Middle East Technical University | Department of Computer Engineering

CENG 483

Introduction to Computer Vision

Spring 2018-2019

Take Home Exam 1

Content Based Image Retrieval

Due date: **29 March 2019, 23:55**

1 Objectives

The purpose of this take home exam is to familiarize yourselves with the fundamental content based image retrieval (CBIR) pipeline along with histogram based descriptors. The assignment is expected to make you gain insight about the computer vision research and evaluation methods.

Keywords: *Content Based Image Retrieval, Color Histogram, Gradient Histogram, Orthogonal Derivative Filters, Grayscale Histogram, Grid Based Feature Extraction, Mean Average Precision*

2 Specifications

In this assignment you are required to implement a CBIR system based on different types of histogram features 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 the format of given LaTeX template.

The text continues with detailed explanations of the methods and requirements.

2.1 Content Based Image Retrieval

The main purpose of the CBIR systems is searching a large database for finding images that matches a query image. The search should be done by comparing semantic contents of the images in the database and the query. However, as we all know, images are represented as a collection of numbers in the lowest level. Hence, there is a difficulty in matching images, which is called the *semantic gap*. In order to overcome this difficulty, images should be described as semantically meaningful feature vectors which are (semantically) higher level representations than collection of numbers. You are going to implement some of these higher level representations in scope of this take home exam, but let us first take a glance of the general structure of the CBIR system.

The CBIR system pipeline starts with feature extraction of the query image and all other images in the database as seen in Fig.1. After obtaining all the features, a similarity test is applied between the features of each image in the database and the query image. Finally, based on the result of this test, the most similar images are identified.

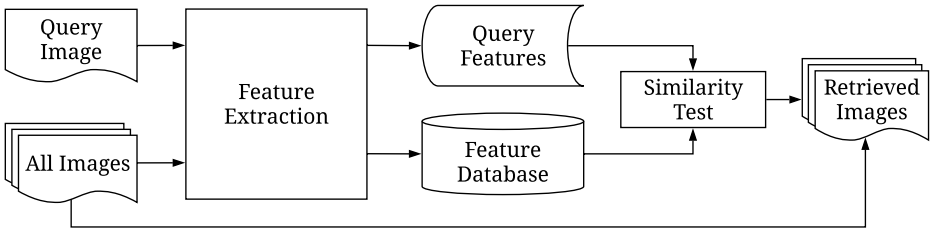

Figure 1: CBIR system pipeline

In this assignment you are required to use several kinds of histograms at different levels of spatial grids as feature extraction method. This is basically representing an image as frequencies of some visual characteristics, such as grayscale intensity, color and edge orientation. After obtaining the histogram representations, a distance metric should be utilized for similarity test. Euclidean distance given in (1) is one of the metrics that can be used. The smaller the distance between feature vectors, more similar the images are.

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

2.2 Histograms

A histogram is a vector that counts how many instances of a given property exist in the image. First step of creating a histogram is to define ranges to determine bins, then each instance is assigned into one of these bins. For example, the property in the hand may be the intensity values, then the length of the histogram will be 256 (for values between 0 and 255, with step size 1) for grayscale images and a histogram is obtained by counting how many instances occur for each bin in the image. It is possible to play and experiment with the level of quantization (number of bins) by defining ranges with different step sizes, you are expected to select several different quantization levels (for example, 1 bin, 128 bins, 256 bins for grayscale intensity) and fill in the corresponding parts in your report with your results.

It is recommended to apply ℓ_1 *normalization* to the histogram as a final step, such that the total count of each histogram sums up to 1.

In this take home exam, you will be implementing histograms for three different types of data sources: (1) grayscale intensities, (2) color values, and (3) grayscale image gradients. These are briefly explained in the following sections.

2.2.1 Grayscale Intensity Histogram

This is simply obtained by quantizing the pixels into histogram bins based on their intensity level and then computing the frequency of each intensity bin in the image.

2.2.2 3D Color Histogram

The color channel histogram can be obtained by quantizing pixels at each color channel separately and then assigning pixels into combination of bins of these three histograms, i.e. treating each combination of the bins of three separate histograms as a single bin of the resulting histogram. For example, if there are 10 bins for each color channel, when we take combinations of all of the bins of all of the histograms, we would result with an histogram with 1000 different bins. You can check https://en.wikipedia.org/wiki/Color_histogram for further information.

2.2.3 Image Gradient Histogram

An edge is basically where a sudden intensity change occur among neighboring pixels, which can be interpreted through gradient magnitudes and orientations at a pixel.

