Computer Vision

Exercises of Lab 14

Exercise 14.1: Human Action Recognition based on human body poses

The goal of this exercise is to perform human action recognition using global human body representations (human body poses). In this exercise we will use (pre-processed) videos from the Weizemann action database. The database also provides binary masks obtained by applying background subtraction (direct link here). We will use these binary masks in order to apply human action recognition based on the method in this paper.

Make also sure that you have downloaded the binary sequences and change the variable Weizemann_dir to the corresponding directory at your hard drive.

Open Exercise14.1.m and read the code. In order to learn a person-independent action recognition model, we will apply person-based evaluation. This means that we will use the sequences of some of the persons in the database for training the action recognition model, and we will evaluate its performance on the sequences depicting different persons. Lines 11-12 include the list of persons used for training and testing, respectively. Lines 13-16 include parameters of the method. Upon completion of the experiment, you can change the values of these parameters in order to observe their effect on performance.

On lines 57 and 108 you are asked to determine the bounding box of the human body a video frame. You can do this by using the function regionprops() of Matlab.

On line 131 you are asked to calculate the prototype poses (also called Dynemes), by clustering the human body poses of the training sequences.

On lines 150-172, each sequence is represented by using the fuzzy histogram of the human body poses forming it using the Dynemes. On line 158 you are asked to calculate the Euclidean distances between all poses and the Dynemes. These distances are later used to calculate the fuzzy histogram with a fuzzification parameter fuzz.

Given the sequence representations based on fuzzy histograms, Linear Discriminant Analysis is applied in lines 175-199. You are asked to calculate the within-class and between-class scatter matrices in lines 180 and 184, respectively.

After projecting the training vectors in the LDA space, a Nearest Class Centroid classifier is trained. You are asked to calculate the projected training data representations and the NCC's parameters in lines 204-205.

Evaluation of the method (using the test sequences) is performed on lines 210-245. Here you are asked to apply the same processing steps as with the training phase in order to obtain the representations of the test sequences. Finally, on line 242, you are asked to classify the test sequences using the NCC classifier trained on the training sequences.

Exercise 14.2: Human Action Recognition based on Improved Dense Trajectories

In this exercise, you will apply human action recognition on the Hollywood2 action database, using the Improved Dense Trajectories video description scheme combined with the Bag of Features model. Due to the (relatively) big size of the database and the high computational cost and storage requirements of the IDT description scheme, the video representations for each descriptor type (Traj, HOG, HOF, MBHx, MBHy) are included in the files of the exercise.

Make also sure that you have changed the variable Hollywood_dir to the corresponding directory at your hard drive.

Open Exercise14.2.m and read the code. The database provides different sets for training and testing, thus, we will use the training video representations to train an action recognition model and we will evaluate it's performance on the video representations of the test videos. Labels of both training and test videos are provided by the database (and are also included in the files of the exercise). You can take a look at the label files, in order to observe that some videos are labeled with multiple actions (this means that in these videos more than one actions appear).

The IDT+BoFs action recognition model is usually combined with kernel-based classification models exploiting the χ^2-distance function. An extensive evaluation of various related approaches can be found here. The RBF kernel function between two histograms H_i and H_j (each having V bins) and the χ^2-distance is defined as:

$$K(H_i,H_j) = exp(-\frac{1}{2A}\sum_{n=1}^{V}\frac{(h_{in}-h_{jn})^2}{h_{in}+h_{jn}})$$

where A is the mean value of distances between all training histograms.

On line 37 you are asked to write a function chiSquare_distance(X,Y) calculating the χ2-distances between the histograms forming the columns of the matrices X and Y. Subsequently, on lines 41, 47, 52, 57 and 62 you are asked to calculate the kernel matrices for the training videos using each video representation type.

After fusing the information encoded by each video representation type using the average kernel matrix, a kernel Regression model is used. On line 78 you are asked to calculate the regression weights Amat.

During testing, the same processing steps are performed in order to calculate the kernel matrices for the different video representation type. On lines 89, 93, 97, 101 and 105 you are asked to calculate the corresponding kernel matrices. Finally, on line 115, you are asked to calculate the output of the kernel regression model for the fused kernel of the test videos (Ktest).

Notice that the performance of the model is measured by calculating the mean Average Precision (mAP) metric. This is because some videos might depict more than one actions.