REU Site Program in CVMA

Lab 1 – Matlab Warm-up Exercises

Problems:

1.
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.
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.
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.
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 the 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.
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. Find **two solutions** to obtain the thresholded binary image and save it in ***bw1*** and ***bw2***.

Use a built-in Matlab function *im2bw* to do the same task and save the resulting thresholded binary image in ***bw3***.

Compare your results ***bw1*** and ***bw2*** with the Matlab's result ***bw3***. If they are equal, display the message "My two methods worked"; otherwise, display the message "My two methods did not work". [Of course, the first message should be displayed when you run the m-file.]

Display ***bw1*** and ***bw3*** side-by-side on figure 5 and label the two images with "my first method" and "Matlab's method", respectively.

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

6.
Close all the five figures and clear all variables.

Matlab Hint: close, clear

7.
Write a Matlab function ReduceResolution to approximately reduce the spatial resolution of any input gray-scale image to its 1/4. That is, if the size of the original image is 256-by-256, the size of the new image should be 64-by-64. **Your ReduceResolution function should not contain any Matlab built-in function.**

8.
Write a Matlab function ReduceGrayScale to approximately reduce the grayscale level of any input gray-scale image to its 1/4.