CS5680/6680 – Fall Semester 2017
Assignment 2 – Image Enhancement in the Spatial Domain
Due: 11:59 p.m. Saturday, September 23, 2017
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 (i.e., 1, 2, 3, etc.).** 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. **If not explicitly specified, you are NOT allowed to call Matlab built-in functions inside your function. For example, you cannot call "imadjust" inside your Scaling function; you cannot call "imhist" or "hist" or "histc" inside your CalHist function; etc.**

Warm-up Exercise:

Matlab provides three useful built-in functions, i.e., imadjust, histeq, and adapthisteq, to enhance the contrast of an image. Please type "edit imadjust", "edit histeq", and "edit adapthisteq" on the command line to bring up the implementation of these three functions. Read the code and try to understand how to correctly use these three functions.

Problems:

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

1. [7 points]

Implement a **Scaling** function to **linearly** rescale (transform) the intensity values of the grayscale input image into new intensity values. The prototype of this function should be:

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

where inputIm is the original grayscale image, range is a vector containing the new range of the scaled image, scaledIm is the rescaled (transformed) image, and transFunc is the transform function. This transform function is a row or column vector of *n* elements, where the value of the first element is the new mapping value for the minimum intensity value after the scaling operation and the value of the last element is the new mapping value for the maximum intensity value after the scaling operation. In other words, *n* = maxIntensity-minIntensity + 1, where maxIntensity and minIntensity are the maximum intensity value and the minimum intensity value of the original image, respectively. Make sure that your function shows an appropriate error message if the range contains the invalid data (e.g., out of range). 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 scale the image **food** into a scaled image **scaledFood** with an appropriate range [newMin newMax] so **scaledFood** has a good quality. Plot the **transFunc** in figure 1 with appropriate titles on both *x* and *y* axes.

2. [3 points]
Use the Matlab built-in function **imadjust** to scale the image **food** into the equivalent range for [newMin newMax] and save the scaled image into **matScaledFood**.

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 normalized histogram or both histogram and normalized histogram of the grayscale input image. Note: I do not provide any function prototype for this function. So you can implement it at your will.

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

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

Display the two *normalized histograms* 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 should be:

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. This transform function is a row or column vector of 256 elements, where the value of the first element is the new mapping value for intensity 0 after histogram equalization and the value of the last element is the new mapping value for intensity 255 after histogram equalization. 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 **equalizedFood** of the original image **food** and the corresponding transform function. Display the running time of using this function to accomplish the task on the Matlab console.

5. [5 points]
Apply the appropriate Matlab built-in function to perform histogram equalization on the original grayscale image **food** to achieve the maximum gray levels and return the corresponding transform function. On the Matlab console, display the following information:

- The running time of using this function to accomplish the task.
- 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 learned, lessons you learned, etc.) after reading the implementation detail of the chosen function for Problem 5.

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.