Assignment 2: Content Based Image Retrieval
Vision and Image Processing

Kim Steenstrup Pedersen and François Lauze

December 2nd, 2013

This is the second mandatory assignment on the course Vision and Image Processing. The goal is for you to get familiar with programming for computer vision with a focus on image descriptors and their use for building image query systems.

You have to pass this and the following mandatory assignments in order to pass this course. There are in total 4 mandatory pass/fail assignments. This is a **group assignment**, i.e., we expect that you will form small groups of 2 to 4 students that will work on this assignment.

The deadline for this assignment is Monday December 16th 2013. You must submit your solution electronically via the Absalon home page. Go to the assignments list and choose this assignment and upload your solution prior to the deadline. Remember to include your name and KU user name (login for KUnet) in the solution. If you do not pass the assignment, having made a SERIOUS attempt, you will get a second chance of submitting a new solution.

A solution consist of:

- Your solution source code (Matlab / Python scripts / C / C++ code) with comments about the major steps involved in each Question (see below).
- Your code should be structured such that there is one main file that we can run to reproduce all the results presented in your report. This main file can, if you like, call other files with functions / classes.
- Your code should also include a README text file describing how to compile (if relevant) and run your program, as well as list of all relevant libraries needed for compiling or using your code. If we cannot make your code run we will consider your submission incomplete and you may be asked to resubmit.
- The code, auxiliary files and README file should be put into a compressed archive file in either the ZIP or tar format (RAR is not allowed - we simply cannot read your archive).
- A PDF file with notes detailing your answers to the questions, which may include graphs and tables if needed (**Max 5 pages** text including figures and tables). Do NOT include your source code in this PDF file.

A note on relevant software

We recommend that you select either Matlab / Python / C / C++ as the programming language you use for your solutions for these assignments. We also recommend that you select the language you are most confident in - the focus should be on learning the methods and not learning a new programming language.

If you wish to use Matlab, the University of Copenhagen has a license agreement with MathWorks that allow students to download and install a copy of Matlab on personal computers. On the KUnet web site you find a menu item called Software Library (Softwarebiblioteket) `https://intranet.ku.dk/Sider/Software-bibliotek.aspx`. Under this menu item you can find a link to The Mathworks - Matlab & Simulink + toolboxes. Click this link and follow the instructions for how to install on your own computer.

If you use Python or C / C++ we recommend that you use the OpenCV library `http://opencv.willowgarage.com/wiki/` and / or the VLFeat library `http://www.vlfeat.org/`. Both libraries provide an extensive collection of implementations of central computer vision algorithms. OpenCV provides an interface for Python. Follow the installation instructions on these websites to install on your own computer. OpenCV is also available via MacPorts on MacOSX.

If you wish to program your solutions in C++ we recommend the Shark machine learning library `http://image.diku.dk/shark/sphinx_pages/build/html/index.html` which should work on Windows, Linux and MacOSX. You need to have CMake and the boost library installed before you install Shark.

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, although you might consider others such as the University of Kentucky Benchmark Data Set (but it is huge and has only four images per categories). The visual words will be built from SIFT descriptors (ignoring position, orientation and scale). To compute the descriptors, we recommend to use VLFeat's `sift`.

1 Codebook Generation

In order to generate the code book, a training set of images must be selected. Extract SIFT features from the training set (without position, orientation and scaling). They should be concatenated into a matrix, one descriptor per row. Then you will run the k-means clustering algorithm on it (or a subset of these descriptors).

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 performed, perform vector quantization and form the bag of words 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` and for C++, we recommend the SHARK implementation, with header file `shark/Algorithms/KMeans.h`.

2 Indexing

The next step consists in indexing the content of the "database" (I put quotation mark as we are more dealing with a collection of data rather than a structured database). The task consist in computing for each image of our collection (and not just the training collection)

- its SIFT descriptors,
- then project them to the codebook, i.e., find for each descriptor the corresponding cluster and cluster center,
- generate the corresponding bag of words, i.e, word histogram.

This should be saved in a table that would contain, per entry at least the file name 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

Implement retrieving via some of the similarity measures discussed in the course (not necessarily all):

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

Report results from a few experiments , successes and failures, and comment.