BIRZEIT UNIVERSITY

Faculty of Engineering and Technology
Electrical and Computer Engineering Department

ENCS5343 Computer Vision
Assignment # 1

Student name:

Student ID:

Notes:

1- **Use this page as a cover for your home work.**
2- **Late home works will not be accepted (the system will not allow it).**
3- **Due date is December 18th, 2023 at 11:59 pm on ritaj.**
4- **Report including Input and Output images (Soft Copy). Include all the code you write to implement/solve this assignment with your submission.**
5- **Organize your output files (images) as well as used codes to be in a folder for each question (Q1, Q2, etc.). Then add the solution of all questions along with this report in one folder named** Assign1**. Compress the** Assign1 **folder and name it as (**Assign1_LastName_FirstName_StudetnsID.Zip**).**
6- **Please write some lines about how to run your code.**
7- **You might need to use OpenCV Python in your implementation.**

Quetsion#1

Look for an image from the internet with the following properties: 8-bit gray-level, 256x256 pixels in size.

1- Show this image. Don't use your friends' ones.
2- Apply a power law transformation with gamma=0.4 to the image and show the image after the transformation.
3- Add a zero-mean Gaussian noise (with variance =40 gray-levels) to the original image and show the resulting image.
4- Apply a 5 by 5 mean filter to the noisy-image you obtained in point 3 above and show the result. Discuss the results in your report.
5- Add salt and pepper noise (noise-density=0.1) to the original image and then apply a 7 by 7 median filter to the noisy-image and show both images.
6- Apply a 7 by 7 mean filter to the salt and pepper noisy-image and show the result. Discuss the results in your report.
7- Apply a Sobel filter to the original image and show the response (don't use ready functions to do this part).

Quetsion#2

Write a function that convolves an image with a given convolution filter

function [output_Image]= myImageFilter(Input_image, filter)

Your function should output image of the same size as that of input Image (use padding). Test your function (on attached images "`House1.jpg`" and "`House2.jpg`") and show results on the following Kernels.

1- Averaging Kernel (3×3 and 5×5)
2- Gaussian Kernel (σ = 1,2,3) Use $(2\sigma+1)\times(2\sigma+1)$ as size of Kernel (You may write a separate function to generate Gaussian Kernels for different values of σ). Discuss the results in your report.
3- Sobel Edge Operators: $\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$ and $\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$
4- Prewitt Edge Operators: $\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$ and $\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & 1 \end{bmatrix}$

Quetsion#3

Attached "`Noisyimage1`" and "`Noisyimage2`" are corrupted by salt and paper noise. Apply 5 by 5 Averaging and Median filter and show your outputs. Why Median filter works better than averaging filter?

Quetsion#4

Compute gradient magnitude for attached image "`Q4_Image`" (using built-in sobel gradients function).

1. Stretch the resulting magnitude (between 0 to 255) for better visualization
2. Compute the histogram of gradient magnitude
3. Compute gradient orientation (the angle of gradient vector)
4. Compute histogram of gradient orientation (angle between 0 and 2*pi)

Quetsion#5

Load *walk_1.jpg* and *walk_2.jpg* images in Python. Convert them to gray scale and subtract *walk_2.jpg* from *walk_1.jpg*. What is the result? Why?

Quetsion#6

Apply canny edge detector on the "`Q_4.jpg`" using OpenCV function "`Canny`". Test different values of '`Threshold`'.

(Good Luck)