Assignment 4

Basic Image Processing Algorithms
Fall 2022

General rules

This is the last Assignment. There will be no one more Assignments during the semester. Remember, you must reach a minimum of 40 / 100 points to get the Teachers' Signature.

Point value: **30 points**, which is 30% of the total assignment points.

Deadline: **December 5, 2022 23:59:59** (grace period ends on Dec. 7)

This is a not-guided exercise. The description of this assignment is general and does not focus on the details as in case of the Lab exercises.

The main task is to provide a good, reasonable solution. You may code "freely" (only minimal restrictions on file names and outputs are given).

Problem formulation

You have to write a custom classifier program that can classify human actions based on small (few frames) video sequences of a single person performing the action.

Input: a set of videos (image-sequences) to be classified

Output: a label (classification result) for each input object

Tasks to do

Write an appropriate feature extractor (3D GLOH descriptor), extract the features from the training data, train a model and classify some test data.

The feature extractor receives an image sequence and returns a “good” feature vector that has small intra-class and large inter-class variance.

The training phase uses the aforementioned extractor on labelled data and trains a model (e.g. SVM) for a classification task.

The trained classifier processes unseen (test) data and gives a class label for each instance in this set.

Key results to be presented:

You may code freely, as there are not so many restrictions on what to use.

However, you should create the followings as the result of your work:

main script which loads training data, trains the model and performs classification with test data

feature extraction function which is used to create 3D GLOH vectors from the input image sequence

model parameters of the trained classifier, and

additional **figures** and **text outputs** as described in the upcoming slides.

Training & testing data

The training (and test) data [1] has 3 classes: walking, clapping, boxing

Each training class has 10 short image sequences in a MAT file, stored as a 4D array. In this structure, the fourth index is the ID of the sequence (1-10), the third one is the frame index within the sequence (1-50), and the first two coordinates are the pixel locations (row, column) in a single frame.

Testing is done with a test data set containing 5 unlabelled sequences stored in the same way as for the training data.

Feature extractor function

This function has exactly one input argument. The extracted feature has to be the 3D GLOH descriptor at the three most significant points.

The 3D GLOH feature is described in articles [2] and [3], additional hints and specification can be found in the upcoming slides.

There is no code package for this assignment.

All scripts and functions must be written entirely by you.

Download the image to be processed from here:
https://beta.dev.itk.ppke.hu/bipa/assignment_04

Submission & hints

You should create a script named `a04_NEPTUN_train.m` where the NEPTUN part is your Neptun ID. This has to be the first script; running that must be able to prepare the training data.

The created feature extractor function should be called `extract_feature` This function must have a single input argument (an H×W×F matrix) where H and W are the frame height and width, and F is the number of frames in the sequence (50). The extracted feature must be the one described in [2]

Submission & hints

The training of the classifier is done using a Matlab App. The trained classifier (model) must be saved as `action_classifier.mat`

A second script file named `a04_NEPTUN_test.m` has to be created as well. This file should load the trained classifier from the .mat file, load the test data and do a prediction with the test data.

You are allowed to create other files (e.g. additional functions) too, if necessary.

Please submit **ALL** files (including the input folder as well) in a **ZIP** file via the Moodle system. **Check the upcoming slides for hints!**

Hint 1

The 3D GLOH feature means a three-dimensional Gradient Location and Orientation Histogram, which is created at “interesting” points in space-time.

Each image sequence (video) is described by a single feature vector that is created from 5 concatenated 3D GLOH features.

