

CENG 483

Introduction to Computer Vision

Fall 2021-2022

Take Home Exam 1

Instance Recognition with Color Histograms

Due date: **8 December 2021, 23:55**

1 Objectives

The purpose of this take home exam is to familiarize yourselves with the simple instance recognition task with color histograms. The assignment is expected to make you gain insight about the computer vision research and evaluation methods.

2 Background

In this assignment you are required to implement an Instance Recognition system based on different types of color histograms and evaluate it with the provided dataset using *top-1 accuracy*. The evaluation results and discussions are required to be reported in a **2-3 pages** long paper using the given L^AT_EX template.

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

2.1 Instance recognition

Instance recognition is a visual recognition task to recognize an instance of an object. The task is to find additional images of a particular object within a dataset, from a given single image of that object. Although nowadays instance recognition is typically done using more sophisticated approaches, in this assignment you will use simple color histograms to measure similarity of image pairs.

2.2 Color histogram

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 interval of quantization (to control number of bins) by defining ranges with

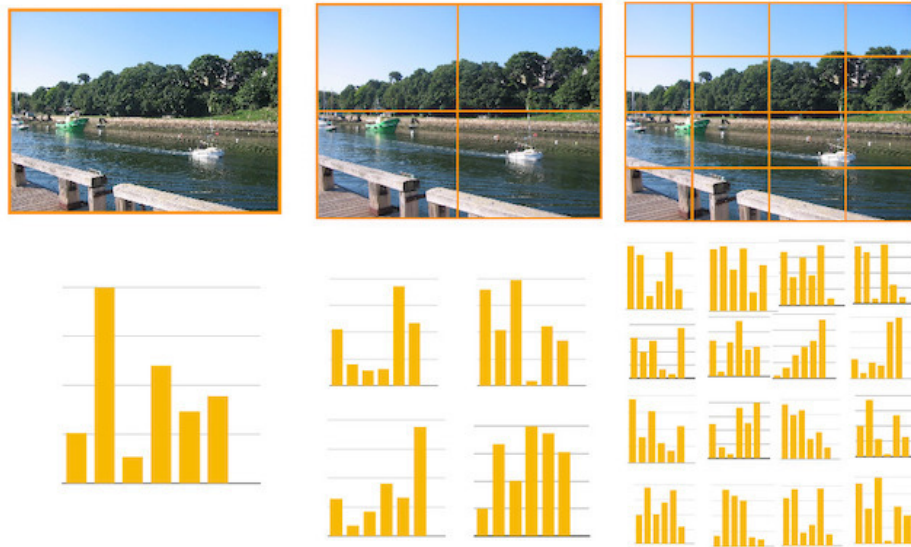


Figure 1: Grid based feature extraction using 1×1 , 2×2 and 4×4 grids. In each case, a histogram with 6 bins is computed at each grid cell. Note that the values in the histograms are selected arbitrarily to visualize the concept, so they are not meaningful.

different sizes, you are expected to select several different quantization intervals (for example, interval of 2 means 128 bins, interval of 32 means 8 bins for grayscale intensity) and fill in the corresponding parts in your report with your results. More detailed information can be found in the lecture videos.

In this take home exam, you will be implementing two types of histograms for color values. This is briefly explained in the following sections.

2.2.1 Per Channel Color Histogram

In the per-channel color histogram scheme, you need to compute a histogram at each channel separately. For a given pair of images, you may first compute similarity of each histogram separately (explained below) and then take the average of the similarity values.

2.2.2 3D Color Histogram

The color channel histogram can be obtained by first 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 quantization of three separate channels 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.

2.3 Grid based feature extraction

A simple option is to compute histograms globally over the whole image. Alternatively, you can split the image into regions (grid cells) within a regular grid, and compute histograms separately at these cells. You can see an overview of the idea in Fig. 1.

2.4 Computing image dissimilarity

Using the schemes described above you will compute multiple histograms at each image. To compute the dissimilarity between two given images, you need to measure the dissimilarity of the corresponding

histogram pairs across the images. In this assignment, you are required to use *Kullback-Leibler divergence* for this purpose, which is defined as:

$$D_{KL}(Q||S) = \sum_{i \in \text{NumHistogramBins}} Q(i) \log \left(\frac{Q(i)}{S(i)} \right) \quad (1)$$

where P and Q are both probability distributions corresponding to the query image (the input image whose other images are being searched for) and support image (an image that is being compared against the query image). There are a few things that you need to be careful about using the KL divergence:

- KL divergence is a *divergence* measure defined for probability distributions. You can convert a histogram into a categorical probability distribution by applying ℓ_1 *normalization* to the histogram. This basically means dividing each entry in a single histogram by ℓ_1 norm of that histogram. You must do this normalization *before* computing KL divergence.
- KL divergence is *asymmetric*. Therefore, the result may change whether you compute query to support, or support to query divergence. For the assignment, you are required to compute query to support divergence, i.e. Q should be a query histogram and S should be a support histogram at each KL divergence calculation.
- When calculating per-channel color histogram and/or using a spatial grid, you will be computing multiple histograms per image. In this case, compute KL divergences between each corresponding pair of histograms, and then take average of the resulting KL divergences to be used as the dissimilarity measure between the images.

2.5 Evaluation methodology

You are provided four sets of images: 1 support set of images (where you do the search) and 3 different query sets (inputs to your search algorithm). At each one of your experiments, you will be using one of the query sets and a particular configuration of your approach (guidance is provided in the template).

In a particular experiment (with a particular query set), for each query image, you need to search the most similar support image as follows: First, calculate average KL-divergence of the query image to each one of the support images, as described above. Then choose the image with the lowest average KL-divergence as the retrieved image. You need to repeat this process for each query image in the query set. Once this process is completed, compute the top-1 accuracy as follows:

$$\text{accuracy}_{\text{top-1}} = \frac{\# \text{ of correctly retrieved images}}{\# \text{ of query images}} \quad (2)$$

You can understand whether a retrieved support (ie. least dissimilar) image is correct or not using the filenames.

2.6 Programming and Interpretation Tasks

You should evaluate your instance recognition system with different configurations by using the provided query datasets. 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.

Along with the implementation of a Instance Recognition system, you are also submit a report that explains your work. A template is given to you, and reports in any other format will not be accepted. In your report you are expected to discuss and highlight relative strengths and weaknesses of certain configurations on each of query datasets (query1, query2 and query3) and explain why these can be the case. Focus on the top performing and worst performing configurations, try to explain why that is the case concisely.

2.7 Support and Query Datasets

Each image set contains 200 images of size 96×96 . Image names in every dataset is same and they will be given to you as a txt file. You are expected to match images with same names in query datasets with their support counterparts. Each query dataset consists of images from support set with various transformations applied on top of them. The report will be based on the observations in the experiments for these queries. **Investigate the query sets to understand what kind transformations applied to them, which is crucial for the discussions that you are expected to present in the report.**

3 Restrictions and Tips

- Your implementation should be in Python 3.
- You should solely use numpy.
- Histogram, grid based feature extraction, and Instance Recognition implementations must be of your own.
- 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.
- Coming up with a correct implementation is important. We may randomly check for random configurations.

4 Submission

- **Late Submission:** As in the syllabus.
- Implement the task in a directory named **the1**. The **implementation** together with a **2-to-3 pages long report** focusing on theoretical and practical aspects you observed regarding this task 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 dataset 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.