DIP Assignment 1 Report

Siddharth Gaur 20171198

Q1. **Pixel Manipulation**

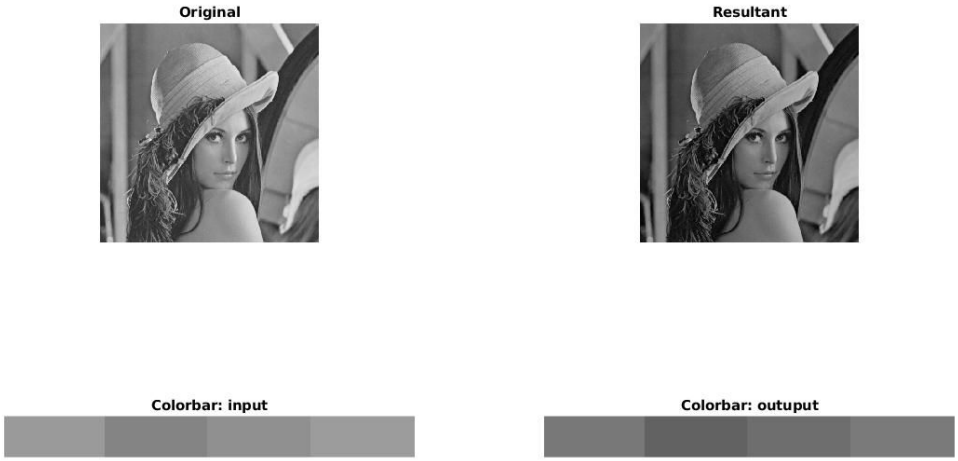
The function ‘maxColor’ gives the most frequently occurring or the dominating color of the input image. It also gives the frequency of that color. ‘maxColor_call1’ is the program used to call this function for ‘fg.jpg’.

‘mergeImage’ merges two images, A and B, it is called by the program ‘merge_call2’ which calls the function for different foreground and background images.

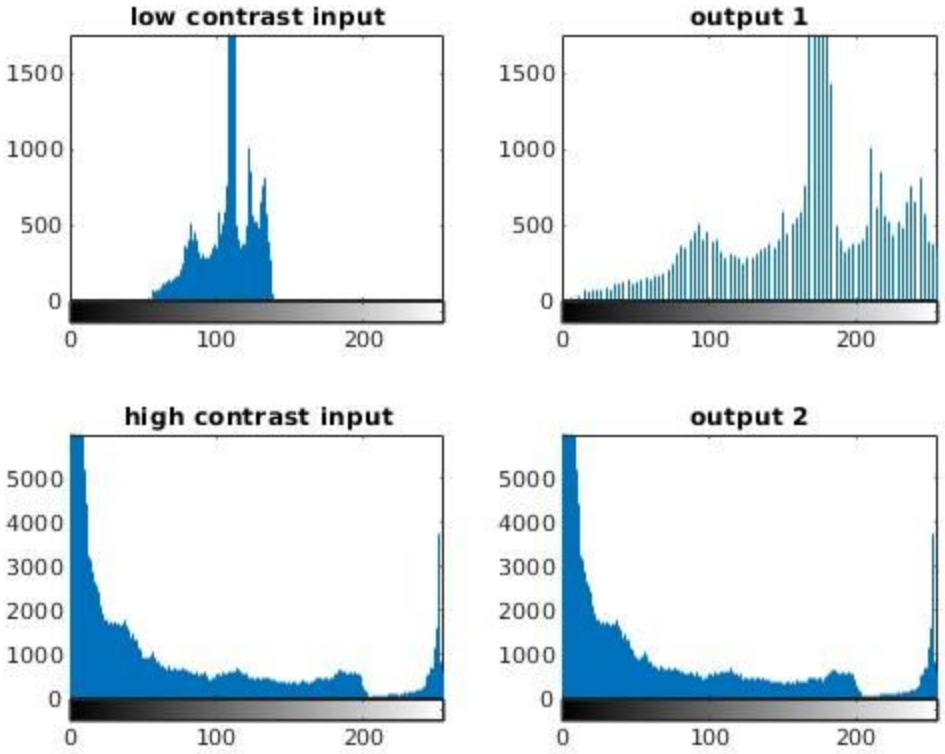
Q2. **Contrast Stretching**

‘linContrastStretching’ function takes a grayscale image ‘im’ and maps its maximum and minimum values such that all the values lie in [a,b]. The program ‘call_1’ takes a grayscale image and displays the input and output images side-by-side along with the colorbars of the corresponding images which are obtained by another function ‘colorbars’.

The reason that the processed image has different colorbars is that the input image had maximum and minimum intensity values which were not mapped to 0 and 255 but the resultant image’s intensity values are mapped to 0 to 255 which increases both the bright and dark part in the image.



On testing with different images we observe that for the same values of ‘a’ and ‘b’, say 0 and 255, the input image with lower contrast changes more than the input image of high contrast. This is due to the fact that the histogram of high contrast image is more spread out as compared to the lower contrast image, hence its values doesn’t change much upon mapping to [a,b].



low contrast input

output 1

high contrast input

output 2

Q3. **Bits Manipulation**

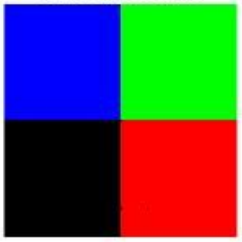
A) Function ‘BitQuantizeImage’ which takes an 8-bit image im and k, the number of bits to which the image needs to be quantized to and returns the k-bit quantized image.

original

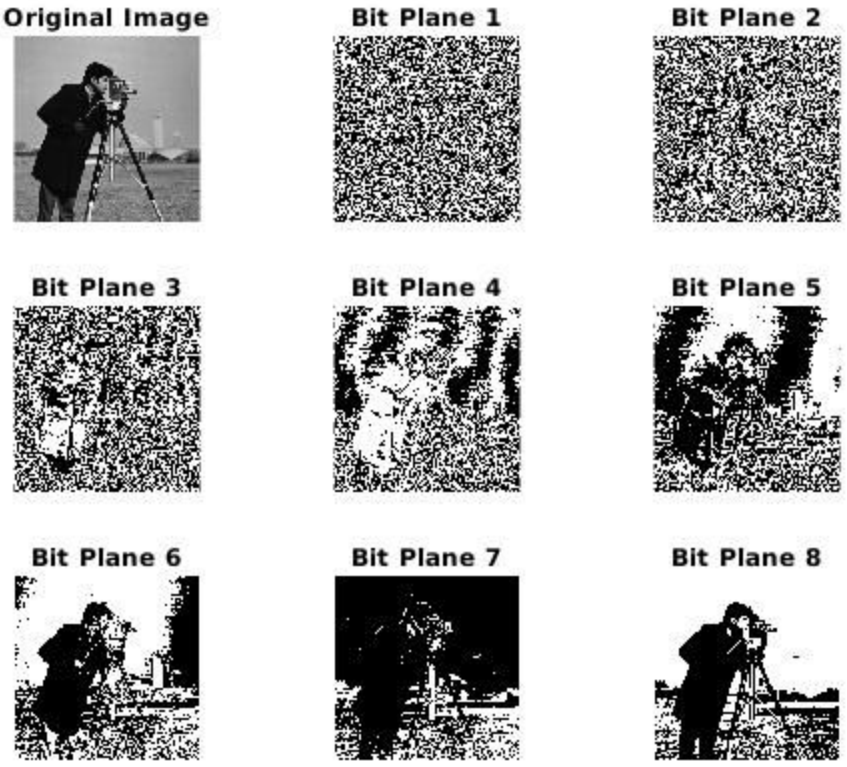
4 bit quantize

2 bit quantize

1 bit quantize



B) The program ‘bitplane’ displays the different bit planes of an 8-bit grayscale image ‘cameraman.png’.



C) The operations done on the given images are as follows:
‘lena1.jpg’ - 6th bit plane of the image ‘lena.jpg’
‘lena2.jpg’ - 2 bit quantized version of image ‘lena.jpg’.
‘lena3.jpg’ - 8th bit plane of the image ‘lena.jpg’.

Q4. **Intensity Transformation**

A) The function ‘negative’ generates the negative of the given in input.
B) The function ‘gammaTransform’ gives transformation for gamma = 0.5, 0.9 and 1.2 .
C) The function ‘piecewiseTransform’ gives transformation for the values of k1,k2,a and b given in the form of graphs.

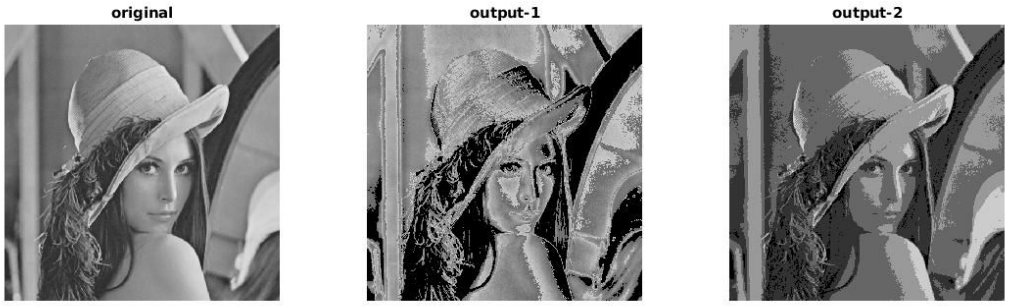
A)


original
1 bit negative
2 bit negative
3 bit negative
4 bit negative
5 bit negative
6 bit negative
7 bit negative
8 bit negative

