

# Assignment - Week 1

June 13, 2025

It's time to put our knowledge to the test. Let's tackle the assignments and practice our skills to reinforce our understanding.

**For submitting all assignments, make a GitHub repo and store the assignments in that repo.**

## Problem 1

Generate a 2-dimensional Numpy array of shape (15,15) filled with random integers between 0 and 255 (inclusive). Perform the following tasks:

1. Create a new array where all the elements less than the threshold(120) are set to zero and the rest of them remain the same.
2. Create a Python function named **boundary\_transversal(matrix)** that will print out boundary elements of the matrix starting from the top-left corner and in clockwise direction.
3. Create a new Numpy array by replacing all the odd elements with zero and even with one.

## Problem 2

Generate a 2-dimensional numpy array of size (20,20) with random integers between 0 and 255(inclusive), then perform the following tasks:

1. Calculate and print the maximum, minimum, and median of each row and column separately.
2. Generate and display the image that has to be created using the above Numpy array after resizing it to (200,200). Use the **nearest neighbor interpolation** that enlarges each pixel to a (10x10) square of the same value, and no smoothing or averaging is applied to that.
3. Create 2 grayscale images, one with a horizontal gradient (0 to 255 left to right) and another with a vertical gradient( 0 to 255 top to bottom) of size (256,256) pixels, and display them.

## Problem 3

Read an image in Grayscale format and perform the following operations:

1. Invert the image (invert all pixel values) and display it.  
**Hint:** For an image with u.int8 datatype, inversion = 255 - image.
2. Read an image in colored format and:
  - Split it into R, G, B channels
  - Zero out the Red channel
  - Merge the channels and display the modified image
3. Read an image in color format and:
  - flip it horizontally and display it.
  - flip it vertically and display it.
  - Mirror the image and display it. (Think of the effect of mirroring an image, considering an image as a 2D matrix, is it the same as horizontal flipping ?)

## Problem 4

Read a colored image and do the following operations:

1. Display the size of the image and the pixel values of different channels. **Note** that cv2.imshow() read image in **BGR** format.
2. Write a Python function that will manually convert a Color image to a Grayscale image. **Don't use cv2.cvtColor() function.** Note that,

$$Gray = 0.299R + 0.587G + 0.114B$$

3. Count the number of Pure white pixels( have value [255, 255, 255] in a color image.

## Problem 5

Read a color image and perform the following operations:

1. Convert it to a grayscale image using OpenCV, and apply a Gaussian Blur with a kernel size of (9,9) and display it.

2. Create a white image of the same size and subtract the blurred grayscale image from the white image. Display the resulting image. Note that this effect is same as the inversion of an image.
3. create a blank image of size (512,512) and do the following:
  - Create a rectangle of blue color with top-left corner at (50,50) and bottom-right corner at (462,462), with a thickness of 3 pixels.
  - Draw a circle of radius = 50, centre at (300,300) and filled with green color.
  - Draw a red diagonal line starting from the top-right to the bottom left corner of thickness = 2 pixels.