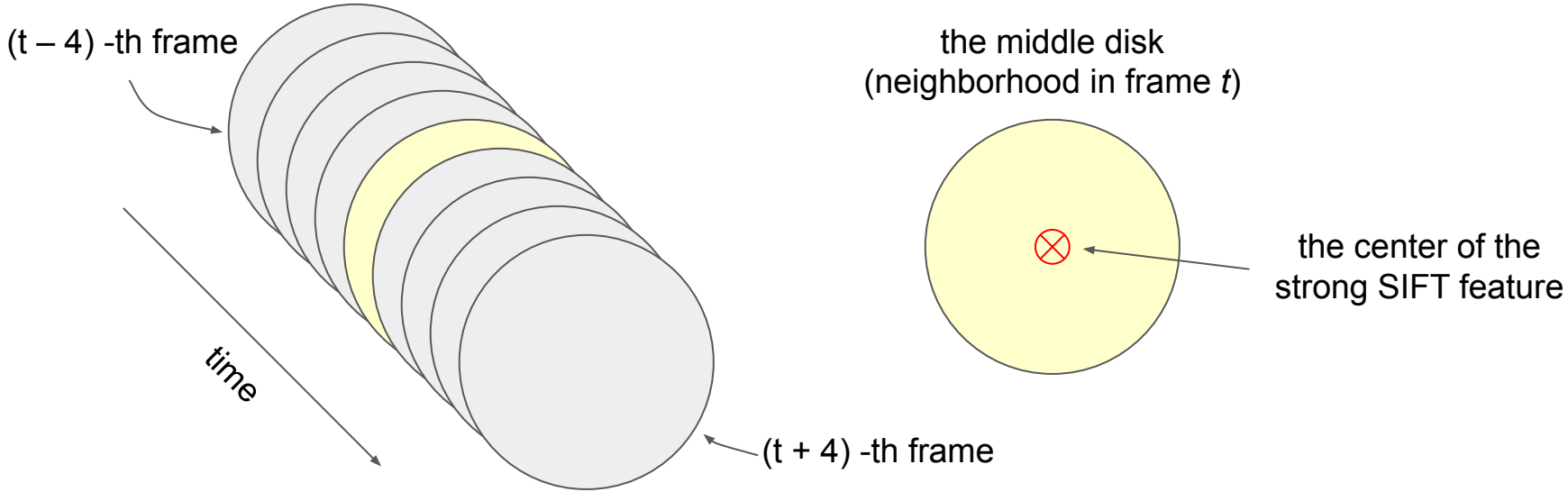
First, “interesting points” should be detected on a single frame (e.g., use the 10th frame, or anything you wish) using the SIFT method. You may use the Matalb’s built-in `detectSIFTFeatures` function.

Next, the 5 strongest feature points has to be selected at which the 3D GLOH feature is to be extracted.

Hint 2

The feature is created from a “cylinder” in space-time. The cylinder is constructed from 9 disks, each having a diameter of 31 pixels. The selected feature point (the strong SIFT feature point) is in the middle of the cylinder.

The disks represent the same neighborhood in the neighboring frames.



Hint 3

The disks are divided into 17 regions. A pixels belongs to a region based on its polar coordinates:

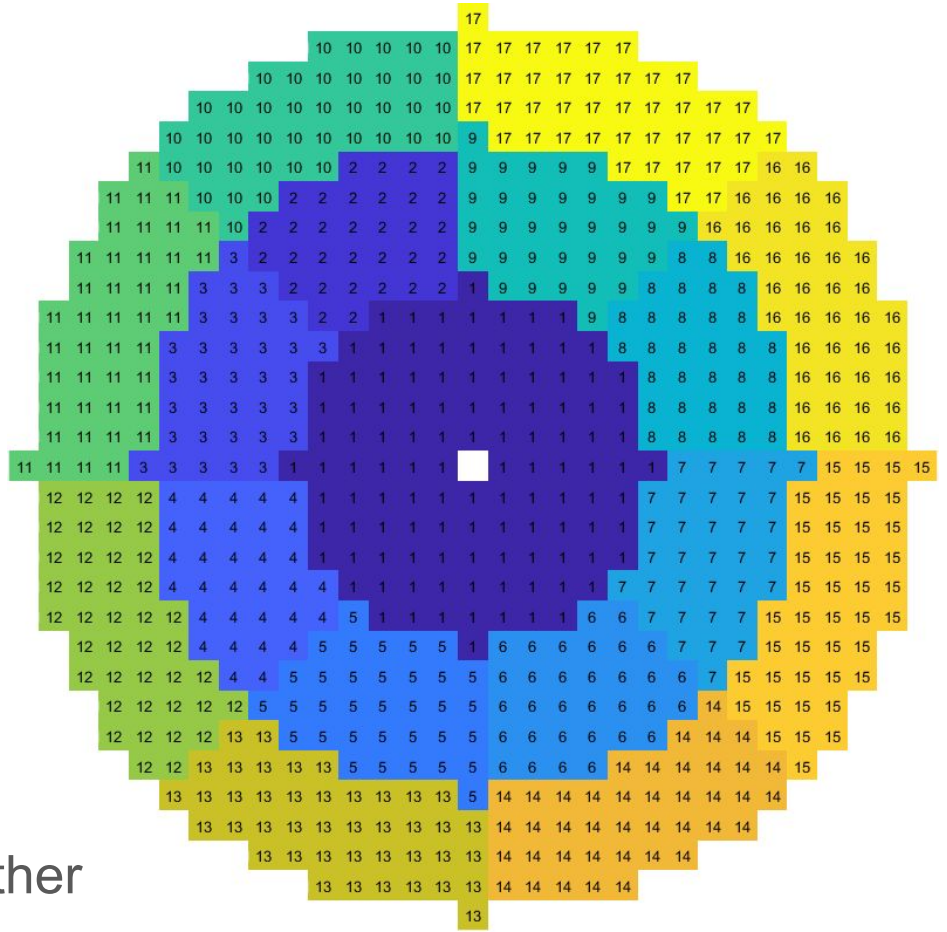
$$r_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2},$$

