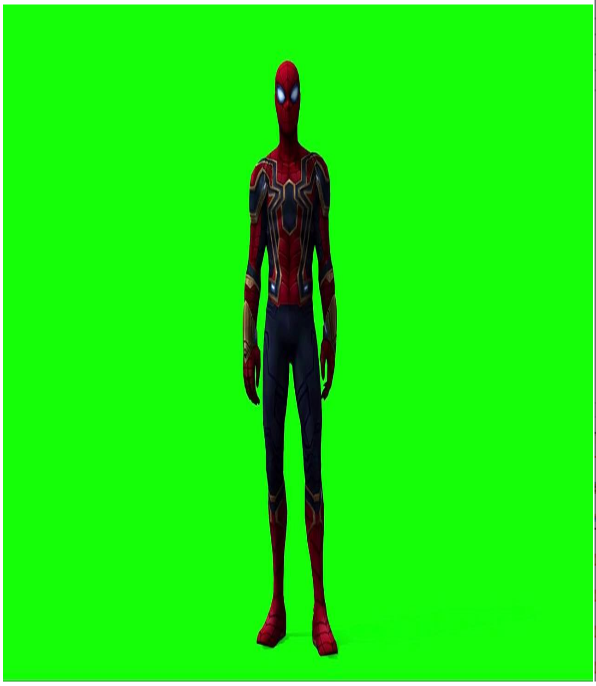
DIP Assignment 1

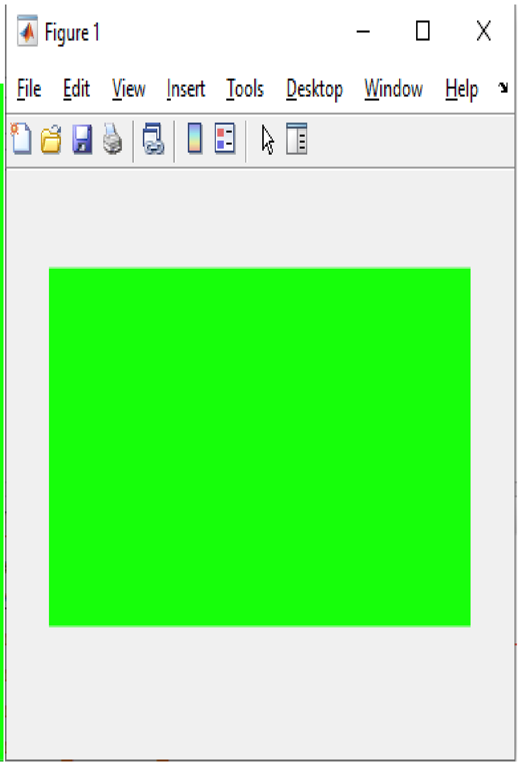
Harsh Sharma

20171157

Q1) To find the most occurring color, we will count the occurring of the pixel using count array and after that we will replace the pixel value with the max occurring pixel so that we will get the entire image consist of only that pixel. And to reduce the error we will use binning i.e., assume 0 to 3 pixel value to bi map to 0 , 4 to 7 map to 1 etc.

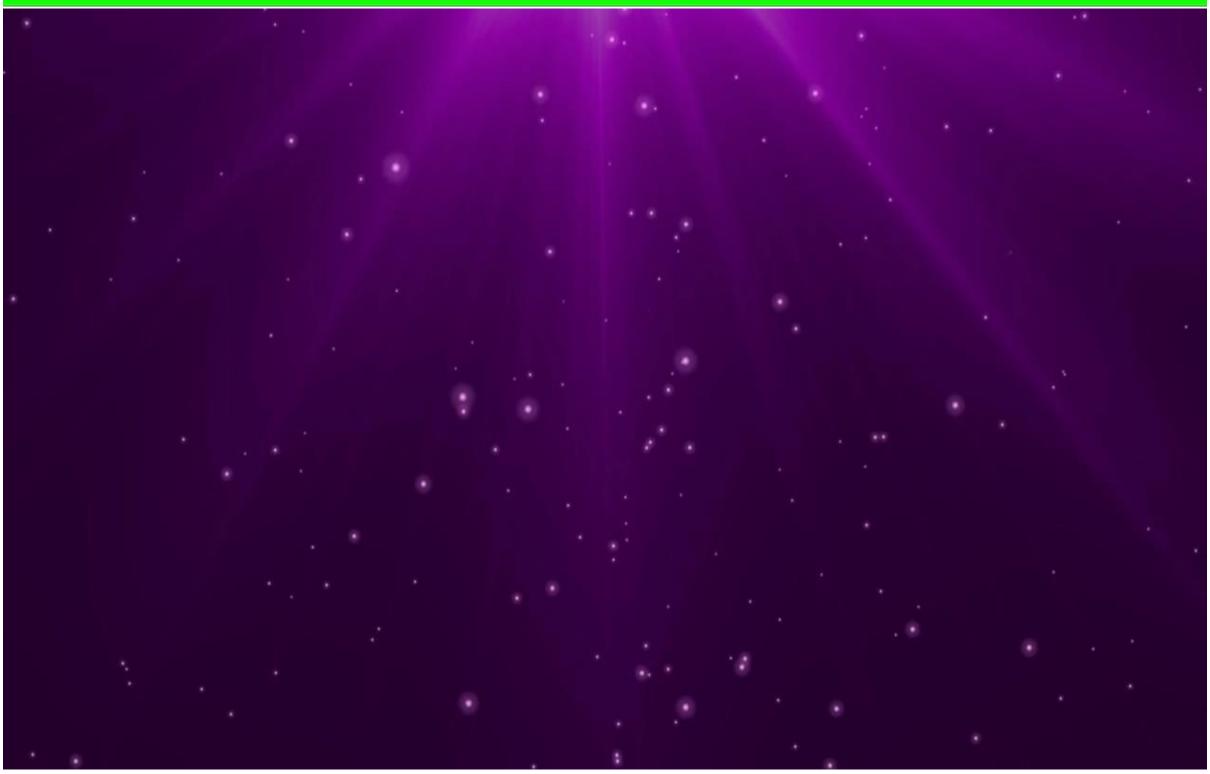
Given Image: Out Image:



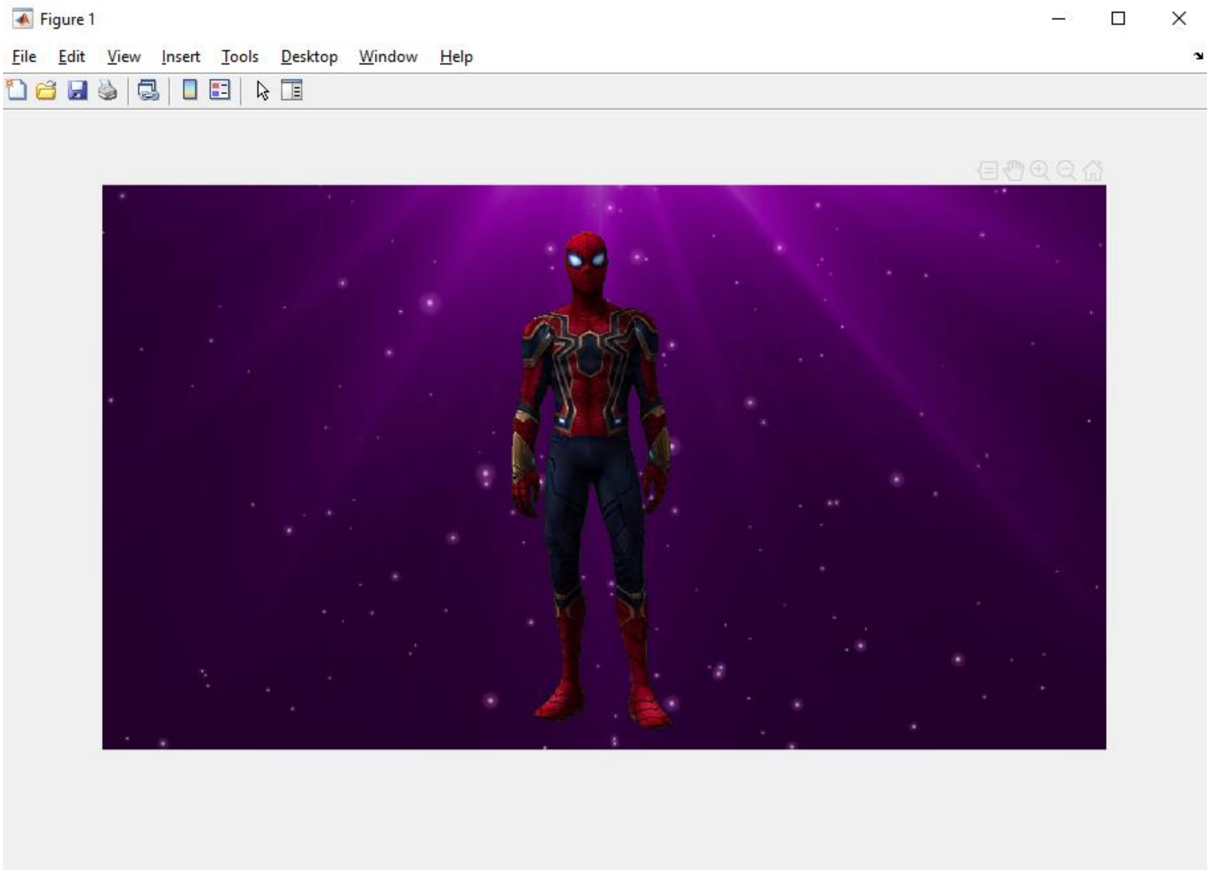


Now in the next part we have to merge 2 images so in this part we will use the 1st part and after getting RGB most occurring pixel we well use the square error to replace the pixel of first image to the other image, we are using square error function because as we can get the some shadow of the object on the background and we want to replace that so to remove that we use square error function.

Input Image:



Out Image:

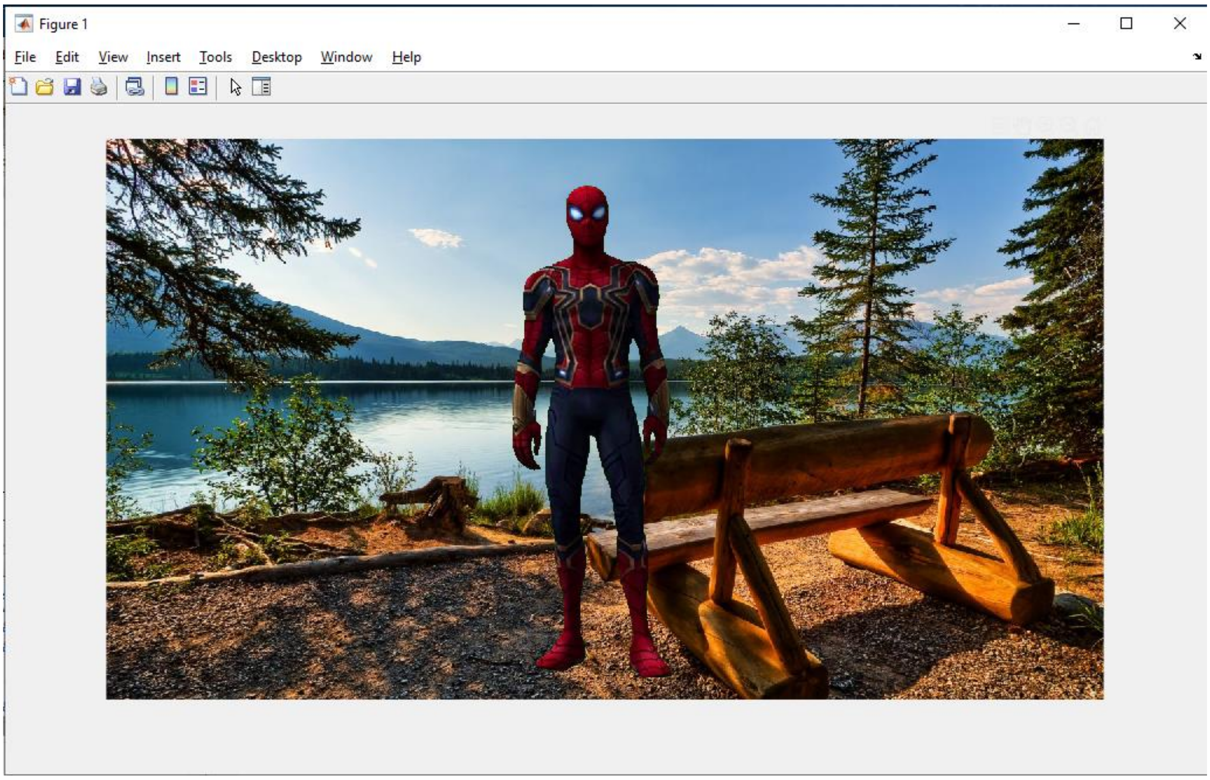


As we can see the green color is completely removed.

With other background,

Input Image:

Out Image:



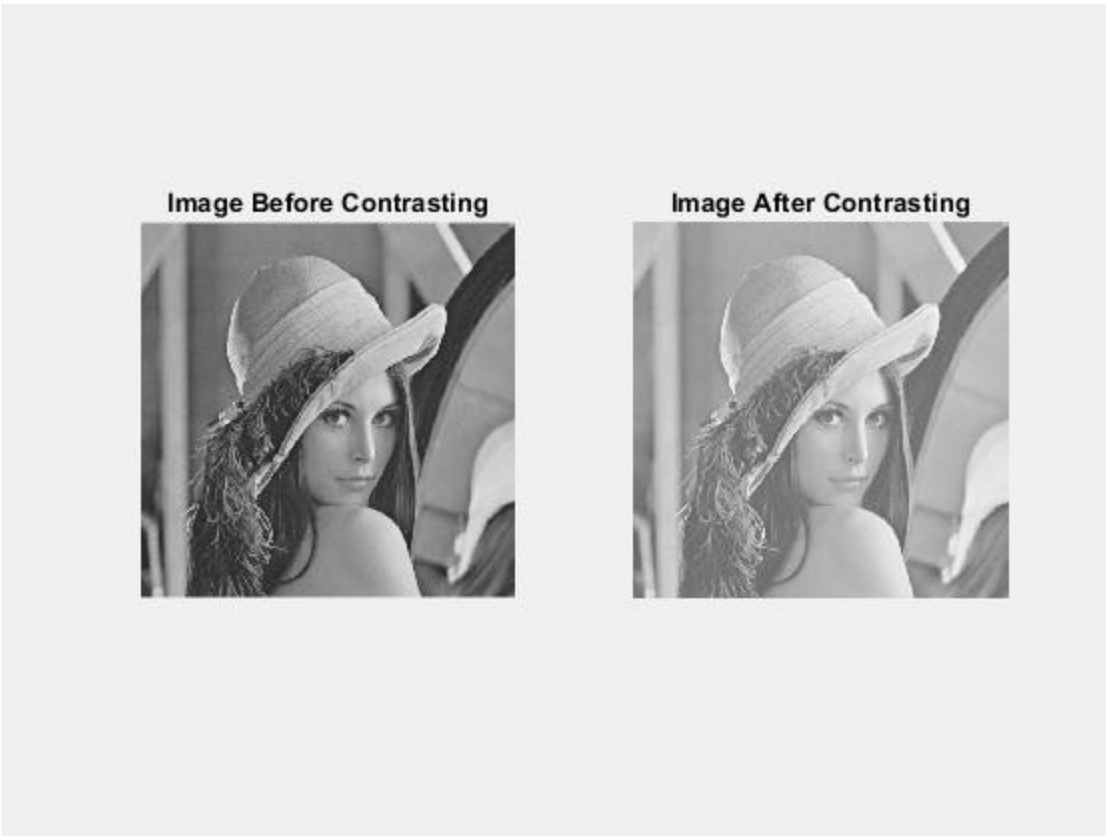
Q2) In this question firstly we have to do linear contrasting with the input as img, a,b i.e., img is the given image a and b are the min and max intensity of the final image so to do that firstly we convert the intensity of the given image to (0 to 1) and than we map the intensity in the interval [a,b] by using liner transformation.

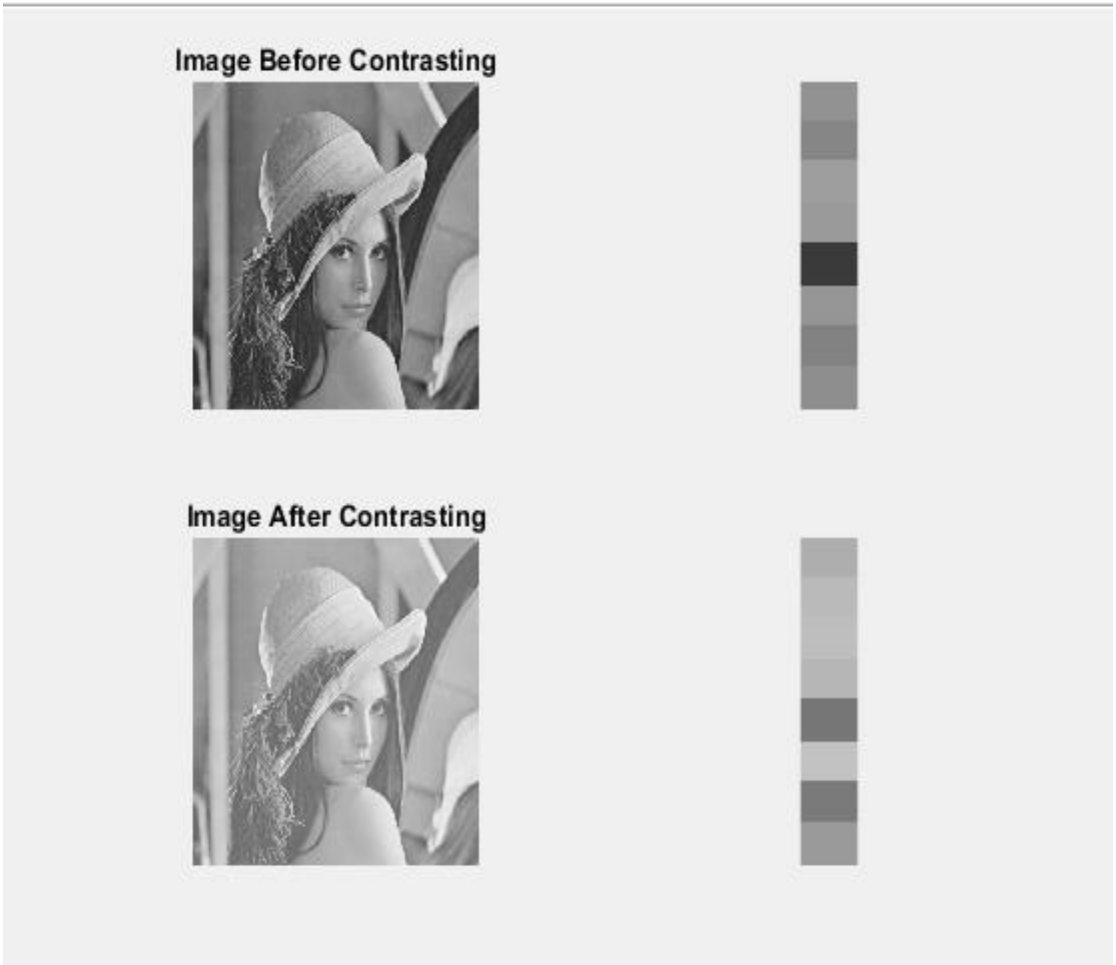
The formula that we use in this is = a + (img - min) * ((b-a)/(max-min)).

And in the next part we have to display the color bar also consists of only k frequnet colors. And to do that we again use the count array and then we build an matrix of 100x20 and divide this matrix into k segments and then map the this matrix elements with the k most occuring pixels. And in this also we will use binning to reduce the error.

