

ATILIM UNIVERSITY

COMPE-464

Image Processing

Homework-2

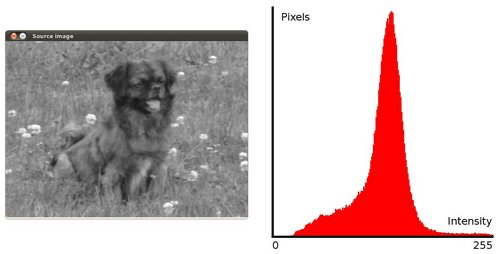
Name: Utku

Surname: Polat

Student ID: 160302001

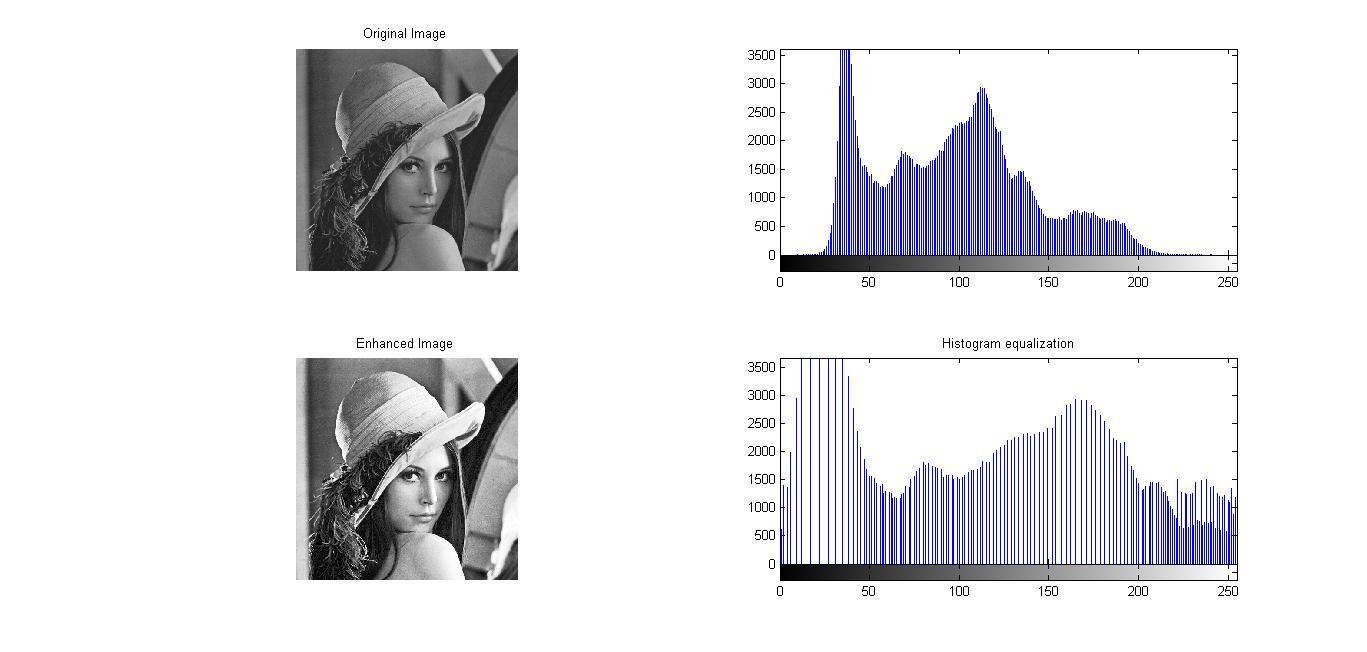
***What is Image Histogram ?***

The histogram image is graphical representation of an image’s property which is the intensity distribution. In a graphic, every column shows us intensity values of pixels. In the graphic, vertical side represent the pixels and horizontal side represents the intensity values.



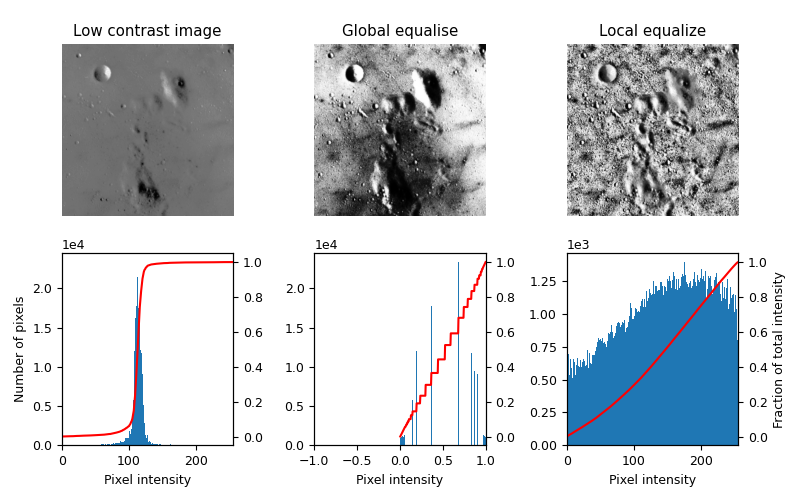
***What is Histogram Equalization and How does it Work?***

Equalization maps one distribution to another distribution because of that the intensity values are sprawled overall region on image. So, the method is usefull for improving the contrast in an image and it using for extending the density range.



