Programming Assignment #3, CSC 416/616/716 Digital Image Processing (Spring 2025)

Due: 11:00 am, Mar. 6

Image files are available on Canvas: Files >> 2_Data. You can choose either *.raw* or *.tif* file.

.raw: No image libraries are needed. However, it does not include any header information (e.g., image dimension and number of intensity levels). Thus, **you need to specify image dimension (# of rows x columns) and data type (e.g., 8 bit) when reading/writing .raw files.**
Also, the simple image viewer in Windows system may not work. For displaying them, use Photoshop or some free software for image display (e.g., ImageJ, https://imagej.nih.gov/ij/download.html).

.tif: You need image libraries depending on the programming language you choose.

Problems 1-2: Assigned to *ALL* students.

Problem 3: Assigned to *Graduate* (M.S. and Ph.D.) students. Undergraduate students who submit this will receive a bonus point.

Spatial filtering – Lowpass, highpass, and nonlinear lowpass filtering

1. Write a program for image denosing using *Gaussian* filtering. You can use the code example for the Gaussian kernel below. Set kernel size = 5x5, σ=2, and K=1.67, where s and t are the coordinates of the 5x5 kernel. The denoised version of the input image should be generated as the output.

```
w[s][t]= exp(s.^2+t.^2) / (2*sigma^2)*(-1))*K
```

- Input: *lena-noise* (512x512, 8-bit image) – Note: This is not the original Lena image.
- Output: *lena-gaussian* (512x512, 8-bit image)

2. Write a program for image sharpening using *Laplacian* filtering. Use the 3x3 kernel below (to extract fine details from the input image). Then, apply the Laplacian result (fine details) to the original image to get a sharpened image, with a weight of -1. The sharpened version (not the Laplacian's output itself) of the input image should be generated as the output.

- Input: *lena* (512x512, 8-bit image)
- Output: *lena-sharpen* (512x512, 8-bit image)

0	1	0
1	−4	1
0	1	0

3. Write a program for image denoising using *Median* filtering. Use a 3x3 kernel. You can use any existing sorting method (e.g., bubble sort) to get median intensities (implementation of a sorting function is not required). The denoised version of the input image should be generated as the output.

- Input*: lena-noise* (512x512, 8-bit image) – Note: This is not the original Lena image.
- Output*: lena-median* (512x512, 8-bit image)

Submission: Submit a single ZIP file including your report, all source code files, executable, output images for all problems. Use *Yourname_HW3.zip* as your submission filename (e.g., *MinjeongKim_HW3.zip*).

In your report, you should include your programming environment (e.g., Visual Studio for C/C++), source code segments (i.e., the core function of your program) for all problems, and output images.

Here are some more requirements.
1) There is no restriction on the programming language (e.g., C/C++, Java or Python). However, extended libraries beyond basic ones (e.g., image file I/O and basic math functions) are not allowed.
2) The instructor and TA should be able to compile and run your source code. The test results by the instructor and TA should be same as your outputs in your submission.
3) The output images submitted by you should fit the given image size (i.e., same to the input image size) unless specified in your report.