CS5680/6680 – Fall Semester 2023
Assignment 2 – Image Enhancement in the Spatial Domain
Due: 11:59 p.m. Sunday, September 24, 2023
Total Points: 35 points

Problems:

Read in the image (***Food.jpg***) and save it in an array ***foodIm.***

1. [7 points]
Implement a **Scaling** function to **linearly** rescale (transform) the intensity values of the grayscale input image to new intensity values. The **Scaling** function has two input parameters, **inputIm** and **range**, where **inputIm** is the original grayscale image and **range** is a two-element vector containing the new minimum and maximum intensity of the rescaled (transformed) image (e.g., **scaledIm**). The **Scaling** function has two output parameters, where **scaledIm** is the transformed image and **transFunc** is the transform function, which is an *n*-element vector that stores the newly transformed intensity for each of the possible grayscale intensities (i.e., the first and last elements of **transFunc** are the newly transformed values for the two intensity values in **range**, respectively). Make sure that your function shows an appropriate error message if **range** contains non-integer values, negative values, and a larger first element than the second element.

Note: Both input and output images of the **Scaling** function should be an array with the same size and the same data type uint8.

Design four **Scaling** function calls to rescale the image ***foodIm*** to a new image ***scaledFoodIm***. The first three of these function calls will have three kinds of invalid data in **range** so appropriate error messages will be displayed. The last function call will rescale the image ***foodIm*** to a new image ***scaledFoodIm*** with an appropriate range [newMin newMax] so ***scaledFoodIm*** has a good quality. Plot **transFunc** in Figure 1 with appropriate titles on both the *x* and *y* axes.

2. [7 points]
Implement a **CalHist** function to calculate either the histogram or the normalized histogram of the grayscale input image.

Call the **CalHist** function to calculate the histogram of the image ***scaledFoodIm*.**

Call the **CalHist** function to calculate the normalized histogram of the image ***scaledFoodIm***.

Display the histogram and the normalized histogram side-by-side in Figure 2 with appropriate titles on both the *x* and *y* axes.

3. [8 points]
Implement a **HistEqualization** function to perform histogram equalization on a grayscale input image to achieve the maximum gray level of 256 **by using the four steps explained in class**. The **HistEqualization** function has one input parameter **inputIm** representing the original grayscale image and two output parameters, **enhancedIm** and **transFunc**, where **enhancedIm** is the histogram equalization result (e.g., histogram equalized image) and **transFunc** is the histogram equalization transform function, which is a vector of 256 elements with the first and last elements being the newly transformed values for intensity 0 and 255, respectively.

Note: Both input and output images of the **HistEqualization** function should be an array with the same size and the same data type uint8.

Call the **HistEqualization** function to generate the enhanced image ***equalizedFoodIm*** of the original image ***foodIm*** and the corresponding transform function. Display the running time of using the **HistEqualization** function to accomplish the task on the console.

4. [2 points]
Call an appropriate built-in function to perform histogram equalization on the original grayscale image ***foodIm*** to achieve the maximum gray level of 256 and return the corresponding transform function. Display the running time of using this built-in function to accomplish the task on the console.

Note: If the built-in function does not return a transform function, you can simply call the built-in function to return the histogram equalized image.

5. [11 points]
Please carefully read the paper titled "Contrast Enhancement Using Brightness Preserving Bi-Histogram Equalization" to understand the basic idea of Brightness Preserving Bi-Histogram Equalization (BBHE). Implement a **BBHE** function to perform a contrast enhancement operation as proposed in the paper and return the transform function, which is a vector of 256 elements with the first and last elements being the newly transformed values for intensity 0 and 255, respectively. Display the running time of using the **BBHE** function to accomplish the task on the console. Make sure to use comments to describe basic steps or ideas of the technique.

Call the **BBHE** function to generate the enhanced image ***BBHEFoodIm*** of the original image ***foodIm***.

Display the enhanced images generated in Problems 3, 4, and 5 side-by-side in Figure 3 with appropriate titles.

Plot the transform functions obtained in Problems 3, 4, and 5 side-by-side in Figure 4 with appropriate titles on both the *x* and *y* axes. If the built-in function does not return a transform function in Problem 4, you plot the transform functions obtained in Problems 3 and 5 side-by-side in Figure 4 with appropriate titles on both the *x* and *y* axes.

Compute and display the Peak Signal to Noise Ratio (PSNR) values to compare ***equalizedFoodIm*** (obtained in Problem 3) with the original image ***foodIm*** and compare ***BBHEFoodIm*** (obtained in Problem 5) with the original image ***foodIm***, respectively.