***What is Local Histogram Equalization?***

This technique improve contrast of an image. Differecance from histogram equalization is that local method use several histograms and computes them. Sometimes, local histogram equalization increase the noise of the images. Because of that I use CHALE method that prevent it by limiting the amplification.



1. import os
2. import numpy as np
3. import cv2
4. from matplotlib import pyplot as plt
5. #READ ALL IMAGES
6. img1 = cv2.imread('/Users/utkupolat/Desktop/COMPE464-Homework2/images\_for\_HW2/im1.bmp',0)
7. img2 = cv2.imread('/Users/utkupolat/Desktop/COMPE464-Homework2/images\_for\_HW2/im2.bmp',0)
8. img3 = cv2.imread('/Users/utkupolat/Desktop/COMPE464-Homework2/images\_for\_HW2/im3.bmp',0)
9. img4 = cv2.imread('/Users/utkupolat/Desktop/COMPE464-Homework2/images\_for\_HW2/im4.jpg',0)
10. #HISTOGRAMS OF ALL INPUT IMAGES
11. plt.hist(img1.ravel(),256,[0,256])
12. plt.show()
13. plt.hist(img2.ravel(),256,[0,256])
14. plt.show()
15. plt.hist(img3.ravel(),256,[0,256])
16. plt.show()
17. plt.hist(img4.ravel(),256,[0,256])
18. plt.show()
19. #APPLY THE HISTOGRAM EQUALIZATION ALL IMAGES AND SHOW THEIR HISTOGRAMS
20. img1HistEgu = cv2.equalizeHist(img1)
21. HistogramImage\_1 = np.hstack((img1,img1HistEgu))
22. cv2.imwrite('HistogramImage\_1.png',HistogramImage\_1)
23. plt.hist(HistogramImage\_1.ravel(),256,[0,256])
24. plt.show()
25. img2HistEgu = cv2.equalizeHist(img2)
26. HistogramImage\_2 = np.hstack((img2,img2HistEgu))
27. cv2.imwrite('HistogramImage\_2.png',HistogramImage\_2)
28. plt.hist(HistogramImage\_2.ravel(),256,[0,256])
29. plt.show()
30. img3HistEgu = cv2.equalizeHist(img3)
31. HistogramImage\_3 = np.hstack((img3,img3HistEgu))
32. cv2.imwrite('HistogramImage\_3.png',HistogramImage\_3)
33. plt.hist(HistogramImage\_3.ravel(),256,[0,256])
34. plt.show()
35. img4HistEgu = cv2.equalizeHist(img4)
36. HistogramImage\_4 = np.hstack((img4,img4HistEgu))
37. cv2.imwrite('HistogramImage\_4.png',HistogramImage\_4)
38. plt.hist(HistogramImage\_4.ravel(),256,[0,256])
39. plt.show()
40. #APPLY THE LOCAL HISTOGRAM EQUALIZATION OF ALL IMAGES AND SHOW THEIR HISTOGRAMS
41. localImage1 = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))
42. localHist1 = localImage1.apply(img1)
43. cv2.imwrite('LocalImage\_1.jpg',localHist1)
44. plt.hist(localHist1.ravel(),256,[0,256])
45. plt.show()
46. localImage2 = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))
47. localHist2 = localImage2.apply(img2)
48. cv2.imwrite('LocalImage\_2.jpg',localHist2)
49. plt.hist(localHist2.ravel(),256,[0,256])
50. plt.show()
51. localImage3 = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))
52. localHist3 = localImage3.apply(img3)
53. cv2.imwrite('LocalImage\_3.jpg',localHist3)
54. plt.hist(localHist3.ravel(),256,[0,256])
55. plt.show()
56. localImage4 = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))
57. localHist4 = localImage4.apply(img4)
58. cv2.imwrite('LocalImage\_4.jpg',localHist4)
59. plt.hist(localHist4.ravel(),256,[0,256])
60. plt.show()

***Comments***

Line 7-10 => Read all images with ‘cv2.imread’ function

Line 13-23 => Histograms graphs of all images ‘plt.hist()’ take all histograms and show them by ‘plt.show()’ and save manually in desktop.

Line 26-52 => Apply the histogram equalization for all images by ‘cv2.equalizeHist’ and output the original image and applied histogram equalized image with ‘hstack’ funtion after that ‘cv2.imwrite’ save all histogram equalized images.

Line 55-81 => Apply the local histogram equalization by “CHALE” method and show histogram graph by ‘plt.hist’ and show them ‘plt.show’.

