CS631T – Spring 2019
Assignment 1 – Matlab Warm-up Exercises
Due Friday, 02/08/2019
Submit electronically before 11:59pm on Blackboard
Total Points: 20 points

General Assignment Instructions: To get this assignment done, you need to have some background of algebra and matrix operations. You can refer to textbook section 2.6 or search online resources for tutorials.

Save all solutions in a **single** m-file. Be sure to place semicolons wherever appropriate, to suppress unnecessary console output, such as when loading images into memory, or operating on them. **You should submit only your m-file via the BB system. Please do not send any image**!

Please include comments at the top of each m-file. The comments should contain at least the following: **your name, your contact email, and assignment number**. **In your main function, place a message "-----Finish Solving Problem X-----" followed by a pause command (i.e., wait for a key to be pressed before continuing) at the end of each solution, where X is the question number (i.e., 1, 2, or 3).**

Problems:

1. **[2 point]**

Load the image ***peppers.bmp*** into a variable ***A***.

Display the loaded image ***A*** on figure 1 with the message "RGB Original Image" as the figure title. {*Think: What is the data type of the variable A?*}

Matlab hints: imread, figure, imshow, title, disp, pause

2. **[3 points]**

Convert image ***A*** into a grayscale image and store it as ***B***.

Transpose image ***B*** as ***TB***.

Display images ***B*** and ***TB*** side-by-side on figure 2.

Display the maximum, minimum, mean, and median intensity value of ***B*** on the command line.

{*Think: What is the data type of the variable B? Can Matlab do calculations on image **B**? Why? See Problem 3 for solution.*}

Matlab hints: rgb2gray, transpose (or '), subplot, max, min, mean, median

3. **[2 points]**

Convert image ***B*** to type **double** and store it again as image ***B***.

Rescale the image ***B*** so that its values range between 0 and 1 and store the rescaled image as ***C***.

Display image ***C*** on figure 3 with the message “Normalized Gray Scale Image” as the figure title. (Note: Image ***C*** should appear the same as the un-scaled image ***B***.)

Matlab hints: double, /, ./

4. **[3 points]**

Raise each pixel in image ***C*** to the powers of 0.25 and 1.25, respectively, and store the results as images (matrices) ***D1,*** and ***D2***.

Display two images ***D1*** and ***D2*** on figure 4 with ***D1*** located at the top and ***D2*** located at the bottom. Label the images with the corresponding powers used.

Explain the effects after applying the above two operations by using display command so the explanation can be shown on the command line when running the m-file.

Save image ***D2*** in jpeg format to a file called “X_D2.jpg” where X should be your first name. Open it using a standard image viewing program to verify that it worked.

Matlab Hint: ^, .^, print, imwrite

5. **[6 points]**

Perform binary thresholding on the original normalized grayscale image ***C***. A threshold 0.3 is chosen and all values in ***C*** greater than or equal to the threshold are set equal to 1, otherwise equal to 0. Save the thresholded binary image in ***bw1***.

Use a built-in Matlab function *im2bw* to do the same task and save the resulting thresholded binary image in ***bw2***.

Compare your results ***bw1*** with the Matlab’s result ***bw2***. If they are equal, display the message “My method worked”; otherwise, display the message “My method did not work”.

Display ***bw1*** and ***bw2*** side-by-side on figure 5 and label the two images with “my method” and “Matlab method”, respectively.

Matlab Hint: find, >=, zeros, ones, &, &&

6. **[1 point]**

Close all the five figures and clear all variables.

Matlab Hint: close, clear

7. **[3 points]**

Write a Matlab function ReduceGrayScale to approximately reduce the grayscale level of any input grayscale image to its 1/4.

A Matlab function should be defined in a separate Matlab file with the title format as

function [output] = FUNCTION_NAME (input) which is followed by the body part of the function. After completing the function, one can call this function from another Matlab file (your main assignment m-file) to test it.

For the ease of grading, copy the function right after your Matlab code of the above problems, in the same m-file. Submit only the single m-file to Blackboard.