



Dhaka University of Engineering & Technology, Gazipur-1707

Department of Computer Science & Engineering

Subject: Pattern Recognition Sessional (CSE-4630)

Lab Name: Image Segmentation.

Submit Date: 13-02-2022

Lab-03

Submitted To:

<p>Dr.Md. Waliur Rahman Miah Professor, Department of Computer Science Dhaka University of Engineering & Technology, Gazipur</p>	<p>Mss. Sabah Binte Noor Assistant Professor, Department of Computer Science Dhaka University of Engineering & Technology, Gazipur</p>
--	--

Submitted By:

<p>Sabbir Hossain Id: 164045 Year:4th Semester: 2nd Section: A</p>
--

Introduction: In this lab experiment we will see how to do image segmentation in OpenCV python by using multiple techniques. To perform image segmentation we need some prior knowledge of image processing.

Image Segmentation: Image segmentation is an image processing task in which the image is segmented or partitioned into multiple regions such that the pixels in the same region share common characteristics. There are two forms of image segmentation: 1) Local Segmentation 2) Global Segmentation

```
import matplotlib as plt
import numpy as np
import cv2
from google.colab.patches import cv2_imshow # for image display

from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount()

```
img = cv2.imread("/content/drive/My Drive/PT_image/babyImage.png")
cv2_imshow(img)
```

K-Means Algorithm: K-means is a clustering algorithm that is used to group data points into clusters such that data points lying in the same group are very similar to each other in characteristics. K-means algorithm can be used to find subgroups in the image and assign the image pixel to that subgroup which results in image segmentation.

```
criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 10, 1.0)
K = 2
attempts=10
```

```
img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
twoDimage = img.reshape((-1,3))
twoDimage = np.float32(twoDimage)
```

```
K=2
ret,label,center=cv2.kmeans(twoDimage,K,None,criteria,attempts,cv2.KMEANS_PP_CENTERS)
center = np.uint8(center)
res = center[label.flatten()]
result_image = res.reshape((img.shape))
cv2.imshow('result_image')

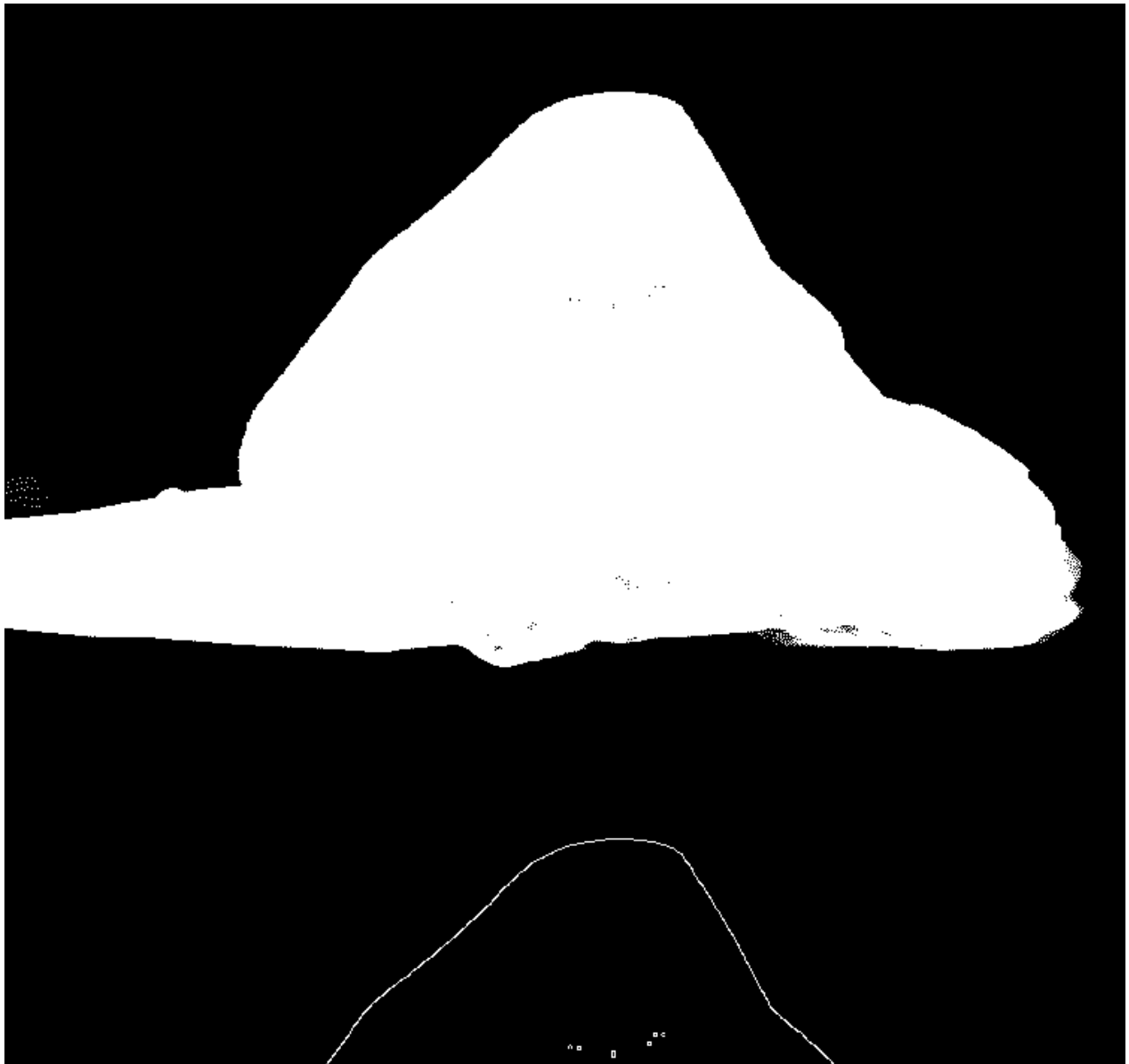
result_image.shape
center
```



Detecing and Drawing Contuours: Use the OpenCV find contour function to find all the open/closed regions in the image and store (cnt). Use the -1 subscript since the function returns a two-element tuple. Pass them through the sorted function to access the largest contours first. Create a zero pixel mask that has equal shape and size to the original image. Draw the detected contours on the created mask.

