fab59a7b1f0>
```



```
plt.imshow(contrast_adjust(q3flowers_array,0.9))
```

```
<matplotlib.image.AxesImage at 0x7fab25f387f0>
```



Comment:

In brightness we just added a offset in the image of intensity of every pixel in the bgr is getting higher and and higher and then we clip it to (0 to 1) and when we decrease the itensity some intensity values gets negative but its again get clipped from (0 to 1)

In contrast we mutiplied a factor p to the image as p get higher the contrast of the image increase as the difference between the instensity value of low and high gets more and more.