Assignment Project Report

K-Means Clustering: Image Segmentation

Name : Meghna Maan

Course : AI & ML (Batch 4)

Given Question

Take a bright colorful image (Eg: image having fruits in it) and implement image segmentation using K-Means. You can first try to implement K-Means on iris dataset to understand its working and then extend the same logic, using the image pixels as the data points. Hint: All the K centroids will represent a color and therefore, you can initialize all the pixels to belong to a cluster randomly and then start the training of the centroids.

Prerequisites

1. Software:

   Python 3

2. Tools:

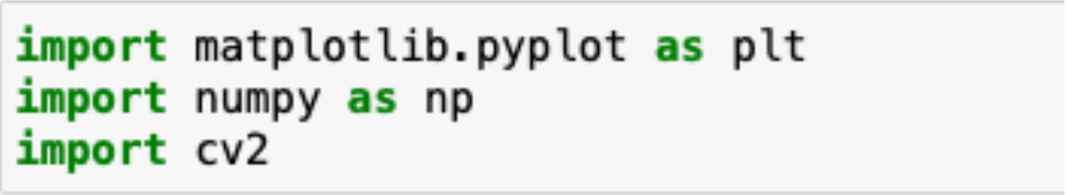
   - Numpy
   - Pandas
   - OpenCv

Methods Used

Image segmentation is the process of partitioning a digital image into multiple distinct regions containing each pixel with similar attributes i.e. classification of an image into different groups. There are different methods, and one of the most popular methods is the k-means clustering algorithm. K-Means clustering algorithm is an unsupervised algorithm, and it is used to segment the interest area from the background.

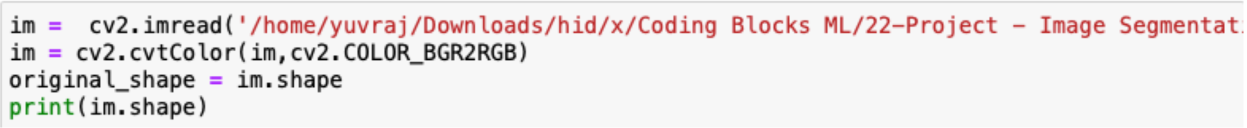
Implementation

1. Loading Libraries and Dataset



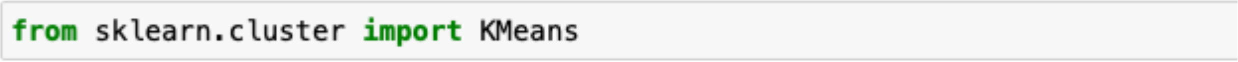
```
import matplotlib.pyplot as plt
import numpy as np
import cv2
```

2. Building Model

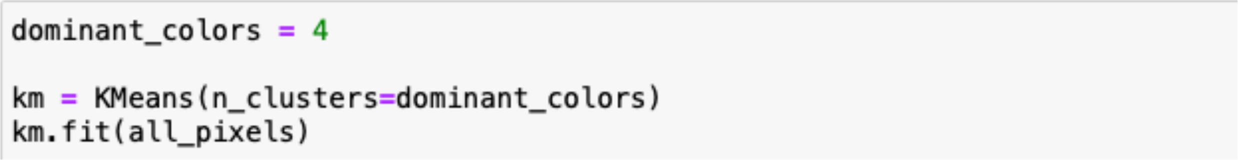


```
im =  cv2.imread('/home/yuvraj/Downloads/hid/x/Coding Blocks ML/22-Project - Image Segmentat
im = cv2.cvtColor(im,cv2.COLOR_BGR2RGB)
original_shape = im.shape
print(im.shape)
```

3. K-Means Clustering and centering data

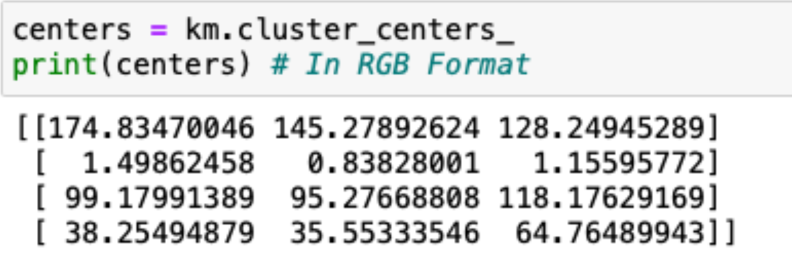


```
from sklearn.cluster import KMeans
```



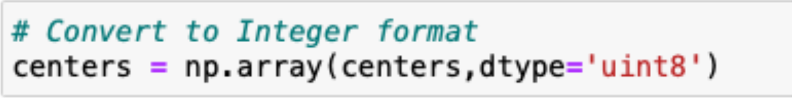
```
dominant_colors = 4

km = KMeans(n_clusters=dominant_colors)
km.fit(all_pixels)
```

