ЛАБОРАТОРНАЯ РАБОТА №11
КЛАСТЕРИЗАЦИЯ ИЗОБРАЖЕНИЙ

Цель работы: Изучение и практическое применение методов кластеризации для анализа и разделения изображений на группы с использованием языка программирования Python.

Необходимое программное обеспечение:

1. Python
2. Jupyter Notebook
3. Библиотеки: OpenCV, NumPy, Matplotlib, scikit-learn (для кластеризации)

Шаги выполнения:

1. Введение:

- Ознакомьтесь с основами кластеризации и ее применением в обработке изображений.

2. Загрузка и визуализация изображения:

- Загрузите изображение, которое будет использоваться для кластеризации.
- Визуализируйте изображение.

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

# Загрузка изображения
image_path = 'path/to/your/image.jpg'
image = cv2.imread(image_path)

# Преобразование BGR в RGB
image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# Визуализация изображения
plt.imshow(image_rgb)
plt.title('Original Image')
plt.axis('off')
plt.show()
```

3. Подготовка данных для кластеризации:

- Преобразуйте изображение в формат, удобный для кластеризации (например, в одномерный массив пикселей).

```python
# Преобразование изображения в одномерный массив
pixels = image.reshape((-1, 3))

# Преобразование значений пикселей в формат с плавающей запятой
pixels = np.float32(pixels)
```

4. Кластеризация с использованием k-средних:

- Примените алгоритм k-средних для кластеризации пикселей изображения.
- Визуализируйте результаты.

```python
from sklearn.cluster import KMeans

# Определение числа кластеров (пример: 3)
num_clusters = 3

# Инициализация и обучение модели KMeans
kmeans = KMeans(n_clusters=num_clusters)
kmeans.fit(pixels)

# Получение меток кластеров и центроидов
labels = kmeans.labels_
centers = kmeans.cluster_centers_

# Присвоение каждому пикселю цвет центроида его кластера
segmented_image = centers[labels].reshape(image.shape)

# Визуализация сегментированного изображения
plt.imshow(segmented_image.astype(np.uint8))
plt.title('Segmented Image (k-means)')
plt.axis('off')
plt.show()
```

5. Кластеризация с использованием других методов:

- Попробуйте другие алгоритмы кластеризации, такие как AgglomerativeClustering, DBSCAN, и т.д.
- Визуализируйте результаты каждого метода.

from sklearn.cluster import AgglomerativeClustering, DBSCAN

```
# Пример AgglomerativeClustering
agg_clustering = AgglomerativeClustering(n_clusters=num_clusters)
agg_labels = agg_clustering.fit_predict(pixels)
agg_segmented_image = centers[agg_labels].reshape(image.shape)

# Пример DBSCAN
dbscan = DBSCAN(eps=5, min_samples=50)
dbscan_labels = dbscan.fit_predict(pixels)
dbscan_segmented_image = pixels.reshape(image.shape)
dbscan_segmented_image[dbscan_labels == -1] = 255  # Присвоение шумовому классу белого цвета

# Визуализация результатов
plt.figure(figsize=(12, 4))

plt.subplot(131)
plt.imshow(agg_segmented_image.astype(np.uint8))
plt.title('Agglomerative Clustering')
plt.axis('off')

plt.subplot(132)
plt.imshow(dbscan_segmented_image.astype(np.uint8))
plt.title('DBSCAN Clustering')
plt.axis('off')

plt.show()
```

6. Эксперименты с параметрами:

- Проведите эксперименты с различным числом кластеров и параметрами алгоритмов.
- Визуализируйте, как изменение параметров влияет на результаты.

7. Анализ результатов:

- Сравните результаты кластеризации различными методами.
- Проанализируйте, какие методы лучше всего подходят для конкретных изображений.

8. Отчет:

- Составьте отчет, описывающий каждый этап работы, включая примеры кода, визуализацию результатов и выводы.