CS5680/6680 – Fall Semester 2019
Assignment 2 – Image Enhancement in the Spatial Domain
Due: 11:59 p.m. Saturday, September 21, 2019
Total Points: 30 points

General Assignment Instructions:

1. Save solutions in appropriate m-files. Be sure to place semicolons wherever appropriate, to suppress unnecessary console output, such as when loading images into memory, or operating on them.
2. Please include comments (e.g., **your name and assignment number**) at the top of each m-file. In your main function, place a message "-----Finish Solving Problem X-----" followed by a pause command at the end of each solution, where X is the question number (e.g., 1, 2, 3, 4, and 5). For this assignment, you should have four .m files (main script, Scaling.m, CalHist.m, and HistEqualization.m).
3. You should submit your zipped m-files via the Canvas system. **Please do not send any image!**
4. **You are NOT allowed to call any Matlab built-in functions except size, disp, and other trivial functions to handle the invalid input data and convert the data types inside your three functions (Scaling, CalHist, and HistogramEqualization).**

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 prototype of this function is:

function [scaledIm, transFunc] = Scaling(inputIm, range)

where **inputIm** is the original grayscale image with the minimum and maximum intensity (e.g., oriMinIntensity and oriMaxIntensity), **range** is a vector containing the new minimum and maximum intensity (e.g., minIntensity and maxIntensity) of the rescaled (transformed) image, **scaledIm** is the transformed image, and **transFunc** is the transform function, which is a vector of *n* (*n* = oriMaxIntensity-oriMinIntensity+1) elements with the first and last elements being minIntensity and maxIntensity, respectively. Make sure that your function shows an appropriate error message if **range** contains the invalid data, such as non-integer values, negative values, and the 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.

Call the **Scaling** function to rescale the image ***foodIm*** to a new image ***scaledFoodIm*** in three ways corresponding to the three kinds of invalid data in **range** so the appropriate error messages will be displayed.

Call the **Scaling** function to 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 *x* and *y* axes.

2. [3 points]
Call the Matlab built-in function **imadjust** to rescale the image ***foodIm*** into the equivalent range used in Problem 1 (e.g., [newMin newMax]) and save the new image in ***matScaledFoodIm***.

Display your scaled image and matlab's scaled image side-by-side in figure 2 with appropriate titles.

3. [7 points]

Implement a **CalHist** function to calculate either the histogram or the normalized histogram or both histogram and normalized histogram of the grayscale input image. Note: You can decide the prototype of this function based on your own implementation.

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

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

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

Display the two normalized histograms at the top row and the two histograms at the bottom row in figure 3 with appropriate titles on both *x* and *y* axes.

4. [8 points]

Implement a **HistEqualization** function to perform histogram equalization on a grayscale input image to achieve the maximum gray levels (e.g., 256 gray levels) **by using the four steps explained in class**. Its prototype is:

function [enhancedIm, transFunc] = HistEqualization(inputIm)

where **inputIm** is the original grayscale image, **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 new mapping value 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 this function to generate the enhanced image ***equalizedFoodIm*** of the original image ***foodIm*** and the corresponding transform function. Display the running time of using this function to accomplish the task on the Matlab console.

5. [5 points]

Call an appropriate Matlab built-in function to perform histogram equalization on the original grayscale image ***foodIm*** to achieve the maximum gray levels and return the corresponding transform function. Display the running time of using this built-in function to accomplish the task on the Matlab console.

Display your enhanced image and Matlab's enhanced image side-by-side in figure 4 with appropriate titles.

Plot the histogram equalization transform functions obtained in Problems 4 and 5 side-by-side in figure 5 with appropriate titles on both *x* and *y* axes.

On the Matlab console, display the following information:

- Comparison of the running times to accomplish the tasks in Problems 4 and 5.
- Comparison of the histogram equalization transform functions obtained in Problems 4 and 5.
- Your findings (e.g., tricks you employed, lessons you learned, etc.) after reading the implementation detail of the chosen function for Problem 5.