*Answer a)*

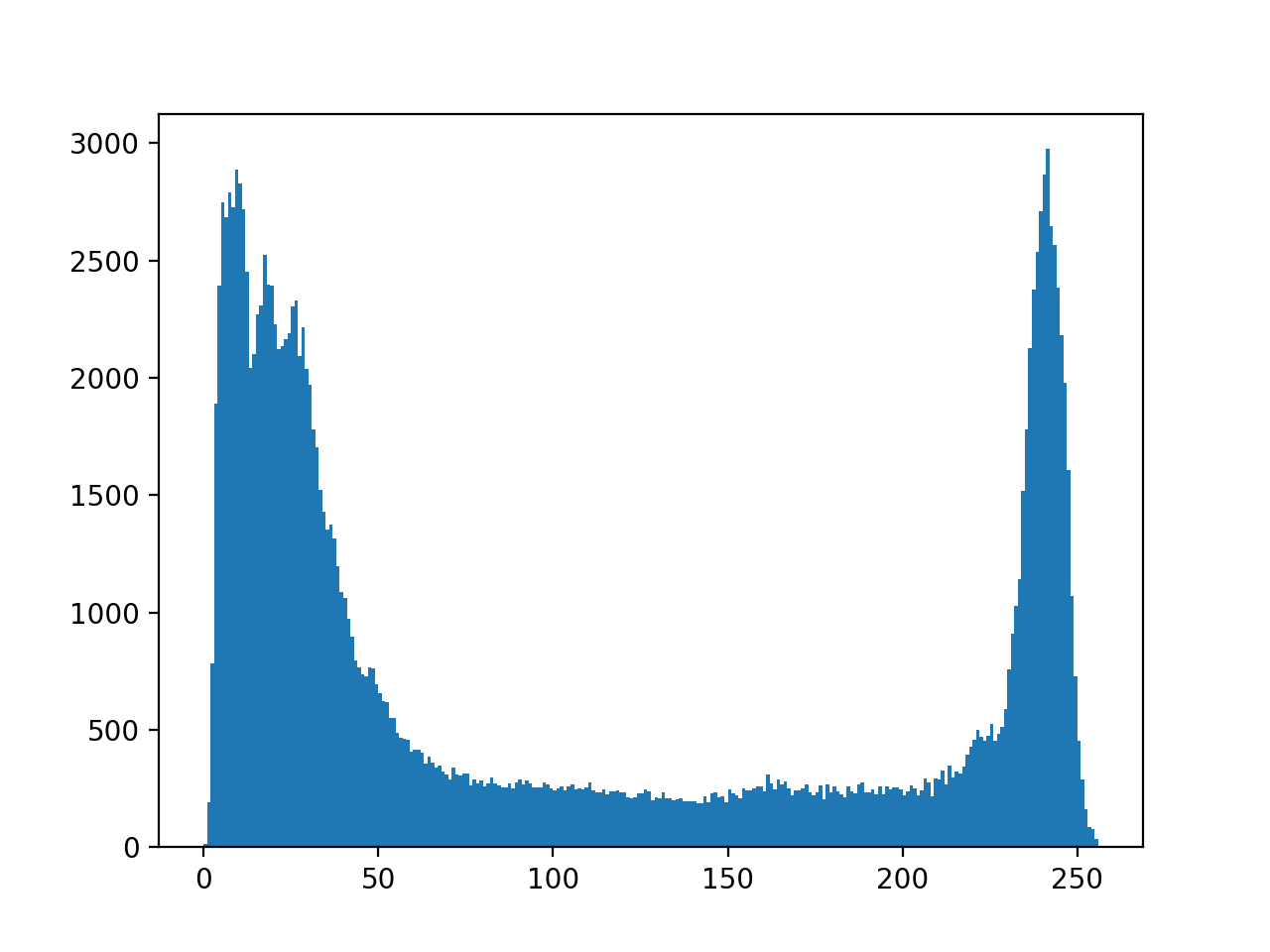


Image 1 Histogram

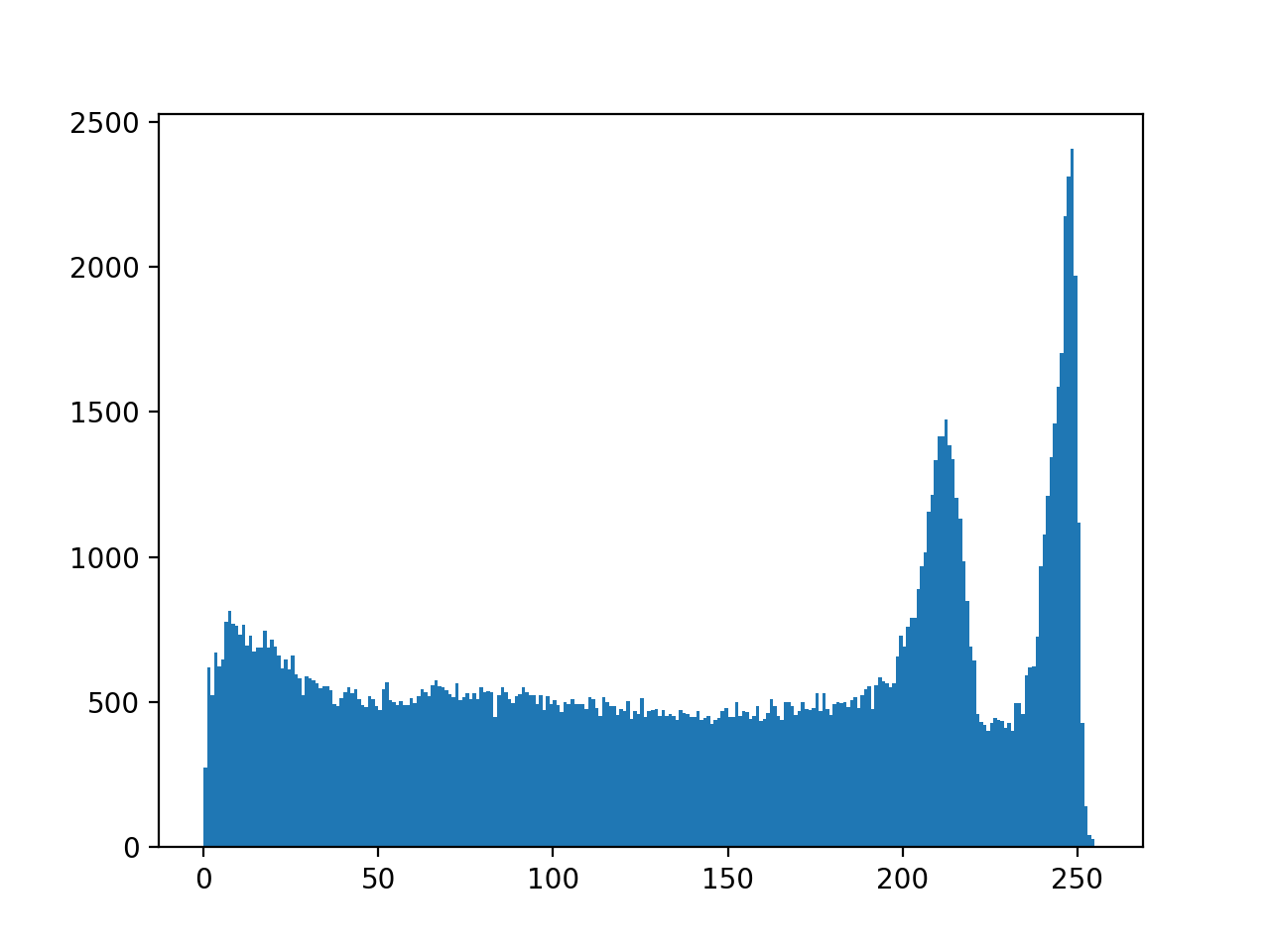


Image 2 Histogram

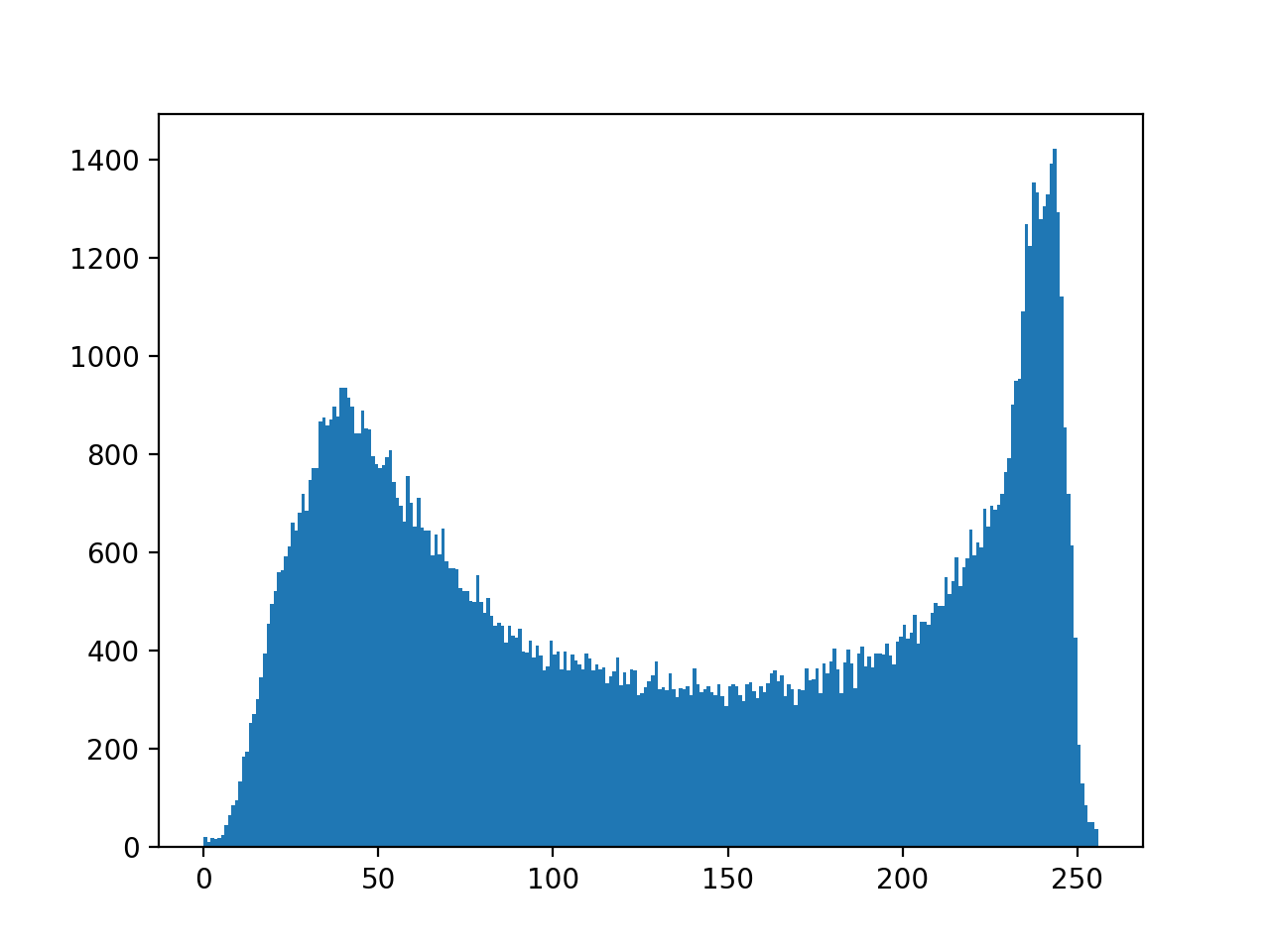


