**Aim:** Simulate Dilation, Erosion, Opening and Closing Operation on Images.

**Theory:-**

* Dilation, erosion, opening, and closing are morphological operations used in image processing to manipulate the shape and size of objects in an image.

**Dilation:-**

* Dilation expands the boundaries of objects in an image by adding pixels to the object's boundaries. It is achieved by sliding a structuring element over the image and assigning the maximum pixel value within the neighborhood of each pixel.

**Erosion:-**

* Erosion shrinks the boundaries of objects in an image by removing pixels from the object's boundaries. It is achieved by sliding a structuring element over the image and assigning the minimum pixel value within the neighborhood of each pixel.

**Opening:-**

* Opening is an erosion operation followed by a dilation operation. It is used to remove noise and small objects from the image while preserving the larger objects' shapes.

**Closing:-**

* Closing is a dilation operation followed by an erosion operation. It is used to close small gaps and holes within objects while preserving the overall object shapes.

**Programm:-**

import cv2

import numpy as np

image\_path = "./Images.jpg"

image = cv2.imread(image\_path, 0)

if image is None:

    raise FileNotFoundError(f"The image '{image\_path}' could not be loaded. Check the file path.")

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

erosion = cv2.erode(image, kernel, *iterations*=1)

dilation = cv2.dilate(image, kernel, *iterations*=1)

opening = cv2.morphologyEx(image, cv2.MORPH\_OPEN, kernel)

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

cv2.imwrite("erosion.jpg", erosion)

cv2.imwrite("dilation.jpg", dilation)

cv2.imwrite("opening.jpg", opening)

cv2.imwrite("closing.jpg", closing)

print("Images saved: 'erosion.jpg', 'dilation.jpg', 'opening.jpg', 'closing.jpg'")

**Output:-**

|  |  |  |
| --- | --- | --- |
| **Original Image** | **Erosion Image** | **Dilation Image** |
|  |  |  |
| **Opening Image** | **Closing Image** |  |
|  |  |  |

**Conclusion :-**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_