CAPSTONE PROJECT REPORT

K-Means Clustering: Image Segmentation

Name - Paleti Samuel Yashaswi
Course - Machine learning and AI
Duration - 24 months
Question - 6

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

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

%matplotlib inline

# Read in the image
image = cv2.imread('her.jpg')

# Change color to RGB (from BGR)
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

plt.imshow(image)
```

```
<matplotlib.image.AxesImage at 0x1065bd60>
```


```
# Reshaping the image into a 2D array of pixels and 3 color values (RGB)
pixel_vals = image.reshape((-1,3))

# Convert to float type
pixel_vals = np.float32(pixel_vals)
```

```
#which will happen is 100 iterations are run or the epsilon (which is the required accuracy)
#becomes 85%
criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 100, 0.85)

# then perform k-means clustering wit h number of clusters defined as 3
#also random centres are initially choosed for k-means clustering
k = 9
retval, labels, centers = cv2.kmeans(pixel_vals, k, None, criteria, 10, cv2.KMEANS_RANDOM_CENTERS)
```

```
# convert data into 8-bit values
centers = np.uint8(centers)
segmented_data = centers[labels.flatten()]

# reshape data into the original image dimensions
segmented_image = segmented_data.reshape((image.shape))

plt.imshow(segmented_image)
```

```
<matplotlib.image.AxesImage at 0x11b9e490>
```