# EE608 - Digital Image Processing

Assignment 3

*Assigned Date: 22/02/2023*

*Due Date:07/03/2023*

## Problem 1: Point Operations

1. Write a program/function to do the "Full scale contrast stretch (FSCS)". Verify your result by printing the min and max pixel values before and after applying FSCS. Also, plot the histogram of the image before and after applying FSCS. (5)
2. Imagine you have an image taken in low light levels and which, as a result, has low contrast. What are the advantages of using contrast stretching to improve the contrast, rather than simply scaling the image by a factor of, say, three? (Theory - write down your explanation) (3)
3. Write a program to perform histogram equalization. Plot the histogram of the image before and after applying histogram equalization. (5)
4. Write a function, which takes an input image and a reference image and applies histogram Matching on the input image by matching the histogram with that of the reference image. Use eye.png and eyeref.png (converted to grayscale) as the input and reference images respectively. (5)

## Problem 2: Spatial Filtering

**Image Filtering**

In this problem, you will denoise images using the linear filtering techniques discussed in class. Use additive white Gaussian noise (AWGN) with zero mean and variance $\sigma^2 = 25$. Write a program to denoise the image using the following techniques:

1. Average filter of size M × M. Vary M from 3 to 15 in increments of 2 and observe the tradeoff between denoising and smoothing. What could be an appropriate window size for this noise level? (5)
2. Gaussian filter of size $5 \times 5$. Experiment with the standard deviation h of the filter from 0.5 to 5 in steps of 0.5 and observe its effect on denoising. (5)
3. Median filter of size $M \times M$. Vary M from 3 to 15 in increments of 2 and observe the tradeoff between denoising and smoothing. What could be an appropriate window size for this noise level? (5)

**Edge Detection**

4. Use Sobel and Prewwit Operators to get the edge map from the given image.(5)
5. Estimate gradient magnitude using the following definitions: (6)

(a) $M(i,j) = \sqrt{\Delta_x^2(i,j) + \Delta_y^2(i,j)}$

(b) $M(i,j) = |\Delta x(i,j)| + |\Delta y(i,j)|$

(c) $M(i,j) = max\{|\Delta x(i,j)|, |\Delta y(i,j)|\}$

6. Laplacian Edge Detectors: Compute the Laplacian using the convolution template (2)

$$\begin{bmatrix} 0 & +1 & 0 \\ +1 & -4 & +1 \\ 0 & +1 & 0 \end{bmatrix}$$

7. Compute the edge map E as the output of a zero crossing detector. (3)
8. Laplacian of Gaussian (LoG):- LoG was motivated by the sensitivity of gradient and Laplacian edge detectors to noise. In this problem, you will implement an edge detector using the LoG operator. Work with the lighthouse.png image that is corrupted with AWGN whose $\sigma^n = 10$. Experiment with different values of $\sigma$ for the Gaussian pre-filter and compute the edgemap E after zero crossing detection. Compare your result with gradient based techniques implemented above and verify the robustness of LoG. (5)

## Problem 3: 2D-DFT

2D-DFT:- Recall the definition of the 2D-DFT from class. Assume that you are working with square images. Write a program, that implements the DFT/IDFT pair using the following techniques.
1. From first principles i.e., as a summation over two dimensions. (5)
Now do the following using three of your favorite images from the above set:
2. Display the image, its DFT magnitude, phase, DFT magnitude subject to the logarithmic transformation. (3)
3. Verify that IDFT[DFT[I]] = I. (3)
4. Show the importance of phase by transposing the phase matrix and then synthesizing the image. Repeat by a) setting the phase to 0, and b) by adding noise. (3)