B)


original
gamma = 0.5
gamma = 0.9
gamma = 1.2

C)



Q5. **Miscellaneous**

A) Making the MSB bits of the bit plane to zero reduces the number of gray levels from 2^8 to 2^{8-i} , at the same time all the pixels greater than 2^{8-i}-1 are lost. Histogram for values less than 2^{8-i}-1 remains the same as that of the original image and the number of pixels at 2^{8-i}-1 will increase in general as the size of the image remains the same. It can be compared with 'negative of an image' as in the case of negative, the values are subtracted from the largest pixel value.

B) If we make the LSB bits of a bit plane zero, the resultant image is a multiple of 2^i, making all values even. We can also think of it as quantizing the intensity values using 2^{8-i} bits as the size of image stays the same and we reduce the number of gray levels from 2^8 to 2^{8-i}. Therefore the number of intensity levels in histogram will reduce, then the histogram will show an increment in the number of pixels at the intensities $n*2^i$, n = 0,1,...,...,2^{8-1}-1.

C) Baud rate = 56k

Size of data = 512x512x10 , 10 = 8 bit(information) + 2bits(start and stop bit)

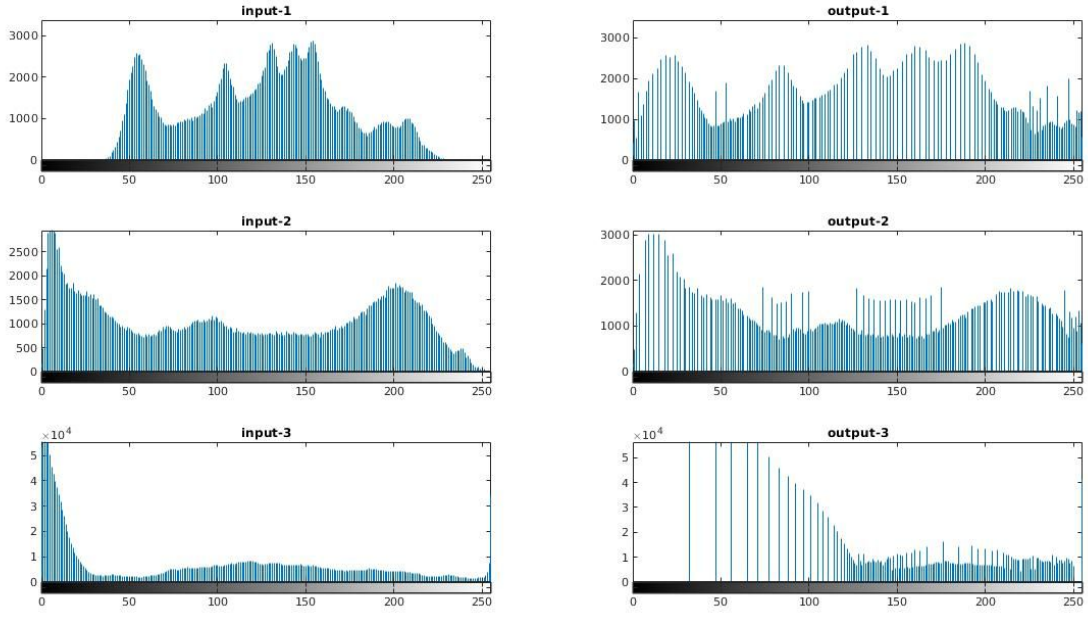
Time required = (512x512x10)/(56x10^3) = **46.8s**

Now, baud rate = 3000k

Time required = (512x512x10)/(3000x10^3) = **0.87s**

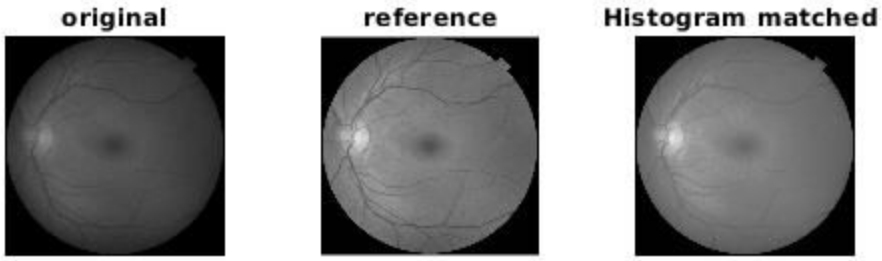
Q6. **Histogram Equalization and Matching**