Image 3 Histogram

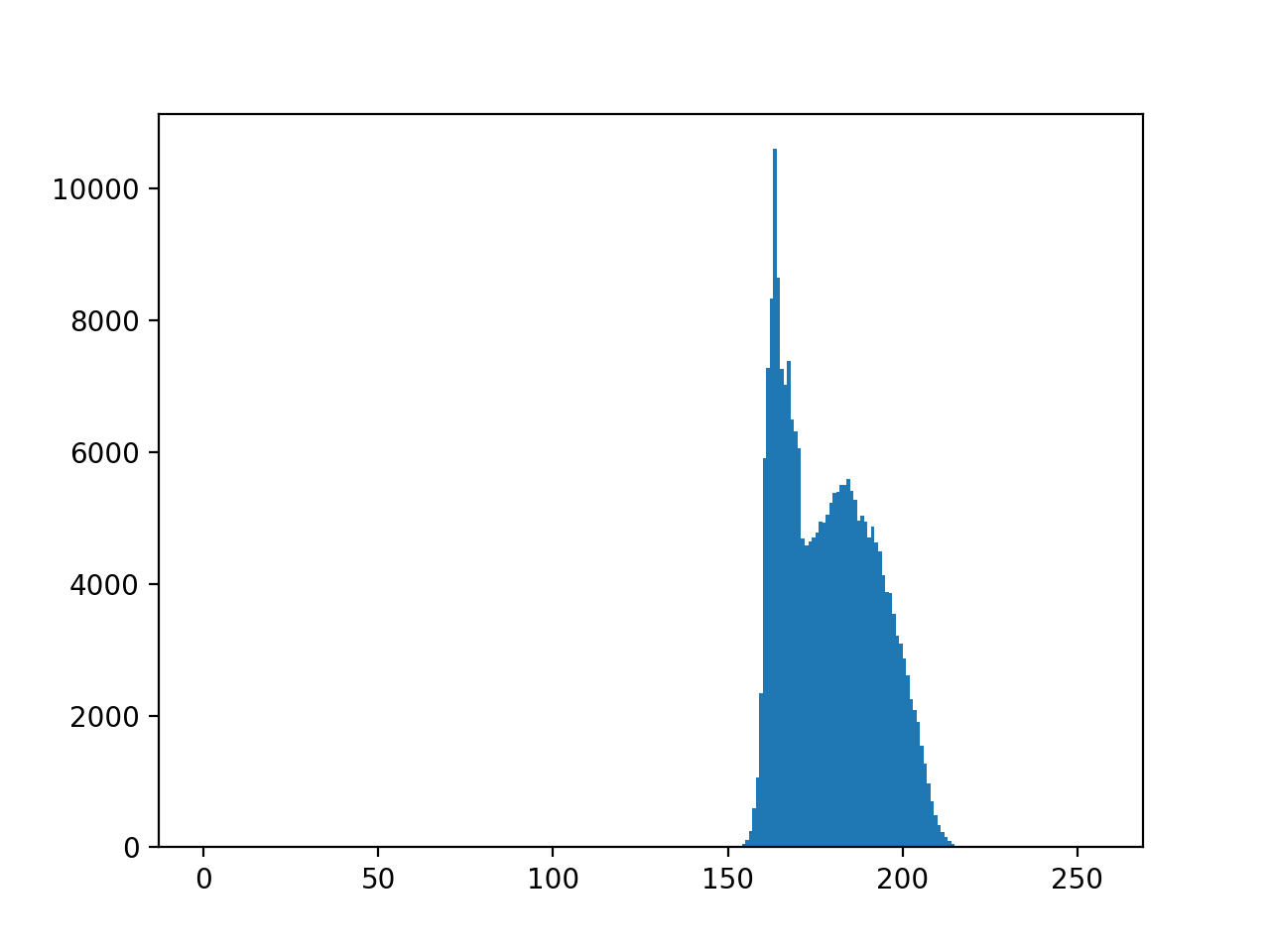
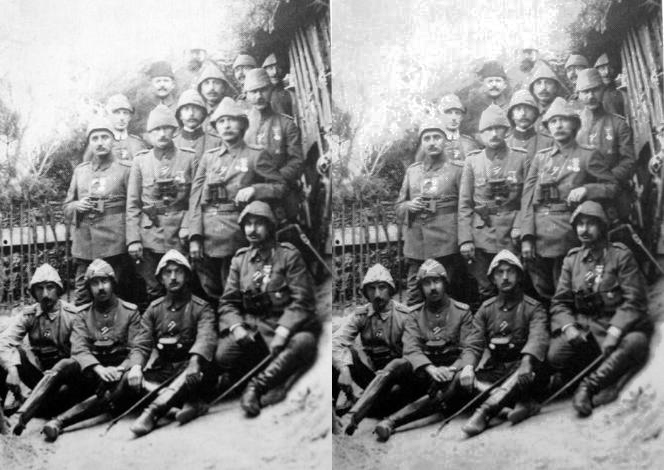


