Bansilal Ramnath Agarwal Charitable Trust's

Vishwakarma Institute of Information Technology

Department of Artificial Intelligence and Data Science

Name: Siddhesh Dilip Khairnar		
Class: TY	Division: B	Roll No: 372028
Semester: V	Academic Year: 2023-2024	
Subject Name & Code: Image Processing: ADUA31205(B)		
Title of Assignment: Perform various morphological operations on an image. (Erosion, Dilation, Skeletonizing, removing small objects, Extracting boundaries etc.)		
Date of Performance: 29-08-2023	Date of Submission: 15-09-2023	

ASSIGNMENT NO. 5

Name : Siddhesh Dilip Khairnar
PRN no : 2210398
Roll no : 372028

Page No.
Date

IP Assignment no 5

Aim : Perform various morphological operation on an image (Erosion, Dilation, skeletonizing, removing small object, extracting boundaries etc)

Learning Objective : →

i) Explore the theory & practical implementation of morphological operation like erosion, dilation, opening, closing & morphological gradient etc.

ii) Gain knowledge about skeletonization algorithm and their application such as thinning the representation of object in an image

Theory : →

Performing various morphological operation on an image using python and opencv, which is a popular computer vision library. Make sure you have opencv installed before proceeding :→

1) Import necessary libraries :→

```
import cv2
import numpy as np
```

2) Load your image

```
image = cv2.imread('')
```

3) **Erosion & Dilation :→**

Erosion & dilation are basic morphological operation you can perform them using opencv erode & dilate function. for example :→

```
kernel = np.ones((5,5), np.uint8)
erosion = cv2.erode(image, kernel, iteration=1)
dilation = cv2.dilate(image, kernel, iteration=1)
```

4) Skeletonizing :
skeletonizing can be achieved using the 'skeletonize' function from the Scikit-image library. you may need to install.
from skimage.morphology import skeletonize
skeleton = skeletonize(image)

5) Removing small objects :→
you can use morphological ~~operator~~ operations to remove small object in an image. for example, to remove small white object on a black background, you can perform opening.
opened = cv2.morphologyEx(image, cv2.MORPH_OPEN, kernel)

6) Extracting Boundaries :→
To extract boundaries, you can use morphological gradient :→
gradient = cv2.morphologyEx(image, cv2.MORPH_GRADIENT, kernel

2) Display & save result.

Conclusion :→ performing various ~~morpho~~ morphological operation on an image is a common task in image processing & computer vision. various operation can be done using libraries like opencv & scikit image in Python. These operation are useful for task like image preprocessing, object detection & image segmentation.

Program Code:

```
import cv2
import numpy as np

# Reading the input image
img = cv2.imread("C:/Users/asus/Downloads/EvXjoAkUYAE5K7O.jpg", 0)

# Taking a matrix of size 5 as the kernel
kernel = np.ones((5, 5), np.uint8)

# The first parameter is the original image,
# the kernel is the matrix with which the image is
# convolved, and the third parameter is the number
# of iterations, which will determine how much
# you want to erode/dilate a given image.
img_erosion = cv2.erode(img, kernel, iterations=1)
img_dilation = cv2.dilate(img, kernel, iterations=1)

cv2.imshow('Input', img)
cv2.imshow('Erosion', img_erosion)
cv2.imshow('Dilation', img_dilation)
cv2.waitKey(0)

# Threshold the image
ret, img = cv2.threshold(img, 127, 255, 0)

# Step 1: Create an empty skeleton
size = np.size(img)
skel = np.zeros(img.shape, np.uint8)

# Get a Cross Shaped Kernel
element = cv2.getStructuringElement(cv2.MORPH_CROSS, (3, 3))

# Repeat steps 2-4
while True:
    # Step 2: Open the image
    opening = cv2.morphologyEx(img, cv2.MORPH_OPEN, element)

    # Step 3: Subtract open from the original image
    temp = cv2.subtract(img, opening)

    # Step 4: Erode the original image and refine the skeleton
    eroded = cv2.erode(img, element)
    skel = cv2.bitwise_or(skel, temp)
    img = eroded.copy()
```

```
    # Step 5: If there are no white pixels left
    # i.e., the image has been completely eroded, quit the loop
    if cv2.countNonZero(img) == 0:
        break

# Displaying the final skeleton
cv2.imshow("Skeleton", skel)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Output: