BMI/CS 567 Medical Image Analysis
University of Wisconsin-Madison
Assignment #2

Instructor: Jeanette Mumford
Due Feb 22, 2018 *before* class begins

For this assignment you should turn in a single pdf file containing your code and the specific output requested for each problem. It shouldn't be necessary to run your code to grade the assignment. Name it `hwk2_yourname.pdf` (html is fine too). It should be submitted via canvas. The late homework will not be graded. Use sections and comments to explain which question and part your code refers to and you may also use comments in the few places I have requested written responses. Provide enough detailed justification and reasoning. For instance, a simple *yes* or *no* answer without justification or code will result in 0 points.

Important!!! You may not use `imagesc()` to answer any of the questions below. You should scale the data, as needed, when plotting.

(1) (2 points) Create an RGB image that is 4x4 where the upper left pixel is red, the upper right pixel is green and lower pixels are blue. No need to plot the image, but print out the values in the variable that makes up the image.

(2) For this problem I recommend first working through the Practical Lesson, 3.7.5. Instead of using the chicken images, you will use `homework_images.mat`. When you read the file in, the variable name for the array will `bucky_image` and this is a 3 dimensional array with 8 Bucky images with varying amounts of noise along the 3rd dimension. Sort the 8 images according to SNR and then CNR. I do not like the trick the author used to sort, since SNR is rounded and in many images I've worked with the SNR can be very low and rounding would set all SNRs to 0. In fact, in this case you will not be able to sort the images well if you round. Instead, use the `sort` function in MATLAB. Also, cut down on the amount of looping used in the author's example. You should:

(a) (2 points) Print the unordered SNRs (according to the order of the images).

(b) (1 point) Plot the Bucky images in a 2×4 grid of images, ordering the images from lowest to highest SNR. Display in grayscale.

(c) (2 points) Repeat the above for CNR. Print out the CNRs and plot the images, ordered by CNR. Did the order change? Explain why it did or did not change.

(3) For this problem use the image `bucky_trio.mat` that was distributed with this assignment. Please use grayscale for all parts. This image contains 3 repetitions of our school mascot in a single image, although all three may not be visible at once when viewing.

(a) (1 point) Display the image in grayscale, properly, using `image`.

(b) (4 points) Create a 3×1 panel plot where panels 1, 2 and 3 display processed image results, using methods from Chapter 4, that reveal the first, second and third Bucky, respectively. The full 351×1920 image should be plotted in each panel. I expect you to use a structured procedure for pulling out each image, as opposed to guessing many numbers until one works. You should be able to make an educated, very good, first try due to the regular structure of the image portions you are trying to extract.

(c) (1 point) When you did the previous part each of the three portions of the image containing the Buckys changed intensity. Describe how they change and *why* they changed in that way.

(4) For this problem you will use `img_029.ppm` from the distributed final project data (available to download in the Canvas Files, `distributed.zip`). The goal of the final project is to pull out features of images that help indicate when a retina shows retinopathy. This is a clear example of such a retina, as indicated by the lighter colored spots on the image. Use windowing (ch 4) and steps for converting a color image to gray scale (as in 3.7.4) to emphasize the spots in the image. Please don't loop through the image as the author does in 3.7.4. Make sure you choose settings that maximize the contrast between these spots and everything else in the image.

(a) (1 point) Describe how you chose your windowing parameters and display the parameters.

(b) (1 point) Describe why you chose the weights you used to create the grayscale image.

(c) (2 points) Display the final result.

(5) This exercise builds off Practical Lesson 5.4.3. The author has supplied code for computing the forward difference in the x direction in a file called, `ForwardDifference_5.m`.

(a) (3 points) Recreate the left panel of Figure 5.7. Also, figure out how to properly use the MATLAB function, `conv2` to carry out the convolution. Display the kernel you used and make a 2 panel plot with the author's result in the left panel and `conv2` in the right panel. Include appropriate figure headings. Be careful! Please refer back to the class notes on the proper definition of convolution.

(b) (1 point) Repeat the previous part, but compute the derivative in the y direction. Again, use both the author's looping strategy and `conv2`. Display the kernel you used as well as the 2 panel plot of the author's result and `conv2` result side-by-side.

(c) (2 points) Compute the length of the gradient (see equation on p. 123 of the book) to estimate the edges of the image. Display the result.