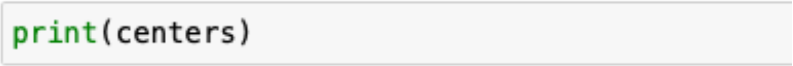


```
centers = km.cluster_centers_
print(centers) # In RGB Format
```

```
[[174.83470046 145.27892624 128.24945289]
 [  1.49862458   0.83828001   1.15595772]
 [ 99.17991389  95.27668808 118.17629169]
 [ 38.25494879  35.55333546  64.76489943]]
```

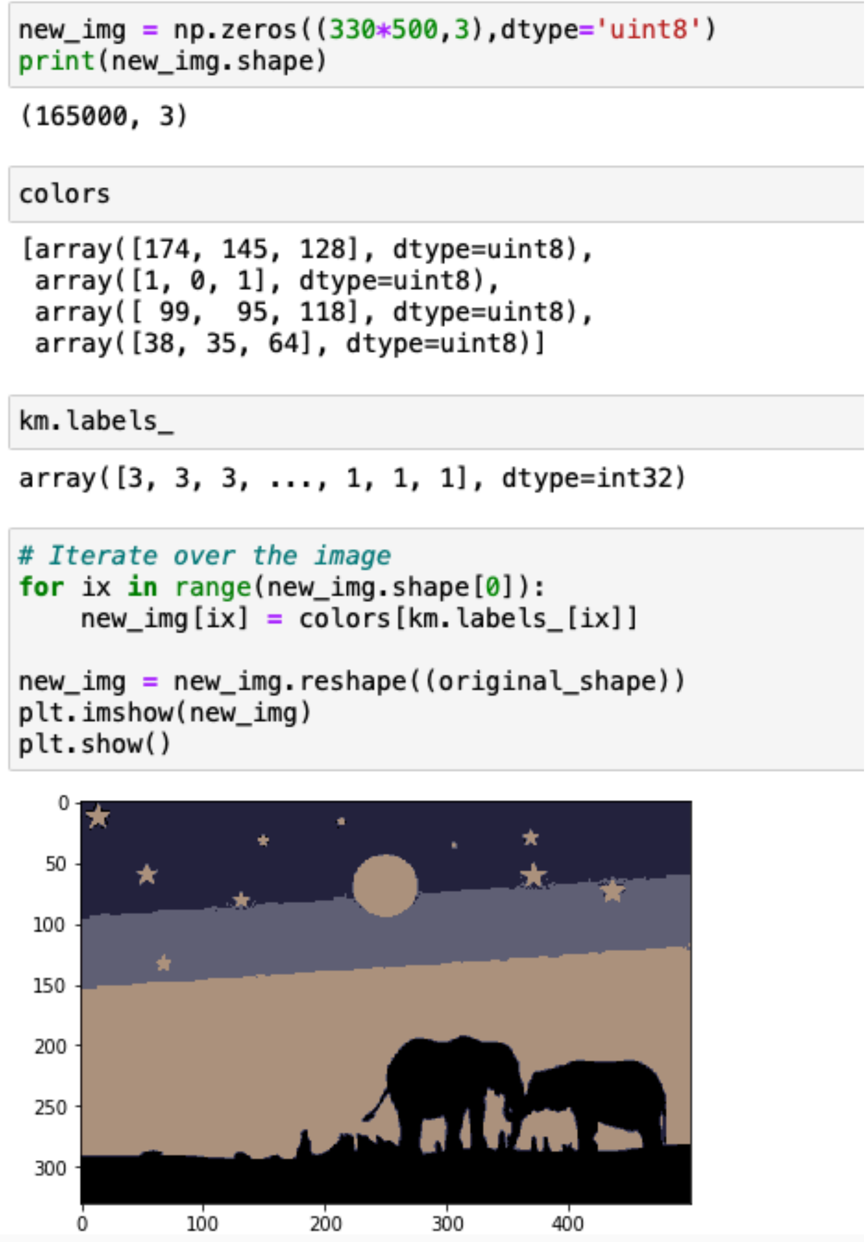


```
# Convert to Integer format
centers = np.array(centers,dtype='uint8')
```



```
print(centers)
```

4. Segmenting our Image + Final Results



```
new_img = np.zeros((330*500,3),dtype='uint8')
print(new_img.shape)
```

```
(165000, 3)
```

```
colors
```

```
[array([174, 145, 128], dtype=uint8),
 array([1, 0, 1], dtype=uint8),
 array([ 99,  95, 118], dtype=uint8),
 array([38, 35, 64], dtype=uint8)]
```

```
km.labels_
```

```
array([3, 3, 3, ..., 1, 1, 1], dtype=int32)
```

```
# Iterate over the image
for ix in range(new_img.shape[0]):
    new_img[ix] = colors[km.labels_[ix]]

new_img = new_img.reshape((original_shape))
plt.imshow(new_img)
plt.show()
```