E9 241 Digital Image Processing

Assignment 04

Due Date: November 8, 2025 - 11:59 pm **Total Marks:** 60

Instructions:

For all the questions, write your own functions. Use library functions for comparison only.

- Your function should take the specified parameters as inputs and output the specified results.
- Also provide the wrapper/demo code to run all your functions and obtain results. Your code should be self-contained i.e., one should be able to run your code as is without any modifications.
- For Python, if you use any libraries other than numpy, scipy, scikit-image, opencv, pillow, matplotlib, pandas, and default modules, please specify the library that needs to be installed to run your code.
- Along with your code, also submit a PDF with all the **results** (images or numbers) and **inferences** (very important: you may not be explicitly asked to give inferences in each question. You should always include your inferences from what you have observed). Include answers to subjective questions, if any.
- Put all your files (code files and a report PDF) into a **single zip file** and submit the zip file. Name the zip file with your name.

1. **Image Downsampling:**

   (a) Downsample the image `city.png` by a factor of 2, 4 and 5. For the factor of 2, downsample the image by selecting every second pixel in both directions (do not use library functions to downsample). Apply a similar procedure for the other factors. What do you observe and why?

   (b) Now, first filter the image with a spatial domain Gaussian Low Pass Filter before downsampling the image. You can use a 5×5 window and $\sigma = 2$. What difference do you notice, and why? Compare your result with a library function.

   (c) For the factor of 5, find out the optimal window size and σ value that minimize the mean squared error between your output and the library function output.

Function	Image Downsampling
Input	Grayscale image, downsampling factor N
Output	Downsampled image

   (10 + 15 Marks)

2. **Edge Detection:**

   For a given set of grayscale images (Checkerboard.png, Coins.png, MainBuilding.png and flowers.png):

   (a) For each image, detect edges using:

      i. Gradient-based approach (e.g., Sobel or Prewitt operator).

      ii. Laplacian of Gaussian (LoG) operator.

(b) Add Gaussian noise to each image. Apply Gaussian smoothing using a 7×7 window and $\sigma = 3$, then repeat the edge detection using both approaches.

(c) Compare and discuss the edge detection performance qualitatively for:

- Clean vs. noisy images
- Gradient-based vs. LoG edge detectors
- With and without Gaussian smoothing

Function	Edge Detection
Input	Grayscale image (clean or noisy)
Output	Edge map (binary or grayscale)

(10 + 15 + 10 Marks)