**DEPARTMENT OF INFORMATION TECHNOLOGY**

**COURSE CODE: DJS22ITL603**

**COURSE NAME: Image Processing and Computer Vision Laboratory CLASS: T Y B. TECH**

**NAME: Anish Sharma ROLL: I011 DIV: IT1-1**

**EXPERIMENT NO. 7 CO/LO:** Apply Image Enhancement Techniques.

**AIM / OBJECTIVE:** To apply Morphological techniques

**EXERCISE**

**Perform Morphological operations such as dilation, erosion, opening, closing.**

CODE:

import cv2 import numpy as np

from matplotlib import pyplot as plt

# Load the image

image = cv2.imread('cameraman.bmp')

# Convert to grayscale

gray\_image = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

# Define kernel for morphological operations kernel = np.ones((5, 5), np.uint8)

# Dilation

dilated\_image = cv2.dilate(gray\_image, kernel, iterations=1)

# Erosion

eroded\_image = cv2.erode(gray\_image, kernel, iterations=1)

# Opening (erosion followed by dilation)

opened\_image = cv2.morphologyEx(gray\_image, cv2.MORPH\_OPEN, kernel)

# Closing (dilation followed by erosion)

closed\_image = cv2.morphologyEx(gray\_image, cv2.MORPH\_CLOSE, kernel)

# Plot results plt.figure(figsize=(8, 8))

plt.subplot(2, 3, 1)

plt.imshow(cv2.cvtColor(image, cv2.COLOR\_BGR2RGB)) plt.title('Original Coloured Image') plt.axis('off')

plt.subplot(2, 3, 2)

plt.imshow(gray\_image, cmap='gray') plt.title('Original Grayscale Image') plt.axis('off')

plt.subplot(2, 3, 3)

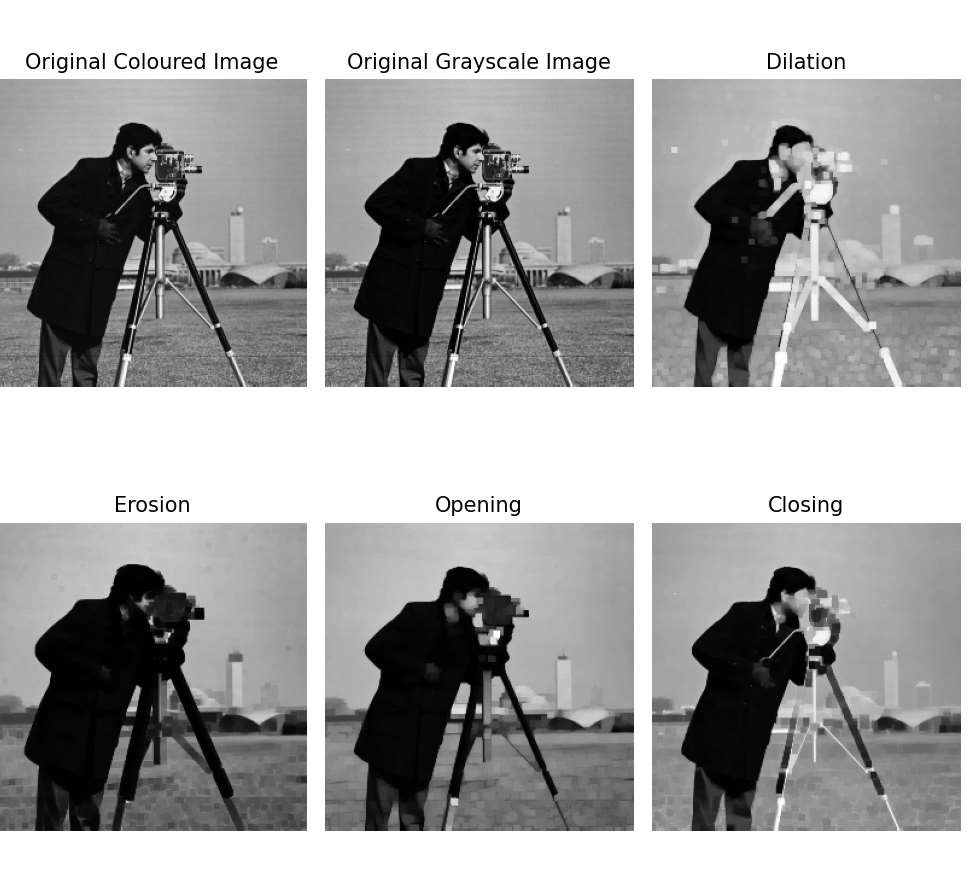
plt.imshow(dilated\_image, cmap='gray') plt.title('Dilation') plt.axis('off')

plt.subplot(2, 3, 4) plt.imshow(eroded\_image, cmap='gray') plt.title('Erosion') plt.axis('off')

plt.subplot(2, 3, 5) plt.imshow(opened\_image, cmap='gray') plt.title('Opening') plt.axis('off')

plt.subplot(2, 3, 6) plt.imshow(closed\_image, cmap='gray') plt.title('Closing') plt.axis('off')

plt.tight\_layout() plt.show() OUTPUT:



**CONCLUSION:**

This experiment demonstrates the effectiveness of morphological operations (dilation, erosion, opening, and closing) in image processing. Dilation expands object boundaries, erosion shrinks them, opening removes small noise, and closing fills small holes in objects. These techniques are widely used in noise reduction, edge enhancement, and shape analysis in computer vision applications.

**REFERENCES:**

**Website References:**

1. Kaggle, “Spatial Filtering OpenCV,” Available: [https://www.kaggle.com/code/bhavinmoriya/spatial-filtering-opencv.](https://www.kaggle.com/code/bhavinmoriya/spatial-filtering-opencv)
2. OpenCV “Image Filtering,” *OpenCV Documentation*. Available: [https://docs.opencv.org/4.x/dd/d6a/tutorial\_js\_filtering.html#:~:text=As%20in%20one%2Ddi mensional%20signals,finding%20edges%20in%20the%20images..](https://docs.opencv.org/4.x/dd/d6a/tutorial_js_filtering.html#%3A~%3Atext%3DAs%20in%20one%2Ddimensional%20signals%2Cfinding%20edges%20in%20the%20images)