Exercises

Machine Learning

Prof. Dr. Peter Ochs

www.mop.uni-saarland.de/teaching/ML25

— Summer Term 2025 —

Data Augmentation

— Challenge —

1 Introduction

In the following, we describe the practical project that we have prepared for the Machine Learning (ML) core lecture of Summer 2025. The idea of this project is that the students put into practice the theoretical concepts introduced during the lectures. This year the project consists on an image segmentation task.

1.1 Participation in the project

The participation to the project is fully voluntary but highly recommended, as it provides an opportunity for students to gain hands-on experience in ML. Students are expected to work in groups of up to 3 members and must register their group via CMS. Participating in the project means submitting both the report and the predictions for the challenge (see below). It is not possible to participate in the challenge but not submitting the report, or the other way around. Refer to Section 6 for the expected timeline.

2 Problem Description

In this challenge, you will work on the task of **binary image segmentation**, where the goal is to separate the foreground (object of interest) from the background in natural images. Unlike fully annotated segmentation masks, you will be provided with **sparse scribble annotations** (scribbles) indicating only rough regions belonging to the foreground and background classes.
Your task is to develop a model that can leverage these partial scribbles to produce accurate, full-resolution binary segmentation masks.

2.1 Mathematical Model of the Task

Since, we do not assume any prior knowledge about image processing, we explain here a mathematical model of the setting[1]. We are given a triplets $(\boldsymbol{b}, \boldsymbol{c}, \boldsymbol{g})$ with the following meaning:

- $\boldsymbol{b} \in \mathbb{R}^{n_x \times n_y \times 3}$ is a RGB-valued natural image, which is understood as a grid with n_x columns, n_y rows, and 3 color channels (3 real values at each gird point). The grid points are called 'pixels', i.e., the image has $n := n_x \cdot n_y$ pixels, each of which is associated with a 3-dimensional vector of RGB values: $\boldsymbol{b}_{ij} \in \mathbb{R}^3$ for each pixel (i, j), where $0 \leq i < n_x$ and $0 \leq j < n_y$.

[1] In the implementation/code, the objects may have different names and may be represented in a different way.

- $\boldsymbol{c} \in \{0, 1, \emptyset\}^{n_x \times n_y}$ contains the scribbles, where for each pixel (i, j) it holds that

$$\boldsymbol{c}_{ij} = \begin{cases} 0\,, & \text{if pixel } (i,j) \text{ is annotated as background pixel}\,; \\ 1\,, & \text{if pixel } (i,j) \text{ is annotated as foreground pixel}\,; \\ \emptyset\,, & \text{if pixel } (i,j) \text{ is not annotated}\,. \end{cases}$$

- $\boldsymbol{g} \in \{0, 1\}^{n_x \times n_y}$ is the ground-truth (binary) segmentation of the image:

$$\boldsymbol{g}_{ij} = \begin{cases} 0\,, & \text{if pixel } (i,j) \text{ belongs to background}\,; \\ 1\,, & \text{if pixel } (i,j) \text{ belongs to foreground}\,. \end{cases}$$

The goal is a classifier that uses the image $\boldsymbol{b}$ and the sparse scribble annotations $\boldsymbol{c}$ to produce a binary segmentation (output in $\{0, 1\}^{n_x \times n_y}$) that matches the ground truth segmentation $\boldsymbol{g}$.

2.2 Python Script Example

We have uploaded a Python script in CMS that contains a description of the data and an example on how to load, preprocess, and visualize the data. Furthermore, we provide an initial simple segmentation model.
The script also includes instructions on how to format and save your predictions for submission to CMS (both for the leaderboard and the final ranking). We highly recommend that you use this script as a starting point for your solution.

2.3 Data and Model Usage Rules

- Data Restrictions: You are only allowed to use the training data provided by us for this project.
- Model Usage: You are free to choose any model architecture for your image segmentation task. However, you must train the model from scratch as part of your project. The use of pretrained models or weights is not allowed. Your training process should be clearly documented and reproducible.

3 Report Instructions

Every student group/team participating in the project should submit a report (.pdf file) using the LaTeX template provided in CMS. The report must be at most 6 pages long (references excluded) and contain detailed information on the methodology applied to select the final model and make the necessary predictions to participate in the challenge. More specifically, the report should contain information on the following aspects:

1. Data analysis and preprocessing: The report should describe any considered approach used for data analysis and preprocessing to prepare the input data (features) to the ML model.
2. ML modeling: The report should include a short description of the different models applied to the data, specifying the used python libraries (if any). Stating only the used libraries is not enough.
3. Model selection: The report should detail the methodology followed to compare the different ML models (and, if applicable, data preprocessing approaches), as well as to select the final model used to make the predictions for the challenge.
4. Empirical results: The report should provide a summary and description of the empirical results that have led the students to select the final model for the challenge.

5. Others: The report may contain any additional analysis performed by the students that may be interesting from a practitioner point of view. Examples of such analysis may i) provide a thorough data analysis (for example, data visualization using unsupervised learning techniques); or account for the robustness explainability or fairness considerations of the different models explored by the students.

