CS540/440 – Digital Image Processing
Lab 2 – Image Enhancement in the Spatial Domain
Due: 16:00 p.m. Wednesday, Feb 7th, 2018

General Lab 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 lab 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.).**
3. **If not explicitly specified, you are not supposed 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.**
4. **You should submit your zipped m-files via the Blackboard system. Please do not send any image!**

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. Carefully read the code and understand how to correctly use these three functions.

Problems:

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

1. [5 points]
Implement a **Mapping** function to **linearly** rescale the grayscale input image into a new intensity range. The prototype of this function should be:

function[scaledIm] = Scaling(inputIm, range)

where inputIm is the original grayscale image, range is a vector containing the new range of the scaled image, and scaledIm is the rescaled (transformed) image. Make sure that your function shows an appropriate error message if the range contains the invalid data (out of range). Note: Both input and output images of the **Mapping** function should be an array with data type uint8.

Call the **Mapping** function to scale the image **food** into an appropriate range [newMin newMax] so the original image is enhanced to a good quality. Save the scaled image into **scaledFood**.

2. [2 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 1 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: Here 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 2 with appropriate titles on both axes.

4. [8 points]
Implement a **HistEqualization** function to perform histogram equalization on a grayscale input image by using the four steps explained in class. Its prototype should be:

function[enhancedIm, transFunc] = HistEqualization(inputIm, level)

where inputIm is the original grayscale image, level is the maximum gray-level of the desired enhanced image, enhancedIm is the histogram equalization result, and transFunc is the 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 data type uint8.

Call this function to generate the enhanced image **equalizedFood** of the original image **food**.

5. [3 points]
Apply the appropriate Matlab built-in function to accomplish the same task in problem 4).

Display your enhanced image and Matlab's enhanced image side-by-side in figure 3 with appropriate titles.

Plot the histogram equalization transformation function obtained in problem 4), Matlab's built-in histogram equalization transformation function obtained in problem 5), and the scaling transformation function obtained in problem 1) side-by-side in figure 4 with appropriate titles on both axes.

6. [2 points]
Obtain the execution time of your HistEqualization function and Matlab's built-in histogram equalization. Display how much faster or slower your solution is in terms of seconds when compared to Matlab's solution**. Faster and right solution will earn a 2-point extra credit!**

7. [4 points]
Analyze the enhancement results obtained from the simple scaling and the histogram equalization in terms of contrast and histogram. Explain the advantage and disadvantage of each solution. Show your analysis on the Matlab console.

8. [4 points]
What are the differences between two Matlab built-in functions **hist** and **histc**? Please write sample function calls to compute the histogram of an image using hist and histc, respectively. Please show your answers on the Matlab console.

9. Close all figures and clear all variables.