NAME:	Anish Gade
UID:	2021700022
BRANCH:	T.Y. CSE Data Science
BATCH:	B
SUBJECT:	FOSIP
EXP. NO.:	8
DATE:	18-11-2023

AIM: - To study image Segmentation

OBJECTIVES:

(I) Image Segmentation using recent published technique based on any one of the following operation.

1. Segmentation using Discontinuity Property

2. Segmentation using Similarity Property

(II) Download the Conference paper from IEEE Xplorer published after 2018

(III) Develop the algorithm and process the following images for different types of images

(IV) Give your conclusion based on results obtained.

INTRODUCTION:

Image segmentation is a fundamental task in computer vision that involves partitioning an image into meaningful and semantically homogeneous regions or segments. The goal is to simplify the representation of an image into distinct and meaningful parts to facilitate analysis, understanding, and interpretation. Image segmentation plays a crucial role in various applications, including object recognition, scene understanding, medical imaging, autonomous vehicles, and more.

The primary objective of image segmentation is to group together pixels or regions that share similar visual characteristics, such as color, intensity, texture, or other relevant features. This process allows for the identification and isolation of individual objects or structures within an image, enabling more advanced analysis and decision-making by computer algorithms.

Here are some key concepts and approaches related to image segmentation:

Pixel-based Segmentation:

Color-based Segmentation: Groups pixels based on their color similarity.
Intensity-based Segmentation: Considers pixel intensity or brightness levels for grouping.

Region-based Segmentation:

Segments based on the similarity of entire regions rather than individual pixels.
Edge-based Segmentation:

Focuses on identifying boundaries or edges between different regions.
Clustering Techniques:

Algorithms like k-means clustering group pixels with similar features into clusters.
Thresholding:

Involves setting a threshold value to separate pixels into different segments based on intensity.
Contour-based Segmentation:

Identifies contours or outlines of objects in an image.
Semantic Segmentation:

Assigns a specific class or label to each pixel, providing a detailed understanding of the image content.
Instance Segmentation:

Identifies and distinguishes individual instances of objects in an image, often by assigning a unique identifier to each instance.
Deep Learning for Segmentation:

Convolutional Neural Networks (CNNs) and other deep learning architectures have shown significant success in semantic and instance segmentation tasks.
Applications:

Medical Imaging: Identifying and segmenting organs or abnormalities.
Object Recognition: Enabling computers to understand and locate objects within an image.
Autonomous Vehicles: Assisting in the perception of the surrounding environment.
Augmented Reality: Enhancing the real-world view by segmenting and understanding the scene.
Image segmentation is a challenging task due to variations in lighting, scale, and the complex nature of real-world scenes. Advances in computer vision and machine learning, especially deep learning, have led to significant improvements in the accuracy and efficiency of image segmentation methods.

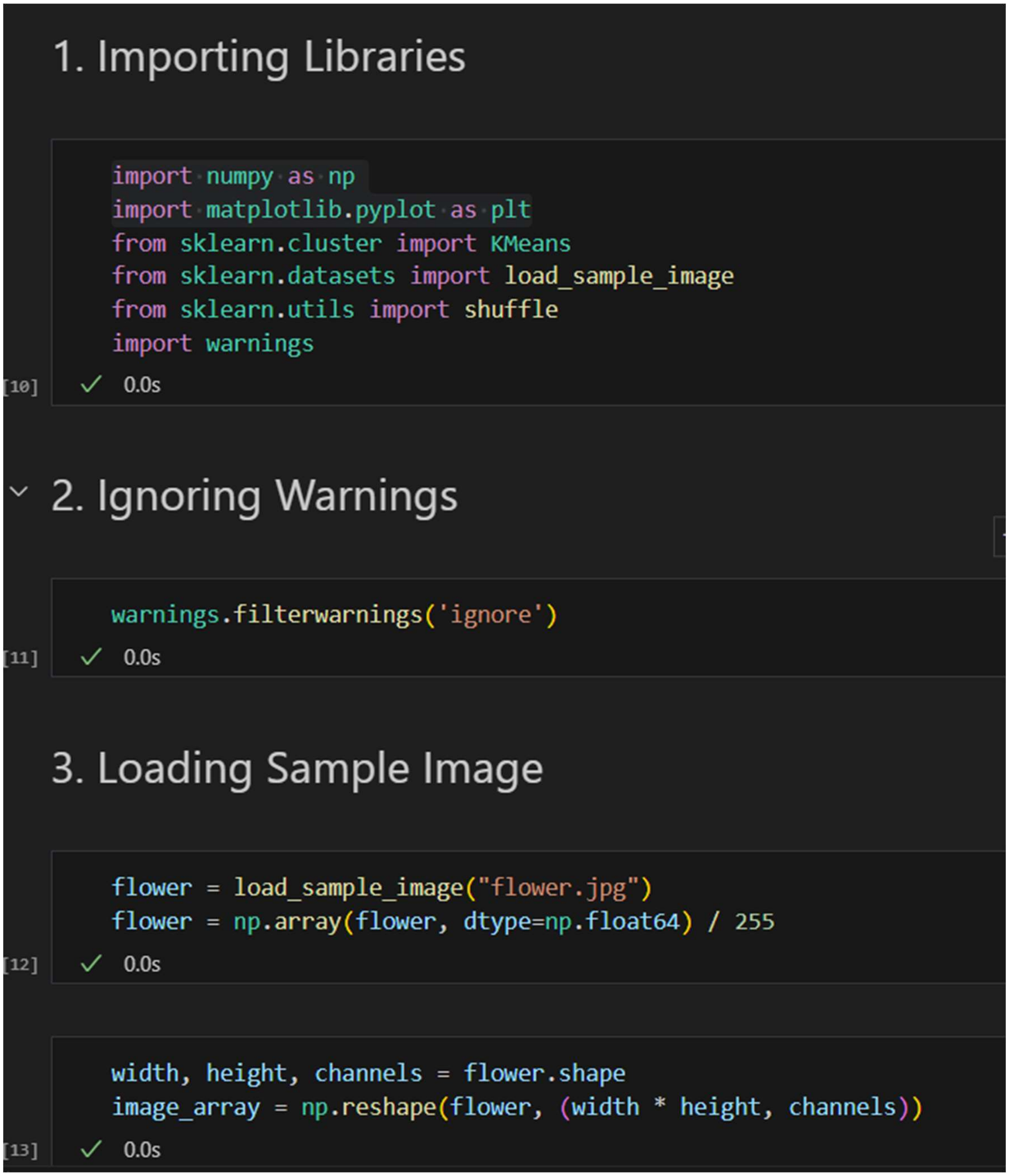
ALGORITHM (K-MEANS):

1. Choose the number of clusters you want to find which is k.
2. Randomly assign the data points to any of the k clusters.
3. Then calculate the center of the clusters.
4. Calculate the distance of the data points from the centers of each of the clusters.
5. Depending on the distance of each data point from the cluster, reassign the data points the nearest clusters.
6. Again, calculate the new cluster center.
7. Repeat steps 4,5 and 6 till data points don’t change the clusters, or till we reach the

assigned number of iterations.

EXPERIMENTATION:

CODE:



1. Importing Libraries

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.datasets import load_sample_image
from sklearn.utils import shuffle
import warnings
```

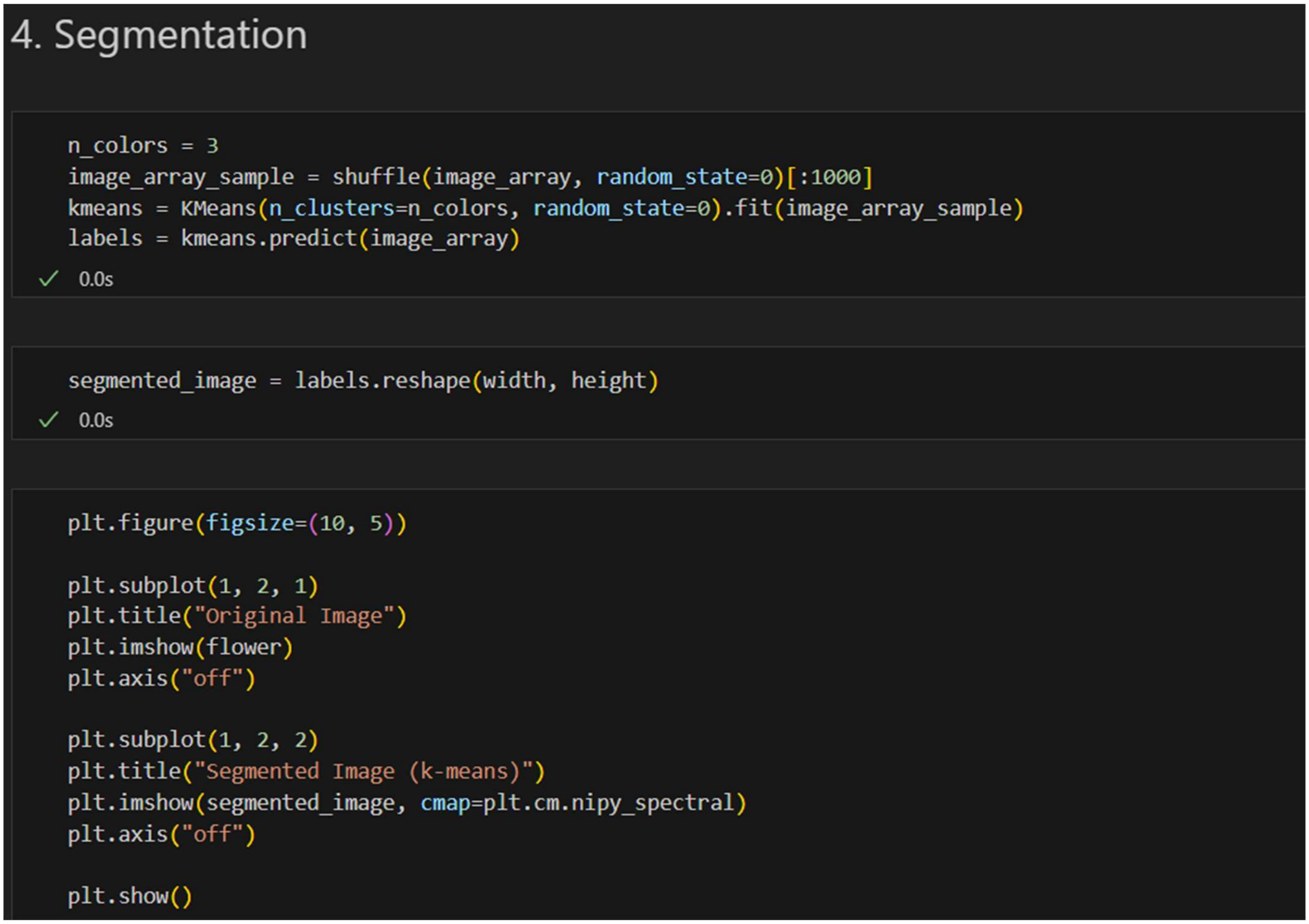
2. Ignoring Warnings

