DIP - Morphological Image Processing

Rifat Hosen

August 2022

Morphological,Dilation, Erosion, Opening, Closing

1 Explanation

Erosion : Morphological erosion removes floating pixels and thin lines so that only substantive objects remain. Remaining lines appear thinner and shapes appear smaller.

Dilation : Morphological dilation makes objects more visible and fills in small holes in objects. Lines appear thicker, and filled shapes appear larger.

Opening : The opening operation erodes an image and then dilates the eroded image, using the same structuring element for both operations.

Closing : The closing operation dilates an image and then erodes the dilated image, using the same structuring element for both operations.

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


def read_image():
    path = "/media/rifat/STUDY/4-1/LAB/Image_Processing/image/font.png"
    img = plt.imread(path)
    gray = cv.cvtColor(img,cv.COLOR_RGB2GRAY)
    gray = gray*255;
    th,binary = cv.threshold(gray,160,255,cv.THRESH_BINARY);
    return img,gray,binary


def morphological_change(binary, kernel):
    img_erosion = cv.erode(binary, kernel, iterations=1)
    img_dilation = cv.dilate(binary, kernel, iterations=1)
    opening = cv.morphologyEx(binary, cv.MORPH_OPEN, kernel)
    closing = cv.morphologyEx(binary, cv.MORPH_CLOSE, kernel)

    plt.figure(figsize=(10,6))

    plt.subplot(2,2,1)
    plt.imshow(img_erosion,cmap = 'gray')
    plt.title("erosion")

    plt.subplot(2,2,2)
    plt.imshow(img_dilation,cmap = 'gray')
    plt.title("dilation")

    plt.subplot(2,2,3)
    plt.imshow(opening,cmap = ’gray’)
    plt.title(”Opening”)

    plt.subplot(2,2,4)
    plt.imshow(closing,cmap = ’gray’)
    plt.title(”Closing”)

    plt.show()



if __name__ == ”__main__”:
    img,gray,binary = read_image()

    print(”Using␣kernel␣1”)
    kernel1 = np.ones((5, 5), np.uint8)
    morphological_change(binary, kernel)

    print(”Using␣kernel␣2”)
    kernel2 = np.array([[1, 0, 0, 0, 1],
                        [0, 1, 1, 1, 0],
                        [0, 1, 1, 1, 0],
                        [0, 1, 1, 1, 0],
                        [1, 0, 0, 0, 1]
                       ],np.uint8)
    morphological_change(binary, kernel2)


    print(”Using␣kernel␣3”)
    kernel3 = np.array([[1, 1, 1, 1, 1],
                        [1, 0, 0, 0, 1],
                        [1, 0, 1, 0, 1],
                        [1, 0, 0, 0, 1],
                        [1, 1, 1, 1, 1]
                       ],np.uint8)
    morphological_change(binary, kernel3)
```