Digital Image Processing HW1 Report

- ## Probelm1

(a) To turn color image into grayscale, I simply adjust all three color channel to the same value for each pixel. To maintain the brightness, I set this value as the average of the value of three channels, i.e. for a pixel with color channel (b, g, r), turn it into (A, A, A) where A = (b + g + r) / 3.
Original Image:

Output Image:

(b) To perform horizontal flipping, I simply swap the channel value of a pixel and that of the corresponding pixel on the other side horizontallly. Explicitly speaking, for a pixel at coordinate (x, y), I swap its channel with pixel at coordinate (W-1-x , y) whrer W is the width of the image.
Original Image:

Output Image:

- Problem2

(a) For a color channel (b, g, r), I set the corresponding pixel of result image into (b/5, g/5, r/5).

(b) For a color channel (b, g, r), I set the corresponding pixel of result image into (b*5, g*5, r*5).

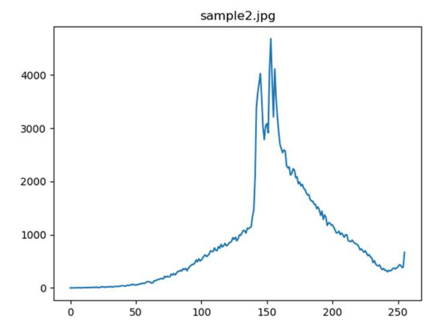
(c) The original image and result image for (a), (b) is shown below.

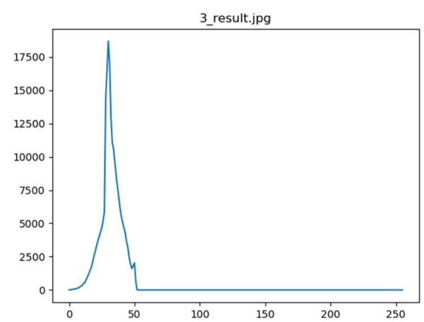
Original Image:

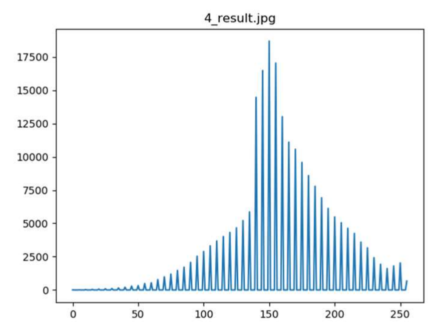
Output Image(for (a)):

Output Image(for (b)):

The histograms are shown below.







Histogram for 3_result.jpg is shifted left, since its brightness decrease uniformly. Histogram for 4_result.jpg is different from original image's, since the implementation of dividing by 5 and multiplying by 5 lose the details. For example, all values ranging in [5, 10) are turned into value 5.

(d) First I compute the cumulative probability function array c[i] and define the following:

c[i]: the number of pixels whose intensity is lower than or equal to i

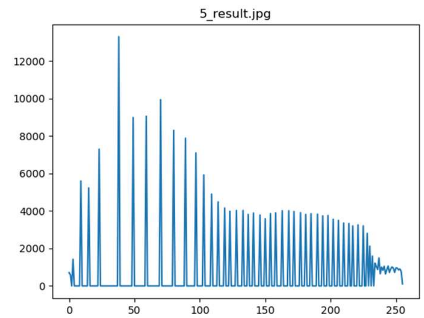
cdf_min: the minimum nonzero value in c[]

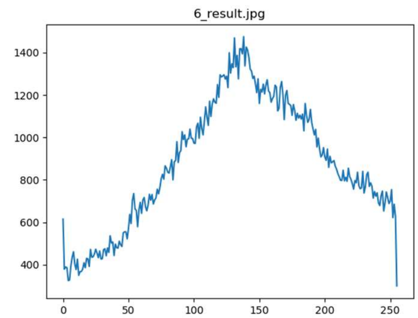
N: the number of pixels(equal to height * width)

Next for any pixel with intensity x, I set the corresponding pixel intensity for result image to be [(c[x] - cdf_min)/(N - cdf_min)]*255. This helps enhance the contrast of image and makes objects clearer.

(e) Local histogram equalization is performed by setting a window for a pixel and do global histogram equalization inside the window. I set the window size to be 20*20.

(f) The histograms are shown below:





Global histogram equalization enhances the constrast in view of all image, while local histogram equalization enhances the constrast in view of only the neighborhood of a pixel image.

(g) The idea for my transfer function is to do two things: enhance the details of the ship on left, and enhance the contrast for the sunlight on right.

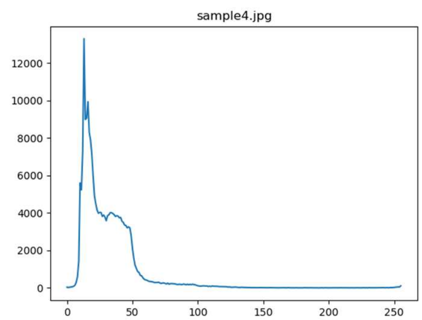
I performed Gamma correction with different power $\gamma_1 < 1$ and $\gamma_2 > 1$ on different range of values. Several parameters were tried, and I found applying Gamma correction with $\gamma_1 = 0.5$ on pixel with intensity < 100 and $\gamma_2 = 3$ on pixel with intensity ≥ 100 help it.

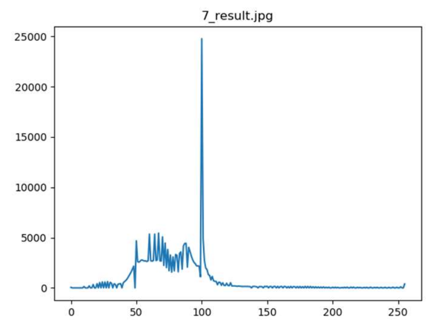
The result image and histogram are shown below.

Original Image:

Output Image:

Histograms:





- Problem3

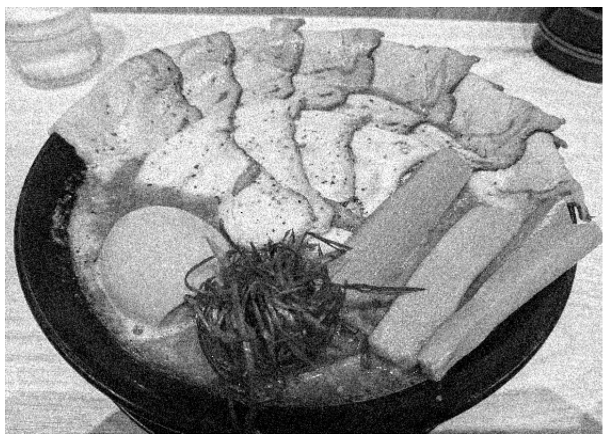
(a) There are Gaussian noise in sample6.jpg, and salt-and-pepper noise in sample7.jpg. Hence I pergorm a low-pass filter on sample6.jpg and a non-linear filter on sample7.jpg.

The mask used for sample6.jpg is as follows:

$$\frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

For border pixel, I copy the center pixel value to extend the image.

Original Image:



Output Image:

For sample7.jpg, I use pseudo-median filtering method. First I extend the image border by copying the border pixel value, and next for any pixel at coordinate (i, j), let its intensity be img[i, j]. The new image intensity at coordinate (i, j), say new_img[i, j] is computed by:

$$MINR = \min(img[i][j-1], img[i][j], img[i][j+1])$$
$$MINC = \min(img[i-1][j], img[i][j], img[i+1][j])$$
$$MAXR = \max(img[i][j-1], img[i][j], img[i][j+1])$$
$$MAXC = \max(img[i-1][j], img[i][j], img[i+1][j])$$
$$MAXMIN = \max(MINR, MINC)$$
$$MINMAX = \min(MAXR, MAXC)$$
$$new\_img[i,j] = PMED = 0.5(MAXMIN + MINMAX)$$

Original Image:

Output Image:

(b)

The PSNR of 8_result.jpg is 8.741650444290936.

The PSNR of 9_result.jpg is 15.749799485485937.

Although the PSNR of 9_result.jpg is higher, it looks like there are still a lot of noise on 9_result.jpg. The value of PSNR can only be referenced, not 100% absolutely be used to determine the quality of a processed image.