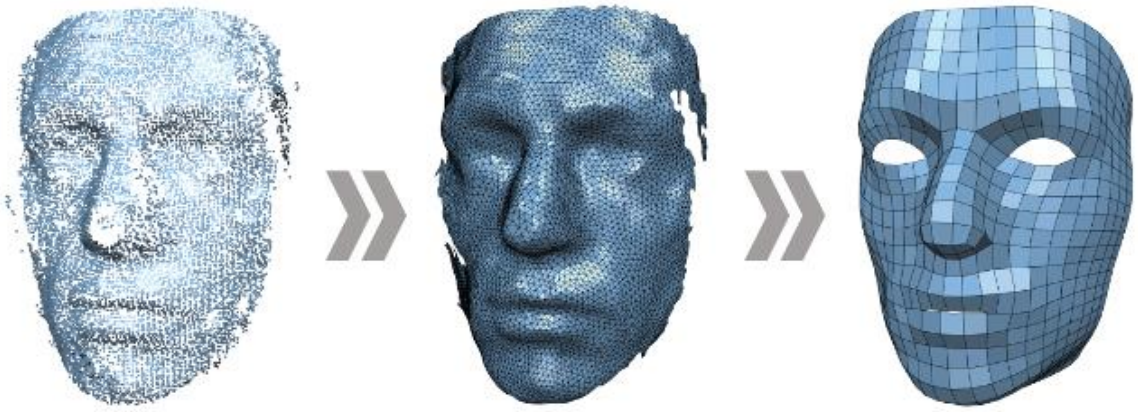
POKHARA UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE

APEX COLLEGE

MID-BANESHWOR, KATHMANDU



COMPUTER GRAPHICS AND IMAGE PROCESSING

REPORT: 4

SUBMITTED TO:

Abhinash Khanal

Bijaya Khadka

SUBMITTER: Piyush Manandhar

Signature

BCIS MAXTHON

HISTOGRAM PROCESSSING

INTRODUCTION

In this lab, we are introduced to histogram and its various operations. Histograms are crucial graphs when studying image processing. The information contained in the graph is a representation of pixel distribution as a function of tonal variation, image histograms can be analyzed for peaks and/or valleys. This threshold value can then be used for edge detection, image segmentation, and co-occurrence matrices. Here, we learn about histogram equalization, histogram matching as well as local histogram equalization.

OBJECTIVES

- To perform Histogram Equalization
- To perform Histogram Matching.
- To perform Local Histogram Equalization.

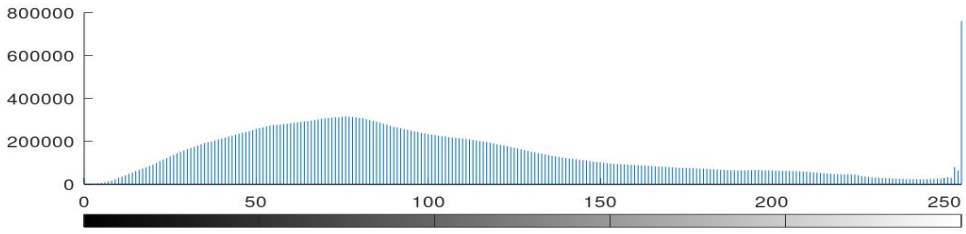
THEORY

1. Histogram:

Histogram of an image, like other histograms also shows frequency. But an image histogram, shows frequency of pixels intensity values. In an image histogram, the x axis shows the gray level intensities and the y axis shows the frequency of these intensities.

For example,

Example of a Image Histogram



In the above figure, it shows the image as well as the histogram under it. The X axis represents the gray level values whereas the Y axis depicts the frequency of these values.

Histograms has many uses in image processing. The first use as it has also been discussed above is the analysis of the image. We can predict about an image by just looking at its histogram.

The second use of histogram is for brightness purposes. The histogram has wide application in image brightness. Not only in brightness, but histograms are also used in adjusting contrast of an image.

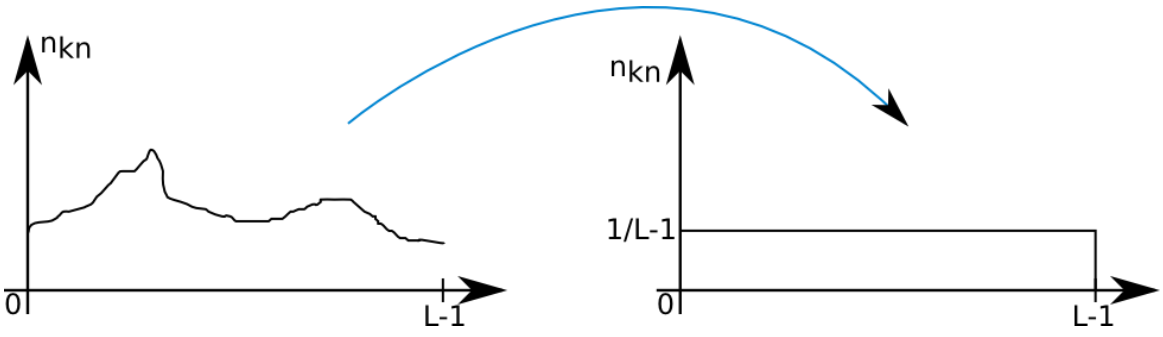
Another important use of histogram is to equalize an image (We will practice this later). Last but not the least, histogram has wide use in thresholding. This is mostly used in computer vision.

2. <u>Histogram Equalization</u>:

Histogram equalization is a method to process images in order to adjust the contrast of an image by modifying the intensity distribution of the histogram to enhance the image.

The processing of histogram equalization relies on the use of the ***cumulative probability function*** (***cdf***). The ***cdf*** is a cumulative sum of all the probabilities lying in its domain.

The idea of this processing is to give to the resulting image a linear cumulative distribution function. Indeed, a linear ***cdf*** is associated to the uniform histogram that we want the resulting image to have.



3. Histogram Matching:

Histogram Matching of an image is a process where an image is modified such that its histogram matches that of another (reference) image's histogram. Histogram matching is also known as histogram specification.

Let us suppose we have two images, an input image and a reference image. We want to use histogram matching to force the input image to have a histogram that is the shape of the histogram of the reference image.

4. Local Histogram Equalization (LHE):

While global histogram equalization enhances the contrast of the whole image, **local histogram equalization** can enhance many image details by taking different transformation of the same gray level at different places in the original image. The algorithm is generally employed when global histogram equalization does not enhance detail in small regions of the image. The algorithm uses the histogram of a local neighborhood or window of a predetermined size to calculate the transformation of each pixel in the image.

However, slow speed and the over enhancement of noise it produces in relatively homogeneous regions are two problems.

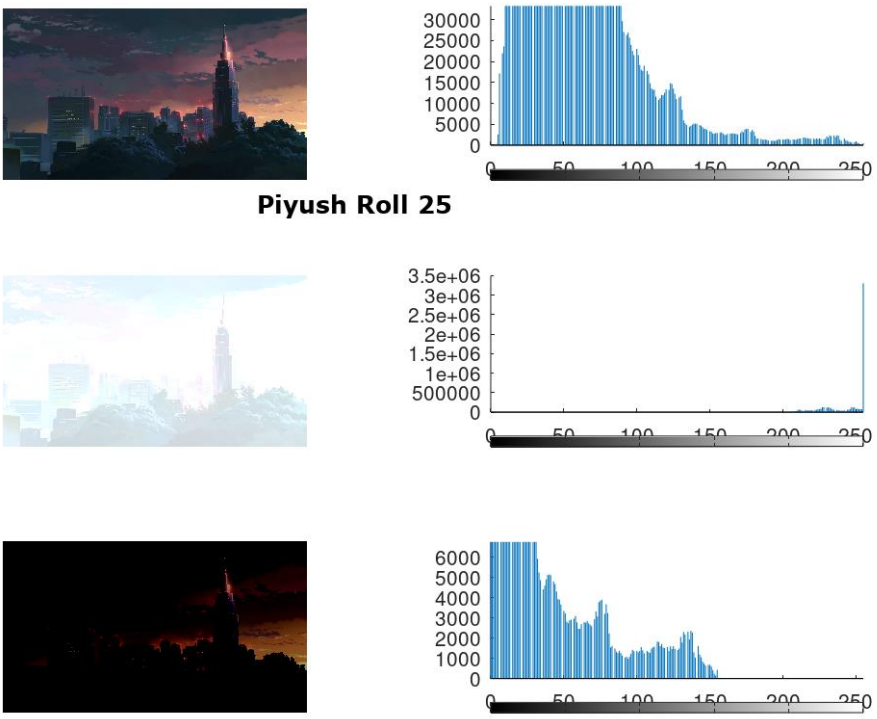
COMMANDS/ SYNTAX

1. **imhist(I):** Produce histogram counts of image I.
2. **J = imhistmatch (I, ref)** transforms the 2-D grayscale or true color image **I** returning output image **J** whose histogram approximately matches the histogram of the reference image **ref**
3. **histeq(I):** Generate equalized histogram of grayscale image I.