Image 4 Histogram

*Answer b)*



## Histograms Equalization Image 1



## Histograms Equalization Image 2

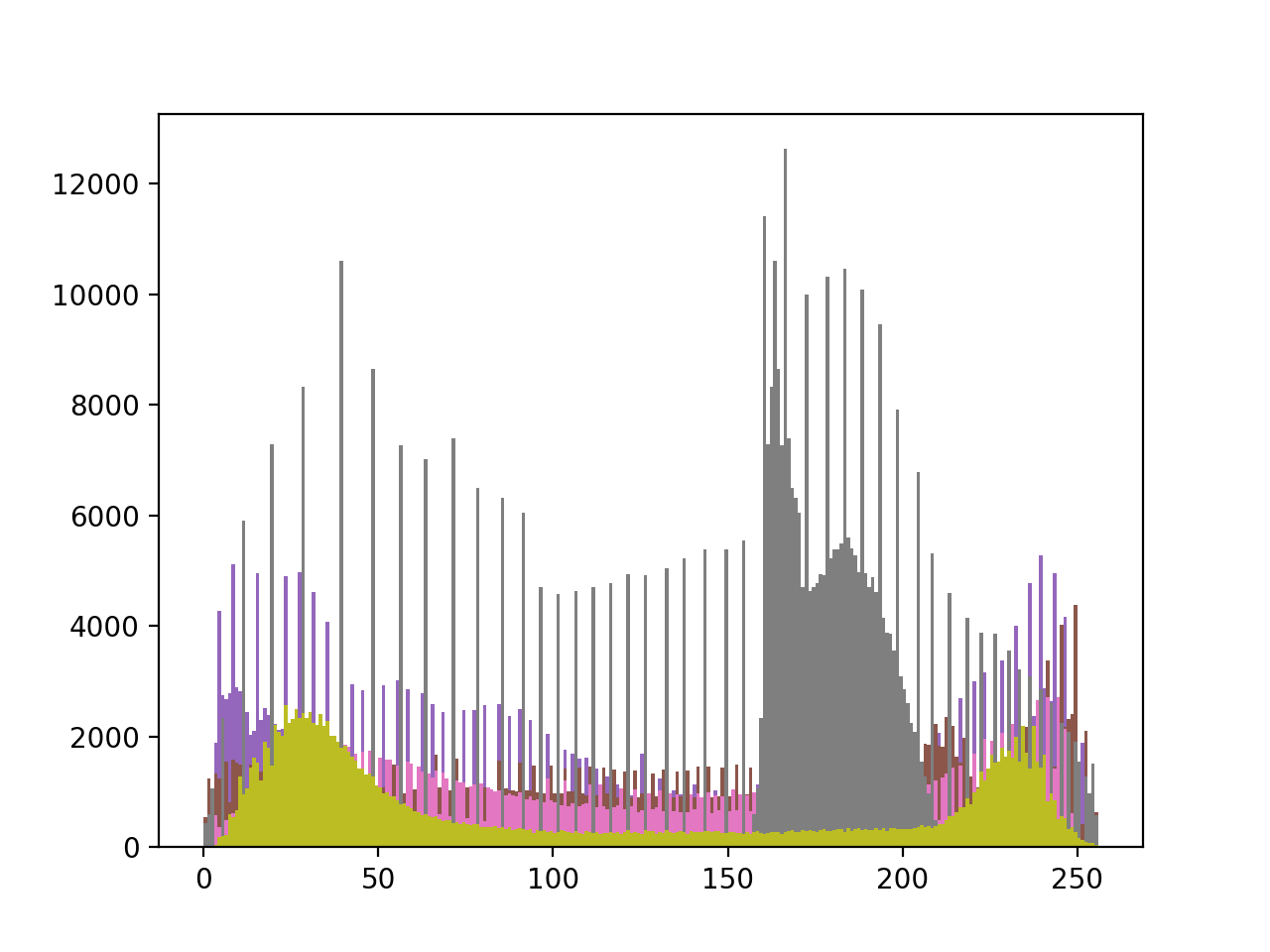
## 

## Histograms Equalization Image 3

