Brandenburgische
Technische Universität
Cottbus - Senftenberg

Database and Information Systems Group
Module: Foundations of Data Mining

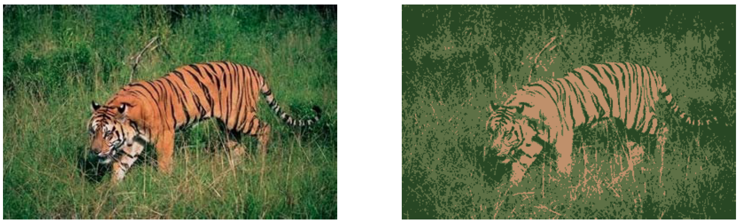
Practical Training - Task Sheet 1
Image Segmentation Using Clustering Algorithms

1 Task Description

This is the first out of two practical training tasks that you will have to fulfill to qualify for the final exam. Work is to be done in groups of two. You may work alone, but the task stays the same.

The task deals with the segmentation of color images using clustering algorithms. For this purpose, the algorithms `k-means` (see Lloyd) [2] and `DBSCAN` [1] are to be implemented (**The algorithms must be implemented by you!**). The objects that are to be clustered are the pixels of the respective image. Each pixel is represented by its vector in the RGB color space (3-dimensional vector).

Design a program with which an input image can be clustered by selecting the appropriate clustering algorithm. The output of the segmentation will be the image divided into segments (cluster). The individual segments are to be represented by different colors. An example can be seen in figure 1:



(a) Non-segmented image (b) Segmented image

Figure 1: Exemplary segmentation (k-means, k=3)

For both algorithms the following parameters should be available and arbitrarily selectable:

- `k-means`: k
- `DBSCAN`: ϵ, $minPts$

2 Submission

1. Submission deadline: 18th Dec. 2022
2. The submission is to be done in Moodle. The submission link deactivates after the deadline. Please only submit once per group. It does not matter which group member does so.

3. The digital submission contains:
    a) Source code of the program
    b) Meaningful documentation of the source code
    c) Short textual description of the program installation, execution and usage (README.txt)
4. Make sure to only use English for the documentation, comments and identifiers.

3 Requirements

In order to pass, you and your submitted program must fulfill *all* of the following criteria:

1. All points listed under ‘Submission’ have to be fulfilled.
2. You must be able to explain *every* part of the source code, even segments that were implemented by your partner.
3. Using the source code and documentation, the program must be compilable without errors.
4. The program has to be able to run without major crashes.
5. The program must be able to load an image given by the user, perform a desired clustering and save or display the resulting image.
6. Both clustering algorithms have to work correctly and in a reasonable amount of time.
7. The clustering algorithms have to be implemented by you yourself.
8. The clustering algorithms have to be implemented universally. This means that they must work with any given feature vector (with any dimensionality) and not just with image data.

4 Non-requirements

The following aspects can be implemented according to your choice.

1. You can choose whether to implement a GUI or a textual interface.
2. The colors of the clusters can be chosen freely. (In Figure 1b the average of the cluster pixels was used.)

5 Discussion

Each submission has to be explained and defended in an individual discussion. The time and date of this discussion will be agreed upon with each group. Each discussion session will take from 15 to 30 minutes.

The discussion meeting can be done in person or via video conference.

6 FAQ

Programming is not part of my curriculum. How can I pass if I cannot program well? This class requires rudimentary programming skills. However, code quality is not our focus, so it does not matter, whether your code is messy or slow. It is important, that you understand the clustering concepts and can apply them.

Which programming languages are allowed? Python, Java, C++. For use of other languages please ask the teacher first.

Is the loading and saving of the image part of the task? Your program needs to be able to do it, but these functions do not have to be implemented by you. Use an image processing library of your choice.

How fast does the program have to be? Depending on your image, it is normal that the clustering (especially DBSCAN) can take a long time to terminate. As an estimation, clustering of a small image, i. e. 100x100 pixels, should not take longer than five minutes. (Although it should usually be faster than that.)

The task demands that our clustering should work universally. Does that mean that the program has to accept input other than images? The task mentions only that the **algorithms** must be universal. The program is still processing images. Only the clustering functions, that you are writing, should be universal. Think of it as writing your own clustering library.
Do not forget, that different data can have more or less dimensions than pixels. You can assume that all points in one data set have the same dimensionality, though.

What about the alpha values of the images? Disregard them.

How many runs do we have to do with k-means? Please use the shown variant of k-means, which runs until there is no more improvement of cluster cost.

Which distance fuction should be used? Euclidean distance would be adequate. It would be ideal to make this function changeable as well, but that is not a requirement.

A final hint: Your feature vectors for the image should have **three** dimensions, not two!

Bibliography

[1] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of the 2nd Int. Conf. on Knowledge Discovery and Data Mining*, pages 226–231, 1996.

[2] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.