RUNNING THE LAB

1. Use the histogram processing methods on images of your choice and discuss the results. Use 3 images, one over-exposed, one under-exposed and one normal image.

```
pkg load image;
img1=imread('demo.png');
over=img1 + 200;
under= img1 -100;
subplot(321),imshow(img1);
subplot(322),imhist(img1);
subplot(323),imshow(over);
subplot(324),imhist(over);
subplot(325),imshow(under);
subplot(326),imhist(under);
```

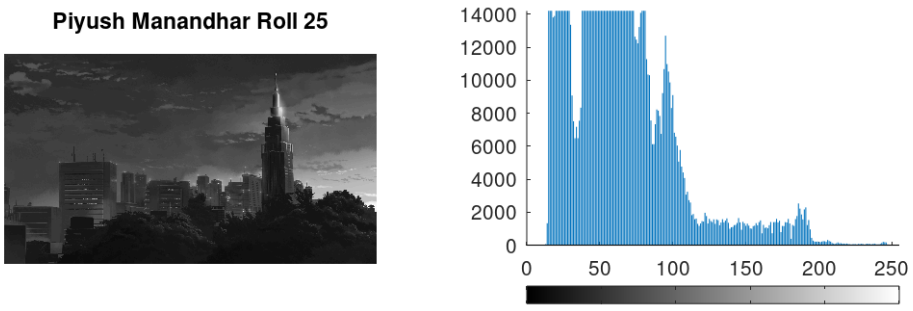


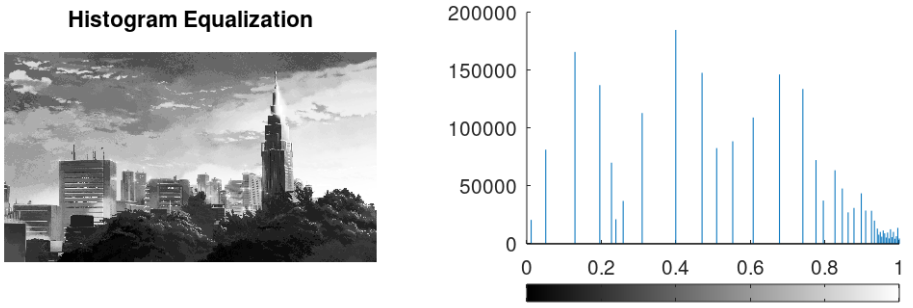
Piyush Roll 25

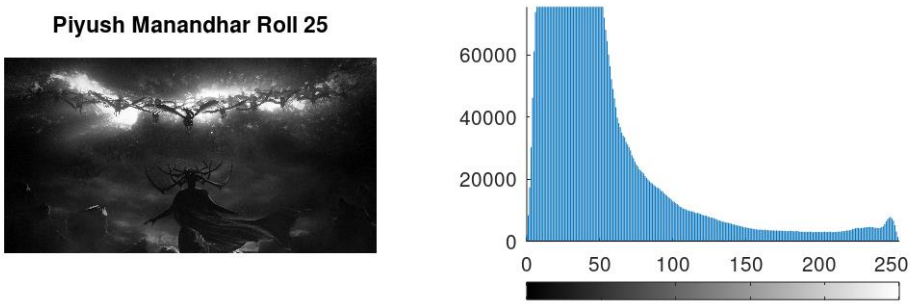
In the above program, we can see that the over- contrasted image has the intensity of the highest gray level far more than the lower gray levels due to exposure of white light. Similarly, in the histogram for the under contrasted image, we can see the intensity of the lower gray levels have more frequency compared to the higher gray levels.

