Prof. Dr. Thomas Schultz
Annika Mikliss (mikliss@cs.uni-bonn.de)

University of Bonn
B-IT / Institute of CS II
November 24, 2025

Winter term 2025/26

Image Acquisition and Analysis in Neuroscience

Assignment Sheet 4

Solution has to be uploaded by December 8, 2025, 10:00 a.m., via eCampus

If you have questions concerning the exercises, please use the forum on eCampus.

- Please work on this exercise in **small groups** of 3 students. Submit each solution only once, but clearly indicate who contributed to it by forming a team in eCampus. Remember that all team members have to be able to explain all answers.
- Please submit your answers in PDF format, and your scripts as *.py/*.ipynb files. If you are using Jupyter notebook, please also export your scripts and results as PDF.

Exercise 1 (Stepsize in Levenberg-Marquardt, *7 Points*)

Chapter 4, Slide 90, proposed to adapt the stepsize of gradient descent based on the second derivatives of the cost function. Observing that a high second derivative indicates rapid local changes in the gradient can provide an intuition for why it makes sense to take more careful steps in such cases. More formally, the slide claimed that this stepsize adaptation makes the progress of the optimization independent from scaling of the independent variable, e.g., whether displacements are expressed in mm or cm. This exercise invites you to convince yourself that this is indeed true.

a) For simplicity, consider minimization of a univariate scalar function $f(x)$. Assuming a fixed stepsize λ^{-1}, how will a single step of gradient descent change the current value x_k? Use a first-order Taylor expansion of $f(x)$ to approximate the corresponding decrease in cost function value. (2P)

b) How could an implementation of gradient descent use the result from part a) to adjust the stepsize? (1P)

c) If we re-scale the independent variable, i.e., consider a cost function $\tilde{f}(x) := f(\alpha x)$ with $\alpha > 0$, how does the resulting derivative $\tilde{f}'$ relate to the derivative f' of the original cost? Considering your result from part a), how will this affect the expected decrease in cost function value if we keep the same stepsize λ^{-1}? (2P)

d) Argue why setting the stepsize to $\lambda^{-1} = \left(\tilde{\lambda} f''\right)^{-1}$ makes the expected progress independent of the scaling factor α. (1P)

e) Instead of using actual second derivatives, the Levenberg-Marquardt algorithm extends Gauss-Newton: Assuming a least-squares cost $f(x) = \mathbf{h}(x)^T \, \mathbf{h}(x)$, it uses the Jacobian $J_\mathbf{h}$ to approximate the Hessian. Based on the simplified case of a scalar function $h(x)$, argue why the scale invariance also carries over to this setting, i.e., if we set $\lambda^{-1} = \left(\tilde{\lambda} \cdot [h'(x)]^2\right)^{-1}$. (1P)

Exercise 2 (GMMs and EM Algorithm for Image Segmentation, *9 Points*)

In this exercise, you will implement a Gaussian Mixture Model (GMM) to produce a probabilistic image segmentation, and find suitable parameters automatically by using the EM algorithm.

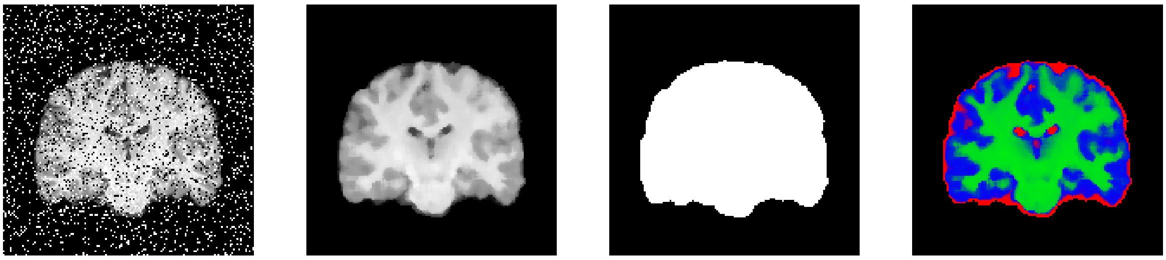

Figure 1: From left to right: The input image, after denoising, the binary mask, and the segmented brain image.

