РАБОТА С КЛАССЕРИЗАЦИЕЙ ИЗОБРАЖЕНИЙ

ПРИМЕНЕНИЕ K-MEANS

Задание основано на применении пакета Python Scikit-learn для кластеризации изображений на примере алгоритма KMeans.

Цели практики:

- овладеть применением алгоритма K-Means
- освоить работать с задачами обучения без учителя
- научиться работать с кластеризацией изображений в Python

Начало работы

Алгоритм KMeans реализован в классе **sklearn.cluster.KMeans**. Поскольку это пример задачи обучения без учителя, для проведения обучения достаточно передать матрицу объектов. (https://scikit-learn.org/dev/modules/clustering.html)

В качестве метрики применяется **PSNR**, представляющий собой адаптацию метрики **MSE** для оценки сходства изображений. Для работы с изображениями рекомендуется использовать пакет **scikit-image**. Для загрузки изображения необходимо выполнить следующую команду:

```
from skimage.io import imread
image = imread('parrots_4.jpg')
```

После выполнения указанных шагов переменная **image** будет содержать изображение в форме массива **numpy** размером ***n * m * 3***, где **n** и **m** соответствуют размерам изображения, а **3** обозначает формат представления **RGB**.

Для отображения изображения на экране необходимо иметь установленную библиотеку **matplotlib**. Для этого воспользуйтесь следующим способом:

```
import pylab
pylab.imshow(image)
```

Задание

1. Загрузите картинку **parrots.jpg.** Преобразуйте изображение, приведя все значения в интервал от **0 до 1** – воспользуйтесь функцией **img_as_float** из модуля **skimage**. Обратите внимание на этот шаг, так как при работе с исходным изображением вы получите некорректный результат.

2. Создайте матрицу ***объекты-признаки***: характеризуйте каждый пиксель тремя координатами - значениями интенсивности в пространстве RGB.

3. Запустите алгоритм K-Means с параметрами **init='k-means++'** и **random_state=241**. После выделения кластеров все пиксели, отнесенные в один кластер, попробуйте заполнить двумя способами: медианным и средним цветом по кластеру.

4. Измерьте качество получившейся сегментации с помощью метрики **PSNR**.

   Пример: https://www.tutorialspoint.com/finding-the-peak-signal-to-noise-ratio-psnr-using-python

5. Найдите минимальное количество кластеров, при котором значение **PSNR выше 20** (***рассмотреть оба способа заполнения пикселей одного кластера, но использовать не более 20 кластеров***).
   Это число и будет ответом в данной задаче.

Ответ на каждое задание — текстовый файл, содержащий ответ в первой строчке (#.txt). Обратите внимание, что отправляемые файлы не должны содержать перевод строки в конце.

Утонения по выполнению задания:

В некоторых случаях необходимо перед выполнением кода

```
import pylab
pylab.imshow(image)
```

в любой ячейке исполнить инструкцию:

```
%matplotlib inline
```

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ

Самой распространенной категорией в задачах машинного обучения являются **задачи обучения с учителем**. В этих задачах имеется обучающая выборка, где для каждого объекта предоставлен ответ, и основной целью является обучение модели предсказывать эти ответы для новых объектов. Такие задачи позволяют строго определить критерии качества.

В случае, если у нас есть только объекты без соответствующих ответов, всё равно возможно пытаться выявить некоторую структуру в данных. Такие задачи относятся к **области обучения без учителя**. Примером такой задачи является кластеризация, где основной задачей является выявление групп похожих объектов в неразмеченных выборках.

Кластеризация может применяться в различных сферах. В данном задании рассматривается задача компьютерного зрения (CV) по нахождению группировки схожих пикселей на изображении. Такой подход позволяет создать суперпиксельное представление изображений, которое является более компактным и эффективным для решения различных задач в области компьютерного зрения.