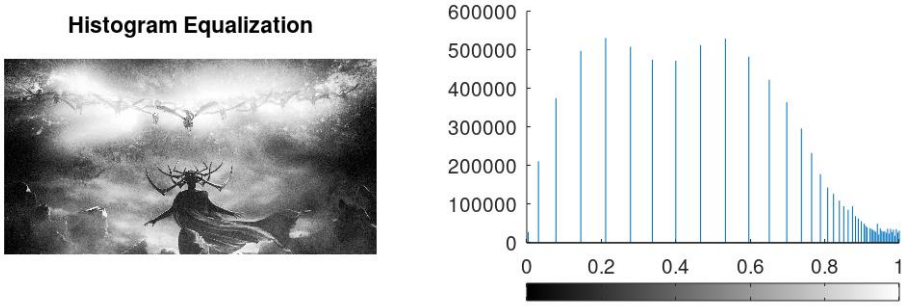
2. Perform histogram equalization on two images of your choice. Compare the resulting histograms. What will happen if the equalized image is applied histogram equalization for the second time?

```
pkg load image;
img2 = rgb2gray(imread('demo.png'));
out1 = histeq(img2);
out2 = histeq(out1);
subplot(321),imshow(img2),title('Piyush Manandhar Roll 25')
subplot(322),imhist(img2)
subplot(323),imshow(out1),title('Histogram Equalization')
subplot(324),imhist(out1)
subplot(325),imshow(out2),title('Histogram Equalization of an Already Equalized Image')
subplot(326),imhist(out2)
```



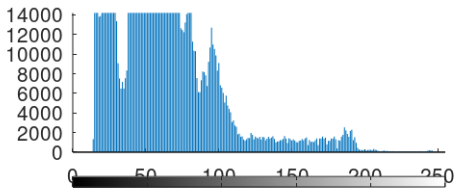




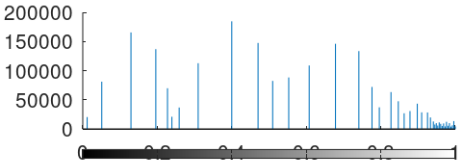


In the above two images, we performed histogram equalization. Our main goal of conducting histogram equalization is to produce an output image that has a flattened histogram. For both images, we can see that the algorithm tries its best to flatten the otherwise unequal histogram of the images.

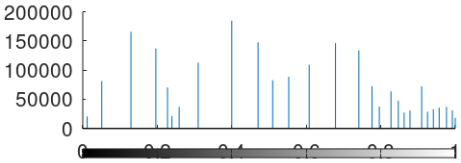
Piyush Manandhar Roll 25



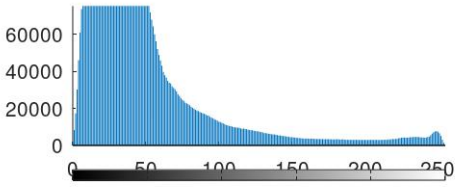
Histogram Equalization



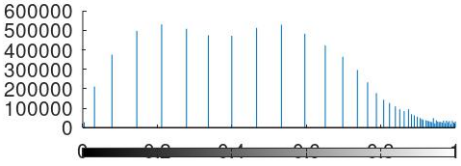
Histogram Equalization of an Already Equalized Image



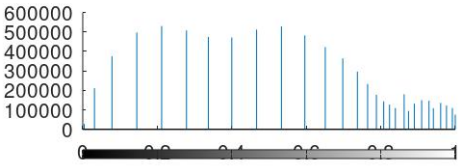
Piyush Manandhar Roll 25



Histogram Equalization



Histogram Equalization of an Already Equalized Image



However, when we perform this same operation on an already equalized image, there isn't much change.

3. Apply histogram matching to any two images taking 'lenna.png' as the reference image. Evaluate the resulting histograms.

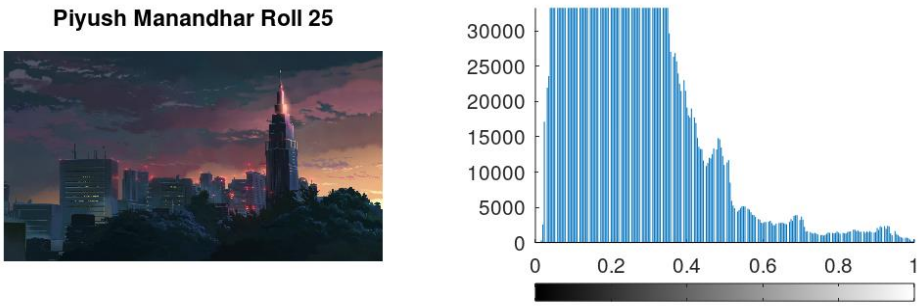
```
40 pkg load image;
41 img1 = imread('demo.png');
42 img2= imread('lenna.png');
43 out = imhistmatch(img1,img2);
44 subplot(321),imshow(img1);
45 subplot(322),imhist(img1);
46 subplot(321),imshow(img1);
47 subplot(322),imhist(img1);
48 subplot(323),imshow(out);
49 subplot(324),imhist(out);
```

