UNIVERSITY OF CALIFORNIA, SANTA BARBARA
Department of Electrical and Computer Engineering

Prof. P. Sen

ECE 178 Introduction to Digital Image and Video Processing

Fall 2017

Homework 4

Due Monday, October 30. No Written portion. MATLAB portion due on GauchoSpace at 9 pm.

This MATLAB assignment consists of two parts. The first part is about Fourier analysis of images and the second part is about complexity of convolution operation.

1. **[15 pts]** In the first part, we are going to transform an image ("test_img.png") from the spatial domain to the fourier domain. Then we are going to observe how the image can be reconstructed in the spatial domain by doing a stepwise inverse fourier transform. The task is to create a video similar to "fftanimation.mov", where the left half represents the reconstructed image and the right part represents the fourier basis which is being weighted and added to the reconstructed image.In the video, you perform reconstruction in descending order of basis weight. (In other words, sort the fourier coefficients in the descending order of their absolute values). For this part you will have to:

   - Write a function which takes the DFT of the input image to obtain its Fourier coefficients **(my_dft_2D.m)**. The supplied code will take care of the sorting by using the inbuilt function from MATLAB. Kindly refrain from using MATLAB built-in function *fft2, ifft2*.
   - Write a function that will generate the specific basis function as requested by the frequency parameters supplied to it **(gen_fourier_basis.m)**. It is advisable to use the functions you have coded for the previous homework.
   - Write another function that will weigh the basis generated above using its matching Fourier coefficient **(get_recon_image.m)**. The code for accumulating the weighted basis has already been supplied in the main script for this part.
   - Finally create an animation/movie/video from the image sequence generated similar to the supplied example **(create_recon_video.m)**. Keep the frame rate as 1 frame per second. Your video should first display top 100 fourier basis and the correspondingly reconstructed image. After that it should show every tenth basis and the reconstructed image at that stage (because understandably the later coefficients would do little to alter the reconstructed image). A link to help you with transforming image sequences into video using MATLAB has been mentioned in the comments with the main script.

   It might be wise to scale the input image to a smaller size for debugging your code.

2. **[10 pts]** In this second part, we are going to implement convolution in the spatial domain between an image ("test_img.png") and a given kernel. The given kernel is separable using the methods described in class and its factorization into vectors has already been done in the script provided. You are to write three functions:

   - **brute_force_conv.m :** This will take the image and the kernel as input and carry out 2D-convolution to calculate the resultant image output. Here the kernel is to be used as is. It is not to be separated. It will also output the time taken for this operation.

- **separated_conv.m:** This function will use the kernel separated into a row and a column and carry out two 2-D convolutions to calculate the resultant image output. As with the earlier function, this function will also output operation time.
- **calc_MSE_2D.m:** This function is supposed to take the resultant arrays from the two functions above and calculate the meansquareerror between them.

The main script for this part is **HW4_part2_main.m**. It will output the time taken for the two convolution operations and will also print the MSE error output. Note that for convolution operation with the kernel, apply similar padding (your choice) in both your convolution functions and return the "valid" part of the resultant convolved image (result should be of same size as input image).

For submitting your code, please upload one zipped file that includes two directories for each part. In part 1, you need the following files: **HW4_part1_main.m**, **HW4_part1_main.m**, **my_dft_2D.m**, **gen_fourier_basis.m**, **get_recon_image.m** and **create_recon_video.m**. You **don't have to attach your resultant video**. For part 2, you need four files: **HW4_part2_main.m**, **brute_force_conv.m**, **separated_conv.m** and **calc_MSE_2D.m**.

Name your zip file in this format: <Perm number>_<First name>_<Last name>_HW4. Good luck!