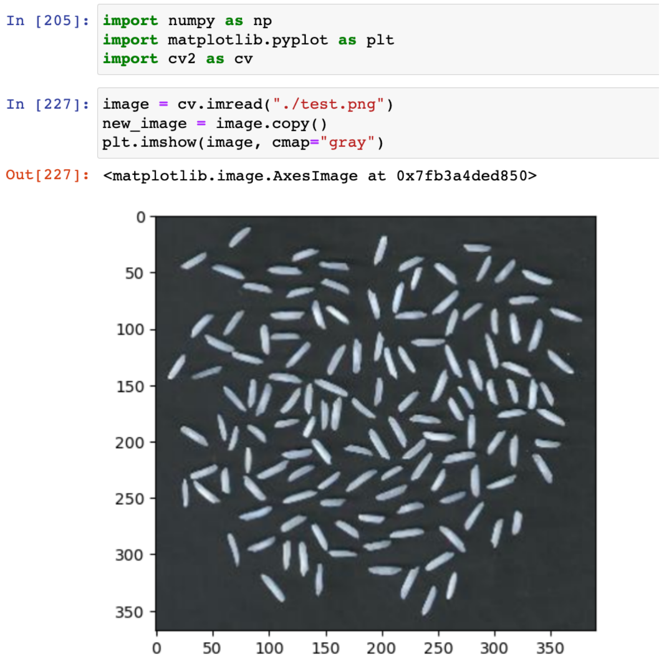
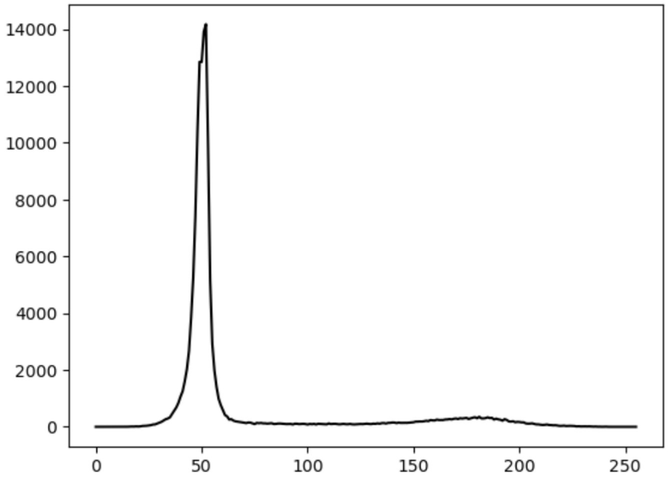
Implementation of basic global thresholding

1- Import necessary libraries to the project
2- Read the image and show it with matplotlib

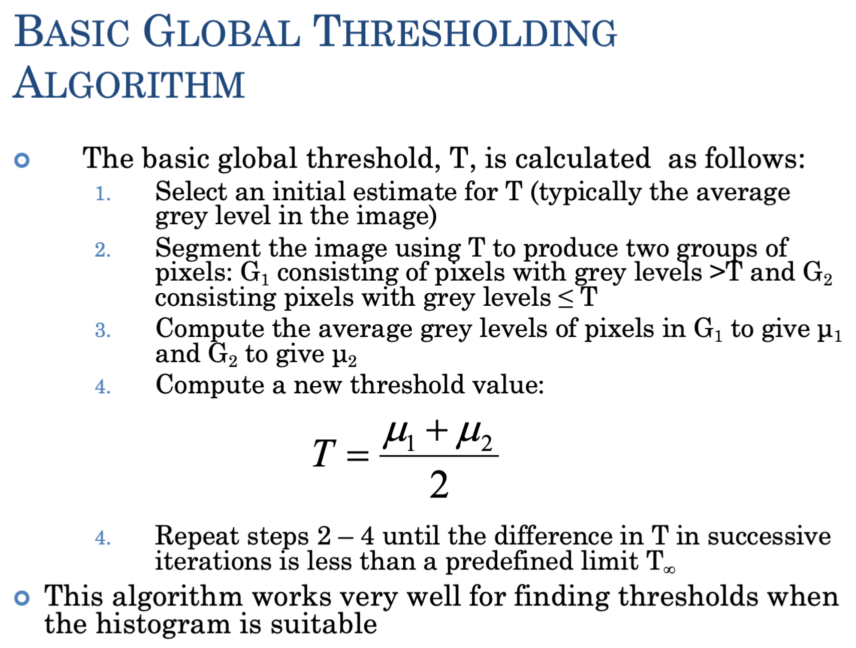


```
In [205]: import numpy as np
          import matplotlib.pyplot as plt
          import cv2 as cv

In [227]: image = cv.imread("./test.png")
          new_image = image.copy()
          plt.imshow(image, cmap="gray")

Out[227]: <matplotlib.image.AxesImage at 0x7fb3a4ded850>
```

3- Plotting the histogram of the image



4 – Basic Global Thresholding Algorithm:



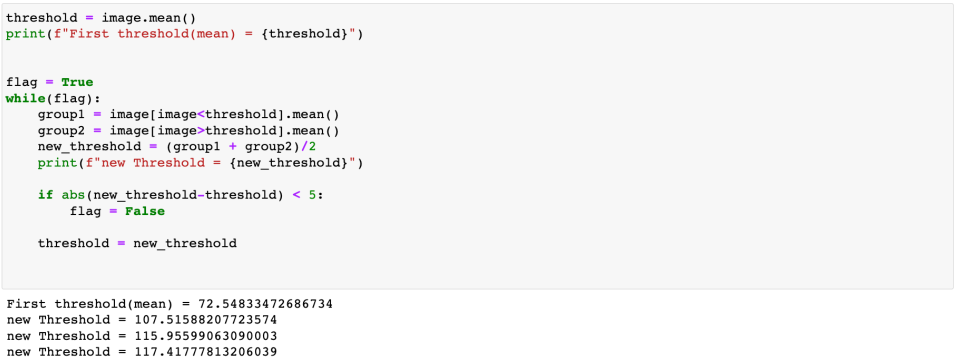
Basic Global Thresholding Algorithm

- The basic global threshold, T, is calculated as follows:
  1. Select an initial estimate for T (typically the average grey level in the image)
  2. Segment the image using T to produce two groups of pixels: G_1 consisting of pixels with grey levels >T and G_2 consisting pixels with grey levels $\leq$ T
  3. Compute the average grey levels of pixels in G_1 to give μ_1 and G_2 to give μ_2
  4. Compute a new threshold value:

$$T = \frac{\mu_1 + \mu_2}{2}$$

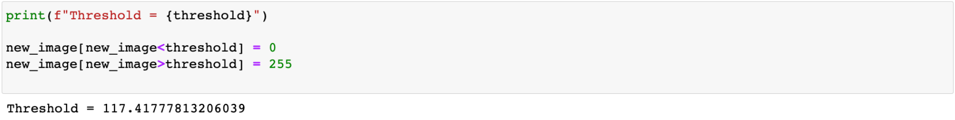
  4. Repeat steps 2 – 4 until the difference in T in successive iterations is less than a predefined limit T_∞
- This algorithm works very well for finding thresholds when the histogram is suitable

4- Implementation of the algorithm above:



```
threshold = image.mean()
print(f"First threshold(mean) = {threshold}")


flag = True
while(flag):
    group1 = image[image<threshold].mean()
    group2 = image[image>threshold].mean()
    new_threshold = (group1 + group2)/2
    print(f"new Threshold = {new_threshold}")

    if abs(new_threshold-threshold) < 5:
        flag = False

    threshold = new_threshold
```

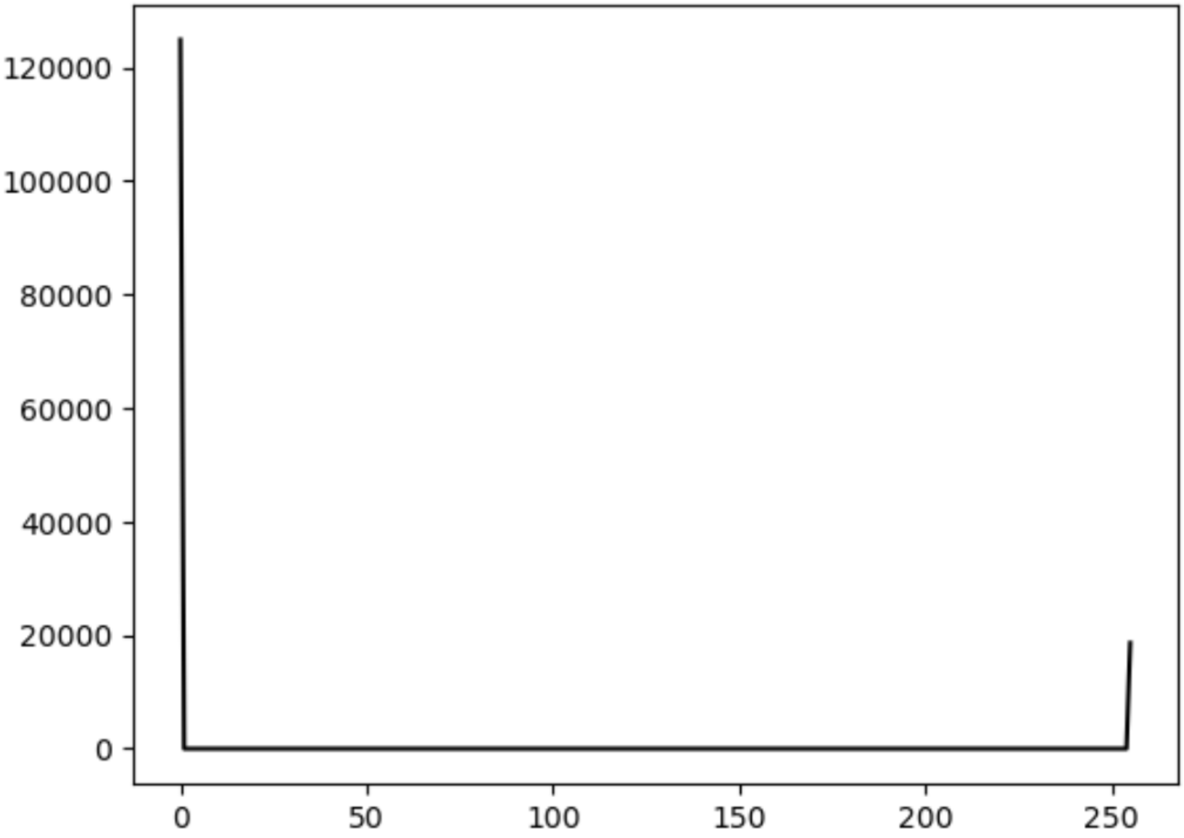
```
First threshold(mean) = 72.54833472686734
new Threshold = 107.51588207723574
new Threshold = 115.95599063090003
new Threshold = 117.41777813206039
```



