ECE4580 FA23 - Prof. Jones – HW 2

Due Thursday, Sep 28, 2023 – 11:59 PM via Canvas

This assignment contains several parts. Some involve code development using CoLab. You should put all of your code into a single python module; in my implementation, my main() calls functions named part1(), part2() and part3().

PART I

In part 1, you will implement and demonstrate a gray-level histogram equalization method.

1. Load a color image from your Google Drive and convert to grayscale by averaging the three color planes.
2. Call a Python function that will implement the histogram equalization method described in lecture 5. The number of bins to use should be a parameter to this function (as well as the image). Your function should return the equalized image. Your function should NOT call a complete histogram equalization method; I want you to implement the method yourself using only Python and numpy array and vector methods!
3. Demonstrate your method on three different images.

PART II

In part 2, you will derive proper kernels for a Gaussian smoothing filter, and demonstrate its use in Python.

1. Using the equation for a Gaussian filter, calculate the coefficients for a Gaussian kernel where $\sigma = 1.75$. You can calculate the coefficients by hand, or using a calculator, or using Excel or some other method – BUT SHOW YOUR WORK! You are NOT to call a function that creates a Gaussian kernel.
2. Also compute and show the two 1-D kernels that are equivalent to the Gaussian kernel above. Multiply the two kernels to show that they are indeed equivalent.
3. Write a Python function to test these kernels. It should:
    a. load a color image and convert to grayscale by averaging;
    b. apply the 2D kernel to the image (you can use a function such as skimage.filters.correlate_sparse() to apply the kernel) and display the result;
    c. apply the two 1D kernels to the image and display the result;
    d. subtract the two results to see the difference between the results of 1D and 2D filtering; compute the mean and standard deviation of this result and print them out.
4. Run this function on two different images.

PART III

In part 2, you will experiment with image sharpening using an unsharp mask.

1. Write a Python function that implements image sharpening by applying a Gaussian unsharp mask and performing the sharpening process. Write your code so that it works on both monochrome and color images; you can check the length of img.shape to see if the image is color or monochrome. As above, you are NOT to call a function that does sharpening!
2. Test your code on a monochrome and a color image. For each one, experiment with several different values of K (the scale factor) to see which result looks best to you. In your report, include the result images for the values of K that you tried, and note the best one.

SUBMISSION:

For your submission, paste your code and the results of running your program into a single Word (or pdf) file. Paste code and program output as plain text (no dark-mode or screenshots). Also submit your final Python notebook (as an ipynb file). Submit two separate files: your Word or pdf submission and your CoLab notebook. Submit your files using Canvas. Do NOT put your files into a zip file for submission; submit them as separate files.