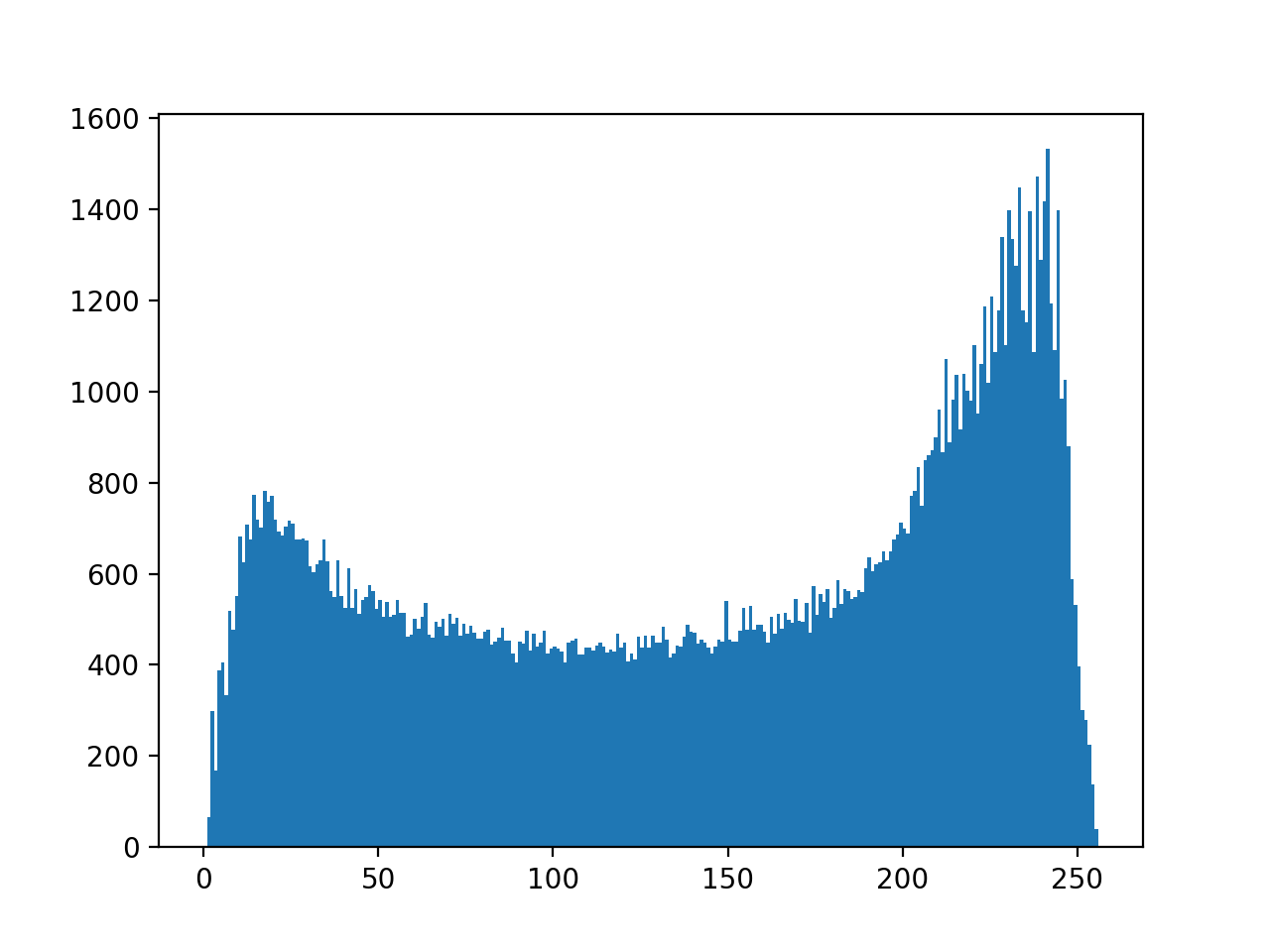


## Histograms Equalization Image 4

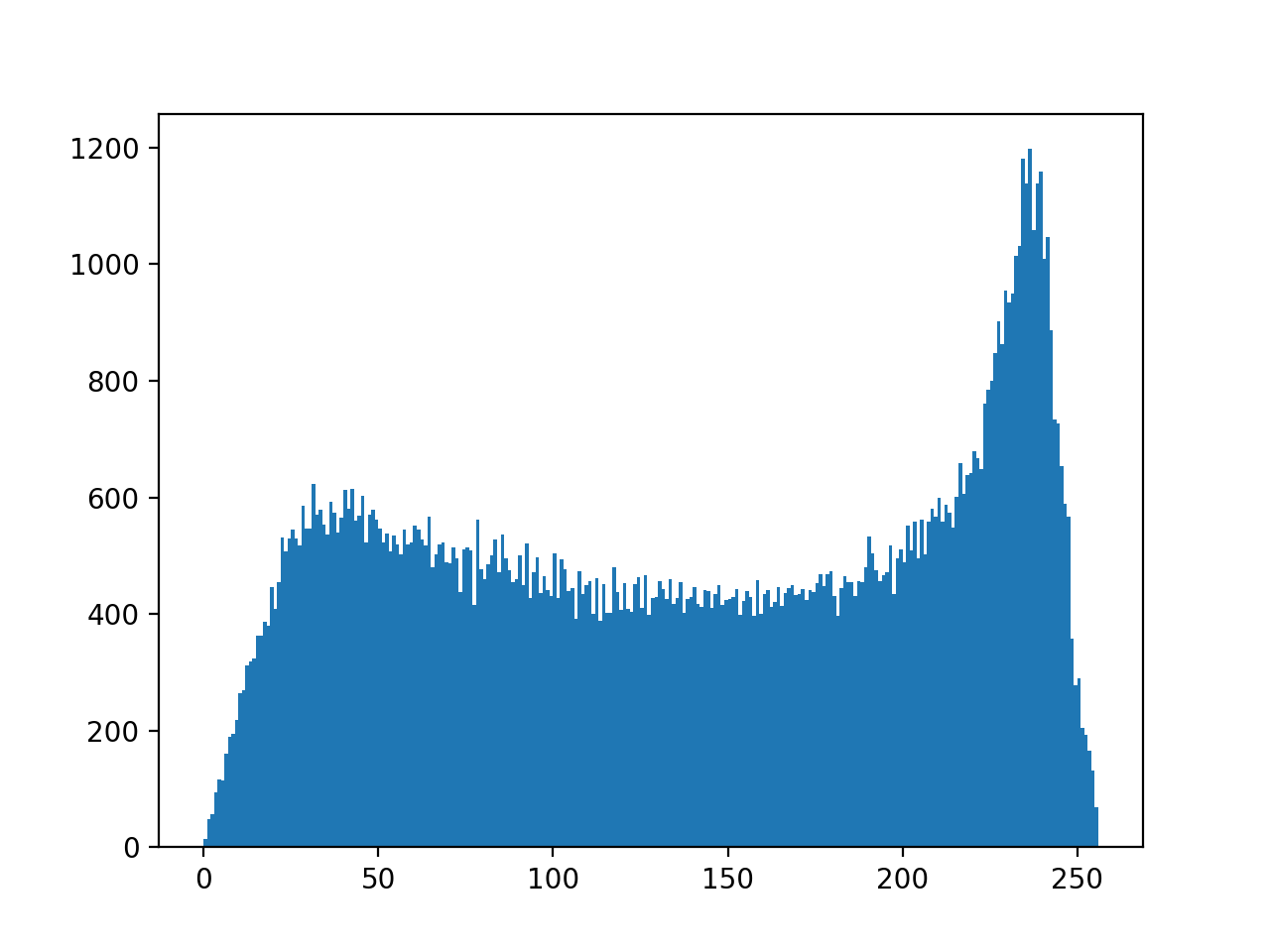
*Answer c)*



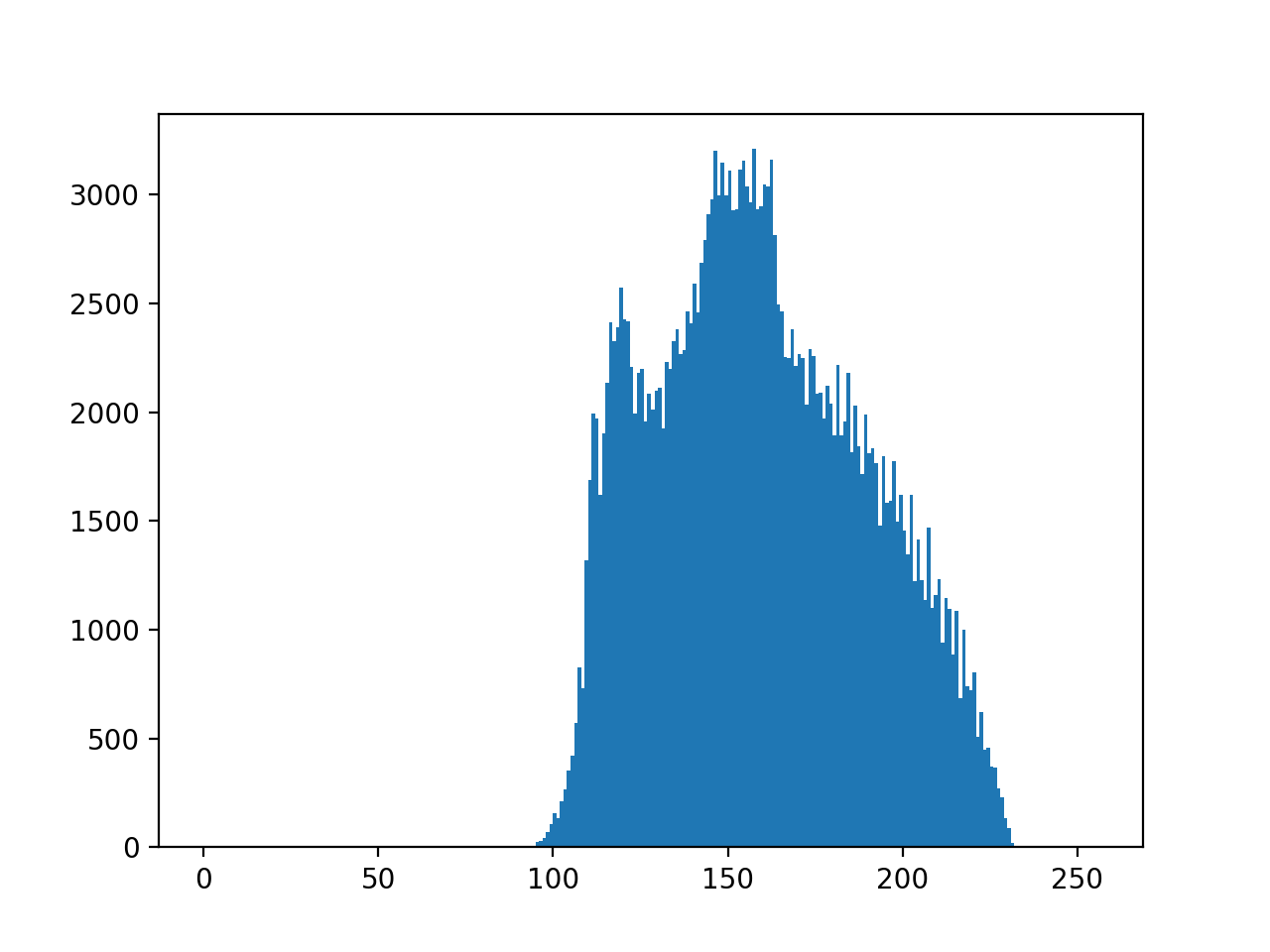
 *Image 1 Local Histogram Equalization and its Histogram*



*Image 2 Local Histogram Equalization and its Histogram*

**

*Image 3 Local Histogram Equalization and its Histogram*

**

*Image 4 Local Histogram Equalization and its Histogram*