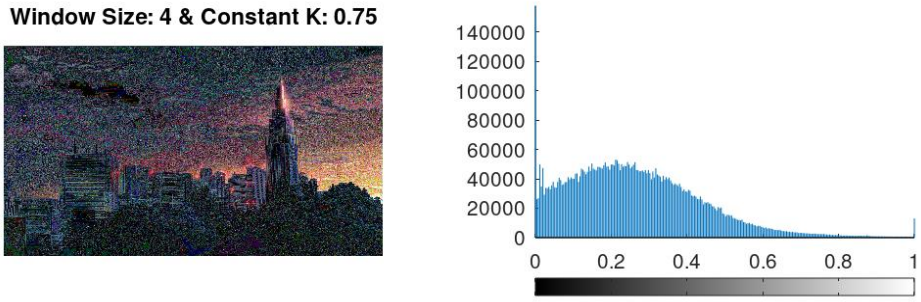
```
Please read <https://www.octave.org/missing.html> to learn how you can
contribute missing functionality.
error: 'imhistmatch' undefined near line 43 column 7
error: called from
    Lab4 at line 43 column 5
>>
```

Since **imhistmatch()** wasn't in the Octave function library, we couldn't run it. However, this function is available on MATLAB and should run without any problem.

4. Apply local histogram processing(equalization) to an image. What is the effect of window size on the results? Compare the result with global histogram equalization.

```
pkg load image;
f=imread('demo.png');
f=im2double(f);
w=input('Enter the Neighborhood or Window size : ');
k=input('Enter the value of the constant k (value should be between 0 and 1) : ');
M=mean2(f);
z=colfilt(f,[w w],'sliding',@std);
m=colfilt(f,[w w],'sliding',@mean);
A=k*M./z;
g=A.*(f-m)+m;
subplot(221), imshow(f), title('Piyush Manandhar Roll 25');
subplot(222), imhist(f);
subplot(223), imshow(g), title('Window Size: 4 & Constant K: 0.75');
subplot(224), imhist(g);
```





Using local histogram equalization, we enhance the above image with window size of 4 and taking constant (K) of 0.75. Just looking at the images, we can see that the enhanced image has more noise compared to the original. The addition of noise helps to bring out details which could have been unseen before.

An image can be considered as a union of homogeneous regions. The size of the window chosen is an important parameter in LHE. Its effects are:

I. **Smoothing Effect**:

Large windows will, more often than not, have a histogram that is a mixture of the histograms from a number of regions. Such a histogram is less likely to have a dominant mode (i.e. a mode that contains a large percentage of the total pixel population) than the histogram obtained from a smaller window. Therefore, there will be restricted stretching (or smoothing) in a few areas of the histogram (and compression elsewhere). On the other hand, smaller windows are more likely to contain pixels from a single region, and to have a dominant mode. Thus, equalizing the histogram will cause large stretching.

