Assignment One

Date: 6/2/2025
Submission Deadline: 12/2/2025

Instructions:

Please answer the questions below using Python and follow the guidelines below:

- *Each assignment must be completed individually.*
- *Submit your working code along with comments explaining how to run it. You may use basic algebra packages (e.g., NumPy, SciPy, etc.), but you are not permitted to use packages that directly solve the problems.*

1. **Adding noise to an image:**

   Given the dog image, write a Python code to add Gaussian noise sampled from a Gaussian distribution with a mean of 0 and a standard deviation of 15 to the image.

2. In this task, you will implement the convolution filtering operation using NumPy. As demonstrated in the lecture, convolution can be seen as a sliding window that calculates the weighted sum of pixel values using a flipped kernel. Use the following kernel for the convolution operation:

   Kernel = np.array([[1,0,-1], [2,0,-2], [1,0,-1]])

   Your implementation will:

   i) apply zero-padding to the image,
   ii) flip the kernel both horizontally and vertically, and
   iii) compute the weighted sum of the neighborhood at each pixel.

3. To remove the noise added from step 1 from the dog image, apply convolution using a 7x7 Gaussian filter with a standard deviation (σ) of 1.0.

   Next, we'll use convolution to sharpen the images. Apply the provided filter to the image using convolution to produce a sharpened result.

   sharpening_kernel = np.array([[1, 4, 6, 4, 1], [4, 16, 24, 16, 4], [6, 24, -476, 24, 6], [4, 16, 24, 16, 4], [1, 4, 6, 4, 1],]) * -1.0 / 256.0

4. Imagine you are a grocery store clerk responsible for checking and restocking shelves when items run out. To automate this tedious task, you decide to develop a computer vision system that tracks items on the shelf.

   Fortunately, from your computer vision lectures, you know that convolution can be used for template matching. This technique involves multiplying a smaller template g with regions of a larger image f to measure how similar each region is to the template.

   When implementing this, remember to flip the filter before passing it to your convolution function so that it remains correctly oriented during comparisons. Additionally, subtract the mean value from both the image and the template (whichever you choose) to prevent bias towards higher-intensity (white) regions.

   You are provided with a product template (**template.jpg**) and an image of the shelf (**shelf.jpg**). Using convolution, locate the product on the shelf.

5. In this task, you will write a function to perform edge detection by following these steps:

   - **Smoothing :** Begin by smoothing the images to prevent noise from being interpreted as edges. Use a 9x9 Gaussian kernel filter with σ = 1.4 for this purpose.
   - **Gradient Computation :** Once smoothing is complete, calculate the image gradient in both the horizontal and vertical directions. Compute the gradient magnitude image as $|G| = \sqrt{G_x^2 + G_y^2}$ and the gradient direction as tan−1(Gy/Gx)

   After computing the images at each step, display and label each intermediate step appropriately. You should have a total of four output images: the original image, the smoothed image, the gradient magnitude image, and the gradient direction image.