Middle East Technical University

Department of Computer Engineering

CENG 483

Introduction to Computer Vision

Spring 2017-2018

Take Home Exam 1
Content Based Image Retrieval

Due date: **12 April 2018, 23:55**

1 Objectives

The purpose of this assignment is to familiarize yourselves with the fundamentals of Content Based Image Retrieval (CBIR) and the interest point based local feature descriptors. The assignment aims to give insights about the computer vision research and evaluation methods.

Keywords: *Content Based Image Retrieval, Intereset Points, Local Feature Descriptors, SIFT, Dense-SIFT, Clustering, Bag of Features Representation, Mean Average Precision*

2 Specifications

In this assignment you are required to implement a CBIR system based on Bag of Features (BoF) representation of SIFT and Dense-SIFT local descriptors, and to evaluate it with the provided dataset using Mean Average Precision (MAP) metric. This evaluation should be reported in a **3-4 pages** long paper prepared in IEEE Conference Proceedings Template (LaTeX is recommended) provided in the following link.

`https://www.ieee.org/conferences_events/conferences/publishing/templates.html`

The text below provides a detailed explanation of the methods and the submission requirements.

2.1 Content Based Image Retrieval

The main purpose of a CBIR system is search for images that are semantically highly similar to a given query image, within a large image database. The search is typically performed by comparing the representation of each images in the database with the representation of the query image. The CBIR system, then, returns the set of images that seem highly similar to the query image.

A crucial step is the determination of the image representation obtained via *feature extraction*. If the image representation provides a good summary of the semantic contents of a given image, then the CBIR

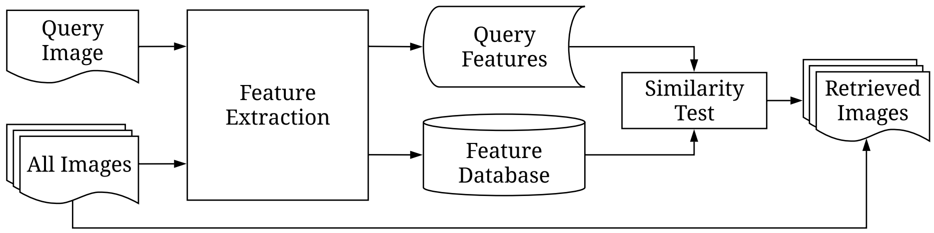

Figure 1: CBIR system pipeline

system is likely to perform well. However, a poor image representation (such as naive histogram of pixel intensities) is likely to lead to poor CBIR results.

A typical pipeline for CBIR systems is shown in Fig.1. The CBIR system pre-extracts the features of each image in its database. When a new query image is provided, first, the feature extraction is performed for the query image. Then, the similarity between the representation of the query image and each image from the database and the query image is measured. Finally, the CBIR system returns the most similar images from its database.

The goal of this assignment is to implement a basic CBIR system, using a simple Bag of Features (BoF, also known as Bag of Visual Words) pipeline. The pipeline that you are going to implement will consist of the following main steps:

- Construct a visual word dictionary for BoF representation (Section 2.2).
- Encode all images in BoF representation using the visual word dictionary (Section 2.3).
- Measure the similarity of each query image to all other images in the dataset (Section 2.4).

In the following subsections, we explain these steps that you are expected to implement in detail.

2.2 Dictionary construction using k-means clustering

BoF representation is based on an adaptation of a natural language processing method called Bag of Words to computer vision. In the bag-of-words approach for natural language processing, each textual document is represented by extracting a histogram of words, i.e. by counting the number of occurrences of each unique word. However, in computer vision, we do not directly have discrete entities like words as in natural languages. Therefore, in order to adapt the bag-of-words representation to computer vision, we need to define *visual words*. In this assignment, we will use *quantized* image patches to extract visual words from images. Here, in order to quantize image patches, you will encode each local patch by extracting a local descriptors and then assign unique ids to them according to a visual word dictionary.

In this section, we briefly explain the steps that you will need to implement to extract local descriptors and then construct a visual dictionary based on them.

Local descriptors In order to define the image patches and extract their local descriptors, you are expected to separately try the following two approaches:

- SIFT: define patches using DoG keypoint detector and extract SIFT descriptor of each interest region.
- Dense-SIFT: define patches at regular intervals (i.e. using a spatial grid) and extract the SIFT descriptor of each local patch.

Dictionary construction Once you extract the local descriptors of all images, you can construct a visual dictionary. For this purpose, load all local descriptors to memory, run *k-means* algorithm, and consider the resulting cluster centers as the visual dictionary. Then, each local descriptor can be mapped to a cluster id by finding the index of the cluster that has the smallest euclidean distance to its descriptor vector.

