**BAHRIA UNIVERSITY, ISLAMABAD**

**Department of Computer Science**

**CEN 444**

**Digital Image Processing**

**Lab Journal 12**

**Student Name: Abdullah Khalil**

**Enrolment No.: 01-134212-007**

[**https://github.com/Abdullah-Khalil123/DIP-LAB**](https://github.com/Abdullah-Khalil123/DIP-LAB)

**Task 1**

def myMorphology(image\_path):

    image = cv2.imread(image\_path, cv2.IMREAD\_GRAYSCALE)

    kernel = np.ones((5, 5), np.uint8)

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

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

    boundaries = closed\_image - eroded\_image

    return boundaries

image\_path = "coins.png"

boundaries = myMorphology(image\_path)

imgGrey = cv2.imread(image\_path, cv2.IMREAD\_GRAYSCALE)

plt.figure(figsize=(10, 5))

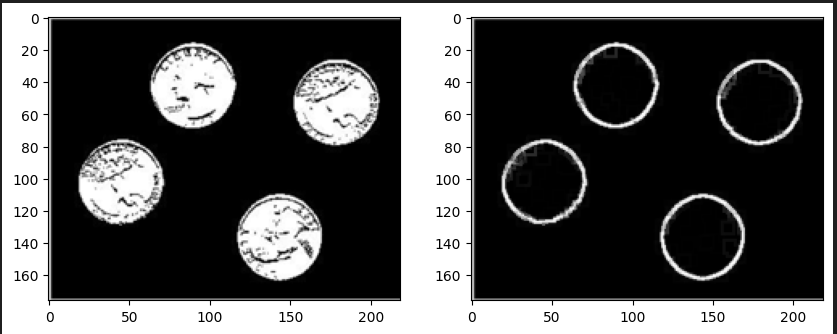
plt.subplot(1, 2, 1)

plt.imshow(imgGrey, cmap="gray")

plt.subplot(1, 2, 2)

plt.imshow(boundaries, cmap="gray")

plt.show()



**Task 2**

image = cv2.imread('lines.png', cv2.IMREAD\_GRAYSCALE)

horizontal\_kernel = cv2.getStructuringElement(cv2.MORPH\_RECT, (11, 3))

vertical\_kernel = cv2.getStructuringElement(cv2.MORPH\_RECT, (3, 11))

horizontal\_opening = cv2.morphologyEx(image, cv2.MORPH\_OPEN, horizontal\_kernel)

vertical\_opening = cv2.morphologyEx(image, cv2.MORPH\_OPEN, vertical\_kernel)

plt.figure(figsize=(15, 5))

plt.subplot(1, 3, 1)

plt.imshow(image, cmap='gray')

plt.subplot(1, 3, 2)

plt.imshow(horizontal\_opening, cmap='gray')

plt.subplot(1, 3, 3)

plt.imshow(vertical\_opening, cmap='gray')

plt.show()

A black and white graph

Description automatically generated

**Task 3**

image = cv2.imread('image.png', cv2.IMREAD\_GRAYSCALE)

\_, binary\_image = cv2.threshold(image, 0, 255, cv2.THRESH\_BINARY\_INV + cv2.THRESH\_OTSU)

kernel\_line = cv2.getStructuringElement(cv2.MORPH\_RECT, (50, 1))

dilated\_line = cv2.dilate(binary\_image, kernel\_line)

num\_lines, \_ = cv2.connectedComponents(dilated\_line)

kernel\_word = cv2.getStructuringElement(cv2.MORPH\_RECT, (10, 1))

dilated\_word = cv2.dilate(binary\_image, kernel\_word)

num\_words, \_ = cv2.connectedComponents(dilated\_word)

print(f"Number of lines: {num\_lines}")

print(f"Number of words: {num\_words}")

cv2.imshow("Binarized Image", binary\_image)

cv2.imshow("Dilated Lines", dilated\_line)

cv2.imshow("Dilated Words", dilated\_word)

cv2.waitKey(0)

cv2.destroyAllWindows()

A screenshot of a computer

Description automatically generated

**Task 4**

Create an image through paint, snipping or even your mobile. The image should have a white background and your name written in it as foreground. Detect the text in the image and make a bounding box around it is using morphological operations.

import cv2

import numpy as np

import matplotlib.pyplot as plt

image = cv2.imread('name.png', cv2.IMREAD\_GRAYSCALE)

\_, binary\_image = cv2.threshold(image, 0, 255, cv2.THRESH\_BINARY\_INV + cv2.THRESH\_OTSU)

kernel = cv2.getStructuringElement(cv2.MORPH\_RECT, (15, 15))

dilated\_image = cv2.dilate(binary\_image, kernel)

contours, \_ = cv2.findContours(dilated\_image, cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_SIMPLE)

bounding\_box\_image = cv2.cvtColor(image, cv2.COLOR\_GRAY2BGR)

for contour in contours:

    x, y, w, h = cv2.boundingRect(contour)

    cv2.rectangle(bounding\_box\_image, (x, y), (x+w, y+h), (0, 255, 0), 2)

fig, axes = plt.subplots(1, 3, figsize=(15, 5))

axes[0].imshow(binary\_image, cmap='gray')

axes[0].set\_title('Binary Image')

axes[0].axis('off')

axes[1].imshow(dilated\_image, cmap='gray')

axes[1].set\_title('Dilated Image')

axes[1].axis('off')

axes[2].imshow(cv2.cvtColor(bounding\_box\_image, cv2.COLOR\_BGR2RGB))

axes[2].set\_title('Bounding Box Image')

axes[2].axis('off')

plt.tight\_layout()

plt.show()

**A white object on a black background

Description automatically generated**