

Multimodal Machine Learning Lab

Winter Semester 2026/2027

Niklas Deckers and Martin Potthast

Agenda

- ❑ Homework
- ❑ Working Mode
- ❑ Classification Models and Attacks
- ❑ ML Foundations

What was the point of the initial homework?

1. Checking what you know
2. Teaching you the basics (esp. important at the start of the semester: bringing everyone to the same page)
3. Having you think about some concepts in depth
4. Seeing how you work

Generating the homework (in extreme cases even without looking at the produced results) makes 1-3 impossible, and will not be possible in later course assignments

GenAI Guidelines

- ❑ Framing work not done by yourself as your own work is (a form of) plagiarism
- ❑ We will have to agree on the way what kind of AI use is acceptable for this course
- ❑ My goal is to teach you efficient yet effective ways (using specific GenAI methods) that still keep you on top of the research
- ❑ This only works when you properly work out the basics
- ❑ Rule of thumb for later in the course: Do not pull any knowledge out of the GenAI model weights and sell it as innovation.
- ❑ From now on, every exercise requires a GenAI disclosure, including: The name of the used model, an abstract description of the steps performed, the used prompts, precise locations of what was edited, precisely what kind of knowledge coming from the GenAI model weights was used
- ❑ Side effect: This will also help you determine what part of your work is innovative (*contributions*) and what is just technical background work (very important step in science and paper writing)

Discussion: Where does the information come from when solving today's homework using AI?

Subtasks of today's homework:

- ❑ How can an external model be used? (API, functionality/capabilities)
- ❑ Coding: Manual/routine tasks (downloading weights, setting up GPU); vs. creative part
- ❑ Pondering about a problem: Real-life inspirations, real-world data and applications; human experience: What do I want, what worked in the past? What is interesting?
- ❑ Understanding the applicability of an approach in a certain domain
- ❑ Understanding foundational/underlying mechanics of the model
- ❑ Language: Grammar issues, spelling, formulation, syntax (code); problems such as wrong variables
- ❑ Coming up with ideas; sampling from an idea space

Systematizing CLIP Application Ideas

- ❑ (see overview table PDF)
- ❑ Shows the most important algorithmic principles
- ❑ We will start with working on classification models

Group Assignment

- ❑ I assigned everyone who submitted the homework to a group (size 3)
- ❑ Based on the homework submission, I tried to assess the way each student prefers to do their work
- ❑ I arranged the groups with the hope of joining students with compatible work styles
- ❑ Swaps possible (if everyone agrees); self-service in Moodle
- ❑ Weekly meetings with me by arrangement (15-20 minutes, ideally on Mondays)

ML Basics

- ❑ Training loop structure (slide ML:I-41 in Introduction)
- ❑ Classification accuracy (slide ML:II-155 in Machine Learning Basics)
- ❑ Losses (slide ML:II-77 in Machine Learning Basics)
- ❑ Thresholds (slide ML:II-96 in Machine Learning Basics)

Attack Approaches

- ❑ Attacks could work via manipulating (1) inference input, (2) model weights, or (3) training data
- ❑ We will focus on manipulating the inference input for now

New Homework: Image Classification

- ❑ Classifiers relying on CLIP could work in two different ways:
 - *Zero-shot* by comparing images to a curated selection of text prompts (via the shared embedding space)
 - *Linear probe*: Using multiple examples to train a linear layer (or a more complex network) on top of the CLIP embedding output (we currently don't want to finetune the CLIP weights)
- ❑ Your task is to develop an image classifier (1) for an arbitrary set of text classes and (independently from that) (2) for the fixed classes *anger*, *disgust*, *fear*, *happiness*, *sadness*, *surprise*, *neutral*. The latter should be trained using a linear probe. You will have to gather your own dataset. Set a reasonable, limited scope of images you would like to input (images of people, faces, stock images, abstract art, ...). This also means that you cannot know what the other groups will optimize their model on.
- ❑ The classifier should be served from a Python script via a REST API with a given specification (see course website)
- ❑ Also build an HTML frontend for testing the classifier with custom uploads and included demo examples (this allows for more GenAI use since learning how to do this is not the main scope of this course)

Attacks on the Classifiers

- ❑ Starting after the group discussion on Monday (but please think about how to do it before), implement three attack methods against the API:
 - *Dictionary attack* by letting an algorithm choose an optimal image from a given set of images to obtain an optimal (winning) score for a given class (have a look into FAISS for working on large datasets)
 - *Pixel-wise optimization*, which manipulates a given image (within reasonable margin) to obtain an optimal (winning) score for a given class
 - *Generative model attack*, by generating an image using, e.g., Stable Diffusion, to obtain an optimal (winning) score for a given class
- ❑ The given class should be configurable: Free text or one of the emotion classes
- ❑ Attacking a classifier that accepts a CLIP embedding of an image instead of an image may form an easy start
- ❑ Think about whether these approaches are *open-box* or *closed-box* (formerly known as *whitebox* and *blackbox*)
- ❑ Be ready to test your approaches on other teams' software

Homework Submissions

- ❑ The homework will be due Tuesday 20:00
- ❑ Upload via Moodle (one per group)
- ❑ Backend as Python file (inside a ZIP, including a documentation of training steps and additional weights that go on top of CLIP), frontend as a standalone HTML file, attacks as ipynbs
- ❑ Please remember to include the GenAI disclosure statements