Project work: A mini segmentation challenge

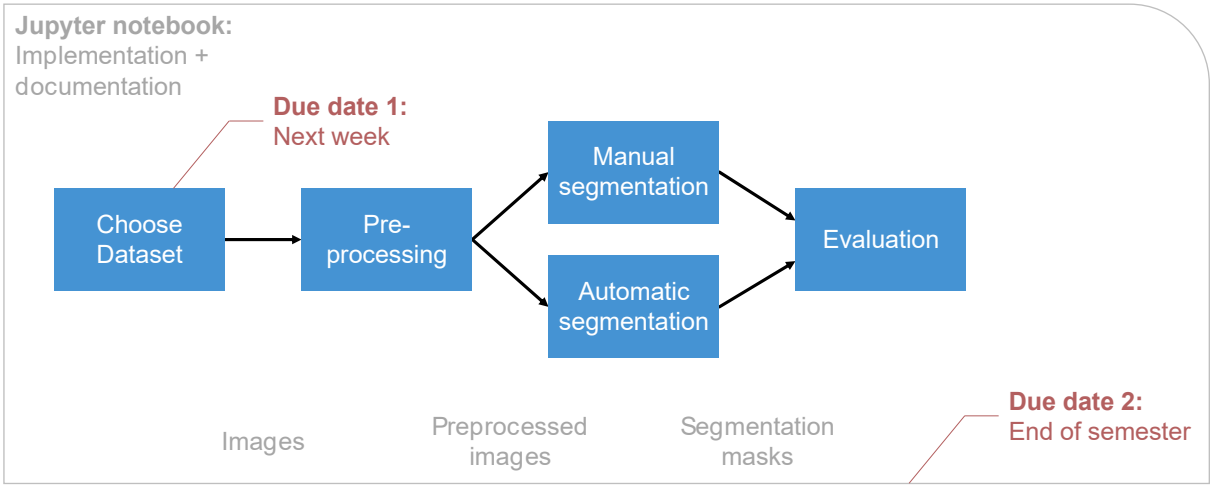
Instructor: Norman Juchler (juch@zhaw.ch)
Module coordinator: Andreas Hock (hocn@zhaw.ch)

The goal of this project is to perform segmentation on an image dataset using two different approaches. First, you will manually label a structure of interest using a suitable tool such as Fiji, ITK-Snap, or QuPath. Second, you will apply a method of your choice to segment the structure automatically. During the course, we will discuss basic ways to achieve these different steps.

Your solution should work reliably for your small dataset. However, try find a solution that, in principle, scales to larger dataset as well. Specifically, avoid manual steps that would not be practical for large-scale data.

Note: This course does not cover machine learning methods in depth. However, if you are already familiar with such methods, you are welcome to use machine learning models (pre-trained or trained from scratch).

This document contains detailed instructions outlining what is expected from you in the project.



Overview of the project work.

Step 1: Choose your dataset:

Search for a publicly available image dataset from an imaging modality of your choice. There is a wealth of open data available – use your favorite search engine, or explore platforms like Kaggle or Google dataset search. You may also refer to this article on public datasets.

For example, you might choose brain MRI, chest X-ray, or light microscopy images of histopathological slides. Try to work with structures of interest that are clearly discernible in the images and relatively easy to segment. We recommend working with 2D images, but if you are accustomed with 3D data, feel free to use it. While the focus of this course is primarily on medical imaging, you are also welcome to choose data from other areas of the life sciences – for instance, aerial imagery of forests or satellite images.

Requirements:

- Dataset of at least 10 image samples.
- The total size of the dataset should not exceed 200 MB.
  If it is larger, provide instructions on how to obtain the data.
- Make sure you have the right to use and share the data (check the usage license!)
- The structures of interest should be clearly visible and suitable for segmentation.
- Avoid datasets with excessive variability that may complicate segmentation.
- Each student or group must work with a different dataset.

Output:

- Image samples in a common machine-readable format
  (e.g., JPG, PNG, TIFF, DICOM, NIfTI, etc.)
- A brief description of the dataset (What does it depict? How was it acquired?)
- A reference to the usage rights or license of the dataset.

Due date 1: 22.04.2025

Step 2: Preprocessing

It may be necessary to preprocess the image before further analyzing the data. Common preprocessing steps include adjusting image contrast to enhance the visibility of relevant structures or applying filters to reduce noise and highlight important features. The specific approach will depend on your data and the goals of your project.

Requirements:

- Choose preprocessing methods that are scalable and suitable for larger datasets.
- Improve the images in a way that facilitates subsequent processing steps.

Output:

- Preprocessed image samples or a script/macro that performs the preprocessing.
- A description explaining what was done, how it was done, and why it was necessary.

Step 3: Manual segmentation:

During the course, you will learn how to use tools like Fiji, ITK-Snap or QuPath for manual segmentation. In this step, use one of these tools to manually segment the structure of interest in your images. If you are using an AI-based tool that performs segmentation automatically but allows manual refinement, you may use it – provided that the results are verified and corrected by you.

The segmentation masks created in this step will serve as the ground truth for evaluating your automated segmentation method in the next step.

Goals / Requirements:

- Create at least 10 *binary segmentation masks*. (If your annotations are stored as 2D polygon coordinates, convert them into binary masks.)
- The segmentation masks may include multiple structures of interest (e.g., if the dataset contains multiple cells, bones, trees, etc.)
- If your dataset includes existing segmentation masks, you may use them for later steps. However, you still need to demonstrate how to create such masks manually using an appropriate tool.

Output:

- At least 10 binary segmentation masks saved as image files (e.g., PNG, TIFF, …).

Step 4: Segmentation in Python:

In this step, you will implement an automatic method to segment the structures of interest using Python. You are free to choose any suitable approach, such as thresholding, region growing, clustering, edge detection, or more advanced methods. You may also use a pre-trained model if you are familiar with how to apply it.

Requirements:

- The segmentation must be implemented in Python.
- You may use external libraries or tools (e.g., OpenCV, scikit-image, PyTorch, etc.).
- Your method should take an image as input and return a binary segmentation mask of the structure(s) of interest.

Output:

- Segmentation masks in the same format as used in the manual segmentation step.

Step 5: Evaluation:

In the final step, you will evaluate how well your automated segmentations match the manual (ground truth) segmentations. A commonly used metric for this purpose is the Dice similarity coefficient, but you may also use other suitable metrics. Briefly explain the metric you choose and implement a function to compute it.

Requirements

- Select an evaluation method that compares two binary segmentation masks and returns a numerical score indicating their similarity (e.g., Dice score).
- Compute the mean and standard deviation of the evaluation scores across your dataset.
- Discuss your results: How well does your method perform? Where does it succeed or fail? What might be improved?

Documentation

Prepare a Jupyter Notebook using the provided template that includes the following elements:

- A description of the **dataset** you used, including the imaging modality, the structure of interest, and the number of images. Provide context about how and why the data was acquired. Help the reader understand the content and relevance of your dataset.
- The description and implementation of the **preprocessing**. For each step, explain what was done and why it was necessary. Include code and visual examples to illustrate the improvements made.
- A description of the **manual segmentation process**. Mention which tool you used, how much time the process took, and what difficulties you encountered. Specify the structure of interest and mention the challenges involved in segmenting it manually.
- A description and the implementation of the automatic **segmentation in Python**. Specify the method you used. Highlight any difficulties or limitations encountered in the process.
- An **evaluation** of the automatic segmentation, including the mathematical formulation of the evaluation metric you used. Present your results (e.g., average performance across the dataset), and visualize them.
- A **discussion** of the results, including the strengths and weaknesses of both manual and automatic segmentation approaches. What challenges did you face? How could the automated method be improved in future work?

Deliverables

You need to turn in the following materials in a single folder:

- Your Jupyter notebook (.ipynb)
- A PDF or HTML export of the Jupyter notebook with all outputs included
- The image data you used (both raw and preprocessed versions)
- The segmentation masks (both manual and automated)
- A requirements.txt file listing all non-standard Python packages used. Note: Only well-maintained and reputable packages will be accepted.
- A summary slide (PowerPoint or similar) that provides an overview of your project.
- A brief video (maximum 3 minutes) summarizing your project. A simple screencast of you presenting the summary slide is sufficient, but feel free to be creative ☺

All materials must be bundled into a single ZIP file. A link for uploading your submission will be provided in due time.

Due date 2: 31.05.2025

Use of generative AI (genAI)

- You are allowed to use ChatGPT, GitHub Copilot, Google Gemini, etc. when preparing the final report, under the condition that you declare where you have used it at the end of the Jupyter notebook.
- genAI can be particularly useful for developing code. We encourage this as long as it supports your learning. In contrast, we discourage any use of genAI that does NOT support your learning. In particular, indiscriminate copying of text and code that you do not semantically understand in order to fulfill the course requirements is NOT allowed. We expect you to be able to explain in your own words what your code does or what your texts mean. We may verify this point individually.

Assessment

The following aspects will be graded by the tutors. See the corresponding grading scheme!

- Content / Implementation (30%)
- Report / Documentation (20%)
- Presentation / Video (30%)
- Creativity / Overall impression (20%)

Try to approach the task with creativity and curiosity. Keep in mind that not all segmentation problems have the same level of difficulty – we will take this into consideration when evaluating your project.