**ASSIGNMENT 3**Logo, company name

Description automatically generated

**Digital Image Processing**

**Sir Muhammad Daud Abdullah Asif**

**Submitted By**

Fatima Seemab

291310

BSCS 9B

21 December 2021

**TASK 1**

**CODE:**

| #packages are imported  import numpy  import cv2  from skimage.morphology import disk  #reading and converting the image to grayscale and then to black and white  image=cv2.imread('A1\_1.png')  img\_gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)  thresh, blackAndWhiteImage = cv2.threshold(img\_gray, 110, 255, cv2.THRESH\_BINARY)  #innverting the image to perform erosion  invert = cv2.bitwise\_not(blackAndWhiteImage)  #morphological operations  erosion=cv2.erode(invert,disk(11),iterations=1)  dilation=cv2.dilate(erosion,disk(1),iterations=1)  num\_labels,labels = cv2.connectedComponents(dilation)  print(“num\_labels”,num\_labels-1)  cv2.imwrite("A1\_1output.png",dilation) |
| --- |

**RESULTS:**

| INPUT IMAGE | OUTPUT IMAGE |
| --- | --- |
|  |  |

**Number of coins**

|  |
| --- |

**TASK 2:**

**Part 1**

**CODE:**

| #packages are imported  import numpy  import cv2  from skimage.morphology import disk  #reading the image and getting its dimension  image=cv2.imread('A2\_1.png')  x=image.shape[0]  y=image.shape[1]  #converting image to grayscale  img\_gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)  #for performing the morphological operation  kernel=numpy.ones((7,7),numpy.uint8)  erosion=cv2.erode(img\_gray,kernel,iterations=1)  output\_image=numpy.zeros\_like(img\_gray)  #iterating over the whole image  for i in range(x):  for j in range(y):  #boundary extraction  temp=img\_gray[i][j]-erosion[i][j]    #adding the image to boundaries  temp=temp+img\_gray[i][j]  #clipping the image  if temp>255:  output\_image[i][j]=255  else:  output\_image[i][j]=temp  cv2.imwrite("A2\_1output.png",output\_image) |
| --- |

**RESULTS:**

| INPUT IMAGE | OUTPUT IMAGE |
| --- | --- |
|  |  |

**Part 2**

**CODE:**

| #packages are imported  import numpy  import cv2  from skimage.morphology import disk  def closing(img,kernel):  dilation=cv2.dilate(img,kernel,iterations=1)  closing=cv2.erode(dilation,kernel,iterations=1)  return closing  def opening(img,kernel):  erosion=cv2.erode(img,kernel,iterations=1)  opening=cv2.dilate(erosion,kernel,iterations=1)  return opening  #reading the image and getting its dimension  image=cv2.imread('A2\_2.png')  #converting image to grayscale  img\_gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)  closingimg=closing(img\_gray,disk(1))  openingimg=opening(closingimg,disk(2))  cv2.imwrite("A2\_2output.png",openingimg) |
| --- |

**RESULTS:**

| INPUT IMAGE | OUTPUT IMAGE |
| --- | --- |
|  |  |

**Part 3**

**CODE:**

| #packages are imported  import cv2  from skimage.morphology import disk  #reading the image and getting its dimension  image=cv2.imread('A2\_3.png')  x=image.shape[0]  y=image.shape[1]  #converting image to grayscale  img\_gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)  #for performing the morphological operation  kernel=numpy.ones((7,7),numpy.uint8)  erosion=cv2.erode(img\_gray,kernel,iterations=1)  output\_image=numpy.zeros\_like(img\_gray)  #iterating over the whole image  for i in range(x):  for j in range(y):  temp=img\_gray[i][j]-erosion[i][j]  temp=temp+img\_gray[i][j]  if temp>255:  output\_image[i][j]=255  else:  output\_image[i][j]=temp  output\_image=cv2.dilate(output\_image,disk(2),iterations=1)  cv2.imwrite("A2\_3output.png",output\_image) |
| --- |

**RESULTS:**

| INPUT IMAGE | OUTPUT IMAGE |
| --- | --- |
|  |  |