```
from google.colab import drive
drive.mount('/content/drive')
```

```
import cv2 # Import opencv library
```

Calculating and plotting histogram

```
img = cv2.imread("/content/drive/MyDrive/FIPLab/Dataset/img11.png")
```

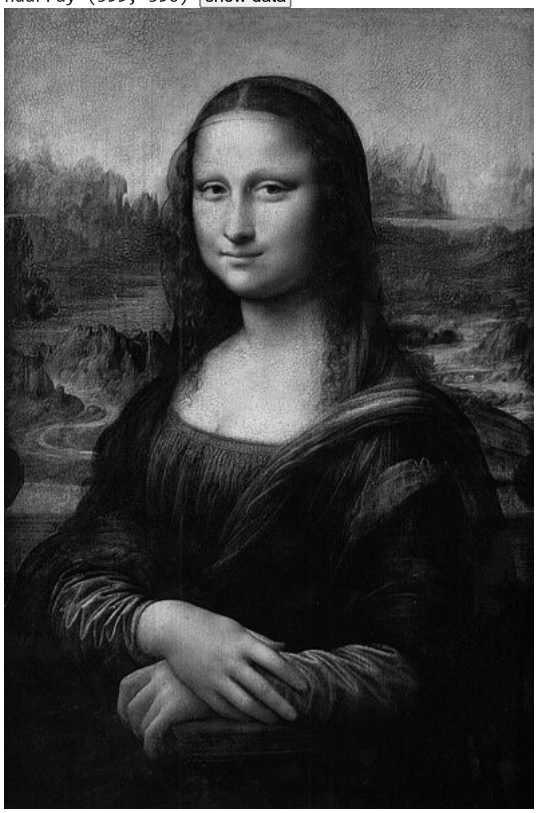
```
img.shape
```

```
(599, 396, 3)
```

```
# Coverting BGR image into gray scale image
gray_img = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
```

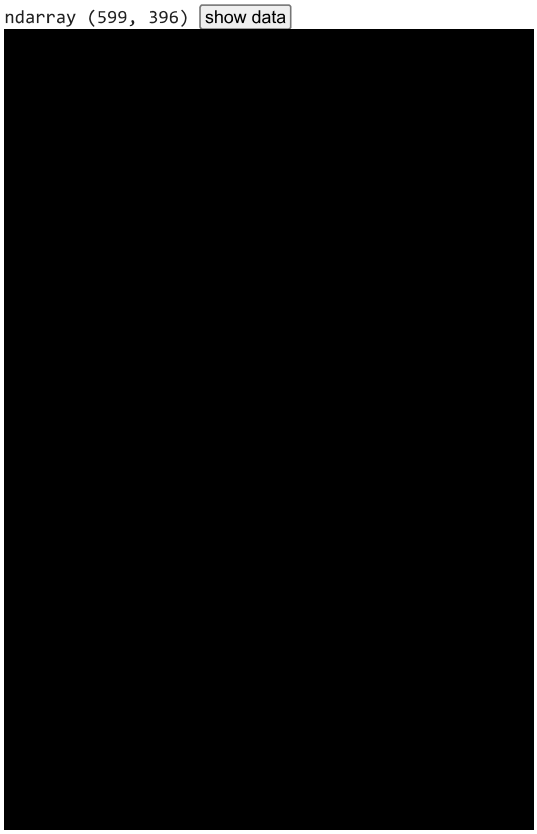
```
gray_img
```

ndarray (599, 396) show data



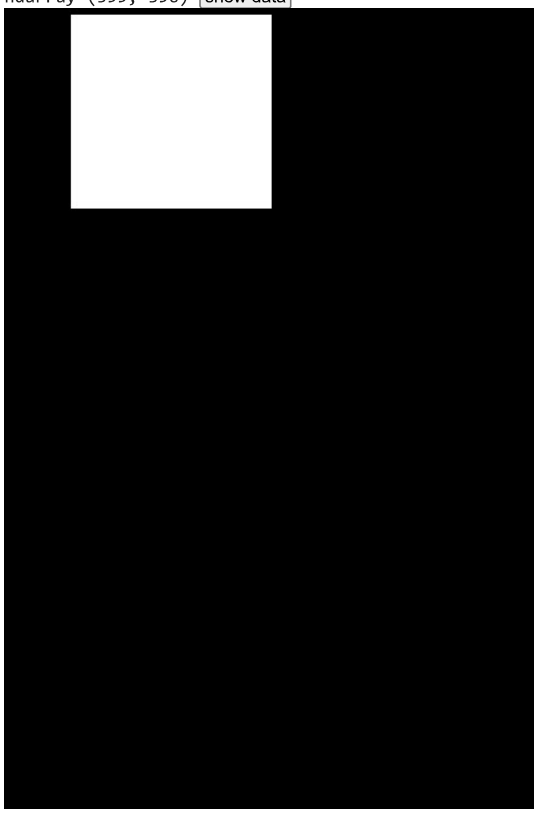
```
import numpy as np
mask = np.zeros(gray_img.shape,"uint8")
mask
```