2.3 Bag of Features representation

At the end of the steps described in the previous section, each image contains a set of visual words. Based on them, we can easily extract the BoF representation of each image by computing the histogram of its visual words, where there is a histogram bin corresponding to each visual word (ie. each cluster center).

Since the number of visual words may vary across the images, particularly due to variations in image sizes, it is important to normalize each histogram by dividing each value by the number of visual words in the corresponding image. (This normalization scheme is known as ℓ_1 normalization.)

2.4 Ranking images by similarity

After obtaining the BoF representations of all images, one can find the images most similar to a given query image by comparing the BoF feature vector of the query image to the BoF feature vector of each other image in the database, and then sorting images in increasing distance order. For this purpose, you are expected to use simple Euclidean distance, given in Equation (1).

$$d_{euclidean}(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{1}$$

2.5 Programming and Interpretation Tasks

You are required to implement the aforementioned CBIR system using K-means clustering of SIFT and Dense-SIFT features with four different number of clusters (K=32,64,128,256). Your system should output a file that has the a result line for each query. A result line starts with the file name of the query followed with a colon. Then, for each image in the database a space, the distance between feature vectors of the images, another space and the name of the image in the database should be written. Each line must end with a newline character. Scripts for converting this into ranking, and evaluating this ranking using Mean Average Precision are provided.

Once your implementation is complete, you should evaluate your CBIR system with different configurations as mentioned before using the provided validation queries and the database. Finally, you are expected to decide the most successful configuration based on your experiments (using the validation queries)

One day before the deadline, you will be provided with test queries. We will use the test queries to check the consistency of the results that you report for the validation queries, and, also to evaluate the overall success of your CBIR system.

Important notes:

- An important **hint** about the implementation is saving results of intermediate steps. Since feature extraction for the whole database and clustering the local descriptors can be time consuming processes, saving the intermediate results for reuse is strongly recommended.

- Another important point is the grid size for Dense-SIFT: if the number of samples is too high, it will most likely lead to a out-of-memory problem. Hence, you should tune the parameters such that the algorithm still samples points in a dense way, but features fit into the memory. Overall, you should aim to maximize the performance of CBIR system according to the MAP score that you obtain using the validation queries.

Along with the implementation of a CBIR system, you are required to prepare a report that explains your work, rationale behind your choices and results of the experiments. It should include at least the following items:

- Detailed comparison of the feature descriptors SIFT vs Dense-SIFT.

- Discussion on the effects of the value of K.
- Discussion of rationale behind your choices of parameters.
- Qualitative evaluation of a few query results.

You are expected to submit a well-written, well-structured, professionally-written report that uses English language properly.

2.6 Database and Queries

The dataset consists of 1491 color images of size 640x480 or 480x640. You are provided 400 validation queries. A query is simply the name of an image whose content will be used for retrieving similar images. For these queries, the ground truth results are also provided. This allows you to evaluate your implementation and do experiments. The report will be based on the observations in the experiments for these validation queries. Ground truth information for a query has the form;

QueryImageName:(␣ResultImageName)*

For example;

dAazkBRVcJ.jpg: oxvGwRiawx.jpg gWyKnFwKBJ.jpg UfzkfCyHnW.jpg

In addition to these, you are provided two scripts *convert_for_eval.py* and *compute_map.py*. The former converts your output to a ranking for each query. The latter computes MAP using this ranking, where higher MAP scores indicate more accurate of retrieval results.

3 Restrictions and Tips

- Your implementation should be in Python 3.
- For SIFT / Dense-SIFT feature extraction, and, k-means clustering algorithm you can use any library that you want. However, BoF and CBIR implementation must be of your own. An Anaconda environment is provided in the homework files that includes OpenCV and cyvlfeat. You can utilize this environment, or create your own with different libraries for the algorithms.
- Do not use any available Python repository files without referring to them in your report.
- Don't forget that the code you are going to submit will also be subject to manual inspection.

4 Submission

- **Late Submission:** As in the syllabus.
- Implement the task in a directory named **the1**. The implementation together with a 3-to-4 pages long report focusing on theoretical and practical aspects you observed regarding this task and the **distance** results for test queries should be uploaded on COW before the specified deadline as a compressed archive file whose name is **<student_id>_the1.tar.gz**, e.g., 1234567_the1.tar.gz.
- The archive must contain **no directories** on top of implementation directory, report and the results document.
- Do not include the database and unmentioned files in the archive.

5 Regulations

1. **Cheating: We have zero tolerance policy for cheating**. People involved in cheating will be punished according to the university regulations.

2. **Newsgroup:** You must follow the course web page and ODTÜCLASS (odtuclass.metu.edu.tr) for discussions and possible updates on a daily basis.