CS631T – Spring 2019
Assignment 3 – Image Enhancement in the Spatial Domain
Due Friday, 02/22/2018
Submit electronically before 11:59pm on Blackboard
Total Points: 50 points

Part 1: Matlab Coding [20 points]

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. **Please submit only your m-file via the BB system. Do not send any image**!

Please include comments at the top of each m-file. The comments should contain **your name, your 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).**

Problem1: Exercises on Low-pass and High-pass Filters in the Spatial Domain [20 points]

(If boundary extension is needed, please pad the boundary with 0's. Convert the image between unit8 and double as needed)

a) Implement a **MeanFilter** function to perform a filtering operation on the input image. [Note: the input of the **MeanFilter** function should be an image array and a mask. The output should be an image array.]

   Call this function to process the noisy image *Circuit* by using an **average** 3-by-3 averaging filter and a **standard** 5-by-5 averaging filter, respectively. You are not allowed to simply use the Matlab "filter2" or "conv2" function in your function implementation. Display original image and two processed images in figure 1 with the appropriate titles. **[3 points]**

b) Use Matlab function **fspecial** and **filter2** to perform the same mean filtering operation on the **Circuit** image. Process the image with a 3-by-3 averaging filter and a 5-by-5 averaging filter, respectively. Compare the two images with the results from your own MeanFilter function. Display if they are same. **[2 points]**

c) Implement a **MedianFilter** function to perform a filtering operation on the input image. [Note: Both input and output of the **MedianFilter** function should be an array with data type uint8.]

   Call this function to process the same noisy image *Circuit* by using a **weighted** 3-by-3 median filter **M** = [1 2 2; 1 1 1; 2 1 1] (see definition below) and a **standard** 5-by-5 median filter, respectively. You are not allowed to simply use the Matlab "medfilt2" function in your function implementation. Display original image and two processed images in figure 2 with the appropriate titles. **[5 points]**

   The standard median filter explained in the class is a special kind of the weighted median filter. Each value in **M** indicates the **number of copies** of the corresponding masked value in the image involved in the standard median filtering.

d) Use the Matlab function **medfilt2** to apply a median filter on the Circuit image. Process the image with a 3-by-3 median filter and a 5-by-5 median filter, respectively. Compare with the resultant images from your own MedianFilter function and display if they are same. **[2 points]**

e) Use the Laplacian mask to filter the image **Moon** by calling an appropriate Matlab function. Find the function by yourself. Use the formula **Enhanced Image = Original Image – Filtered Image** to get the final enhanced image. Use **imshow** to display four images including the original image, the filtered image, the scaled filtered image, and the enhanced image, in a new figure with the appropriate titles (Refer to the Figure 3.38 in the textbook). **[3 points]**

f) Close all figures and all variables in the workspace.

Part2: Exercise Problems. Submit the answers of this part as a word file or a pdf file separately from the Matlab m-file above. [30 points]

Problem1: Suppose we apply the following operations to input image f:

a. 1x3 mean filter (centered about origin) applied to f to give output f_1.
b. 3x1 mean filter (centered about origin) applied to f to give output f_2.
c. Take the mean (average) of f_1 and f_2 to get the final output.

These three operations can be done all at once with one mask operator. Derive this operator, carefully specifying the origin and mask weights. You must justify your answer.

Problem2: Given original image:

Original Image:

4	6	8	9	10
1	3	5	6	9
2	4	6	12	12
4	8	10	14	6

And mask

1	2	3
1	4	1
1	3	2

a) What is the resultant image of convolution (as defined in Eq. (3.4-2))?
b) What is the resultant image of correlation (as defined in Eq. (3.4-1))?

Problem3:

a) Suppose that you filter an image, f(x, y), with a spatial filter mask, w(x,y), using convolution, as defined in Eq. (3.4-2), where the mask is smaller than the image. Prove the important property that, if the coefficients of the mask sum to zero, then the sum of all the elements in the resulting convolution array will be zero also. You may assume that the border of the image has been padded with multiple layers of zeros.

b) Would the result to (a) be the same if the filtering is implemented using correlation, as defined in Eq. (3.4-1)?