```
gray = cv2.cvtColor(img,cv2.COLOR_RGB2GRAY)
cv2_imshow(gray)

_,thresh = cv2.threshold(gray, np.mean(gray), 255, cv2.THRESH_BINARY_INV)
cv2_imshow(thresh)
canny = cv2.Canny(thresh,0,255)
cv2_imshow(canny)
edges = cv2.dilate(canny,None)
cv2_imshow(edges)
```





Segmentation using Color Masking:

```
img = cv2.resize(img,(256,256))
cnt = sorted(cv2.findContours(edges, cv2.RETR_LIST, cv2.CHAIN_APPROX_SIMPLE)[-2], key=cv2.con
mask = np.zeros((256,256), np.uint8)
masked = cv2.drawContours(mask, [cnt],-1, 255, -1)
cv2_imshow(masked)
```



```
dst = cv2.bitwise_and(img, img, mask=mask)
segmented = cv2.cvtColor(dst, cv2.COLOR_BGR2RGB)
cv2.imshow('dst', dst)
cv2.imshow('segmented', segmented)
```

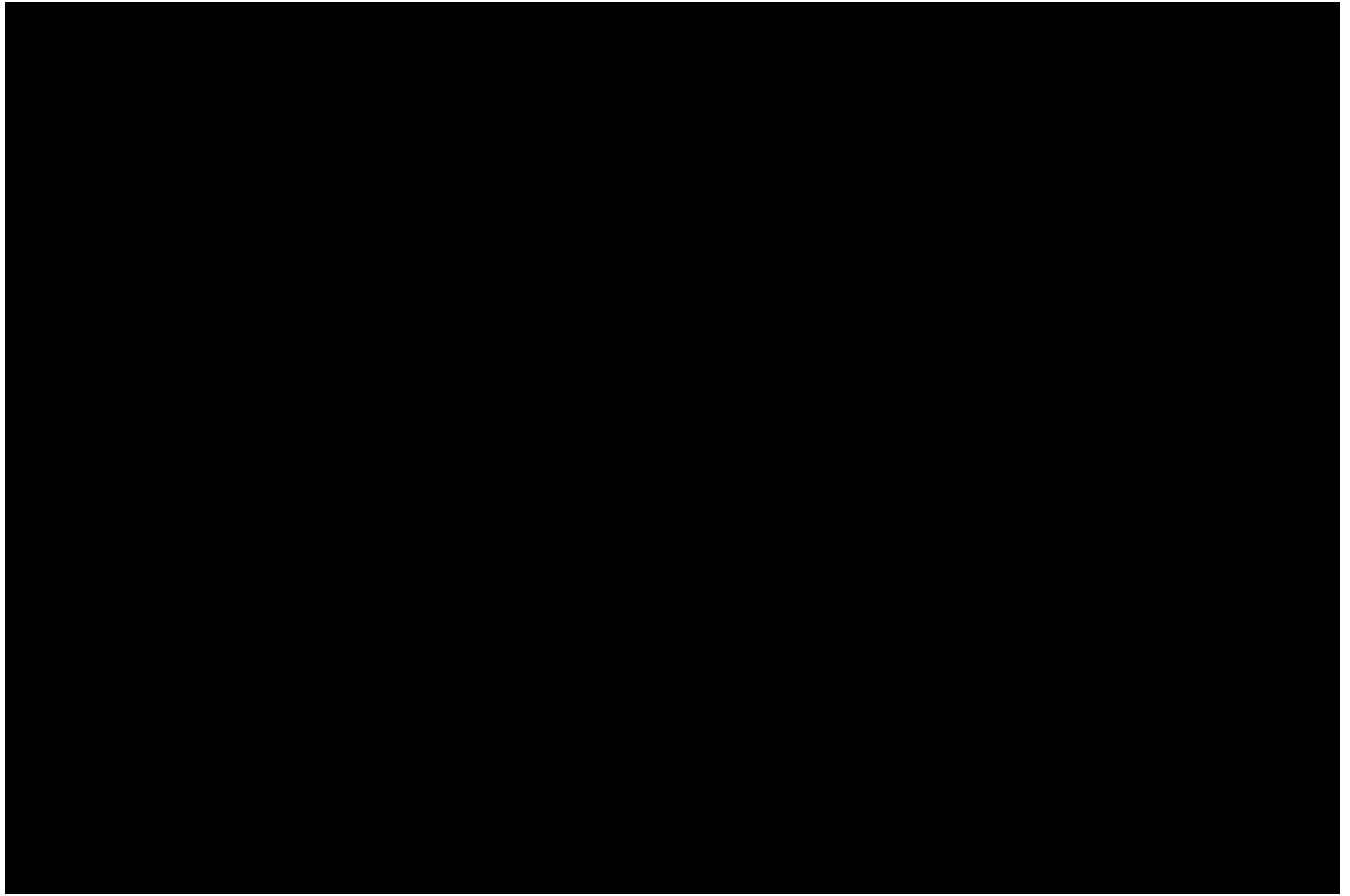


```
img = cv2.imread("/content/drive/My Drive/PT_image/babyImage.png", 1)
rgb_img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
hsv_img = cv2.cvtColor(rgb_img, cv2.COLOR_RGB2HSV)
