ÜBUNG 183.651

Video Analysis

2nd Assignment

SS 2017

Sebastian Zambanini

Computer Vision Lab
Institut für Rechnergestützte Automation
http://www.caa.tuwien.ac.at/cvl/course/video-analysis-ue/

vidana@caa.tuwien.ac.at

1 Social Interaction Analysis

The second assignment consists in the analysis of short pieces of video in order to classify the social interaction ongoing. The expected result is a software that reads some input videos and proposes the final classification decision among a set of possible classes.

You will receive a training set that you will use to train and tune your classifier. You are expected to produce a software/matlab function that will be tested on a brand new test set. In both, training and test sets, the classes will be 4: *Kiss*, *Hug*, *HighFive* and *HandShake*. You can develop your algorithm on the training set that can be downloaded at: `ftp://scruffy.caa.tuwien.ac.at/Lehre/vidana13/Assignment2.rar`.

When you submit your work (code + report) it will be evaluated on a test set of different videos. The program can be coded in Matlab or C++ (OpenCV). For sake of practicality, you must follow the next conventions:
Input

- Test sequences will be placed in a `test` folder. However, at this stage, the class name will not appear in clear in the file name (i.e. `test/001.avi` as `001.avi` might belong to either one of the four classes). Your function/software should be able to read all the files and produce the following output file.

Output

- A `result.txt` file formatted as `'videoname.avi\tclassname\n'` i.e.:
  ```
  'test/001.avi HandShake'
  'test/002.avi Hug'
  'test/003.avi Kiss'
  ```

The choice of the parameters and the methodology, are at your discretion. Please highlight them with appropriate motivation in the final report. You can perform the validation on the given training set, using the *leave-4-out* cross-validation where 4 is the number of classes (this means one sample per class). You should comment the performance of your method on the given training data in your report. Please describe carefully how you get to those results using the evaluation methodology reported in Section 1.3.

1.1 Description of a Sample

In order to be able to classify a video sequence it is important to extract the discriminant information from the sequence itself. For this particular task the most successful technique is a descriptor based on visual features.

Visual features differ among each other essentially on two aspects: The way the information is collected (i.e. globally as in Long Displacement Optical Flow (LDOF) or locally like in Dense Trajectories and Spatio Temporal Interesting Points (STIP)), and on what type of information is fetched (i.e. spatial as in the Histogram of Gradients or temporal as in Histogram of Optical Flow).

In order to have a brief overview on the dynamic visual features we suggest you to take a look to the following paper:

- *Tamrakar, Amir, et al. "Evaluation of low-level features and their combinations for complex event detection in open source videos." Computer Vision and Pattern Recognition (CVPR), 2012.*[1]

1.2 Classification

Once you have transformed the video in a feature vector, your task is now to perform the decision on the video. In this case we are dealing with a supervised classification task. There are many methods that can be employed, in the following we are going to suggest some of them:

- **Support Vector Machine (SVM)** is probably the most used "black box" classifier. It is essentially a two-class classifier that can be easily extended to multiclass using a *one-against-the-rest* approach. In literature there are many implementation of the SVM, as LibSVM[2] that is also implemented in OpenCV.

- **K-Nearest Neighbors (K-NN)** is a simple classification method based on the properties of the K closest samples. This method can be easily implemented by your own.

- **Citation Nearest Neighbor (C-KNN)** is a variation of the previous method that considers not only the closest training samples but also those samples that would "cite" the test sample as one of the neighbors.

- **Neural Networks** are nowadays one of the most fashionable tool for image classification. The main reason for its success is given not only by the possibility of using strong computational power and GPU processing but also for the huge availability of data (i.e. ImageNet) in order to build reliable inter-layer connections. Neural Network is composed by nodes linked by weighted connections divided in a certain number of layers. The input layer is represented by the features (or the raw images) while in the last layer lie the final decision nodes. Also in this case there are many toolboxes which are implemented for Matlab or Python (Matlab Deep Learning Toolbox[3], Caffe[4], Theano[5], TensorFlow[6], etc.).

- **Random Forest** is also a much valuable classification tool that relies on decision trees and information theory to perform the final inference. Also in this case an implementation is available in the OpenCV libraries.

  The classification procedure is not supposed to be limited to the previously mentioned methods, any other successful method is also encouraged.

1.3 Evaluation

The evaluation will consider accuracy, precision and recall. All of these parameters are computed from the confusion matrix (Fig. 1).

[1] `http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6248114`

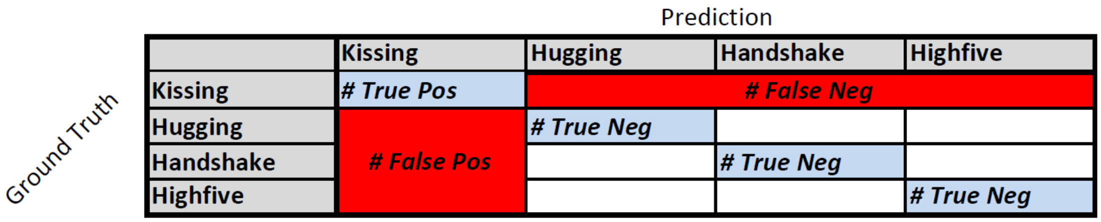
[2] `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

[3] `https://github.com/rasmusbergpalm/DeepLearnToolbox`

[4] `http://caffe.berkeleyvision.org/`

[5] `http://deeplearning.net/software/theano/`

[6] `https://www.tensorflow.org/`



		Prediction			
		Kissing	Hugging	Handshake	Highfive
Ground Truth	Kissing	# True Pos	# False Neg		
	Hugging	# False Pos	# True Neg		
	Handshake			# True Neg	
	Highfive				# True Neg

Figure 1: An example of classification performance evaluation for *kissing* class.

Accuracy is the overall correctness of the model and is calculated as the sum of correct classifications divided by the total number of classifications.

Precision is a measure of the accuracy provided that a specific class has been predicted. It is defined by:

$$Prec = \frac{Tp}{Tp + Fp} \tag{1}$$

Recall is a measure of the ability of a prediction model to select instances of a certain class from a data set. It is commonly also called *sensitivity*, and corresponds to the true positive rate. It is defined by the formula:

$$TPR = \frac{Tp}{Tp + Fn} \tag{2}$$