Problem Set 2

Due: October 16, 2023

Make sure to follow the **Homework Submission Instructions** on the last page prior to submitting.

1) [7 pts] In this exercise, you are tasked with designing a 3×3 convolutional kernel for edge detection in an image.

a) [3 pts] Design a 3×3 convolutional kernel that focuses on detecting horizontal edges while suppressing vertical edges in a given image. (*Hint:* Design separable 1D kernels for vertical and horizontal operations.)

b) [2 pts] Find the equivalent filter $H(u, v)$ in the frequency domain.

c) [2 pts] Determine whether this filter is a low-pass or high-pass filter.

2) [12 pts] In class we saw both mean and median filters as examples of neighborhood operations. Let $\mathcal{N}(i)$ denote a given neighborhood of pixel i. For the rest of the exercise, you may assume that the number of pixels in the neighborhood, i.e. $|\mathcal{N}(i)|$ is odd. Let I_j denote the intensity of pixel j.

a) Show that the output of the mean filter applied to the neighborhood $\mathcal{N}(i)$ is the solution to

$$\min_{I_i^{new}} \sum_{j \in \mathcal{N}(i)} |I_i^{new} - I_j|^2$$

b) Show that the output of the median filter applied to the neighborhood $\mathcal{N}(i)$ is the solution to

$$\min_{I_i^{new}} \sum_{j \in \mathcal{N}(i)} |I_i^{new} - I_j|$$

c) Suppose we try to solve

$$\min_{I_i^{new}} \sum_{j \in \mathcal{N}(i)} w_j |I_i^{new} - I_j|$$

for some weights $\{w_j\}$. Then how can I_i^{new} be calculated from the knowledge of I_j for $j \in \mathcal{N}(i)$?

3) [9 pts] a) Consider Gaussian random vectors $\mathbf{x}_1 \sim \mathcal{N}(\boldsymbol{\mu}_1, \mathbf{V}_1)$ and $\mathbf{x}_2 \sim \mathcal{N}(\boldsymbol{\mu}_2, \mathbf{V}_2)$. Let their pdfs be denoted by $p_1(\mathbf{x})$ and $p_2(\mathbf{x})$ respectively. Show that $p_1(\mathbf{x})p_2(\mathbf{x})$ also correponds to a Gaussian pdf.

b) Generalize the result to $\Pi_k p_k(\mathbf{x})$ when $\mathbf{x}_k \sim \mathcal{N}(\boldsymbol{\mu}_k, \mathbf{V}_k)$.

4) [18 pts] Programming Exercise: In this exercise, you will re-generate three examples we saw in class.

a) [3 pts] Use the phase of the Fourier spectrum of Lena image, and the magnitude of the Fourier spectrum of the kneeling man image (provided to you), to generate a combined image with the corresponding phase and magnitude Fourier spectra. Display the input images and their relevant Fourier specta, and the final output image.

b) [6 pts] Take the DCT of each distinct (i.e. not sliding) 8×8. Keep the largest (in magnitude) 10 DCT coefficients, and set the rest to zero. Take the inverse DCT of each block to generate a new image. Do the same with DFT and inverse DFT (i.e. FFT). Display the images.
(*Hint:* In Python, it will be helpful to define a function to extract/process non-overlapping sliding blocks and store the results in a new array. For MATLAB, the built-in function blkproc may be helpful. Also defining a function that performs the given transform, then keeps the largest 10 coefficients and then does the inverse transform will also be useful.)

c) [9 pts] Perform the sharpening exercise with the coins image ('eight.tif'). Read-in the image, then perform lowpass filtering with an averaging filter (use imfilter/signal.convolve2d). Generate the high-pass image by subtracting this low-pass image from the original. Generate the sharpened image as original image plus 2 times the high-pass image. Display the original, low-pass, high-pass and sharpened images.

Homework Submission Instructions:

- Please upload two separate files:

  1. A pdf (please name it as **name_hw2**.pdf) containing your solutions and **also the output of your coding implementations**. Submitted pdf file has to be **readable**, please double check readability before submission.

  2. Your codes (in case of multiple scripts, you can rar them but single python scripts are preferred) **should be runnable** and **generate only all the figures requested in the homework** with proper title & captions. These should be consistent with the ones reported in the pdf file. Make sure to use plt.figure() for each figure to avoid overwriting, use plt.subplot to reduce the number of generated figures.

- Submit your codes as a python file (**name_hw2**.py) or as a jupyter notebook file (**name_hw2**.iypnb). Please use the markdowns to separate the answers.

- **As per the syllabus:** You may discuss solution strategies with other students taking the class. *However, all of the submitted work must be entirely yours.*