Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Department of Computer Science and Engineering (Data Science)
Image Processing and Computer Vision I (DJ19DSL603)

Name: Yash Thakar **SAP ID.: 60009210205**

Experiment 7

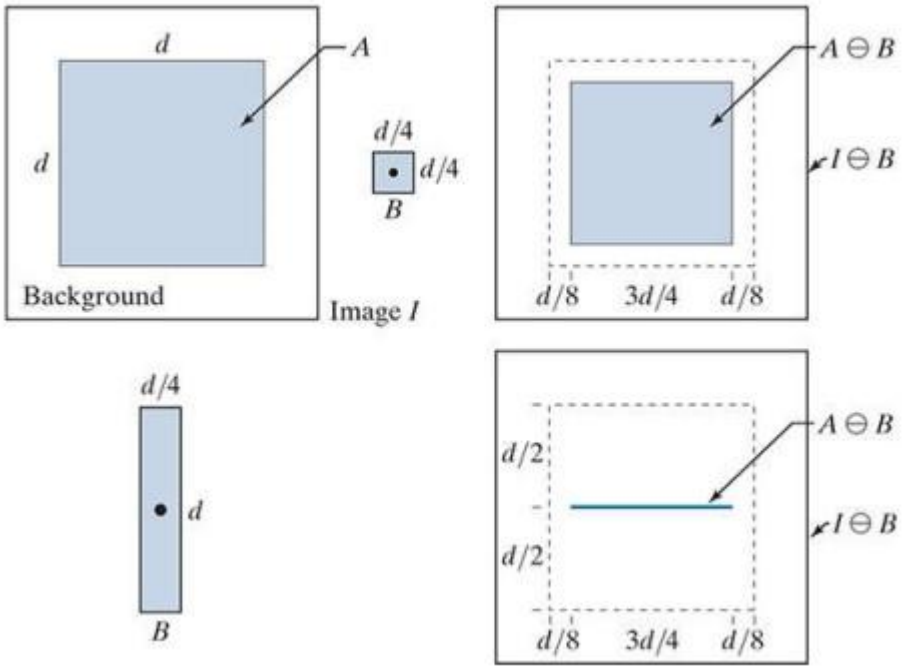
Aim: To perform morphological operations on image (erosion, dilation, opening, closing and hit and miss transform) **Theory:**

1. **Erosion:**

Morphological expressions are written in terms of structuring elements and a set, A, of foreground pixels, or in terms of structuring elements and an image, I, that contains A. We consider the former approach first. Erosion of A by B is defined as:

$$A \ominus B = \{z \mid (B)_z \subseteq A\}$$

In words, this equation indicates that the erosion of A by B is the set of all points z such that B, translated by z, is contained in A. (Remember, displacement is defined with respect to the origin of B. The result of erosion is controlled by the shape of the structuring element. The image is eroded by two different structuring elements (B) giving the outputs as seen:



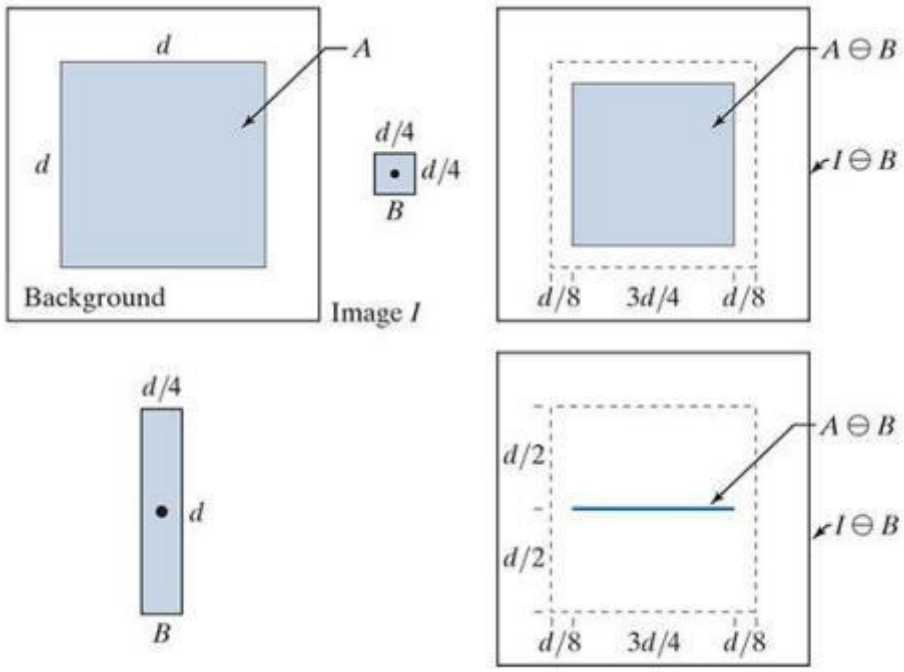
Department of Computer Science and Engineering (Data Science)
Image Processing and Computer Vision I (DJ19DSL603)

2. Dilation:

Dilation of A by B is defined as:

$$A \oplus B = \left\{ z \middle| [(\hat{B})_z \cap A] \subseteq A \right\}$$

The dilation of A by B then is the set of all displacements, z, such that the foreground elements of overlap at least one element of A. (Remember, z is the displacement of the origin of $\hat{B}$. The image is dilated by two different structuring elements (B) giving the outputs as seen:

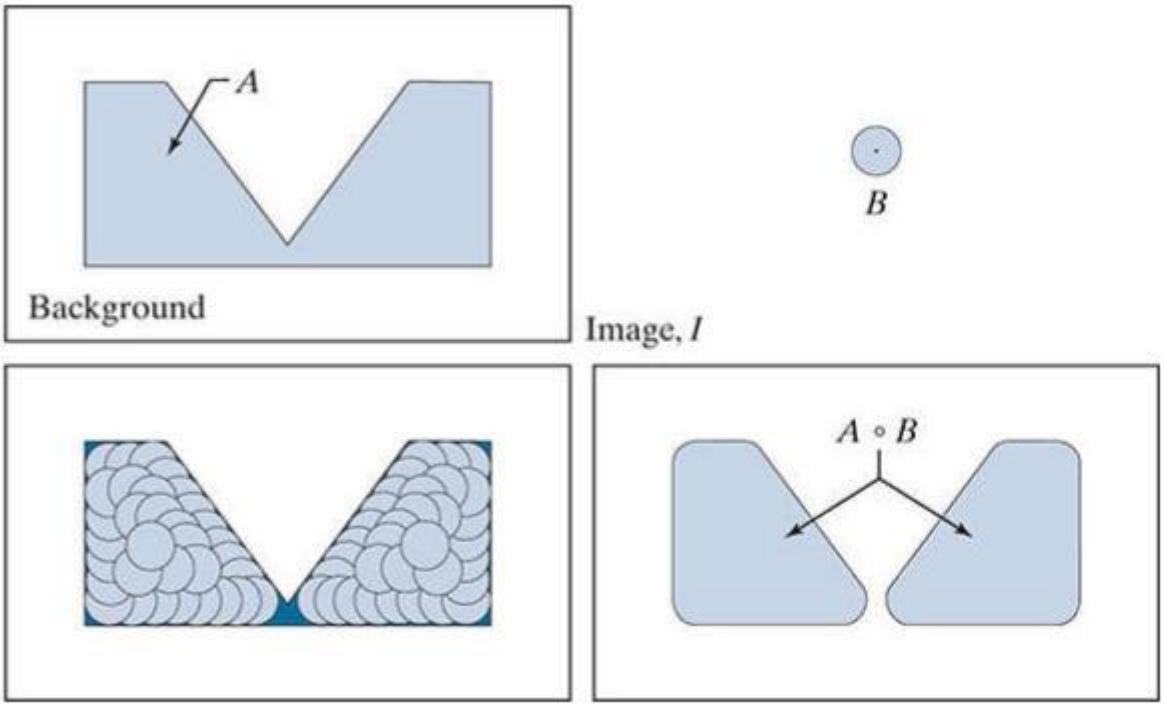


3. Opening:

Opening generally smoothens the contour of an object, breaks narrow isthmuses, and eliminates thin protrusions. The opening of set A by structuring element B is defined as:

$$A \circ B = (A \ominus B) \oplus B$$

Thus, the opening A by B is the erosion of A by B, followed by a dilation of the result by B. The opening of A by B is the union of all the translations of B so that B fits entirely in A

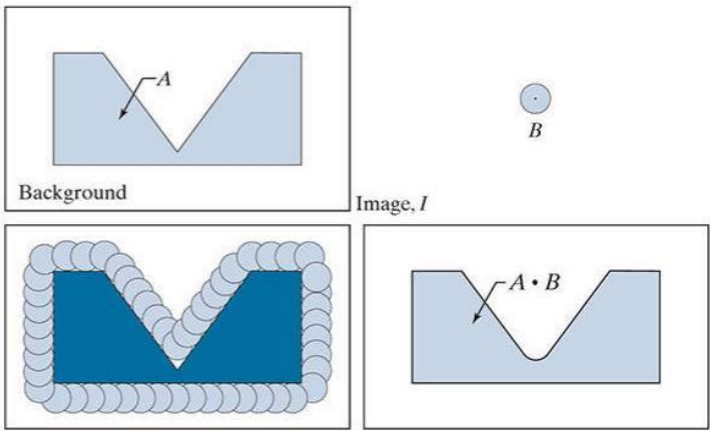


4. Closing:

The closing of set A by structuring element B is defined as:

$$A \bullet B = (A \oplus B) \ominus B$$

The closing of A by B is simply the dilation of A by B, followed by erosion of the result by B. The opening of A by B is the union of all the translations of B so that B fits entirely in A.



5. Hit and miss transform:

The morphological hit-or-miss transform (HMT) is a basic tool for shape detection. Let I be a binary image composed of foreground (A) and background pixels, respectively. Unlike the morphological methods discussed thus far, the HMT utilizes two structuring elements: B_1 for detecting shapes in the foreground, and B_2 for detecting shapes in the background. The HMT of image I is defined as

$$I \circledast B_{1,2} = \{z \mid (B_1)_z \subseteq A \text{ and } (B_2)_z \subseteq A^c\}$$
$$= (A \ominus B_1) \cap (A^c \ominus B_2)$$

In words, this equation says that the morphological HMT is the set of translations, z, of structuring elements B_1 and B_2 such that, simultaneously, B_1 found a match in the foreground (i.e., is contained in A) and B_2 found a match in the background (i.e., is contained in A^c).

Department of Computer Science and Engineering (Data Science)

Image Processing and Computer Vision I (DJ19DSL603)

Lab Assignments to complete in this session

Problem Statement: Develop a Python program utilizing the OpenCV library to morph the images in spatial domain using the morphological operations explained. The program should address the following tasks:

1. Read any image from COVID 19 Image Dataset. Take it as object **Dataset Link:** [Covid-19 Image Dataset]
2. Create a structuring element of any form that is much smaller in size than the image considered in step 1.
3. Display the before & after image(s) used in every task below:
    a. Apply erosion and show the before and after image
    b. Apply dilation and show the before and after image
    c. Apply opening and show the before and after image
    d. Apply closing and show the before and after image
    e. Apply hit and miss transformation and show the before and after image. Take the structuring element to be negation of the one used till now.

The solution to the operations performed must be produced by scratch coding without the use of built in OpenCV methods.

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

def perform_morphological_operations(image_path):
  try:
    # Load grayscale image
    img = cv2.imread(image_path, 0)

    # Create structuring element
    se = np.ones((3, 3), dtype="uint8")
```

```
    # Morphological operations
    eroded_img = cv2.erode(img, se)
    dilated_img = cv2.dilate(img, se)
    opened_img = cv2.morphologyEx(img, cv2.MORPH_OPEN, se)
    closed_img = cv2.morphologyEx(img, cv2.MORPH_CLOSE, se)

    # Hit-or-Miss Transform
    se_neg = cv2.bitwise_not(se)
    hit_or_miss_img = cv2.morphologyEx(img,
cv2.MORPH_HITMISS, se_neg)

    # Create Matplotlib figure with subplots
    fig, axes = plt.subplots(2, 3, figsize=(12, 6))

    # Original image
    axes[0, 0].imshow(img, cmap="gray")
    axes[0, 0].set_title("Original")
    axes[0, 0].axis("off")

    # Eroded image
    axes[0, 1].imshow(eroded_img, cmap="gray")
    axes[0, 1].set_title("Eroded")
    axes[0, 1].axis("off")

    # Dilated image
    axes[0, 2].imshow(dilated_img, cmap="gray")
    axes[0, 2].set_title("Dilated")
    axes[0, 2].axis("off")

    # Opened image
```

```
    axes[1, 0].imshow(opened_img, cmap="gray")
    axes[1, 0].set_title("Opened")
    axes[1, 0].axis("off")

    # Closed image
    axes[1, 1].imshow(closed_img, cmap="gray")
    axes[1, 1].set_title("Closed")
    axes[1, 1].axis("off")

    # Hit-or-Miss image
    axes[1, 2].imshow(hit_or_miss_img, cmap="gray")
    axes[1, 2].set_title("Hit-or-Miss")
    axes[1, 2].axis("off")

    # Adjust layout
    fig.suptitle("Morphological Operations Results",
fontsize=14)
    plt.tight_layout()

  except (FileNotFoundError, cv2.error) as e:
    print("Error:", e)

# Replace with your actual image path
image_path = "images/covid19_data_samples/08.jpeg"
perform_morphological_operations(image_path)

plt.show()
```

Department of Computer Science and Engineering (Data Science)

Image Processing and Computer Vision I (DJ19DSL603)