**Department of Computing**

**Digital Image Processing**

**Class: BSCS-9ABC**

**Lab 9**

**Submitted By: Fatima Seemab and** [**Mahum Samar**](mailto:msamar.bscs19seecs@seecs.edu.pk)

**CMS ID: 291310,290647**

**BSCS 9B**

Image Morphology

**Note:- You have to submit this lab in groups of two,just make one submission per group and write the names of both group members in it.But both the member should have the complete understanding of the lab because viva will be taken in next lab.**

**Task #1: Effect of binary opening and closing**

Consider the following signatures. First, apply binary erosion and dilation. Later, apply binary opening and closing and analyze the difference with respect to erosion and dilation. You may use appropriate functions available in various Python packages.



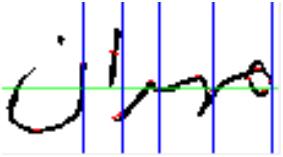
**CODE:**

|  |
| --- |
| def opening(img,kernel):  #dilation after erosion    erosion = cv.erode(img,kernel,iterations = 1)    result = cv.dilate(erosion,kernel,iterations = 1)    cv.imwrite("opening.png",result)  def closing(img,kernel):  # erosion after dilation    dilation = cv.dilate(img,kernel,iterations = 1)    closing = cv.erode(dilation,kernel,iterations = 1)    cv.imwrite("closing.png",closing)  import cv2 as cv  import numpy as np  img = cv.imread('Lara\_Craft.png')  kernel = np.ones((5,5),np.uint8)  # built in function for erosion  erosion = cv.erode(img,kernel,iterations = 1)  # built in function for dilation  dilation = cv.dilate(img,kernel,iterations = 1)  opening(img,kernel)  closing(img,kernel)  # built in function for erosion  cv.imwrite("erosion.png",erosion)  cv.imwrite("dilation.png",dilation)  cv.waitkey(0) |

|  |  |
| --- | --- |
| Erosion | Dilation |
|  |  |
| Opening | Closing |
|  |  |

**Task #2: Application**

Can you segment the foreground (Urdu characters) in the following image from its background (lines and other background)? Consider using proper binarization methods in conjunction with various morphological operations. Does your algorithm face any particular difficult areas while segmenting? Again, use of built-in functions is allowed. Note: Binarization threshold may result in ‘holes’ in the foreground which might be filled with some morphological operations.



Please make sure that you submit the following contents:-

A single lab reporting per group,containing the code and screenshots of output.

**CODE:**

|  |
| --- |
| import cv2  def opening(img,kernel):    erosion = cv.erode(img,kernel,iterations = 1)    result = cv.dilate(erosion,kernel,iterations = 1)    cv.imwrite("opening+erosion.png",result)    return result  def closing(img,kernel):    dilation = cv.dilate(img,kernel,iterations = 1)    closing = cv.erode(dilation,kernel,iterations = 1)    cv.imwrite("Mehran\_closing.png",closing)  originalImage = cv2.imread('Mehran.PNG')  #Converting to gray scale  grayImage = cv2.cvtColor(originalImage, cv2.COLOR\_BGR2GRAY)  #Converting to binary  (thresh, blackAndWhiteImage) = cv2.threshold(grayImage, 35, 255, cv2.THRESH\_BINARY)  #For different operations to restore the image, different kernel are assigned  kernel = np.ones((9,9),np.uint8)  kernel2 = np.ones((3,3),np.uint8)  kernel3 = np.ones((5,5),np.uint8)  #Series of operation to restore the image  erosion = cv.erode(blackAndWhiteImage,kernel,iterations = 1)  result=opening(erosion,kernel2)  dilation = cv.dilate(result,kernel3,iterations = 1)  cv.imwrite("result.png",dilation)  cv2.waitKey(0) |

**OUTPUT**

|  |
| --- |
|  |