



National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Department of Computing

EE 433: Digital Image Processing

Name: Amal Saqib

CMS: 282496

Class: BSCS 9C

Lab 12: Image Segmentation

Date: 13th December 2021

Time: 2.00Pm to 5.00Pm

Instructor: Dr. Imran Malik

```
In [1]: from PIL import Image
import numpy as np
import cv2 as cv
```

read image

```
In [2]: inputImg = Image.open("image_lab12.png").convert('L')
inputImg
```

Out[2]:



```
In [3]: inputImgArray = np.asarray(inputImg)
```

binarize image using otsu threshold method

```
In [4]: _, thresh = cv.threshold(inputImgArray, 127, 255, cv.THRESH_BINARY + cv.THRESH_OTSU)
```

```
In [5]: binarizedImg = Image.fromarray(thresh)
binarizedImg
```

Out[5]:



clean image using morphological operations

```
In [6]: import skimage.morphology  
SE = skimage.morphology.square(22)
```

```
In [7]: cleanedImgArray = cv.morphologyEx(thresh,cv.MORPH_CLOSE,SE)  
cleanedImg = Image.fromarray(cleanedImgArray)  
cleanedImg
```

Out[7]:



connected component labelling

```
In [8]: ccl = cv.connectedComponentsWithStats(cleanedImgArray, 8, cv.CV_32S)
        number_of_labels = ccl[0]
        labelMatrix = ccl[1]
        number_of_labels
```

Out[8]: 3

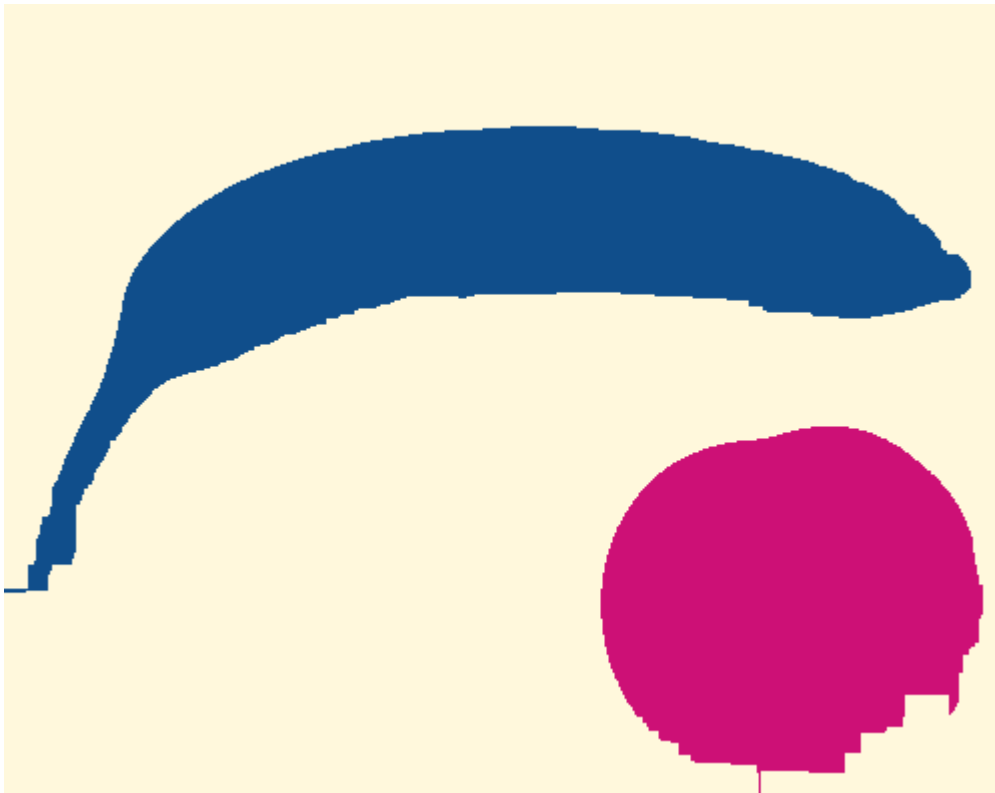
```
In [9]: colors = [(255,248,220), (16,78,139), (205,16,118)]
```

```
In [10]: def colorLabels(array, colors, labelMatrix):
        height, width = array.shape
        output = np.zeros(shape = (height, width, 3))

        for x in range(height):
            for y in range(width):
                output[x][y] = colors[labelMatrix[x][y]]
        output = output.astype('uint8')
        return output
```

```
In [11]: coloredImgArray = colorLabels(cleanedImgArray, colors, labelMatrix)
        coloredImg = Image.fromarray(coloredImgArray)
        coloredImg
```

Out[11]:



```
In [12]: def showObjectWithColor(array, color):
height, width = array.shape[0], array.shape[1]
output = np.zeros(shape = (height, width, 3))
for x in range(height):
    for y in range(width):
        check = (array[x][y][0], array[x][y][1], array[x][y][2])
        if check == color:
            output[x][y] = color
return output.astype('uint8')
```

```
In [13]: imageToSee = input("We have 3 colors\n1. cornsilk1\n2. dodgerblue4\n3. deeppink3:\nEnter your choice: ")
imageToSee = int(imageToSee)
color = colors[imageToSee - 1]
objectSegmentationArray = showObjectWithColor(coloredImgArray, color)
objectSegmentationImg = Image.fromarray(objectSegmentationArray)
objectSegmentationImg
```

Out[13]:



- Every object has its own label, hence is given a unique color. - No. of objects of one color = 1