

BAHRIA UNIVERSITY, ISLAMABAD
Department of Computer Science

CEN 444
Digital Image Processing
Lab Journal 12

Student Name: M MUNEEB AHMED
KIANI_____

Enrolment No.: _01-135212-063_____

Title: Morphological Image Processing

Objectives: To introduce fundamental morphological operations such as dilation and erosion and using them in combination: opening and closing. To perform morphological operations on an image in Python and extracting image components that are useful in representing and describing shapes of a region.

Tools Used: Python

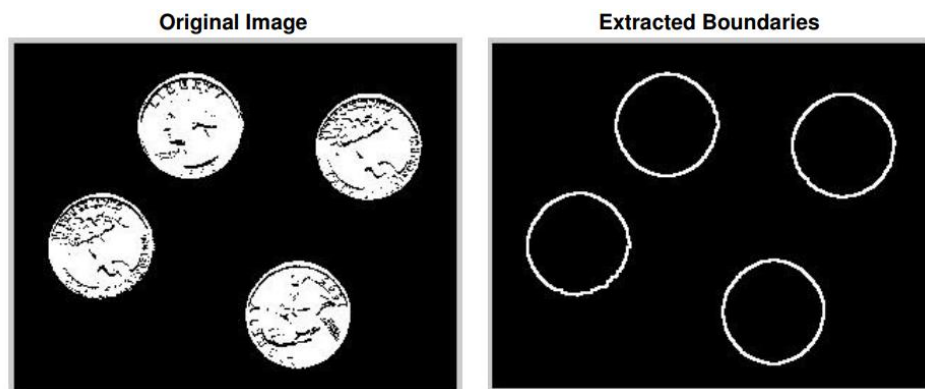


Task 1

Read the image 'eight.tif'. Write a function named 'myMorphology' to extract the boundaries of coins from the read image.

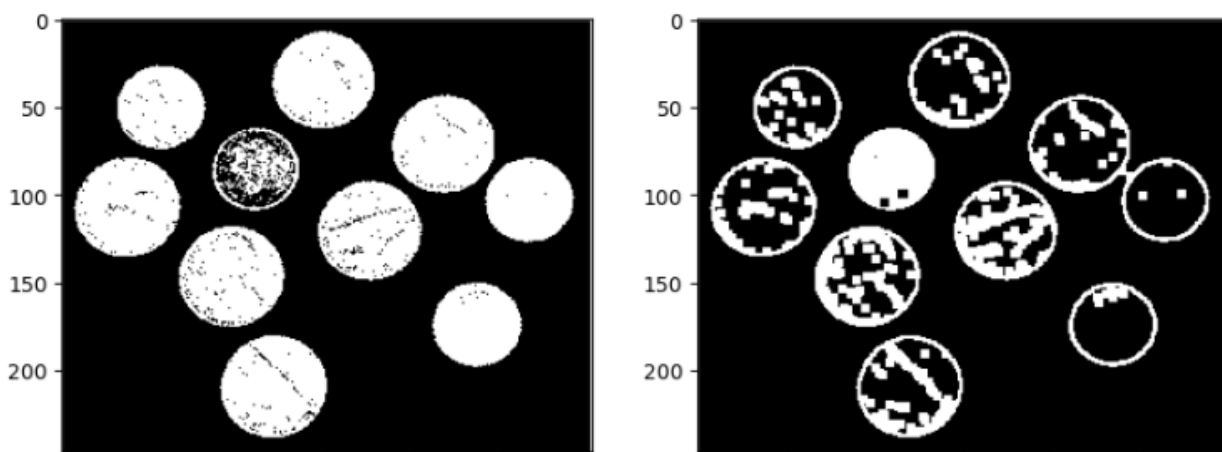
[HINTS]:

1. First close the image then perform erosion.
2. Take the difference of two images to find boundaries.



```
import cv2
import numpy as np
import matplotlib.pyplot as plt

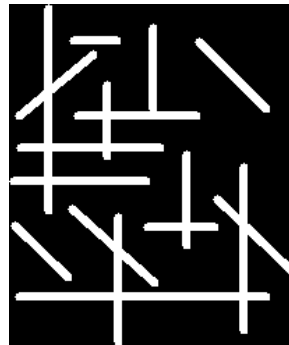
def myMorphology(image_path):
    img = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
    _, binary_img = cv2.threshold(img, 127, 255, cv2.THRESH_BINARY)
    kernel = np.ones((5, 5), np.uint8)
    closed_img = cv2.morphologyEx(binary_img, cv2.MORPH_CLOSE, kernel)
    eroded_img = cv2.erode(binary_img, kernel, iterations=1)
    boundaries = cv2.subtract(closed_img, eroded_img)
    plt.figure(figsize=(10, 5))
    plt.subplot(1, 2, 1), plt.imshow(binary_img, cmap='gray')
    plt.subplot(1, 2, 2), plt.imshow(boundaries, cmap='gray')
    plt.axis()
    plt.show()
    return boundaries
boundaries = myMorphology('eight.JPG')
```



Task 2

Read the image 'lines.png'. Use the opening operator to separate horizontal and vertical lines.

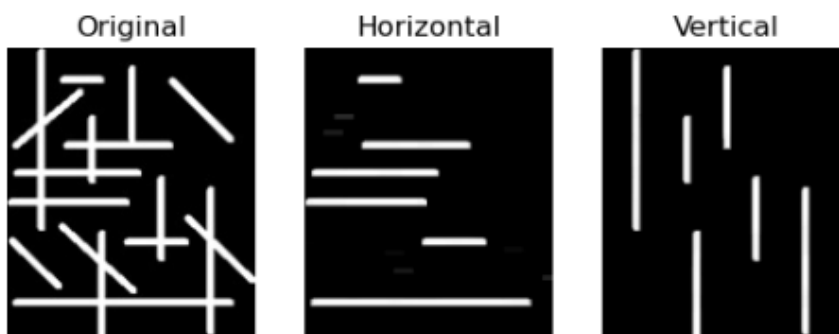
[HINTS]: Experiment with structuring elements of sizes 7x3, 9x3, 11x3 etc. to remove horizontal lines. Use the transpose of these structuring elements to eliminate vertical lines.



```
import cv2
import matplotlib.pyplot as plt

image = cv2.imread('lines.jpg', cv2.IMREAD_GRAYSCALE)
horizontal = cv2.morphologyEx(image, cv2.MORPH_OPEN, cv2.getStructuringElement(cv2.MORPH_RECT, (11, 3)))
vertical = cv2.morphologyEx(image, cv2.MORPH_OPEN, cv2.getStructuringElement(cv2.MORPH_RECT, (3, 11)))

plt.subplot(131), plt.imshow(image, cmap='gray'),
plt.title("Original"), plt.axis('off')
plt.subplot(132), plt.imshow(horizontal, cmap='gray'),
plt.title("Horizontal"), plt.axis('off')
plt.subplot(133), plt.imshow(vertical, cmap='gray'),
plt.title("Vertical"), plt.axis('off')
plt.show()
```



Task 3

You are provided with a printed document image (Image.png). You need to find the approximate number of lines and words in the given image.

Hints:

- Binarize the image
- dilation with a horizontal structuring element to merge all characters in a line.

- Apply connected component labeling algorithm (connected component) to find the number of lines.
- Use a smaller horizontal structuring element to merge characters in a word together and again use the CC labeling algorithm to find the number of words in the image.

```
import cv2
import numpy as np

def count_lines_and_words(image_path):
    img = cv2.imread(image_path, 0)
    _, binary = cv2.threshold(img, 127, 255, cv2.THRESH_BINARY_INV)
    lines = cv2.dilate(binary, np.ones((1, 15), np.uint8))
    words = cv2.dilate(binary, np.ones((1, 5), np.uint8))
    num_lines, _ = cv2.connectedComponents(lines)
    num_words, _ = cv2.connectedComponents(words)
    return num_lines - 1, num_words - 1

lines, words = count_lines_and_words('lines.jpg')
print("Lines:", lines, "Words:", words)
```

Lines: 1 Words: 1

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
image = cv2.imread('MUNEED.png', 0)
_, binary = cv2.threshold(image, 127, 255, cv2.THRESH_BINARY_INV)
dilated = cv2.dilate(binary, None, iterations=2)
contours, _ = cv2.findContours(dilated, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
output = cv2.cvtColor(image, cv2.COLOR_GRAY2BGR)
for contour in contours:
    x, y, w, h = cv2.boundingRect(contour)
    cv2.rectangle(output, (x, y), (x + w, y + h), (0, 0, 255), 2)
cv2.imshow('Text Detection', output)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

M MUNEER AHMED KIANI