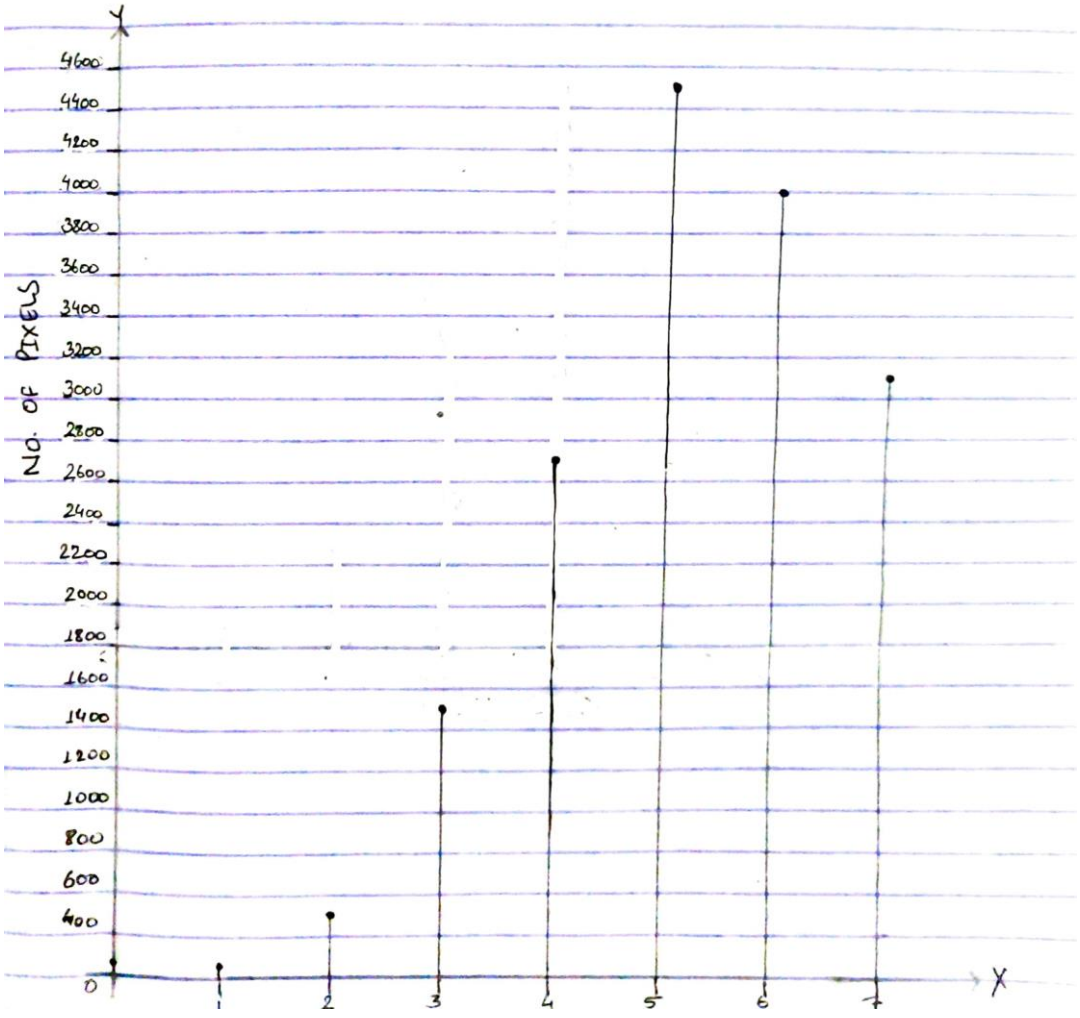
II. **Boundary Variation:**

The size of the window used in LHE also affects the transformation near boundaries. As the window size increases the transition (edge) between the semi-conductor object and the background becomes broader, producing a distinctive smoothed gradient.

5. Suppose that you have a 128 × 128 square pixel image with an 8 gray level intensity range, within which the lighter intensity levels predominate as shown in the table below.

Gray Level	0	1	2	3	4	5	6	7
No. of Pixels	34	50	500	1500	2700	4500	4000	3100

a) Sketch the histogram (no. of pixels vs gray Level) to describe this distribution.



b) How many pixels/gray pixels would be there in an equalized version of this histogram?

Soln

We know,

Total No. of pixels (n) = 16,384

Now,

Gray Level (r_k)	No. of Pixels (n_k)	PDF $P_r(r_k) = n_k/n$	CDF $S_k = \Sigma P_k(r_k)$	$(L-1)S_k = 7 \times S_k$	Rounding off
0	34	0.002	0.002	0.014	0
1	50	0.003	0.005	0.035	0
2	500	0.030	0.035	0.245	0
3	1500	0.091	0.126	0.882	1
4	2700	0.164	0.29	2.03	2
5	4500	0.275	0.565	3.955	4
6	4000	0.244	0.809	5.663	6
7	3100	0.190	1	7	7
	16,384	1			

Finally,

Equating Gray levels to No. of Pixels.

0 ⇒ 584

1 ⇒ 1500

2 ⇒ 2700

3 ⇒ 0

4 ⇒ 4500

5 ⇒ 0

6 ⇒ 4000

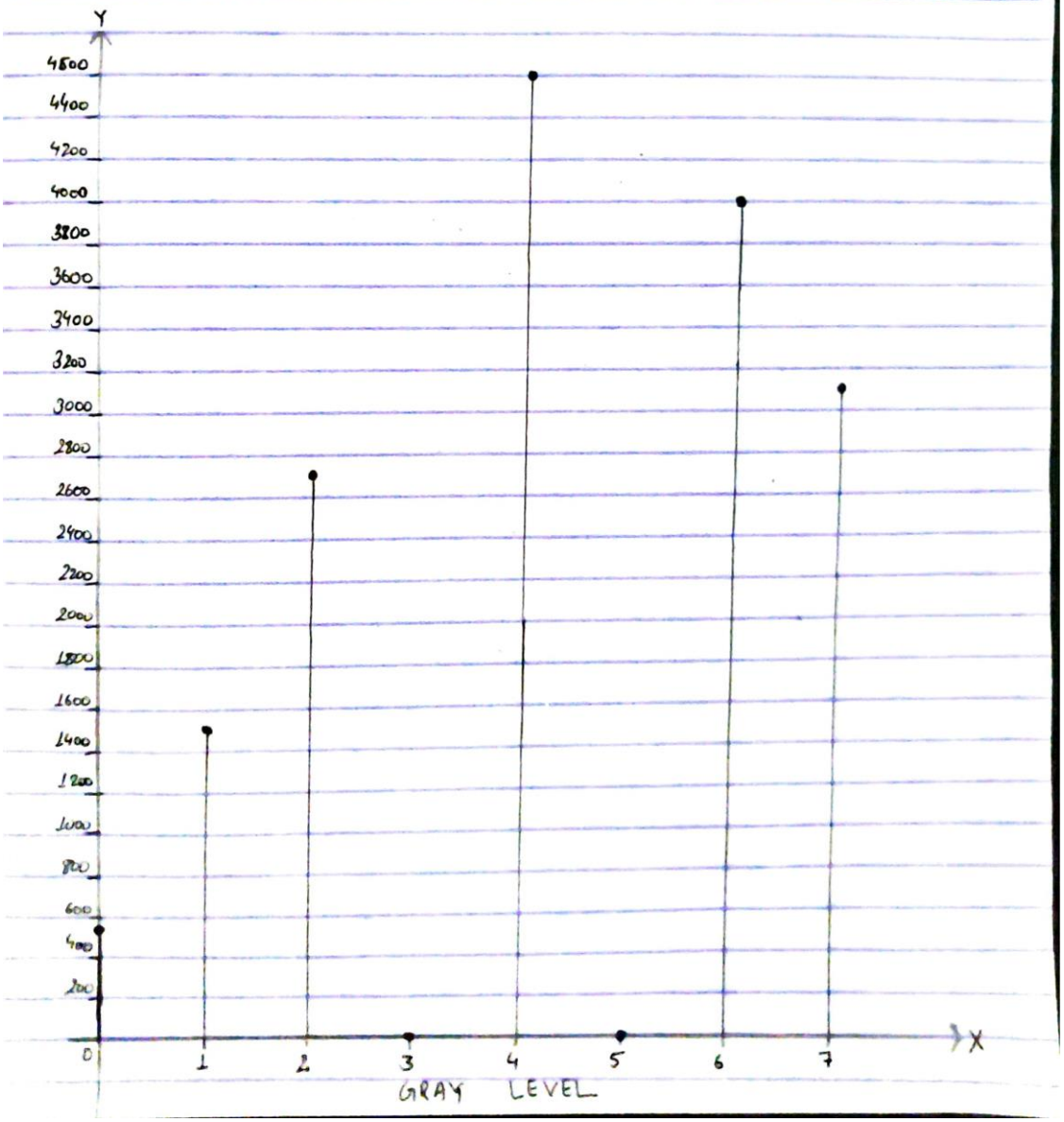
7 ⇒ 3100

Total ⇒ 16,384

Hence, the new equalized version of this histogram is verified.

c) Plot the equalized histogram.

Gray Level	0	1	2	3	4	5	6	7
New No. of Pixels	584	1500	2700	0	4500	0	4000	3100



DISCUSSION AND CONCLUSION

From this lab, we come to learn that using various operations done through histogram to enhance images. We performed histogram equalization, both global and local and checked the variations on the results. We also performed histogram matching but wasn't able to see the results on Octave. Histogram equalization showed us how to enhance the contrast of a dark image but also showed us that sometimes if the histogram is already distributed on all the range of levels, the enhancement is not very effective and for that we use local histogram equalization with our own values.