3.1 Report Grading

The report will be graded. There are four possible grades for the project report:

- **[0 (out of 10) points]** If a major methodological mistake (e.g., selecting the ML model on the data used to train it) is detected.
- **[5 points]** If a subset of the models introduced in the lectures and tutorials are correctly applied, evaluated and reported.
- **[7.5 points]** If a comprehensive application of the techniques covered in the lectures are correctly applied, evaluated and reported.
- **[10 points]** If the students go one step beyond the course material. They may, e.g., provide additional content in the report covering data analysis or robustness/fairness aspects (see point 5. above) and/or apply methodology that goes beyond what has been introduced in the lectures to train excellent models.

4 Challenge

The ML project will be maintained in a challenge format similar to a Kaggle-like competition[2]. This means you will see your performance and ranking in a leaderboard which will be updated at multiple time stamps during the semester. To begin with, we plan to update the leaderboard about once every week, and towards the end of the semester (i.e. after the main exam until the project deadline) we will update the leaderboard about once every 2 days. After each update of the leaderboard you will be able to see your model's performance and your ranking in the whole competition. In the following part, we will deliver the details about the challenge.

There are 3 data-sets: (Training, Test1 and Test2).

- Training: This is a collection of triplets (image, scribble, ground truth), which is used to train and evaluate your models.
- Test1: This is a collection of tuples (image, scribble), for which your model's predictions (binary classifications for each image) are submitted (via CMS) to the leaderboard as often as you want.
- Test2: Another collection of tuples (image, scribble), which is released two weeks before the deadline. In these two weeks, you apply your model exactly once and upload the predictions exactly once (via CMS).

Throughout the semester, you are encouraged to work with your training data, to update your models, try different approaches, or perform better model selection and hyper-parameter tuning. Your participation in the challenge will be evaluated in two stages:

[2] https://www.kaggle.com/

1. Leaderboard: For the first part of the challenge, we have a leaderboard in CMS which will be updated multiple times during the semester. Throughout the semester, you can submit your predictions on the test set (Test1) in CMS. In this leaderboard, you will be able to see the performance evaluation of your team's model and its ranking among the models of other teams. The idea of the leaderboard is twofold: i) give you a realistic estimate of the team raking for each of the tasks to incentivize healthy competition; and ii) get you familiar with the challenge evaluation process.

2. Final Assessment: For the second and final evaluation of the challenge, there will be another test set (Test2) to which only exactly one submission is admitted. You need to submit your predictions again through CMS. Your participation in the challenge will be assessed by your model's performance in this second test set alone, independent of the performance on the data from Test1.

4.1 Performance Measure:

To evaluate the performance of your binary image segmentation model, we use a metric called **Intersection over Union (IoU)**. Intersection over Union measures how well the predicted segmentation aligns with the ground truth. It is defined as:

$$\text{IoU} = \frac{|\text{Prediction} \cap \text{Ground Truth}|}{|\text{Prediction} \cup \text{Ground Truth}|}$$

Where:

- The numerator is the number of pixels where both the prediction and the ground truth label the pixel as belonging to the object.
- The denominator is the number of pixels labeled as object in the prediction or the ground truth (or both).

In this challenge, segmentation is binary:

- **Class 0**: Background
- **Class 1**: Object (of interest)

We compute IoU for both classes, and then report the **mean IoU (mIoU)**:

$$\text{mIoU} = \frac{1}{2}\left(\text{IoU}_{\text{background}} + \text{IoU}_{\text{object}}\right)$$

Your goal is to maximize the mean IoU across the test set. A perfect segmentation would yield an mIoU of 1.0.

Bonus points:
The top 5% of the teams, ranked according to the above metrics evaluated on the dataset Test2, will get a bonus (one extra point in the German grading system, i.e. "grade minus 0.3") on their final grade.

4.2 Submission

To submit your predictions for both the leaderboard and final submission, put your .png files containing the predictions for the segmentation task in the zip file. Please name files inside the .zip as stated in the Python scripts.

5 Effect on Overall Course Grade

The project is fully voluntary but it can help your grade, and did so for many students of previous iterations. Thus, it is highly recommended to do the project. To pass the course, you will have to pass either the main or re-exam. If you do not submit a project report, then your exam grade (i.e. the better grade of main exam and re-exam) will be your final grade. If your team submits the report, your overall course grade will be the better of the following two grades: your exam grade alone, or a 75% exam + 25% report mixture grade. To put in in math terms:

$$\text{Course grade} = \max(\text{Exam grade}; 0.75 \cdot \text{Exam grade} + 0.25 \cdot \text{project report grade})$$

All members of a team get an identical report grade. Note again that to participate in the project (and get a potential grade boost), your team must submit both the report and the predictions for the challenge.

Bonus points: As noted at the end of Section 4.1, the best-performing teams in the challenge will receive a bonus that improves their final grade by 0.3 points in the German grading system (e.g., a 2.7 becomes a 2.3, a 2.0 becomes a 1.7, and a 1.3 becomes a 1.0). An example, taken from the previous iteration: if you have a grade of 2.0 in the exam, and a good report (10 out of 10 points), and you also do well enough in the challenge to get the bonus, your final course grade would be a 1.3.

6 Timeline

In the following, we detail the key dates that must not be missed if interested in joining the ML project:

20.06.2025 Team registration: Teams of up to 3 students should be registered in CMS by then.

31.08.2025 Final project submission via CMS. The students will need to submit both their report and final predictions for the dataset Test1 by 23:59. We will not accept any late submissions.

01.09.2025 Release of the dataset Test2.

07.09.2025 Submission of final predictions for the dataset Test2 by 23:59. We will not accept any late submissions.