**ECE-GY 6123 Image and Video Processing, Spring 2023**
**Computer Assignment 2: 2D Convolution and Image Noise Removal**
Due 2/23/2023 at 11:59 AM

Please submit your solutions, including Python code and outputs (when relevant), **as a single PDF file to GradeScope**.

In this assignment, you will implement 2D convolution and perform image denoising in Python.

**Problem 1** (Implementing 2D convolution in Python).

(a) Write 2D convolution function `conv2`, which implements "same"-padding convolution[1]. For simplicity, you can assume the filter has an odd length in both vertical and horizontal directions. The input image should be grayscale. Assume pixel values outside the image are zero.

(b) Write a function `plot_filtering` that uses your `conv2` function to filter a given input image with a given input filter and plots the following:

- The original input image and filtered image
- The log-magnitude spectrum of the original image, filter, and output image.

Use a grayscale colormap and a colorbar for each plot. For your filter-response, use an FFT size equal to the size of your image.

(c) Use `plot_filtering` on an image of your choice with the following 3 filters:

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

For each filter, discuss the result of filtering in both spatial and frequency domains. Explain how the filtering effect in the image domain correlates with the filter and its frequency response.

[1]see MATLAB's `conv2` for example

**Problem 2** (Image denoising with average and Gaussian filters).

(a) Write a function `awgn` that takes in an input image and noise-level $\sigma_n$ and adds i.i.d[2] zero-mean Gaussian random noise with standard-deviation $\sigma_n$.
(see `numpy.random.randn`)

(b) Write a function `gaussian_filter` which returns a 2D Gaussian filter of size $m \times m$ with standard deviation $\sigma$, where $m = \lceil 5\sigma \rceil$.Be sure to normalize your filter to sum to 1 so that filtering does not shift the mean of your image.

(c) Generate a noisy version of an image using `awgn`, with a noise-level $\sigma_n = 0.1$ on an image intensity scale of $[0, 1]$. Apply separately a Gaussian filter of size $5 \times 5$ and average-filter of size $5 \times 5$, and comment on how the filters compare in their noise-removal, qualitatively. Repeat this for a few different noise-levels and filter sizes. Each time, calculate the PSNR[3] of your noisy and denoised images via the formula,

$$\begin{aligned} \text{PSNR} &= 10 \log_{10}\left(\frac{\text{max}^2}{\text{MSE}}\right) \\ &= -10 \log_{10}(\text{MSE}), \quad \text{(when images are on range [0,1])}, \end{aligned}$$

where MSE is the mean-squared-error with respect to your original image, and max is the maximum of your image intensity range (1 for images on range [0,1], 255 for images on range [0,255]).

**Help!**: If your `conv2` function from problem 1 isn't working, you can use the scipy package's 2D convolution function (see `scipy.signal.convolve2d`).

[2] independent identically distributed
[3] see `https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio` for more info.