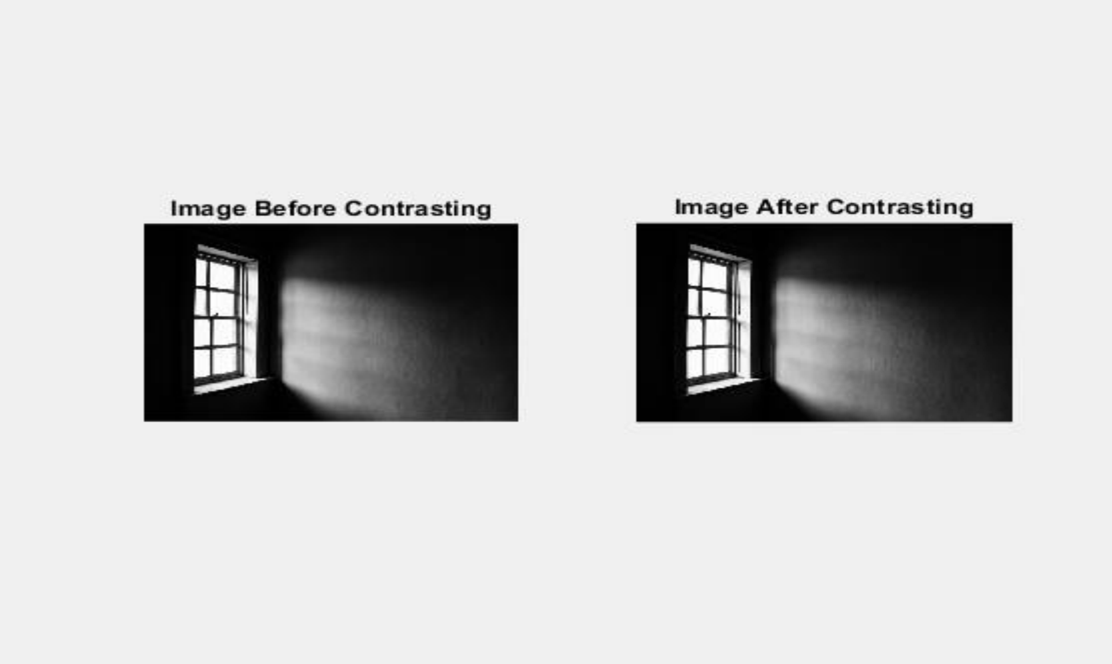
So if a =0 and b = 255, k=6, then,

If a = 100,b=255,k=8, then,





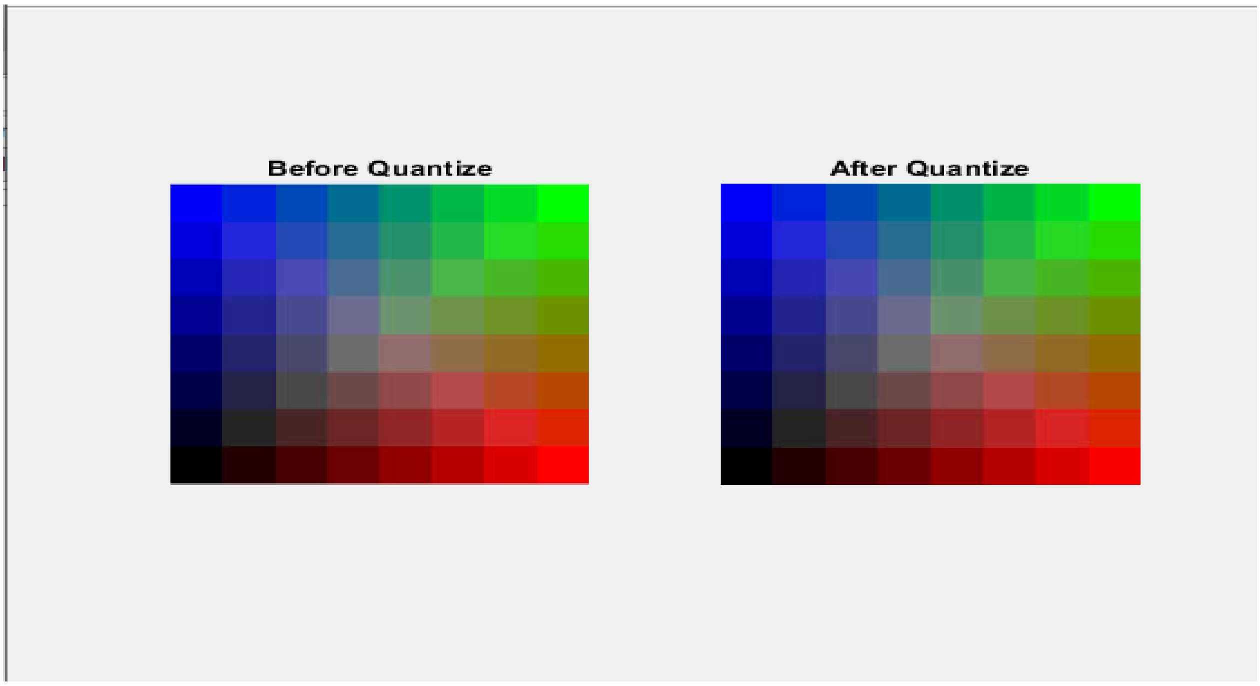
And for the 3rd part, if we have the image that have min and max intensity not equal to 0 and 255 initially or in other words if have 0 value shown on histogram of the image then if we do the linear contrasting from 0 to 255 then we get the image which will be differ from the initial image but if we have the some non zero value on the 0 and 255 intensity value then if we do linear contrasting from 0 to 255 but this time the final image we will same as the initial image because if we see on the formula max – min will going to be equal to b-a so in this case it will failed.



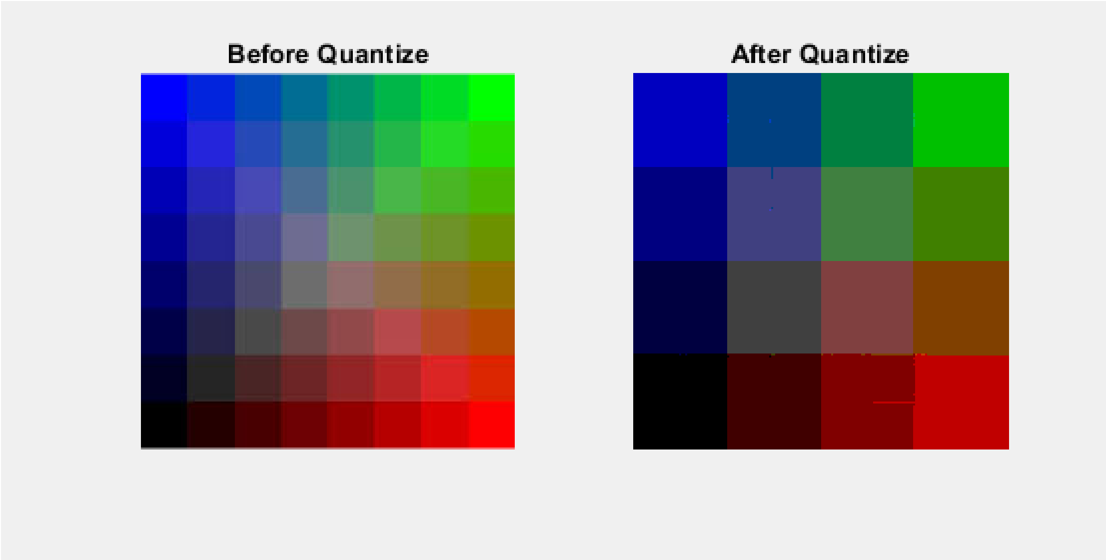
Q3) For Bit Quantization we will do

If we increase the bits then the color increases.

