EE475 Fall 2020 HW2 **Sezen Perçin**

I. Extracting the bit planes of images (3.4 G&W)

0	1	8	6
2	2	1	1
1	15	14	12
3	6	9	10

Integer values of the pixels are stored by bits and an image can be extracted into its planes that contain the bit values with plane 1 storing the lowest order bit. For this example, I allocated the space needed for plane matrices, performed division and mod operations by 2 to obtain binary representation and store the bits in separated places accordingly. The results can be seen below:

```
plane1 =

     0     1     0     0
     0     0     1     1
     1     1     0     0
     1     0     1     0

plane2 =

     0     0     0     1
     1     1     0     0
     0     1     1     0
     1     1     0     1

plane3 =

     0     0     0     1     0     0     1     0
     0     0     0     0     0     0     0     0
     0     1     1     1     0     1     1     1
     0     1     0     0     0     0     1     1

plane4 =
```

2. Nonlinear Monotonic Point Operations:

Since R is defined as the the range over which the gray-level histogram is non-zero, i investigated the gray level histograms of the both page.

lenahist — 256x1 double

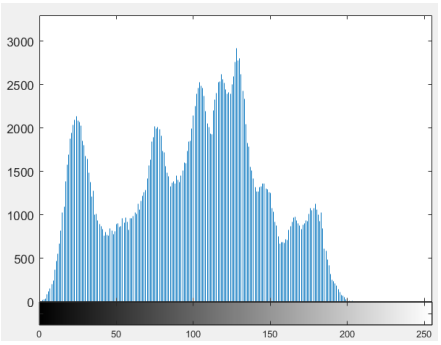
	1
199	50
200	32
201	19
202	14
203	7
204	9
205	4
206	4
207	1
208	1
209	1
210	2
211	0
212	0
213	0
214	0
215	0
216	0
217	0

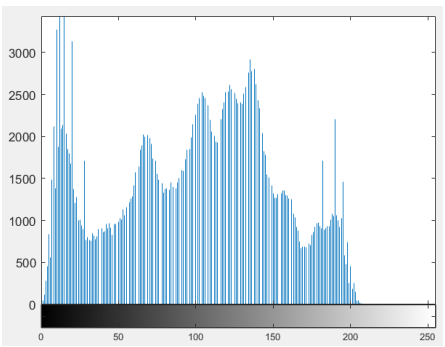
lenahist — 256x1 double

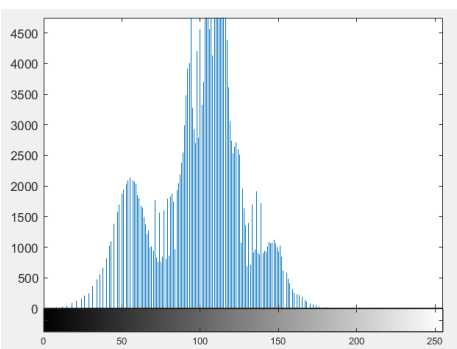
	1
1	1
2	7
3	19
4	27
5	37
6	84
7	123
8	157
9	205
10	248
11	367
12	470
13	555
14	668
15	817
16	1029

As it can be seen on the left the maximum non-zero gray-level for the lena.bmp was 210 and for Gray Bars.png it was 256. The lowest non-zero level fort he lena.bmp was 1 and for Gray Bars.png it was 1. Therefore i find Rlena=210 and Rgraybar=256. The original, f transformed g transformed image (in order) and the histograms can be seen below:









As it can be seen from the imagesand the histograms, f transformation caused image to experience more contrast since it took some of the gray levels and added to other gray levels. This cause,d histogram to have a more uneven distribution and increased the contrast. On the other hand, g transformed image have its gray values cumulated in the middle of the spectrum causing pixels to have similar gray level values. Therefore, it decreased the contrast of the image. The same effect can be seen below fort he gray bar image (original-f-g):



III.Intensity Transformations

For the first part with the log transformation I got the best visual enhancement when c=45. For the bigger values the image got darker and lost quality. For the lower values it was too dark inspect the image. The corresponsing image can be found on the left.

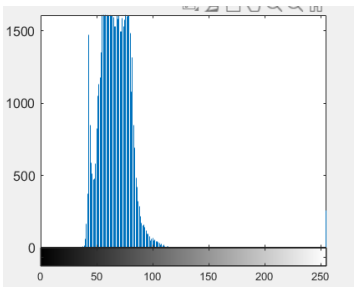
For the second,d image I got the most detailed picture at gama=0.75 and c=12. The more I raised the gama the more contrast introduced to image therefore I balanced them at those values. The corresponding image can be seen on the right.

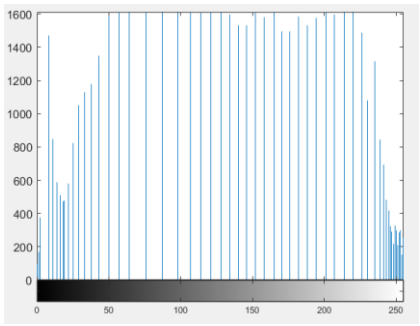
The major difference between them was the existing of the contrast determining element.

IV. Histogram Equalization

a) I wrote the program by calculating the sum of the logical values that indicate the position of the pixels with spesific gray level value.
b) After the calculation of the histogram, I calculated the normalized histogram by dividing values to the total pixel numbers. Then i calculated the cumulative normalized histogram by summing the normalized, histogram values in every step.

The rest was done by assigning every values to its corresponding values on the cumulative normalized histogram. Then those values are multiplied with 255 and 0.5 was added. After performing the floor() operation, the image was equalized. The original image, its histogram, equalized image and new histogram:

The result shows that the more gray level values are distributed, the more color is added to the image and therefore we were able to perceive the image as more clear and somewhat bright.

V. Image resisizing

To achieve the goal of resizing the image, first I shrunk and project the 400x400 image on the 256x256 sized image plane . By the ratios between the dimensions, I determine the fractional coordinates of the pixel in the new image. Then for every unknown pixel 4 nearest pixel in the original image is used and by performing 2 interpolation operation (on y and x axis) the unknown value is estimated. The original and new image in order: