Practical Work Handout: Data Structures for Image Processing

# 🎯 Objective

To understand and apply key data structures used in image processing through hands-on programming and conceptual activities.

# 🧪 Task 1: Matrix Representation

- Load an image using Python (OpenCV or PIL).  
- Convert it to grayscale.  
- Print the pixel matrix.  
- Apply thresholding and display the binary image.

Code Example (Python):  
import cv2  
import matplotlib.pyplot as plt  
  
img = cv2.imread('sample.jpg', cv2.IMREAD\_GRAYSCALE)  
\_, binary = cv2.threshold(img, 127, 255, cv2.THRESH\_BINARY)  
  
plt.subplot(1,2,1); plt.imshow(img, cmap='gray'); plt.title('Grayscale')  
plt.subplot(1,2,2); plt.imshow(binary, cmap='gray'); plt.title('Binary')  
plt.show()

# 🧪 Task 2: Chain Code Extraction

import cv2

import numpy as np

import matplotlib.pyplot as plt

#example of chain codes

# Load the image and convert to grayscale

img = cv2.imread('test\_image.jpg')

gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

# Apply binary thresholding

\_, binary = cv2.threshold(gray, 127, 255, cv2.THRESH\_BINARY\_INV)

# Find contours

contours, \_ = cv2.findContours(binary, cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_NONE)

# Draw the contour on original image

contour\_img = img.copy()

cv2.drawContours(contour\_img, contours, -1, (0,255,0), 2)

plt.imshow(cv2.cvtColor(contour\_img, cv2.COLOR\_BGR2RGB))

plt.title('Detected Contour')

plt.axis('off')

plt.show()

# 🧪 Task 3: Image Pyramid Visualization

- Use OpenCV to build a Gaussian Pyramid.  
- Display all levels.

Code Example (Python):  
import cv2  
  
img = cv2.imread('sample.jpg')  
layer = img.copy()  
for i in range(3):  
 layer = cv2.pyrDown(layer)  
 cv2.imshow(f'Level {i+1}', layer)  
cv2.waitKey(0)  
cv2.destroyAllWindows()