Computer Vision Fall 2022
Programming Assignment # 1
(Issue date: 29/11/2022)

1. Write a function ***def* C*onvolutionOperation (image, kernel):*** that has arguments

   a. Image (Images may be of varying sizes and you may want to give the size as arguments. You can use the *shape* property in python.)

   b. Kernel *H* (Again, you should allow varying size Kernels.)

The output of the function should be the convolution of *I* with *H*. Test your function and show results on the following Kernels, using the provided sample images within the assignment.

i. Averaging Kernel (3×3, 5×5, and 7×7)

ii. Gaussian Kernel (σ =1 with kernel size 3×3, 5×5, and 7×7)

iii. Sobel Edge Operators: $\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$ and $\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$

iv. Prewitt Edge Operators: $\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$ and $\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & 1 \end{bmatrix}$

v. Laplacian Operators: $\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

Note: You must perform convolution by writing your own code and comparing it with the result of the **cv2.filter2D()**.

2. Convert the **balloon.jpg** image to a grayscale image and add random noise to it (assume the noisy image name is **balloonGrayNoisy.jpg**). Apply the generated Averaging and Gaussian Kernels on the "**balloonGrayNoisy.jpg**" image to perform noise filtering and show the outputs. Test different filter sizes.

3. Perform edge detection on the "**linux.jpg**" using the Sobel and Prewitt Operators and show the outputs (Convert the image to grayscale, smooth the image using a Gaussian filter, compute horizontal and vertical gradients and then the magnitude of the gradient. Apply a threshold.)

4. Perform edge detection on the "**linux.jpg**" using the Laplacian Operators and show the outputs (Convert the image to grayscale, smooth the image using a Gaussian filter, Compute the output image using the Laplacian operator and then apply a threshold.)

5. Perform edge detection on the "**linux.jpg**" using the Canny edge detection algorithm and show the outputs (Convert the image to grayscale, smooth the image using a Gaussian filter, Compute the magnitude and gradient using the Sobal operator, apply non-maximum suppression, and then apply double thresholding (Hysteresis)).

Deliverables:

1. Report including Input and Output images (Soft Copy)
2. Code (Soft copy)
3. Your analysis for the best results in all questions
4. Please send your assignments to google classroom as a Jupiter notebook (Assignment1.ipynb file)
5. Submission Deadline: 06/12/2022 (23:59)