Middle East Technical University   Department of Computer Engineering

CENG 483

Introduction to Computer Vision

Spring 2018-2019

Take Home Exam 2
Object Recognition

Due date: **5 May 2019 23:55**

1 Objectives

The main purpose of this assignment is to gain experience on object recognition task, which basically is to tell which one of the predefined object types that a given object instance belongs to. This requires to represent objects in semantically meaningful ways (bag of words for this assignment) and to derive a way to determine which class is most likely for that object.

Keywords: *Object Recognition, Bag of Words, Classification, k-Nearest Neighbors Classifier, SIFT, Dense-SIFT*

2 Object Recognition

In object recognition problem, we have more than one type of objects (i.e. apples, cars, books. etc) and the goal is to be able to tell type of a previously not seen object instance. In other words, we need to classify a test object. To achieve this goal, one should be able to have an idea for what are the properties that a particular class has, preferably in a semantically meaningful way. As such a feature, we are going to use Bag of Features (BoF) in this assignment. Later, we need to devise a classifier, which will be used to predict the class of a given input image. In this assignment, we will be using k-nearest neighbors classifier for this purpose.

3 Bag of Features

As mentioned before, we will be using Bag of Features to represent images in this homework. BoF representation originates from texture recognition and is an adaptation of a natural language processing method called Bag of Words. Textures can be detected using interest point detectors, sampled randomly or densely. Then, local feature descriptors (such as SIFT or Dense SIFT) are used to describe the texture in the neighborhood of an interest point or a sample point. Finally, all of the texture descriptions in the database are clustered to form a dictionary. This dictionary is basically a codebook consisting of repetitive local properties of textures and can be used to represent images. When Bag of Features is used

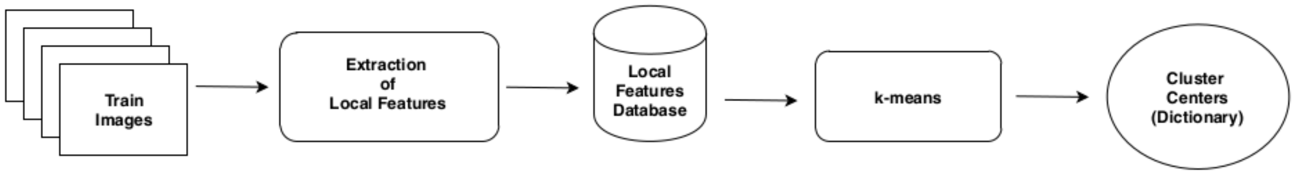

Figure 1: Pipeline for dictionary construction.

for object recognition, these properties will most likely be object parts (e.g. ears of dogs or tires of cars, etc.), where each property corresponds to a cluster center.

Once the dictionary is created, any image can be represented as the frequencies of the words in the dictionary. To this end, one should first detect interest points or sample some random/dense points in the image as in the previous part. Then, these should be described and assigned to most similar word in the dictionary. The final step is creating an histogram of these where bins represents the visual words in the dictionary. The un-normalized histogram is obtained by counting the occurrence of each visual word in an image is counted. This histogram is then normalized to make the representation invariant to the number of interest regions. The resulting normalized histogram, which is a fixed length vector, is then used as the image representation for the corresponding image.

In this assignment. you are required to use interest point based SIFT and grid sampling based *Dense-SIFT* local feature descriptors along with K-means Clustering algorithm to create BoF representations. Note that you should first create a database consisting of all local feature descriptors in the training set and then cluster this database using K-means algorithm. Resulting cluster centers will compose our dictionary.

4 K-means Clustering

K-means clustering is a very intuitive and easy to implement algorithm. It consists of an initialization step and an alternating procedure that stops when there is no change as a result of alternations. At the initialization step K random points are chosen to be cluster means. The alternation process is as follows

- Assign data instances to the closest cluster center using a distance metric (e.g., Euclidean Distance given in Eq. 1.
- Update each cluster center as the mean of its assigned data instances.

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

During the iterations, you may encounter *empty cluster centers*, where an empty center is the one to which no points have been assigned to. To handle such cases, you can implement a heuristic that *drops* empty clusters or re-initialize them with random values.

