Image Understanding and Processing (OpenCv-Python)

Lab Exercise – 10

Year 4 **Semester 1, 2024**

Goal

- Learn different morphological functions like cv2.morphologyEx( ) and cv2.getStructuringElement( )
- Learn how to extract horizontal and vertical lines by using morphological operations
- Learn how to extract circles by using morphological operations
- Learn how to apply hit or miss transformation to detect shape

The cv2.morphologyEx( ) function can be applied for few morphological transformations such as: Opening, Closing, Morphological Gradient, Top Hat and Black Hat.

Exercise 01:

Opening operation open up the gap between the objects connected by thin protrusions that are of size less than that of the structuring element. Below is the input image and output image. Write a small code to enclose the gaps between objects.

Exercise 02:

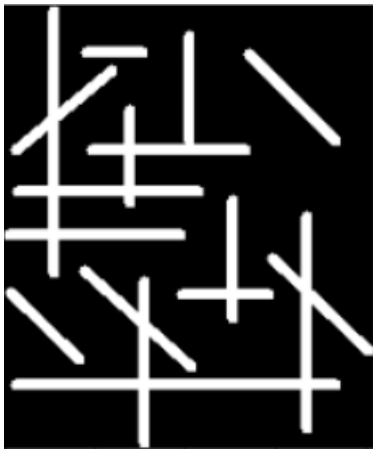
Closing operation closes the holes/gaps present in the object while keeping the initial object size the same. Below is the input image and output image. Write a small code to enclose the gaps between objects.

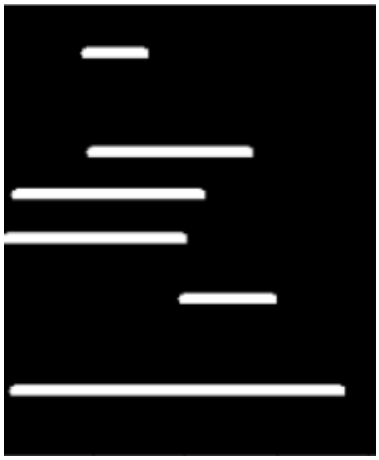
Exercise 03:

Try to obtain the output image from the original image

Exercise 04:

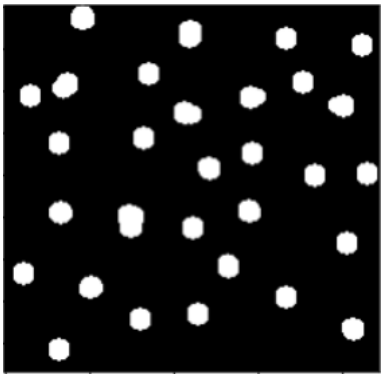
The image below shows another binary image. Suppose that this time we wish to separately extract the horizontal and vertical lines. Try to obtain the output image from the original image.





Exercise 05:

The image below shows another binary image. It contains a mixture of circles and lines. Suppose that we want to separate out the circles from the lines, so that they can be counted. Try to obtain the output image from the original image.



Hit or miss transform:

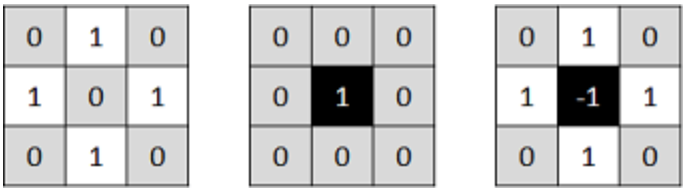
The Hit-or-Miss transformation is useful to find patterns in binary images. In particular, it finds those pixels whose neighborhood matches the shape of a first structuring element B1 while not matching the shape of a second structuring element B2 at the same time. Execute the following code:

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

input_image = np.array((
    [0, 0, 0, 0, 0, 0, 0, 0],
    [0, 255, 255, 255, 0, 0, 0, 255],
    [0, 255, 255, 255, 0, 0, 0, 0],
    [0, 255, 255, 255, 0, 255, 0, 0],
    [0, 0, 255, 0, 0, 0, 0, 0],
    [0, 0, 255, 0, 0, 255, 255, 0],
    [0,255, 0, 255, 0, 0, 255, 0],
    [0, 255, 255, 255, 0, 0, 0, 0]), dtype="uint8")
kernel = np.array((
        [0, 1, 0],
        [1, -1, 1],
        [0, 1, 0]), dtype="int")


output_image = cv2.morphologyEx(input_image, cv2.MORPH_HITMISS, kernel)


plt.imshow(input_image, cmap= 'gray')
plt.show()

plt.imshow(kernel, cmap= 'gray')
plt.show()

plt.imshow(output_image, cmap= 'gray')
plt.show()
```

The hit-or-miss operation comprises three steps:

1. Erode image A with structuring element B1.
2. Erode the complement of image A (Ac) with structuring element B2.
3. AND results from step 1 and step 2.

The structuring elements B1 and B2 can be combined into a single element B. Let's see an example:



Structuring elements (kernels). Left: kernel to 'hit'. Middle: kernel to 'miss'. Right: final combined kernel

Exercise 06:

Find the output results of applying different kernels to the same input image used before: