Assignment Project Report

K-Means Clustering: Image Segmentation

Name: Arun Govind
Course: AI and ML
(Batch 4)

- **Problem Statement**

  Take a bright colorful image (Eg: image having fruits in it) and implement image segmentation using K-Means.

- **Prerequisites**

  - Software:
    - Python 3 (Use anaconda as your python distributor as well)

  - Tools:
    - Numpy
    - OpenCV
    - Matplotlib

  - Dataset used: A photo of a palm tree

1. **Method Used**

   In computer vision, image segmentation is the process of partitioning an image into multiple segments. The goal of segmenting an image is to change the representation of an image into something that is more meaningful and easier to analyze. It is usually used for locating objects and creating boundaries.

   Uses:
   1. Used in self-driving cars. Autonomous driving is not possible without object detection which involves segmentation.
   2. Used in the healthcare industry. Helpful in segmenting cancer cells and tumors using which their severity can be gauged

- **Implementation:**

1. Load all required libraries and load the image in RGB space

```
import numpy as np
import cv2
import matplotlib.pyplot as plt
```

```
original_image = cv2.imread("palm_tree.jpeg")
```

```
img=cv2.cvtColor(original_image,cv2.COLOR_BGR2RGB)
```

2. Convert the RGB space to HSV and then convert unit8 values to float

We convert the unit8 to float as it is a requirement of the k-means method of OpenCV

```
vectorized = img.reshape((-1,3))
```

```
vectorized = np.float32(vectorized)
```

3. Apple K-Means clustering

OpenCV provides cv2.kmeans(samples, nclusters(K), criteria, attempts, flags) function for color clustering.

1. **samples**: It should be of np.float32 data type, and each feature should be put in a single column.
2. **nclusters(K)**: Number of clusters required at the end
3. **criteria**: It is the iteration termination criteria. When this criterion is satisfied, the algorithm iteration stops. Actually, it should be a tuple of 3 parameters.

```
K = 3
attempts=10
ret,label,center=cv2.kmeans(vectorized,K,None,criteria,attempts,cv2.KMEANS_PP_CENTERS)
```

4. Convert back to unit8

```
center = np.uint8(center)
```

5. Access to labels to regenerate the clustered image

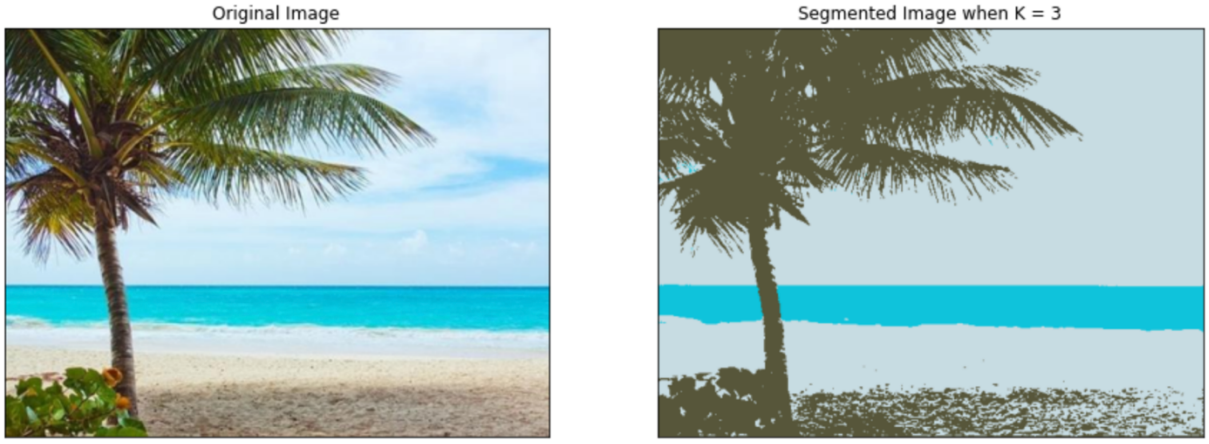
```
res = center[label.flatten()]
result_image = res.reshape((img.shape))
```

6. Code for visualizing output result

```
figure_size = 15
plt.figure(figsize=(figure_size,figure_size))
plt.subplot(1,2,1),plt.imshow(img)
plt.title('Original Image'), plt.xticks([]), plt.yticks([])
plt.subplot(1,2,2),plt.imshow(result_image)
plt.title('Segmented Image when K = %i' % K), plt.xticks([]), plt.yticks([])
plt.show()
```

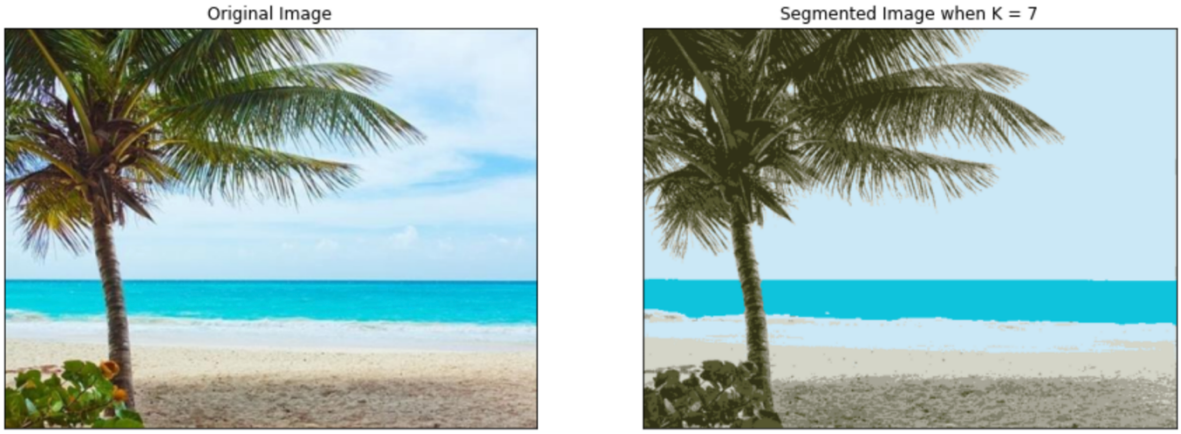
- **Results:**

1. Output result when k=3

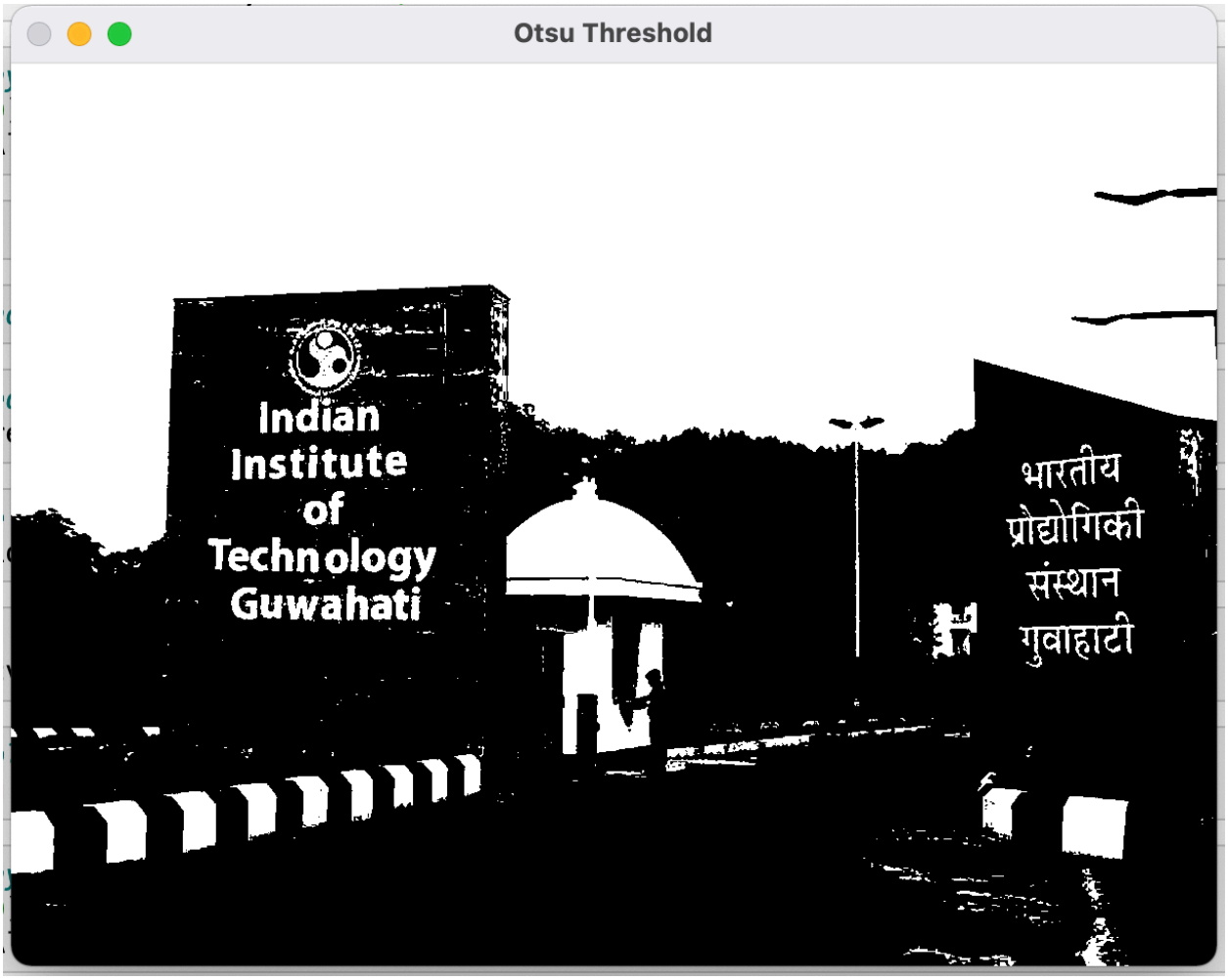


2. Output result when k=5

3. Output result when k=7



So, the algorithm has categorized our original image into three dominant colors.


Otsu Threshold
Indian
Institute
of
Technology
Guwahati
भारतीय
प्रोद्योगिकी
संस्थान
गुवाहाटी