Edge Detection – Coding Practice Exercises

Based on: First Principles of Computer Vision (FPCV) – Edge Detection Module

General Constraints (Apply to All Exercises)

1. All images must be converted to grayscale before processing.
2. You must implement convolution logic explicitly or via allowed low-level functions.
3. No high-level edge detection APIs are allowed (e.g., cv2.Canny, cv2.Sobel).
4. All thresholds must be chosen and justified experimentally.
5. Visualize intermediate results for every major step.

Allowed Libraries

1. Python 3.x
2. NumPy (array operations, math)
3. SciPy (signal.convolve2d ONLY)
4. OpenCV (cv2.imread, cv2.cvtColor, cv2.imshow / matplotlib display ONLY)
5. Matplotlib (visualization)

Exercise 1: Sobel Edge Detection from Scratch

1. Implement Sobel 3×3 kernels manually.
2. Compute Ix and Iy using convolution.
3. Compute gradient magnitude and orientation.
4. Apply single-threshold and hysteresis-based thresholding.

Forbidden: cv2.Sobel, cv2.filter2D with Sobel kernels.

Exercise 2: Laplacian of Gaussian (LoG) Edge Detection

1. Implement a Gaussian kernel parameterized by σ.
2. Construct a Laplacian-of-Gaussian (LoG) kernel.
3. Convolve image with LoG.
4. Detect edges via zero-crossings with magnitude thresholding.
5. Repeat for multiple σ values and compare results.

Forbidden: cv2.Laplacian, cv2.GaussianBlur combined with Laplacian.

Exercise 3: Simplified Canny Edge Detector

1. Gaussian smoothing.
2. Gradient computation using Sobel kernels.
3. Gradient magnitude and orientation.
4. Non-maximum suppression along gradient direction.
5. Hysteresis thresholding with edge connectivity.

Forbidden: cv2.Canny or any built-in Canny implementation.

Submission Guidelines (Recommended)

1. One Python file per exercise OR one notebook with clear sections.
2. Include figures for all intermediate outputs.
3. Brief written explanation of parameter choices.