Exercise 3: Clustering

Objective

To implement K-Means clustering and demonstrate their applications in image segmentation.

Introduction

Clustering is an unsupervised learning technique that groups similar data points together based on their features. Different clustering algorithms have different strengths and applications. In this exercise, we cover:

- **K-Means Clustering**: Groups data into K clusters by minimizing intra-cluster variance. Commonly used in image segmentation.

K-Means Clustering and Image Segmentation

Algorithm Steps

1. Choose the number of clusters (K).
2. Initialize K cluster centroids randomly.
3. Assign each data point to the nearest centroid.
4. Recalculate centroids based on current cluster memberships.
5. Repeat steps 3 and 4 until centroids converge or a maximum number of iterations is reached.

Python Program

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from skimage import io

# Load image
image =
io.imread('https://upload.wikimedia.org/wikipedia/commons/thumb/2/24/Ca
t03.jpg/320px-Cat03.jpg')
plt.imshow(image)
plt.title('Original Image')
plt.axis('off')
plt.show()
```

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

# Apply KMeans clustering
k = 4  # Number of clusters
kmeans = KMeans(n_clusters=k, random_state=42)
kmeans.fit(pixels)
segmented_img =
kmeans.cluster_centers_[kmeans.labels_].reshape(image.shape).astype(np.
uint8)

# Display segmented image
plt.imshow(segmented_img)
plt.title(f'Segmented Image with {k} colors')
plt.axis('off')
plt.show()
```

Sample Output

1. **Original Image** – Shows the unprocessed image.
2. **Segmented Image** – Shows the image segmented into 4 color regions using K-Means clustering.

Tasks

Task 1 [CO2] [BTL 2] [2 mark]
Run the program and visualize the original and segmented images for different K

Output : 2025007855-Task-1

Inference of task-1

Initially, we import all the necessary libraries like importing K-means clustering algorithm from sk-learn.cluster, skimage import io (scikit image library) ..,etc

Step-1: Load and show the original image

Before we proceed with initializing and clustering, we first need to load an image **(cluster1.jpg)** in the local drive .Once it is done title it as “Original Image” and display the image(using skimage library).

Step-2: Initialization

Now in this step we start by randomly selecting K point from the dataset. These points assigned to K acts as the cluster centroid .

K=4 (lets initially consider)

Step-3: Assignment of data points

Now,we calculate the distance between each data point with the initialized K cluster centroids. And assign the data point to their nearest clusters based on the distance calculated between them.This step effectively forms K clusters.

Step-4: Update Centroids

From the previous step, once all the data points are assigned to the each cluster they form a membership among them.This completed the first iteration,now we re-calculate the centroids of the clusters by taking the mean of all data points assigned to each cluster.

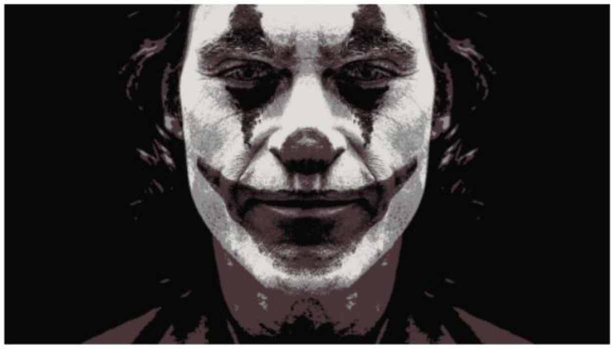
After Iteration 1: K=4 so on ..after some “n” iterations value of K gets updated to K=10,20,30,40,50

Step-5: Repeat the steps-3,4

In this step, we just repeat step-3 and 4 till we notice convergence (no signification change in cluster centroids after some iterations).

At K=50,we cannot see any further changes between the original image and segmented image.

Conclusion: With K-Means clustering algorithm,we have learned how the image segmentation is carried out and how the initializing K clusters helps in grouping similar data points with similar characteristic which in turn helps us discover the underlying data structure within the data.In the program executed we notice similar data pattern followed by which colors are segmented in the image.And, how the color intensity of the image enhances on increasing the value of “K”.

Original Image	Segmented Image with 4 colors
	

Segmented Image with 10 colors	Segmented Image with 20 colors
Segmented Image with 30 colors	Segmented Image with 50 colors

Task 2 [CO2] [BTL 3] [3 mark]

Using Elbow technique, find the optimal number of clusters K. Visualize the original and segmented images.

Output: 2025007855-Task-2

Inference task-2:

Elbow Method:

This method helps us pick the no.of clusters(k) by measuring how inertia changes as “k” grows.Here, inertia is the sum of squared distance between each data point and its assigned cluster center.

A sharp “elbow” in the inertia vs k plot suggests us the most natural grouping.

Step-1: Load and show the original image

Before we proceed with initializing and clustering, we first need to load an image **(cluster1.jpg)** in the local drive .Once it is done title it as “Original Image” and display the image(using skimage library).

Step-2: Computing Inertia for Different k values

In this step,we first initialize an empty list to store inertia values.we have kept k in loop from 1 to some max.

Step-3: Plotting Insertia vs k

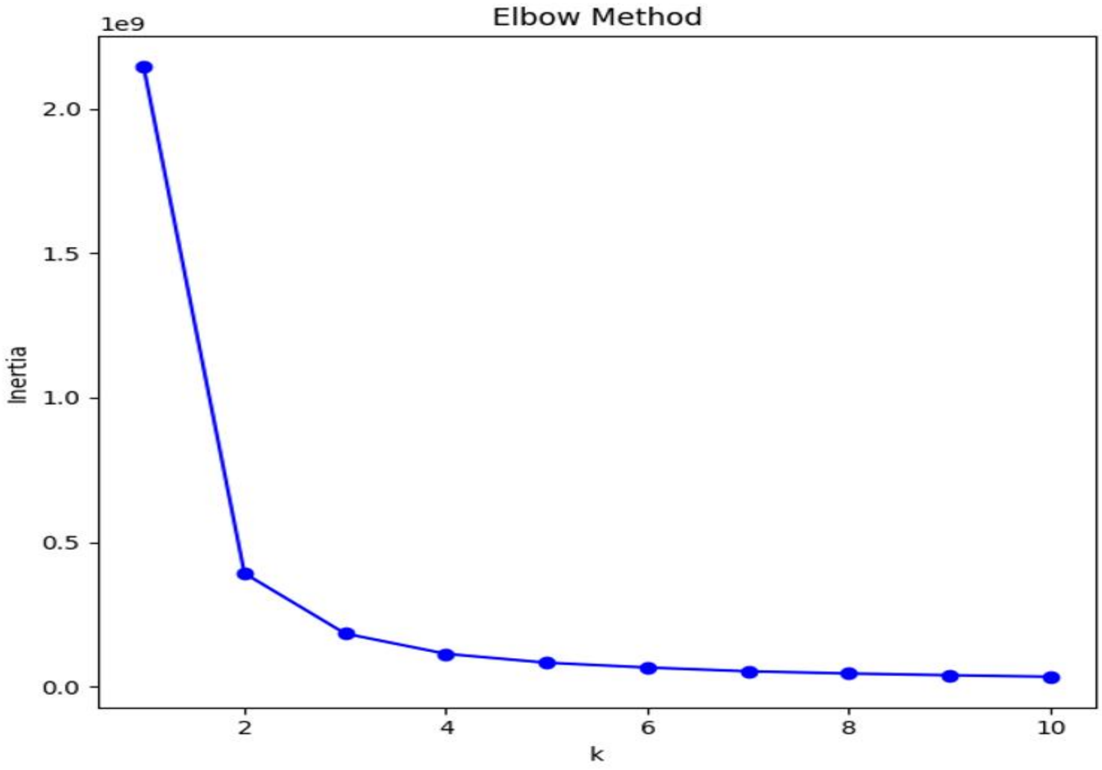
Here,we create a line plot with x-axis as no.of clusters(k) and y-axis as corresponding inertia.And now we look for a point in the plot where the inertia decreases slowly. That's where the **elbow curve** take into shape leading to notate the elbow point .

Step-4: Identifying the Elbow

From the above step , we can get to know that "at small k , adding another cluster drastically reduces inertia.After certain k value we can see that convergence takes place.And further increasing the value of k does show much impact on the image segmentation.

Step-5: Choose k value and execute

Finally, we have a point k in the elbow and initialize K-Means clustering.Now we use cluster centers and labels to segment the images.


Elbow Method
1e9
Inertia
k