project 3:

binary classification using (tensor) LDA

solution(s) due:

July 20, 2017 at **12:00** via email to **bauckhag@bit.uni-bonn.de**

problem specification:

general remarks

- the file `uiucTrain.tgz` contains positive and negative training data ($m \times n$ images of cars (class Ω_1) and background scenery (class Ω_2))
- set the class labels y_i according to the discussion in lecture 11
- do not forget to normalize the data to zero mean before training

task 3.1: LDA Implement a binary classifier for car detection based on the idea of conventional linear discriminant analysis; proceed according to lecture 12; that is, vectorize the images in the training data to obtain a set $\{(\boldsymbol{x}_i, y_i)\}$, and determine a projector $\boldsymbol{w}$ such that

$$\boldsymbol{w} = \underset{\boldsymbol{w}}{\operatorname{argmax}} \frac{\boldsymbol{w}^T \boldsymbol{S}_B \boldsymbol{w}}{\boldsymbol{w}^T \boldsymbol{S}_W \boldsymbol{w}}$$

Once $\boldsymbol{w}$ is available, do the following:

1. visualize $\boldsymbol{w}$, i.e. turn $\boldsymbol{w}$ into an $m \times n$ matrix and plot it
2. create $k = 1, \ldots, 10$ different classifiers

$$y(\boldsymbol{x}) = \begin{cases} +1, & \text{if } \boldsymbol{w}^T \boldsymbol{x} \geq \theta_k \\ -1, & \text{otherwise} \end{cases}$$

   using 10 different choices of θ_k where $\theta_k \in [\mu_1, \mu_2]$

3. evaluate your classifiers *on the training data*[1], i.e. determine precision and recall for each classifier and then create a precision/recall diagram
4. apply your best performing classifier on the test images in the file `uiucTest.tgz`

[1] **Note:** We only do this for simplicity! In real life, we never evaluate on the training data!

task 3.2: tensor LDA Implement a binary classifier for car detection based on the idea of separable tensor LDA; proceed according to lecture 13; that is, given a labeled set of image patches $\{(\boldsymbol{X}_i, y_i)\}$, determine a projector $\boldsymbol{W}$ such that a new image patch $\boldsymbol{X}$ can be classified using

$$y(\boldsymbol{X}) = \begin{cases} +1, & \text{if } \langle \boldsymbol{X}, \boldsymbol{W} \rangle \geq \theta \\ -1, & \text{otherwise} \end{cases}$$

where

$$\boldsymbol{W} = \sum_{r=1}^{\rho} \boldsymbol{u}_r \boldsymbol{v}_r^T$$

Do this for $\rho \in \{1, 3, 9\}$ and visualize the resulting projectors. Moreover, for the case where $\rho = 9$, do the following:

1. create $k = 1, \ldots, 10$ different classifiers using 10 different choices of θ_k where $\theta_k \in [\mu_1, \mu_2]$
2. evaluate your classifiers *on the training data*, i.e. determine precision and recall for each classifier and then create a precision/recall diagram
3. apply your best performing classifier on the test images in the file `uiucTest.tgz`

Note: you may find further details in the two following papers

- C. Bauckhage and T. Käster, *Benefits of Separable, Multilinear Discriminant Classification*, Proc. ICPR, 2006.
- C. Bauckhage, *Robust Tensor Classifiers for Color Object Recognition*, Proc. ICIAR, 2007.

which are available at

- www.researchgate.net/publication/220927873_Benefits_of_Separable_Multilinear_Discriminant_Classification
- www.researchgate.net/publication/221472251_Robust_Tensor_Classifiers_for_Color_Object_Recognition

general hints and remarks

- IF YOU DO HAVE QUESTIONS, DO NOT HESITATE TO ASK THEM DURING THE LECTURE!
- Send all your solutions (code, resulting images, slides) in a ZIP archive to bauckhag@bit.uni-bonn.de

  note: if you are implementing in C/C++, you may use whatever libraries appear useful. It is, however, suggested you use python and the Image and scipy modules for all your practical work. If you insist on using a language other than python, you have to figure out elementary image processing functions/toolboxes in these languages by yourself. **Implementations in MATLAB will be rejected.**
- Remember that you have to successfully complete all three practical projects (and the tasks therein) to be eligible to the written exam at the end of the semester. Your grades (and credits) for this course will be decided based on the exam only, but –once again– you have to succeed in the projects to get there.
- Not handing in a solution implies failing the course.
- Your project work needs to be *satisfactory* to count as a success. Your code and results will be checked and your presentation needs to be convincing.
- If your solutions meets the above requirements and you can demonstrate that they work in practice, it is a *satisfactory* solution.
- A *good* to *very good* solution requires additional efforts especially w.r.t. to elegance and readability of your code. If your code is neither commented nor well structured, your solution is not good! A very good solution requires additional efforts towards the quality of your project presentation in the colloquium. Your presentation should be well timed, consistent, and convincing. Striving for very good solutions should always be your goal!