When k = 6, output will be,

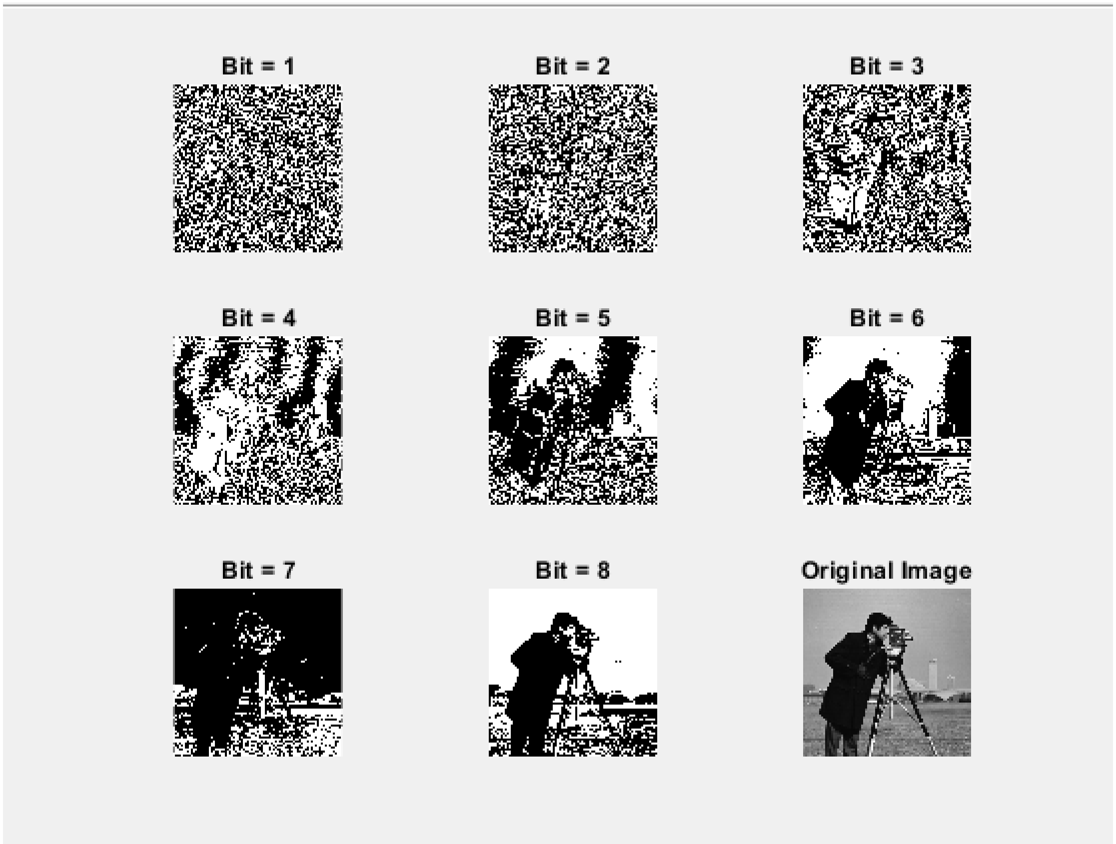


When k=2,

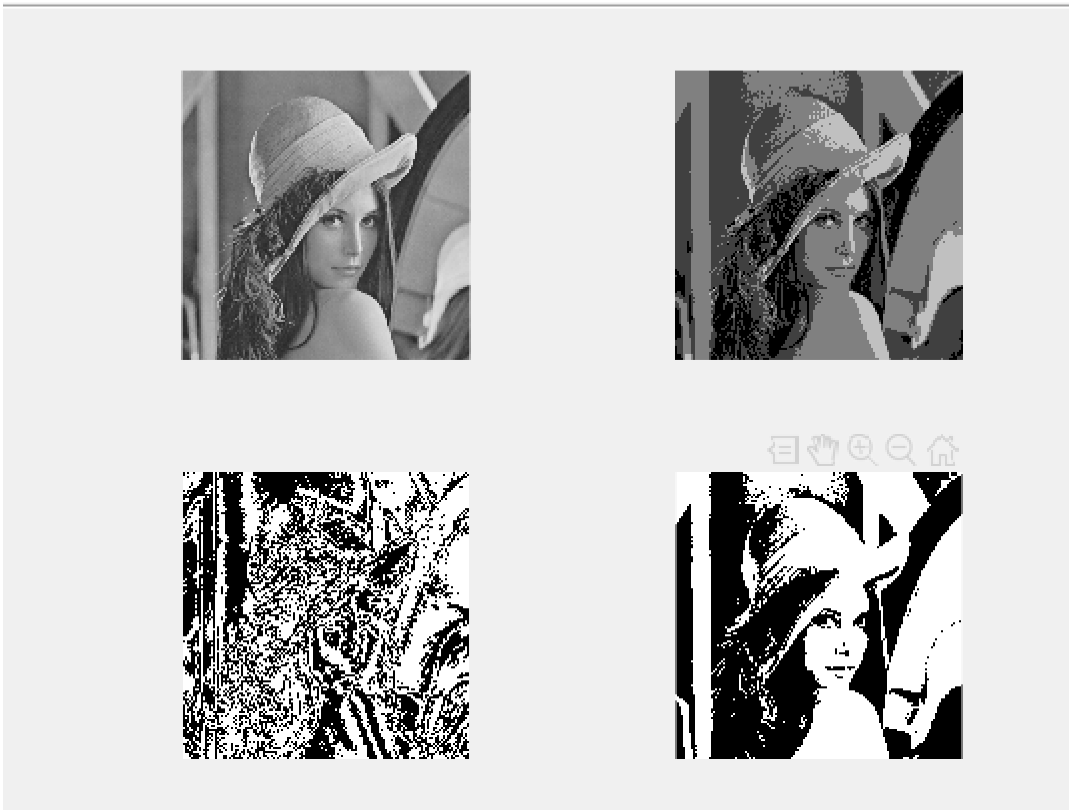


Now to display the bit planes of an image from bit =1 to bit = 8

We simply take logical AND with 2^(bit-1) because the image is the combination of all the 8 bit planes.



And for the 3rd part we get lena1 by doing bit plane with bitplane = 5 , lena2 by doing bit optimization with k=2 and lena3 by bit plane 8.



Q4) To create the negative of the image with the max intensity as maxi we firstly do liner contrasting from 0 to maxi because we want the max intensity of the final image to maxi and after that we will do img = maxi-img, it will give the negative of the image with given specification.

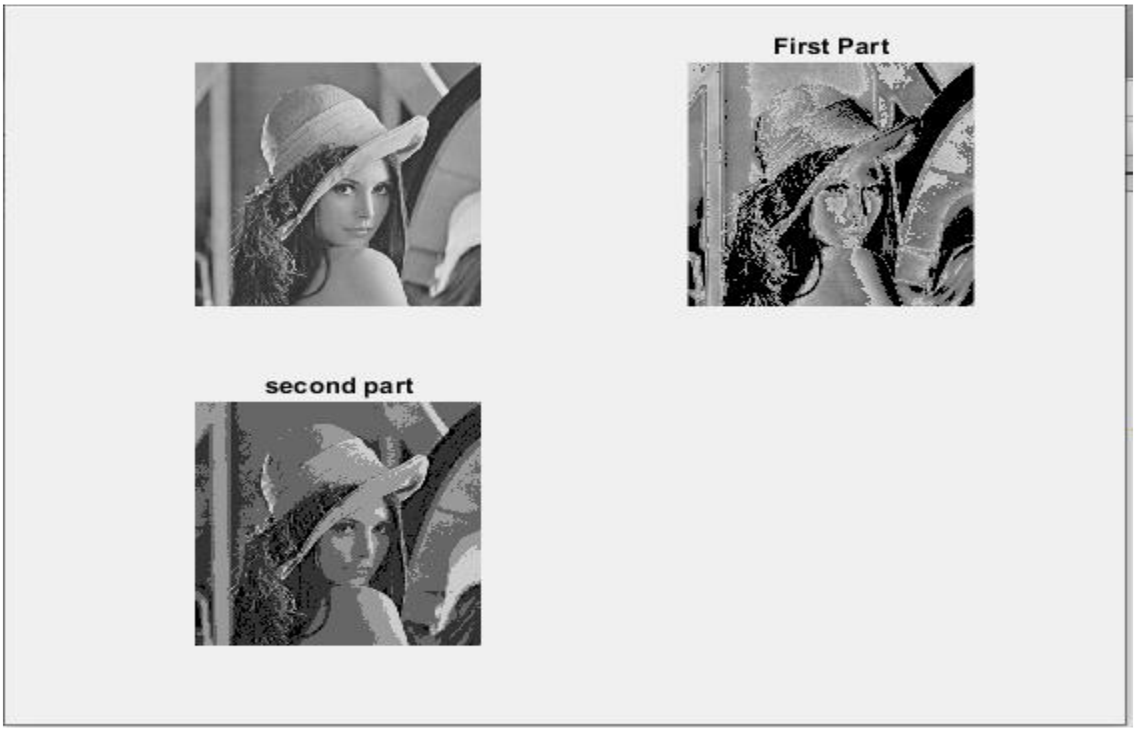
Taking img as lena.jpg and maxi as 255,

Now for the gamma tranformation we use the equation S = A* (r^y)