a) Read the grayscale image `brain-noisy.png` and the brain mask `mask.png`, which are provided along with this sheet on eCampus. Reduce the salt and pepper noise in the image using a median filter. (1P)

   Hint: You do not have to implement the median filter yourself, you may use a suitable Python package.

b) Plot a log-scaled histogram of the pixels within the brain mask. It should show how frequently different intensity values occur in the image. Based on this histogram, what are approximate intensity ranges for cerebrospinal fluid, gray matter, or white matter? What are their approximate mixing weights? (2P)

c) Now, we will use a three-compartment Gaussian Mixture Model (GMM) for image segmentation: Based on the histogram from b), find a plausible initialization of the GMM parameters. Use them to compute the responsibilities ρ_{ik} of cluster k for pixel i. (2P)

d) Visualize the responsibilities by mapping the probabilities of belonging to the CSF, gray matter, and white matter clusters to the red, blue, and green color channels, respectively. Please submit the resulting image. (1P)

e) Use the update rules provided in the lecture to re-compute the parameters μ_k, σ_k, and π_k. (1P)

f) Iterate the E and M steps of the algorithm until convergence. Please submit the final parameter values, a visualization of the final responsibilities, and your code. Remember to constrain your algorithm to the brain mask. (2P)

Exercise 3 (Markov Random Fields, *9 Points*)

In this exercise, you will extend your GMM-based segmentation to account for spatial structure via a Markov Random Field.

a) Based on your implementation of the EM algorithm from Exercise 3, but leaving out the median filtering, create a discrete (hard / non-probabilistic) label image that contains the most likely material for each pixel. Output it as an RGB image. (1P)

b) Implement one iteration of the Iterated Conditional Modes (ICM) algorithm for a Markov Random Field that uses the Potts model and $\beta = 0.5$. Use your EM parameters to initialize the unary potentials and use the labels of the neighbouring pixels to compute the pairwise potentials. Finally, for each pixel pick the label that minimizes the energy. Output the result as an RGB image. (3P)

c) Apply your ICM iteration five times overall. Output the number of pixels whose label changes in each iteration, and output the final labels as an RGB image. (2P)

d) Integrate your implementation of the ICM into the EM algorithm and run it until convergence. Output the final result as an RGB image. (2P)

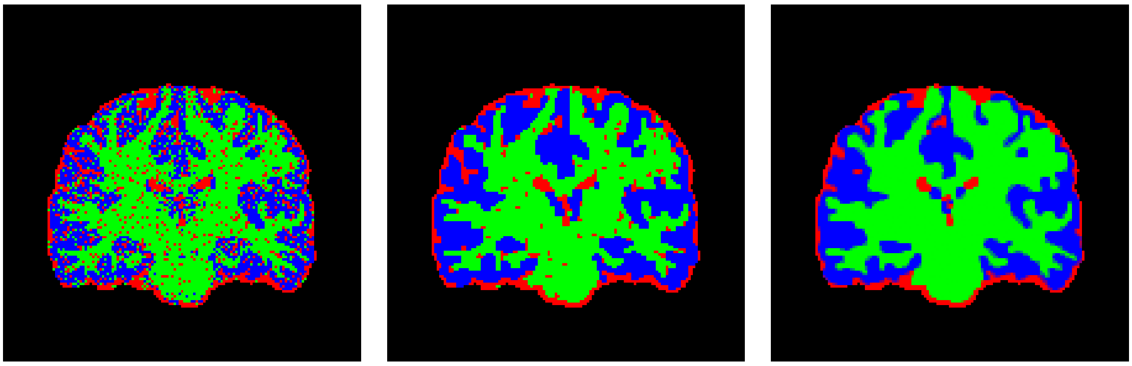

Figure 2: A hard labeling of the noisy image based on the algorithm from Exercise 2 (left), the result after repeated application of Iterated Conditional Modes with higher values of β (center), and the final probabilistic segmentation result with the MRF term (right).

e) Increase the β parameter and repeat task d). At which value of β is the final segmentation almost noise-free similar to Fig. 2 (right)? Output the final result. (1P)

Good Luck!