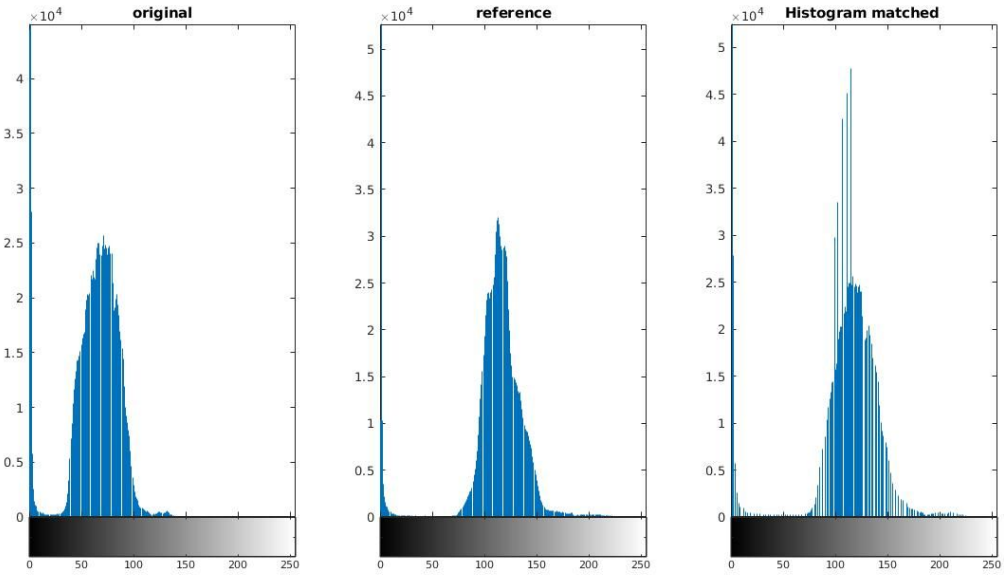
A). Function ‘histEqualization’ takes a grayscale image and applies histogram equalisation on whole image



Histogram equalisation works really well for the 1st and 3rd image, for the 2nd one we don’t observe many changes as it is clear from the histograms that it is widespread for 2nd image while compact in case of 1st and 3rd images, which causes contrast enhancement for 1 and 3 while 2 remains pretty much the same.

C). Function ‘histMatching’ which takes an input image and a reference image and applies histogram Matching on the input image by matching the histogram with that of the reference image.

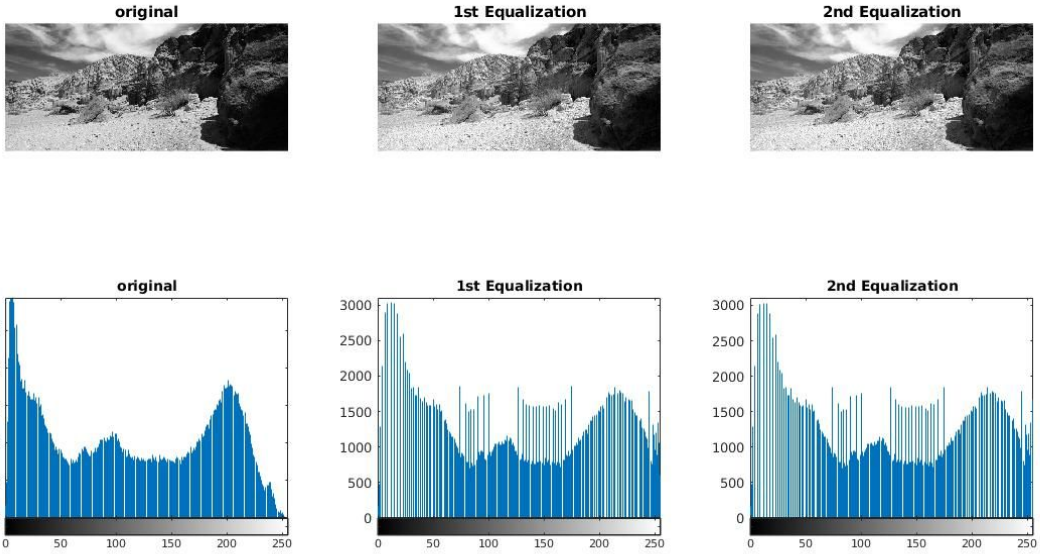




D). part1.png, part2.png, part3.png and part4.png are recombined and to retrieve the original image ‘canyon.png’.

Q7. **Histogram Transformation**

A). Histogram equalisation is applied on an image, then again applied on the resultant image.



B).


original
reference
resultant-1
original
reference
resultant-2
original
reference
resultant-3