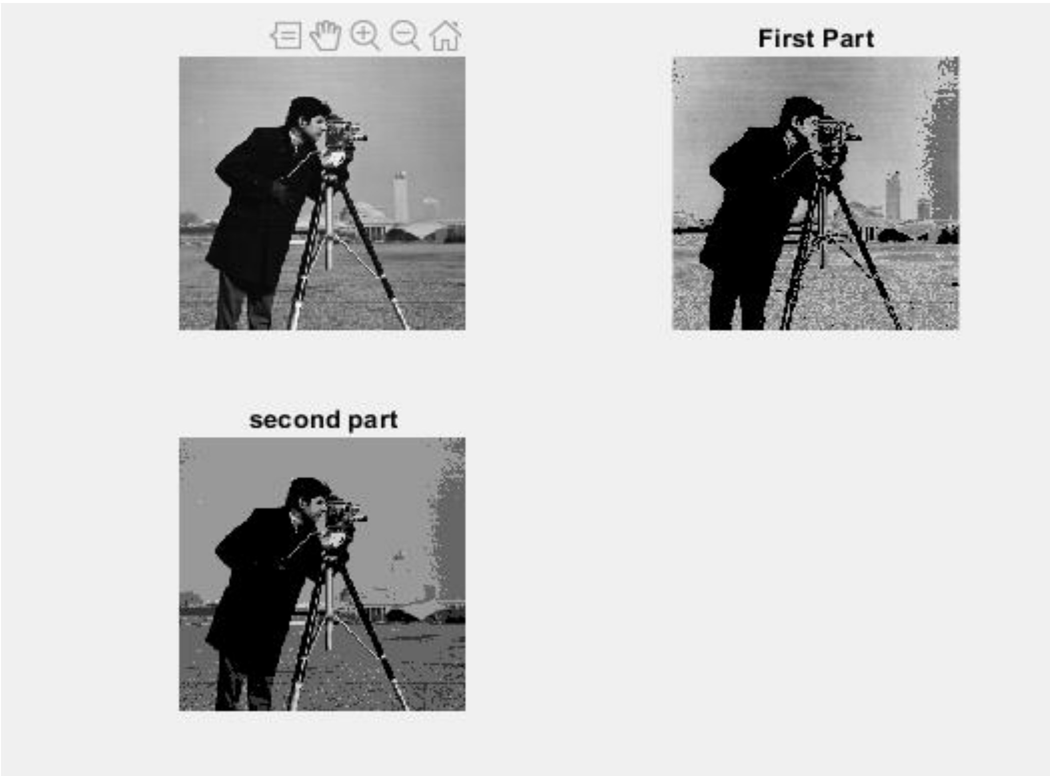
Here y is gamma , r is the given image and A is the positive constant.

Now for the piece wise transformation we will do, we firstly check the pixels with have the intensity in the range of a*255 and b*255 and if that was true then wec replace the intensity with the val = k1*intensity + k2*255. We are multipying a, b,k2 with 255 because we want to map the 0 to 1 graph to 0 to 255.

If the image is lena.jpg then the output will be,



And if the image is cameraman.png then,



Q5) For the first part, setting MSB to 0 will lead to reduction in the number of gray scale levels i.e., from 2^8 to 2^{8-i} where i is the number of bits that are 0, and as the size of the image remains same, then the frequency for the lower intemsity increases and then the image will apear darker.

2.

If we set to zero the LSB in Bitplane to zero . Therefore, we are generating integer numbers that are multiples of 2*i. So, we are representing all the integer values in the range [n∗2i,(n+ 1)∗2i−1], n= 0,1, . . . ,28−i−1 by the number n∗2i. We can think of the situation as quantizing the intensity values using 28−i bits because in terms of the gray-levels we reduce the number of intensities from 28 to 28−i.Therefore, the histogram will reduce the number of intensity levels and since the size of the image does not change, then the histogram will show an increment in the number of pixels at the intensities n∗2i, n= 0,1, . . . ,28−i−1

In the second part if we set the LSB to 0 then we have all the intensity with even values so we are representing all integers values in the range of [$n*2^i$, $(n+1)*2^{i-1}$] for n = 0 to 2^{8-i}. Now because of this the no. of gray levels decreases so the histogram will reduse the number of intensity levels and also since the size of of image does not change, then the histogram will show some increment in the number of pixels at the intensity of $n*2^i$, n = 0, 2^{8-i-1} .

In the 3rd part,

To transmit 512x512 image over 56k baud link

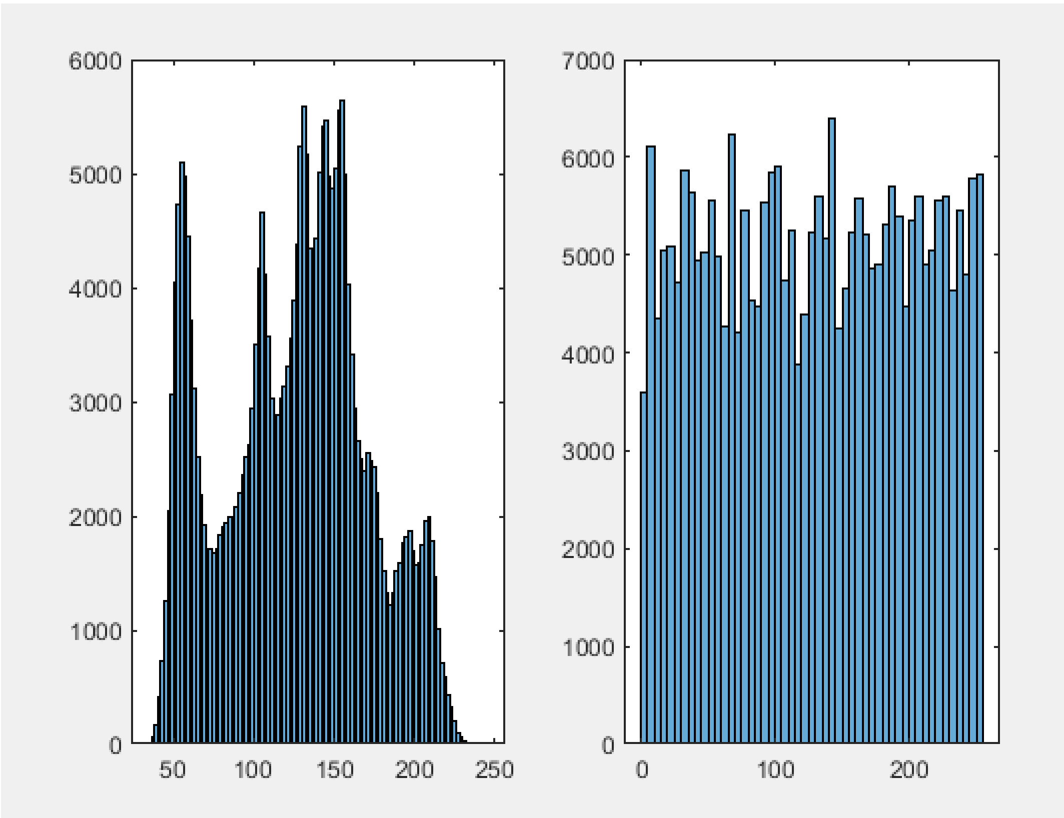
Time Required=512 x 512 x (1+8+1)/56 x 10^3=46.8s