ndarray (599, 396) show data



```
mask[5:150,50:200]=255
mask
```

ndarray (599, 396) show data



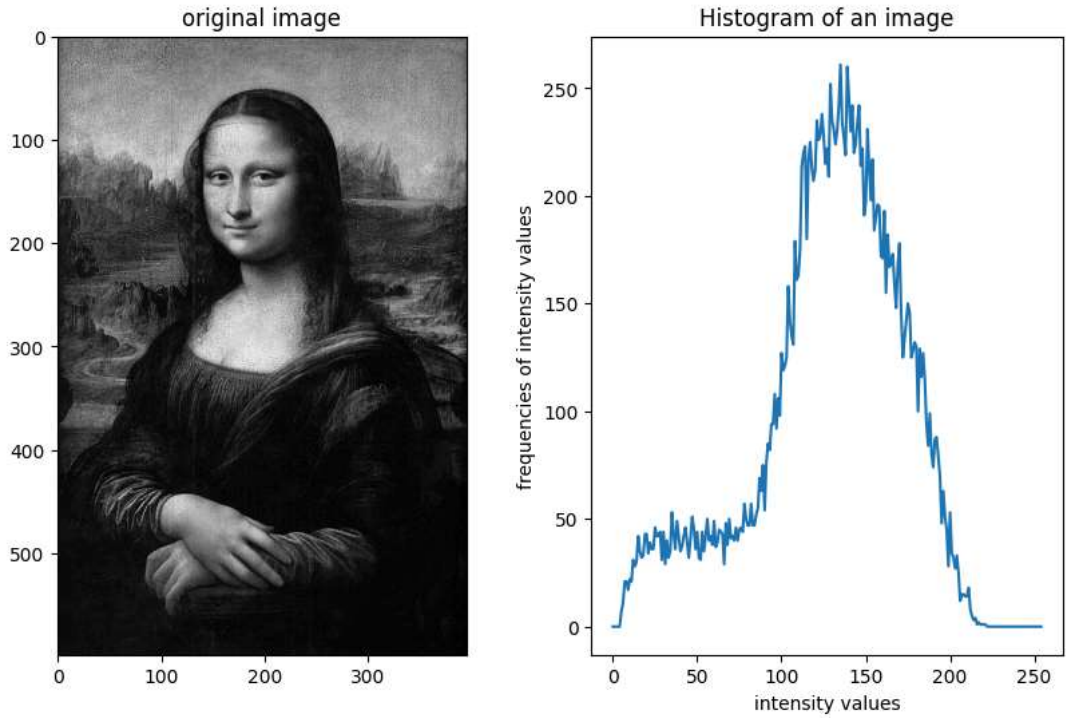
```
# calcHist function
# 1st input -> image for which you want to plot histogram
# 2nd input -> is channnel number
# 3rd input -> mask if you want to calculate histogram for small part of image
# 4th input -> histsize indicate size of histogram i.e. x-axis size
# 5th input -> indicates range of intensity values for which you want to calculate histogram
histr = cv2.calcHist([gray_img],[0],mask,[255],[0,256])
```

```
help(cv2.calcHist)
```

```
Help on built-in function calcHist:

calcHist(...)
    calcHist(images, channels, mask, histSize, ranges[, hist[, accumulate]]) -> hist
    .   @overload
    .
    .   this variant supports only uniform histograms.
    .
    .   ranges argument is either empty vector or a flattened vector of histSize.size()*2 elements
    .   (histSize.size() element pairs). The first and second elements of each pair specify the lower and
    .   upper boundaries.
```

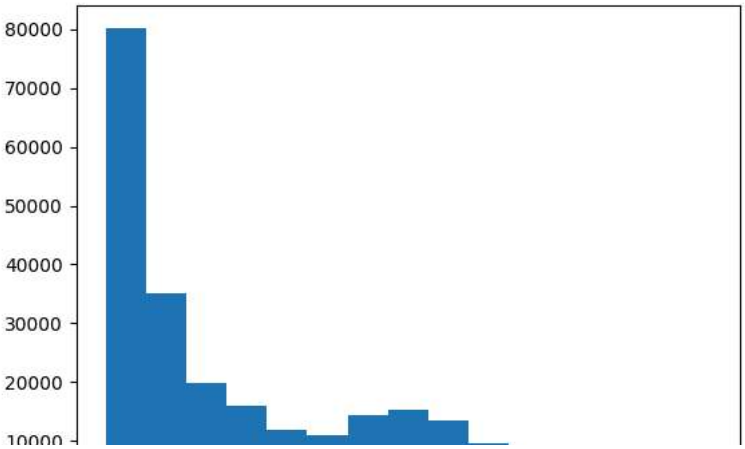
```
import matplotlib.pyplot as plt

plt.figure(figsize=(10,6))
plt.subplot(121)
plt.imshow(gray_img,"gray")
plt.title("original image")
plt.subplot(122)
plt.plot(histr)
plt.xlabel("intensity values")
plt.ylabel("frequencies of intensity values")
plt.title("Histogram of an image")
```

```
Text(0.5, 1.0, 'Histogram of an image')
```




```
plt.hist(gray_img.ravel(),15,[0,256])
```

```
(array([8.0084e+04, 3.5139e+04, 1.9949e+04, 1.6012e+04, 1.1912e+04,
        1.0992e+04, 1.4377e+04, 1.5373e+04, 1.3565e+04, 9.6950e+03,
        5.8090e+03, 3.3480e+03, 9.0500e+02, 4.4000e+01, 0.0000e+00]),
 array([  0.        ,  17.06666667,  34.13333333,  51.2       ,
         68.26666667,  85.33333333, 102.4       , 119.46666667,
        136.53333333, 153.6       , 170.66666667, 187.73333333,
        204.8       , 221.86666667, 238.93333333, 256.        ]),
 <BarContainer object of 15 artists>)
```




```
hist_eq = cv2.equalizeHist(gray_img)
```

```
plt.subplot(221)
plt.imshow(gray_img,"gray")
plt.subplot(222)
plt.hist(gray_img.ravel(),255,[0,256])
plt.subplot(223)
plt.imshow(hist_eq,'gray')
plt.subplot(224)
plt.hist(hist_eq.ravel(),255,[0,256])
plt.show()
```