$$\theta_i = \tan^{-1}(y_i - y_c / x_i - x_c),$$

The regions are where the radius is

$0 < r \leq 6$ and $6 < r \leq 11$ and $11 < r$

Every region but the center one is further divided into 8 parts based on the angle.



Hint 4

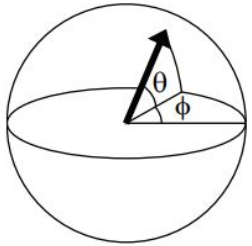
In each region of the consecutive disks, the 3D gradients are calculated as described in [3]. For each cropped region, compute the 3D magnitude and the two angle values as

$$m_{3D}(x,y,t) = \sqrt{L_x^2 + L_y^2 + L_t^2},$$
$$\theta(x,y,t) = \tan^{-1}(L_y/L_x),$$
$$\phi(x,y,t) = \tan^{-1}\left(\frac{L_t}{\sqrt{L_x^2 + L_y^2}}\right).$$

where the L values are computed using finite difference approximations, for example L_t is approximated by $L(x,y,t+1) - L(x,y,t-1)$

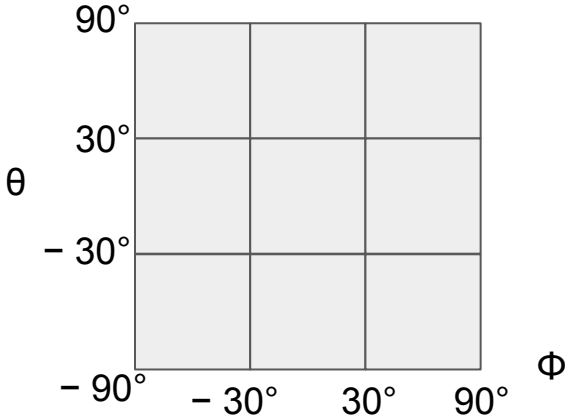
Hint 5

For each region the orientation histogram has to be calculated. This is done in a similar way that is used in the HOG method.



The two angles define a vector in the sphere:

For orientation binning, this sphere is divided into nine parts, where the theta and phi angles are in the [-90°, -30), [-30°, 30°), [30°, 90) regions:



For each vector, find the corresponding bin and add the magnitude value to the bin.

Hint 6

The spatio-temporal feature vector is the linearized version of the 17 × 9 3D orientation gradients. It is wise to perform block-normalization on the individual regions of the cylinder.

In this application, normalization means that the sum of the bins in a 9-bin histogram must be equal to one, thus the sum of the whole feature vector is 17.0

The image sequence is described by the 5 strongest feature points. Hence the descriptor of the sequence is a single row vector containing 5 × 17 × 9 numbers.

Ordering of the features based on spatial data seems to be a smart idea!

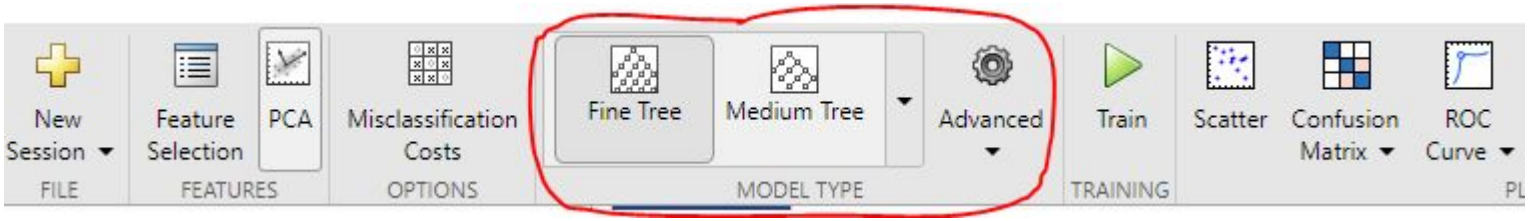
Hint 7

Training a classifier should be done by the **Classification Learner** MATLAB application. This is part of the *Statistics and Machine Learning Toolbox* which you may have to download (using the License you got from PPCU).

