**EL-GY 6123 Image and Video Processing, Spring 2018**

**Programming Assignment 2 (Due 2/8/2018)**

1) Implement 2D convolution using Python. Complete the following tasks:
   a) Write a 2D convolution function. The inputs of this function should be an image array and a filter array, and the output should be the convolved image. Your output image should have a size equal to the complete filtered image (similar to the "full" option of MATLAB conv2 function). For simplicity, you can assume the filter has an odd length in both horizontal and vertical dimensions and has an origin at the center. Also you can assume pixel values are zero outside the original input image.
   b) Write a main function, which will read in an image and allow a user to specify the filter size and filter coefficients. It will then call the convolution function from part 1) to filter the image. The program will show the original image and the filtered image and save the filtered image into a file after properly normalizing the pixel value range to 0 to 255. The program will also calculate and display the magnitude of Fourier transform of the original and filtered image, as well as frequency response of the filter. (In Python, you can use fft2() in numpy package)
   c) Using your main program to filter an image with the following three filters:

$$H_1 = \frac{1}{16}\begin{bmatrix} 1, 2, 1 \\ 2, 4, 2 \\ 1, 2, 1 \end{bmatrix} \qquad H_2 = \begin{bmatrix} -1, -1, -1 \\ -1, 8, -1 \\ -1, -1, -1 \end{bmatrix} \qquad H_3 = \begin{bmatrix} 0, -1, 0 \\ -1, 5, -1 \\ 0, -1, 0 \end{bmatrix}$$

   In your report, for each filter, you should discuss the difference between the original and filtered image, both in the spatial and frequency domain and explain how does the filtering effect correlate with the filter and its frequency response.
   The input to the convolution function should a single channel image (e.g. gray scale). Your main program can be flexible. You may allow both color and gray image as input, but if the input is color, you convert it to gray. You can use built in functions to calculate the transform and freq response. For the FT of the images, you can use the fft2 function. For the freq response, you can use the fft2 function.

2) Image noise removal using averaging and Gaussian filters.
   a) Create a noisy image, by adding zero mean Gaussian random noise to your image (In Python, the noise can be generated by using "np.random.normal(mean,sigma,(img.shape))").
   b) Apply your previous program to filter the noise-added image using an average filter of size n x n, where n=odd
   c) Apply your previous program to filter the noise-added image using a Gaussian filter of size n x n (Recall that with Gaussian filter, the filter size and the Gaussian filter parameter $\sigma$ should satisfy $n>=5\sigma$)

   Try two different noise levels (0.01 and 0.1) and for each noise level different filter sizes (ex: 5x5 to 9x9 in step size of 2).

   Your report should show the original, noise-added, and filtered images for each combination of noise level and filter size. Comment on for each noise level, which filter size is best for each filter and how does the two filters compare in their noise removal capability.