Assignment 3: Content Based Image Retrieval Vision and Image Processing

Søren Olsen, Francois Lauze, and Hong Pan

December 1, 2014

This is the third mandatory assignment on the course Vision and Image Processing. The goal for you is to get familiar with the Bag Of Words (BoW) principle and its use in content based image retrieval systems.

This assignment must be solved in groups. We expect that you will form small groups of 2 to 4 students that will work on this assignment. You have to pass this and the other 3 mandatory assignments in order to pass the course. If you do not pass this assignment, but you have made a SERIOUS attempt, you will get a second chance of submitting a new solution.

The deadline for this assignment is Monday 4/1, 2016 at 20:00. You must submit your solution electronically via the Absalon home page. For general information on relevant software, requirement to the form of your solution including the maximal page limit, how to upload on Absalon etc, please see the first assignment.

Bag of Visual Words Content Based Image Retrieval

The goal of the assignment is to implement a prototypical CBIR system. We recommend the use of the CalTech 101 image database. We recommend that you (for a start) select a subset of say 10-20 categories. When you have checked that everything works you may extend to more categories. For each category, the set of images should be split in two: A training set and a test set. The test set must not include images in the training set.

You should extract visual words using SIFT descriptors (ignoring position, orientation and scale) or similar descriptors extracted at interest points. To compute the descriptors, we recommend to use VLFeat's `sift`, but other options are possible.

1 Codebook Generation

In order to generate a code book, select a set of training images. Then Extract SIFT features from the training images (ignore position, orientation and scaling). The SIFT features should be concatenated into a matrix, one descriptor

per row. Then you should run the k-means clustering algorithm on the subset of training descriptors to extract good prototype (visual word) clusters. A reasonable k should be between 500 and 3000, depending on the complexity of your data and the size of the training set. You should experiment with k (but beware that this can be rather time-consuming).

Once clustering has been obtained, classify each training descriptor to the closest cluster centers) and form the bag of words (BoW) for each image in the image training set.

Note that Matlab has its own implementation of k-means, called `kmeans`, and so has Python, in the module `scipy.cluster.vq`. For C++, we recommend the SHARK implementation, with header file `shark/Algorithms/KMeans.h`. The Shark machine learning library, which should work on Windows, Linux and MacOSX, is available at: `http://image.diku.dk/shark/sphinx_pages/build/html/index.html` You need to have CMake and the boost library installed before you install Shark.

2 Indexing

The next step consists in indexing the content of the "database" (the quotation mark indicate that we are dealing more with a collection of data rather than a structured database). For each image both in the test set and in the training set you should:

- Extract the SIFT descriptors of the feature points in the image,
- Project the descriptors onto the codebook, i.e., for each descriptor the corresponding cluster prototype should be found,
- Construct the generated corresponding bag of words, i.e, word histogram.

The result should be saved in a table that would contain, per entry at least the file name, the true category, if it belongs to the training- or test set, and the corresponding bag of words / word histogram. You may want to set up a database, if you know how to do it, but this is not a requirement as simple structures will be enough for this assignment.

3 Retrieving

Finally, you should implement retrieving of images using some of the similarity measures discussed in the course slides. This may include one or two of:

- common words
- tf-ifd similarity
- Bhattacharyya distance or Kullback-Leibler divergence

Your report should show commented results from a few experiments. These should be repeated using images from the test set only and for training images only. The results from the two set of experiments´ts should be compared.

How often do you retrieve the correct image and how often does it fail. What/how many failures might be explained by some obvious circumstances. It is very important that you report what you have done, the statistics of your results, and that you comment on the ability of your system to perform CBIR.