UNIVERSITY OF CALIFORNIA, SANTA BARBARA

Department of Electrical and Computer Engineering

Prof. P. Sen **ECE 178 Digital Image Processing** Fall 2017

Homework 5 MATLAB part

No Written portion. MATLAB portion due on GauchoSpace at Monday November 6th, 9:00pm.

This MATLAB assignment consists of only one part.

1. **[5 pts]** In this part, we are going to implement convolution in the frequency domain between an image "$elaine\_512.png$" and a given kernel.

Under suitable conditions, the convolution in the spatial domain equals to point-wise multiplication in the frequency domain.

(a) **spatial_conv.m:** You could use the MATLAB's inbuilt function *conv2* to perform the convolution. Make sure that you pad the input image using periodize padding before performing convolution. You could use the MATLAB's inbuilt function *padarray* with *'circular'* option to perform the periodize padding on the input image. The convolution theorem for the discrete-time Fourier transform indicates that a convolution of two infinite sequences can be obtained as the inverse transform of the product of the individual transforms. In order to make the input image become an infinite sequence, we need to apply periodize padding to it. (More details: `https://en.wikipedia.org/wiki/Discrete_Fourier_transform#Circular_convolution_theorem_and_cross-correlation_theorem`)
The output of your spatial_conv.m function should be of the same size as the input. You could achieve this by specifying appropriate *shape* option for the *conv2* function.

(b) **frequency_conv.m:** In this part, you have to implement the convolution in frequency domain. For the memory and computation time issue, we do not use the **my_dft_2D.m** in HW4. Since we know that the convolution in the spatial domain equals to point-wise multiplication in the frequency domain, we use the MATLAB's built-in *fft2* to calculate the Fourier coefficients of the input image and convolution kernel. After we got the Fourier coefficients, we perform point-wise multiplication for these two Fourier coefficient maps to get the product of Fourier coefficients, and then reconstruct the convolved image from the product of two Fourier coefficient maps using inverse DFT. This is achieved by using MATLAB's *ifft2* function.

Here comes a question: the size of input image (512×512) is different from the size of the kernel (27×27), so the size of Fourier coefficient maps should be different. How do we calculate the point-wise multiplication? We can put the kernel onto a canvas (zero matrix with size 512×512)

and then do the calculation of Fourier coefficient. One straight forward way to put the kernel is to put the kernel at the top-left of the canvas. However, it will give you a shifted convolved result like following:

The final result should look like the image shown below.

You can notice that the first image is a circularly shifted version of the expected result. So you need to perform a circular shift using MATLAB's inbuilt function *circshift*. You need to figure out how much circular shift should be done and in which direction to get the expected result.

In the **HW5_part1.m**, the program will output the MSE between two methods of convolution. If it shows value other than 0, you should check your code to find the bug. Most likely cause of error could be that your output is circularly shifted.

For submitting your code, please upload one zipped file includes only 2 files: **spatial_conv.m** and **frequency_conv.m**. We will run your code and generate results using the wrapper code. So make sure that your functions run properly with the wrapper code provided without making any changes to the wrapper code. Name your zip file in this format: <Perm number>_<First name>_<Last name>_HW5. Good luck!