Prepare the data for training as described in the support article: https://www.mathworks.com/help/stats/select-data-and-validation-for-classification-problem.html

Hint 8

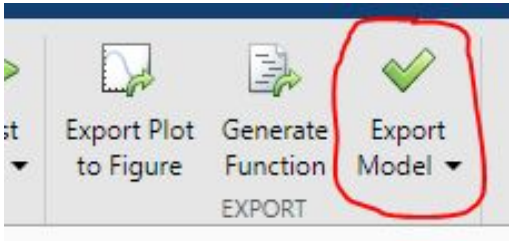
Chose a model type that is best for the task based on your opinion. Tune the model parameters if necessary.



When everything is ready, start the training using the green “play” button.

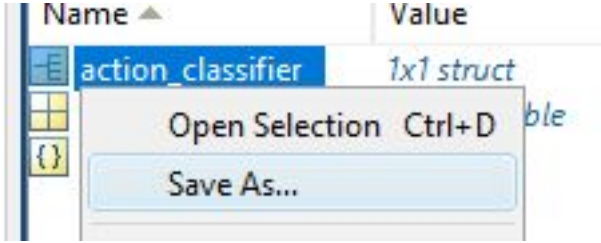
Hint 9

When the training is finished, use the Export Model button to save the trained classifier. (Do not use the Export Compact Model; the training data should be included in the saved object.)



Name the model as `action_classifier`

Save the variable as a .mat file. Right click the variable in the workspace and use the Save As… option in the context menu.



Hint 10

In the second script (`a04_NEPTUN_test.m`) you have to load the test data as well as the trained model.

Use the trained model to predict the actions performed in the test image sequences.

The result of the script has to be a console output showing the predicted label:

```
Test sequence 1: walking
Test sequence 2: clapping
Test sequence 3: walking
Test sequence 4: clapping
Test sequence 5: boxing

fx >>
```

Grading

The final score of this assignment is the sum of the following points:

The uploaded ZIP contains everything (scripts, function, input data, model)	2 points
Training script loads the training data	2 points
Training script calls the feature extraction function for each sequence	1 point
Training script creates good training data (shape, size, labels)	1 point
Feature extraction is fully contained by the appropriate function	1 point
SIFT is used for strong feature detection, 5 strongest features are selected	2 points
Cylinder is constructed according to the description (spatio-temporal position)	1 points
Cylinder regions are constructed as required (17 regions with good limits)	2 points
3D gradient computation neighborhood is correct	2 points
3D gradient vector computation is correct	2 points

Continues on next slide...

Grading

3D gradient orientation vector binning is OK	2 points
Descriptor for a cylinder is correct, normalized	1 point
Classification method selection “makes sense”	1 point
Classifier is trained, accuracy is acceptable	2 point
Trained classifier is saved as a .mat file	1 point
Testing script loads the test data	1 point
Testing script data pre-processing is OK (feature extraction)	2 points
Testing is performed, results are shown in the command window	2 points
Code quality (readability, understandability, good comments and structure)	2 points
TOTAL:	**30 points**

Contact

If you have any further questions regarding this assignment, contact

Márton Bese NASZLADY

via **Teams** (in private chat) or write an email to

naszlady@itk.ppke.hu

References

[1] Schuldt, Laptev and Caputo, *Proc. ICPR'04, Cambridge, UK* (2005) https://www.csc.kth.se/cvap/actions/

[2] Abdulmunem, Ashwan & Lai, Yu-Kun & Sun, Xianfang. (2016). 3D GLOH features for human action recognition.
DOI: 10.1109/ICPR.2016.7899734

[3] Paul Scovanner, Saad Ali, and Mubarak Shah. 2007. A 3-dimensional sift descriptor and its application to action recognition. In Proceedings of the 15th ACM international conference on Multimedia (MM '07). Association for Computing Machinery, New York, NY, USA, 357–360.
DOI: 10.1145/1291233.1291311

THE END