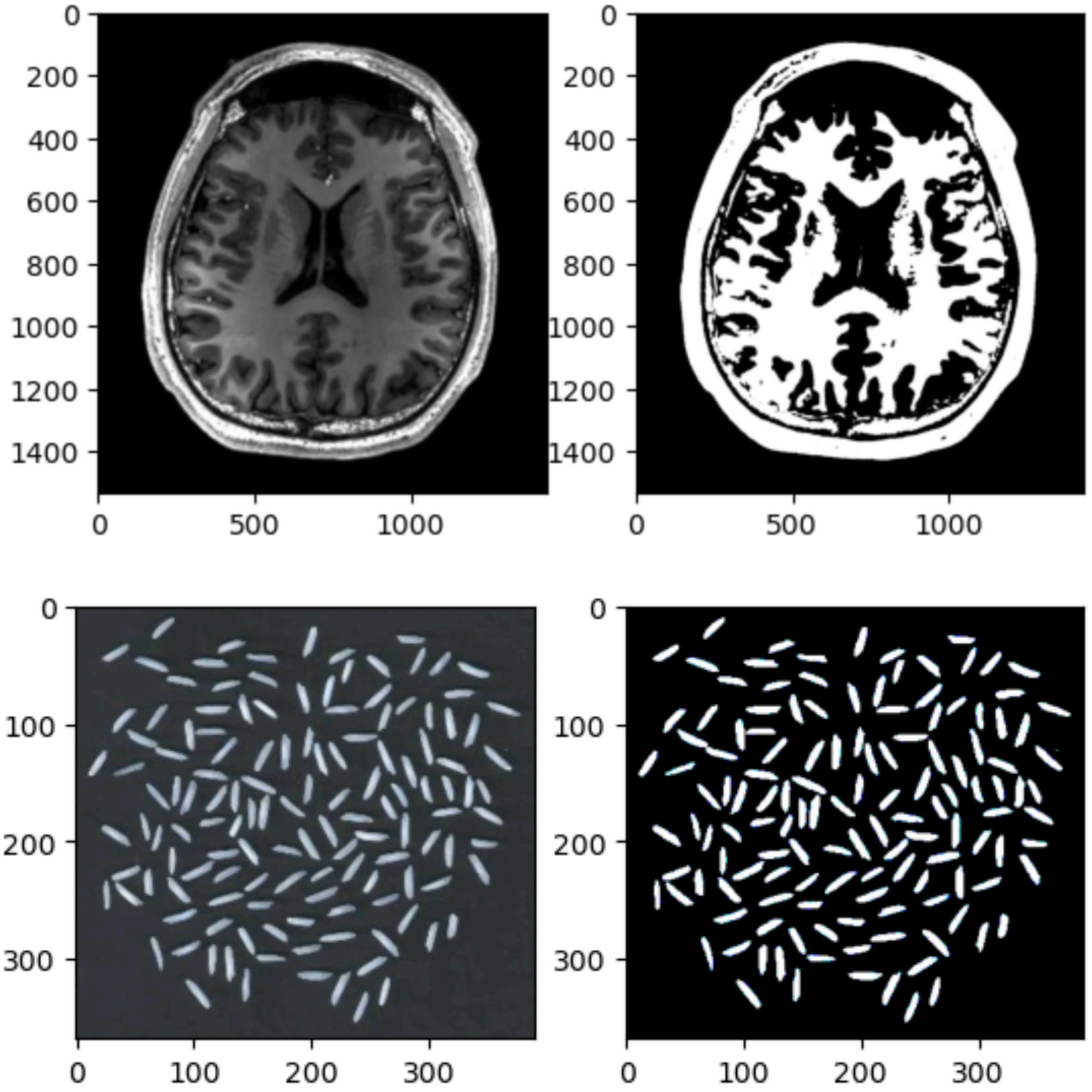
```
print(f"Threshold = {threshold}")

new_image[new_image<threshold] = 0
new_image[new_image>threshold] = 255
```

```
Threshold = 117.41777813206039
```

5- Plotting the histogram of the final image



Result:


0
200
400
600
800
1000
1200
1400
0
500
1000
0
200
400
600
800
1000
1200
1400
0
500
1000
0
100
200
300
0
100
200
300
0
100
200
300
0
100
200
300