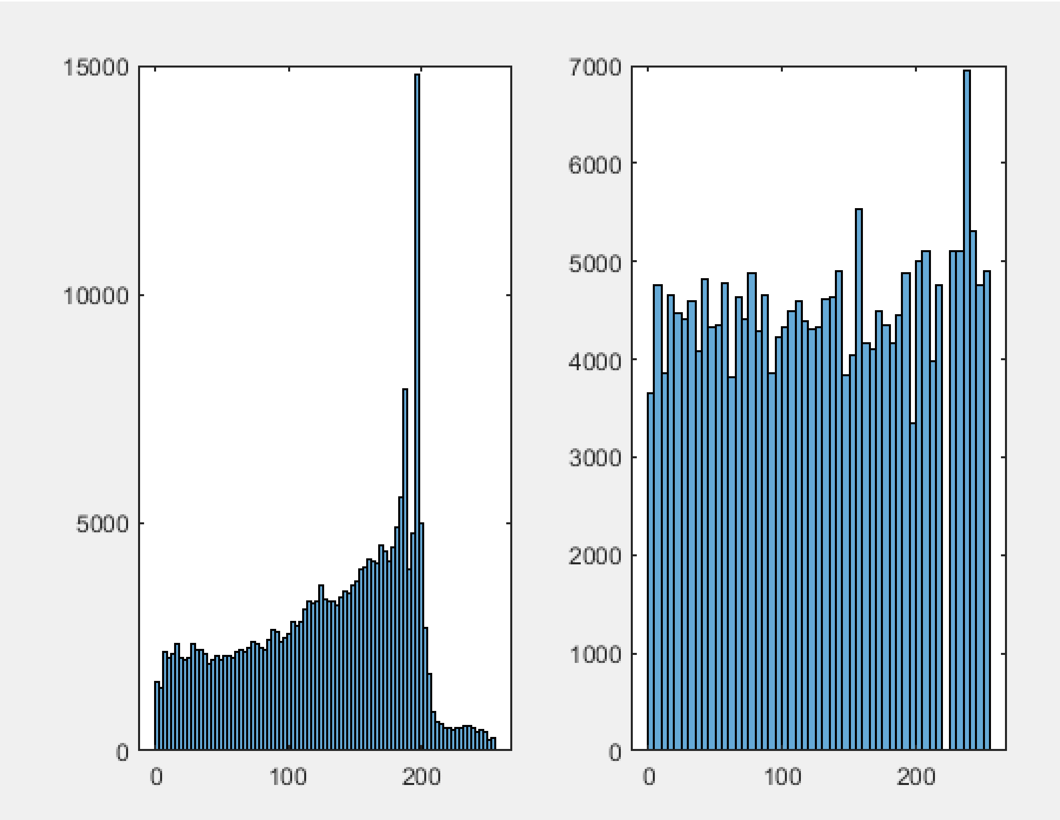
To transmit 512x512 image over 3000k baud link

Time Required=512 x 512 x (1+8+1)/56 x 10^3=0.87s

Q6) For the grayscale image, we count the number of times each intensity occurs, then find the cdf of the intensities, and as cdf is between 0 and 1, we multiply it by 255, to bring the intensities in a proper range, and then rounding off to make it an integer.