You may limit the number of iterations if convergence takes too long. Avoid stopping the algorithm too early, though.

5 Classification and Evaluation

As explained in Section 2, you are expected to use k-Nearest Neighbor Classifier to decide the class of an input image (from the validation or the test set). As features, you are expected to use your own

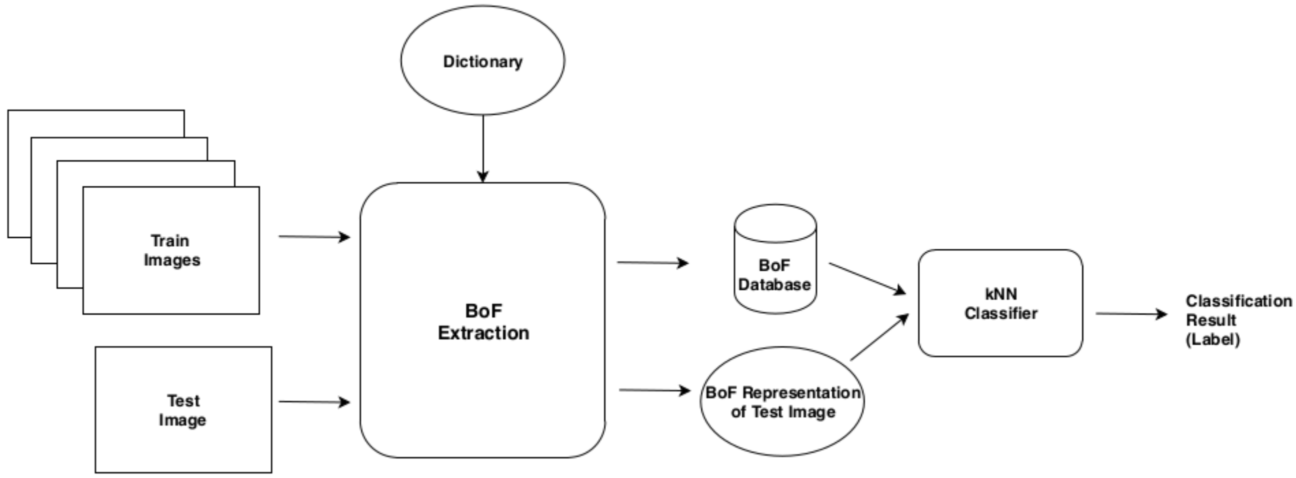

Figure 2: Pipeline for classification.

BoF representations to represent objects. Detailed expectations about the experiments that you need to conduct and report are given in the report template as in the first assignment. As the evaluation method, we will be using accuracy, which basically is the ratio of correctly classified instances in the validation / test set.

6 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. Bag of Features and k-Nearest Neighbors implementation must be of your own.
- You may use np.histogram.
- You may run k-means algorithm by collecting SIFT/Dense-SIFT samples from a subset of the training images. Usually a larger sample set leads to a better vocabulary. Therefore, try to take a large set of examples, but avoid using a too large set that would require virtual memory access during the k-means iterations.
- You probably do not want to write SIFT / Dense-SIFT to disk. Instead, store only the bag-of-words histograms.
- Dataset is available **here**.
- Do not use any available Python repository files without referring to them in your report. Ask for permission if you will use any external code / repository.
- 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 classification results, so please do not forget to mention your setup explicitly.
- 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.

- **Be careful** about k values of k-means and k-nearest neighbor algorithms. These two are distinct parameters and have no relation at all. When you use them in your report, do not forget to explicitly state which one do you mean.
- Test set will be given to you one day before the deadline (keep checking ODTÜCLASS discussions).
- Upload your classification results in a text file for all test images in the following format:
  test_image_name: label
  an example: "00065.png: apple"

7 Submission

- **Late Submission:** As in the syllabus.
- Implement the task in a directory named **the2**. The implementation together with a report and your classification results for the test set should be uploaded on ODTÜCLASS before the specified deadline as a compressed archive file whose name is **<student_id>_the2.tar.gz**, e.g., 1234567_the2.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.

8 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.