Exercise for MA-INF 2218 Video Analytics SS18
Submission on 16.05.2018
Dense Trajectories

You have to implement your solution using python2.7. For your implementation, you are allowed to use OpenCV 2.x, numpy, scikit-learn to its full extend. You do not need to implement every single component like Optical Flow on your own. Please comment your code appropriately. You will continue with the code (or the model solution) for the next exercises, so try to organize it neatly.

1. Extract dense trajectories for each video (only for single spatial scale, you don't need to apply any smothing), and calculate a 30 dimensional trajectory shape descriptor for each trajectory. (see Section 3.1, 3.2 from [1]).
   Hint: Use the video `dummy.avi` for debugging. Your trajectories should start near corners (but not on homogeneous regions) of the rectangle when it moves. *(7 Points)*

2. Around each extracted 15 frame long dense trajectory, create a volume of $32 \times 32 \times 15$ (where 32×32 is spatial window, and 15 is the length of trajectory). Further divide this volume into tubes of size $2 \times 2 \times 3$. For each tube create

   - HoG with 8 bins
   - HoF with 9 bins
   - MBHx with 8 bins.
   - MBHy with 8 bins.

   Concatenate extracted feature for each tube, this will result in a 96 dimensional HoG, a 108 dimensional HoF, a 96 dimesnional MBHx, a 96 dimensional MBHy (see Section 3.3 from [1], also Figure 2 right from [1]). Finally, concatenate 5 descriptors into one vector as (Trajectory Shape, HoG, HoF, MBHx, MBHy). Thus, the final vector should be of 426 dimensions. *(6 Points)*

3. Calculate the video representation, this involves.

   - Apply PCA on your calculated features, to reduce dimension to 64 from 426.
   - Calculate Fisher Vector representation for each video using (See slide 90 (action recognition) from lecture).

   (5 Points)

4. On the provided dataset, train your system with the videos specified in `train.txt` and report the accuracy on the test set (`test.txt`). Each line of the (train.txt or test.txt) contains file name and its label separated by a blank space. With Fisher vector approach, you will have one feature vector per video, so you have a simple classification task. Train a support vector machine and classify the test videos, use a linear kernel for your SVM. *(2 Points)*

Since video processing is expensive, make sure to write efficient code. Please make sure to write comment your code, also make sure to write well structured code. If you have any questions, feel free to contact me (**iqbalm@iai.uni-bonn.de**).

[1] H. Wang: Dense trajectories and motion boundary descriptors for action recognition, 2013