```
warnings.filterwarnings('ignore')
```

3. Loading Sample Image

```
flower = load_sample_image("flower.jpg")
flower = np.array(flower, dtype=np.float64) / 255
```

```
width, height, channels = flower.shape
image_array = np.reshape(flower, (width * height, channels))
```



4. Segmentation

```
n_colors = 3
image_array_sample = shuffle(image_array, random_state=0)[:1000]
kmeans = KMeans(n_clusters=n_colors, random_state=0).fit(image_array_sample)
labels = kmeans.predict(image_array)
```

✓ 0.0s

```
segmented_image = labels.reshape(width, height)
```

✓ 0.0s

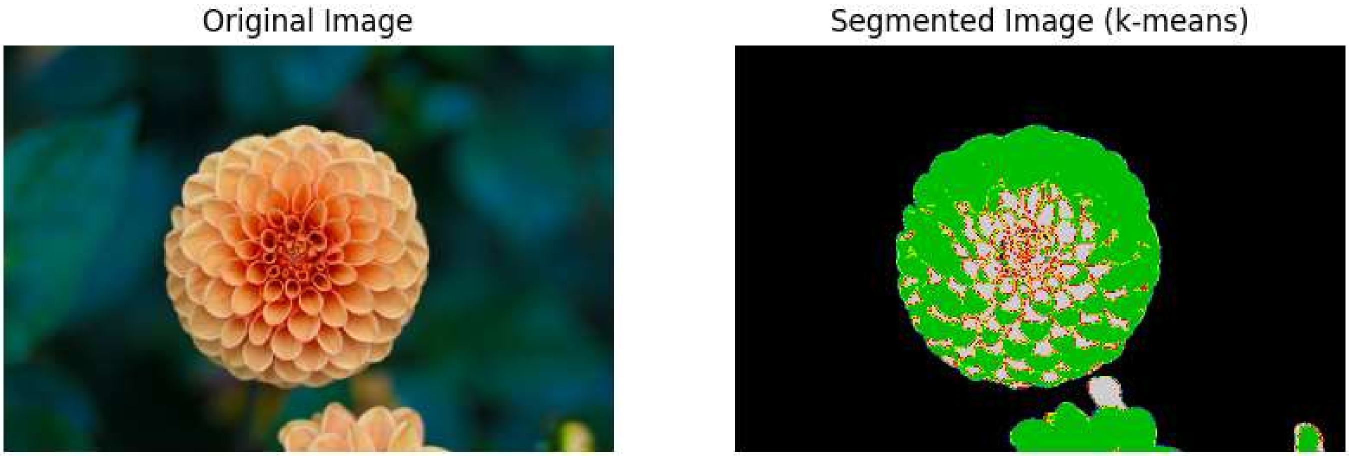
```
plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)
plt.title("Original Image")
plt.imshow(flower)
plt.axis("off")

plt.subplot(1, 2, 2)
plt.title("Segmented Image (k-means)")
plt.imshow(segmented_image, cmap=plt.cm.nipy_spectral)
plt.axis("off")

plt.show()
```

RESULT:



CONCLUSION:

I learned about the application and implementation of Image segmentation using similarities using k-means clustering.

REFERENCES:

1. https://www.geeksforgeeks.org/image-segmentation-using-k-means-clustering/