In this take home exam, you are expected to extract gradient histogram for this edge map where each bin value is the total magnitude of the pixels whose orientations are quantized into that bin. In order to obtain magnitude and orientation values, you need to apply horizontal and vertical filters (orthogonal derivative filters) to the map in hand for getting the gradient at each pixel. Assuming v and h are the responses of the vertical and horizontal filters respectively, orientation(θ) and magnitude(a) of a pixel at coordinates i and j can be obtained as:

$$\theta_{ij} = arctan[v_{ij}/h_{ij}] \tag{2}$$

$$a_{ij} = \sqrt{v_{ij}^2 + h_{ij}^2} \tag{3}$$

2.3 Grid Based Feature Extraction

The features we have mentioned above can be extracted at different spatial levels. For example, you can create an histogram for the image itself (level 1) which results in a single histogram; or you can divide the image into a grid and extract the histogram for each cell of the grid individually, and then concatenate the resulting histograms. For this take home exam, level 2 corresponds to constructing a 2×2 grid and level 3 corresponds to constructing a 4×4 grid. You can see an overview of the idea in Fig. 2.

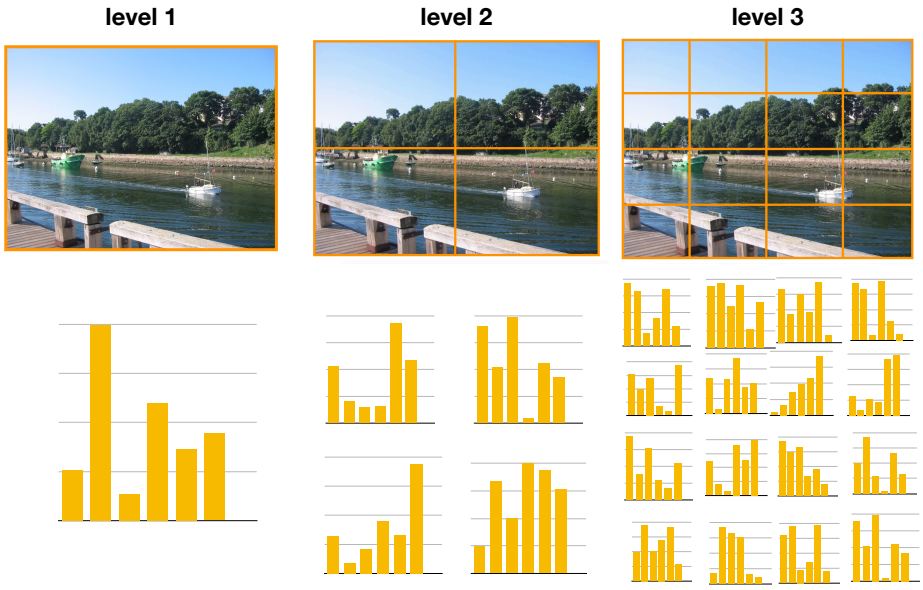

Figure 2: Grid based feature extraction, visualized with histogram with 6 bins as a feature. Note that the values in the histograms are selected arbitrarily to visualize the concept, so they are not meaningful.

2.4 Programming and Interpretation Tasks

You are required to implement the aforementioned CBIR system using histograms of grayscale intensity, color intensity and edge magnitude & orientation (magnitude and orientaion combined). 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 will be provided. After implementation, you should evaluate your CBIR system with different configurations as mentioned before using the provided validation queries and the database. The configurations are given in detail in your report template. Although you are expected to do experiments with these configurations, you are free to add your own configurations as well, just do not forget to mention it in your report. Finally, you will decide on the most successful configuration based on your experiments and submit to ranking results. One day before the deadline, you will be provided with test queries.

An important **hint** about the implementation is saving results of intermediate steps. Since feature extraction for the whole database is a time consuming process, saving the results for reuse is strongly recommended. Also, you may want to visualize intermediate steps for edge detection process, so you can monitor if you are going well or not.

Along with the implementation of a CBIR system, you are required to prepare a report that explains your work. A template will be given to you, and reports in any other format will not be accepted.

2.5 Database and Queries

The dataset consists of 1491 color images of size 640x480 or 480x640. For validation part it includes 400 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.

3 Restrictions and Tips

- Your implementation should be in Python 3.
- Histogram, grid based feature extraction, gradient extraction and CBIR implementations must be of your own.
- You may use any available convolution function for filtering (scipy.signal.convolve2d etc.).
- 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.
- Stick with the given template for your report. We will be running your codes for your best configuration found on the validation set in order to reproduce your ranking results, so please do not forget to mention your setup explicitly.

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 ODTÜCLASS 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.