```

Define the Color Range to be Detected: Define the RGB range for the color we want to detect. Use the OpenCV in range function to create a mask of all the pixels that fall within the range that we defined. It will later help to mask these pixels.

```
light_blue = (90, 90, 90)
dark_blue = (128, 255, 255)
```

```
mask = cv2.inRange(hsv_img, light_blue, dark_blue)
cv2_imshow(mask)
```

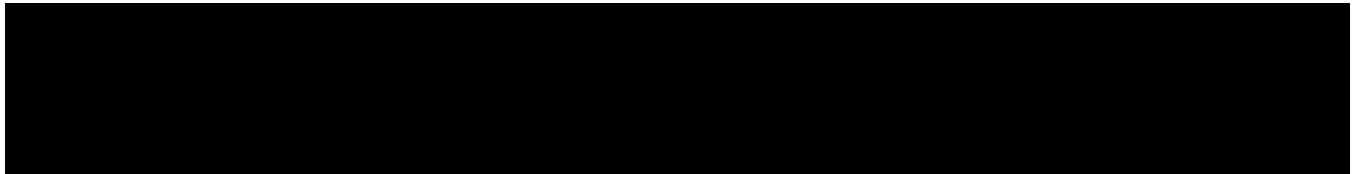


Apply the Mask: Use the bitwise AND operation to apply our mask to the query image.

```
result = cv2.bitwise_and(img, img, mask=mask)
cv2_imshow(result)
```




Conclusion: In this lab experiment I have learnt how to segment a image in different way and different segment. Here I have use K-Means, Detecting and Drawing Conturs and Segmentation using color masking. Colab was my editor and I have run all my code here.



✓ 0s completed at 11:02 PM