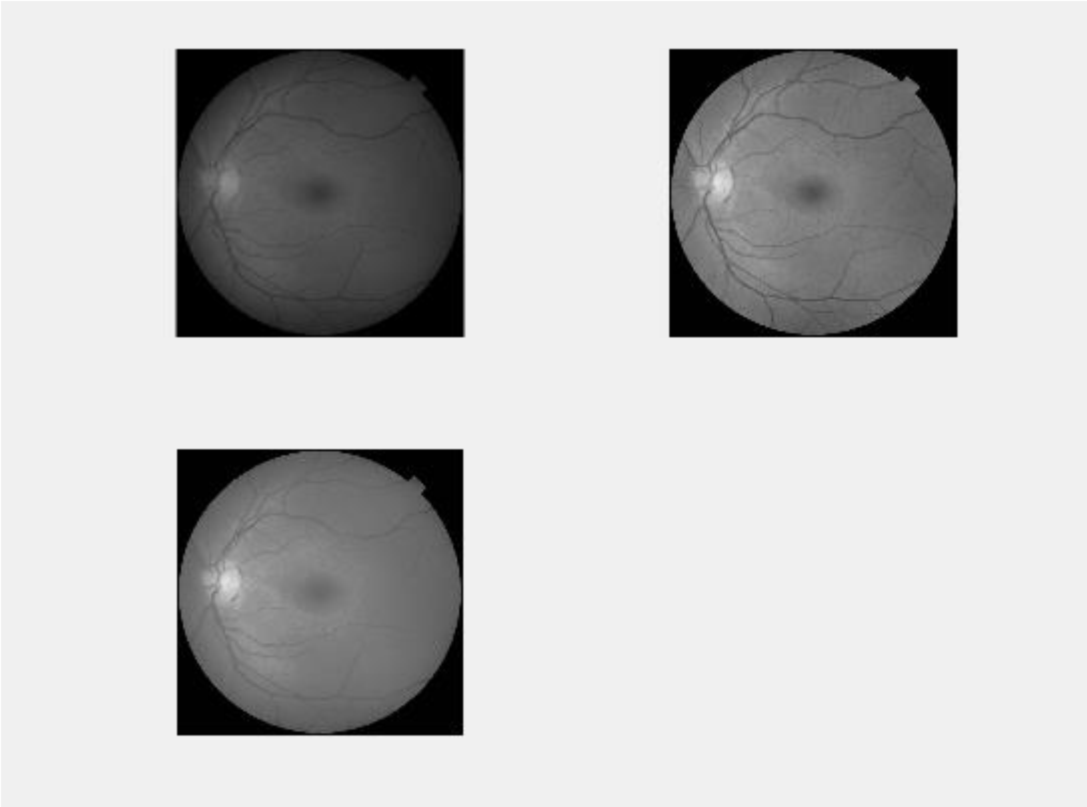

Original Image
Histogram equalization

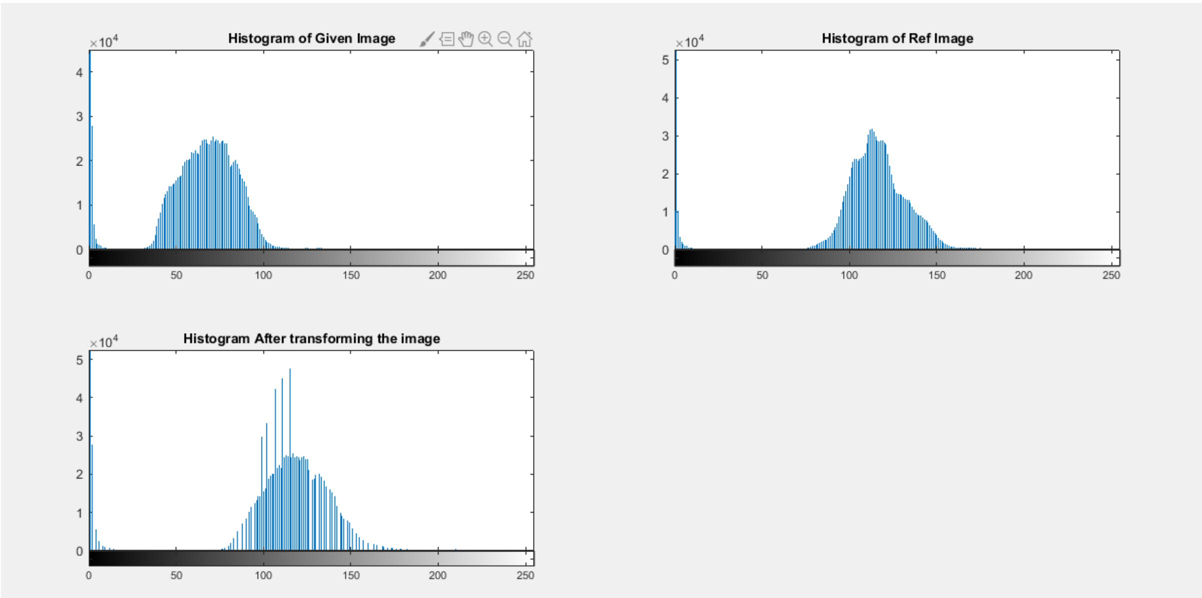

6000
5000
4000
3000
2000
1000
0
50
100
150
200
250
7000
6000
5000
4000
3000
2000
1000
0
0
100
200



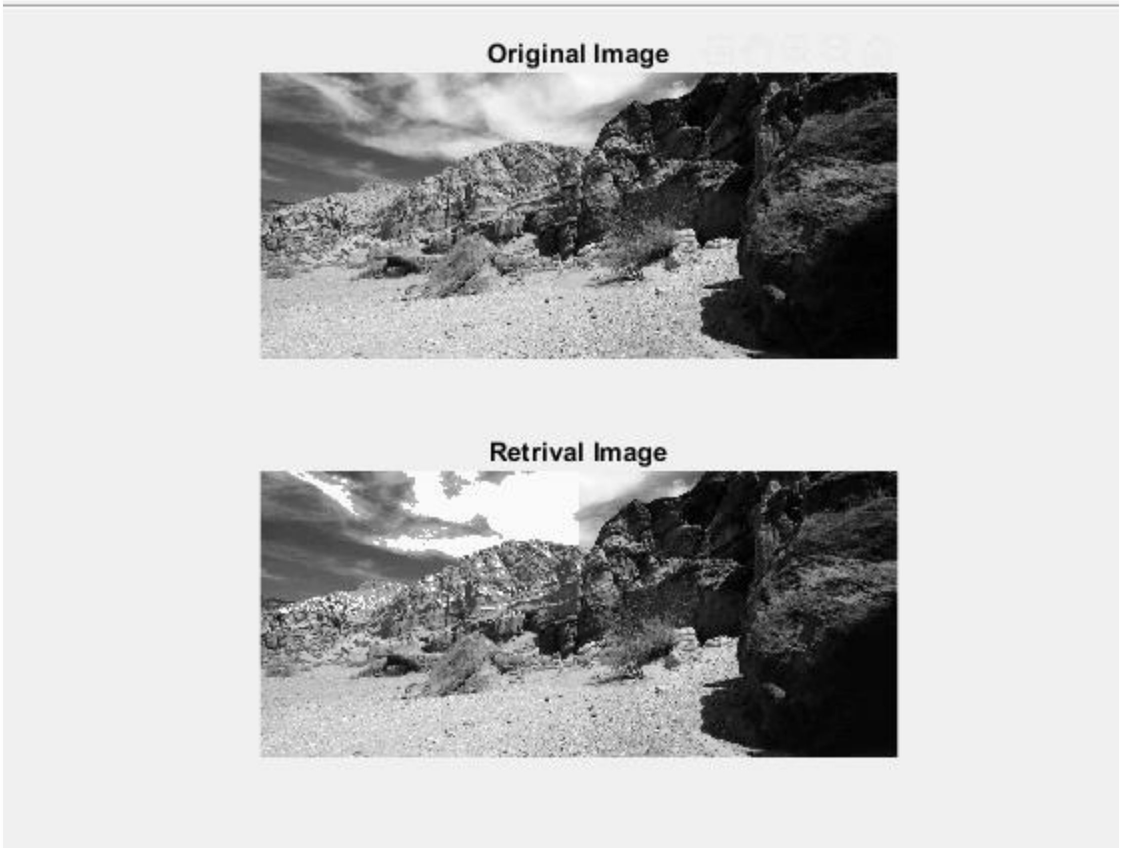
By doing histogram equlization the image can become noise free or can become noisy it depends on the input image like if we have bright image which is good and noise free and if run hist equaisation then it might become dark and can have some noise.

And for histogram matching,

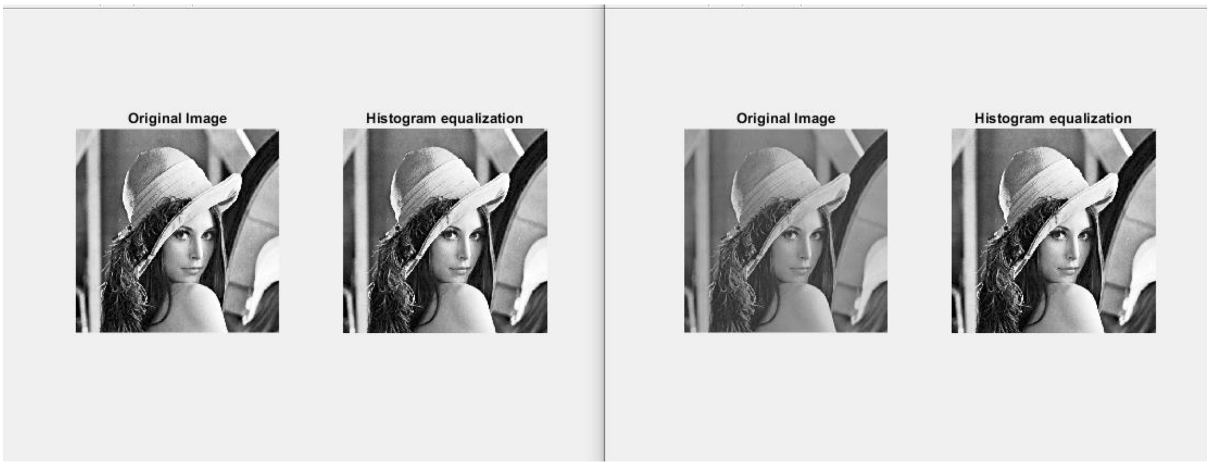




And for the retrive the final image, we firstly divide the original image into 4 equal parts and then do hist matching with the corresponding parts and then concantenate it .



Q7)



If we run hist equalization for the first time then the image become smoother but if we run again with the output image we see that there is not much difference between the out1 image and out2 image because we already equalise the histogram so there is no sense to run again hist equalization on that image.

Similar Histogram leads to no difference in the intensity of the image.

Dark to Bright make the darker image to some brighter image .

Bright to dark.