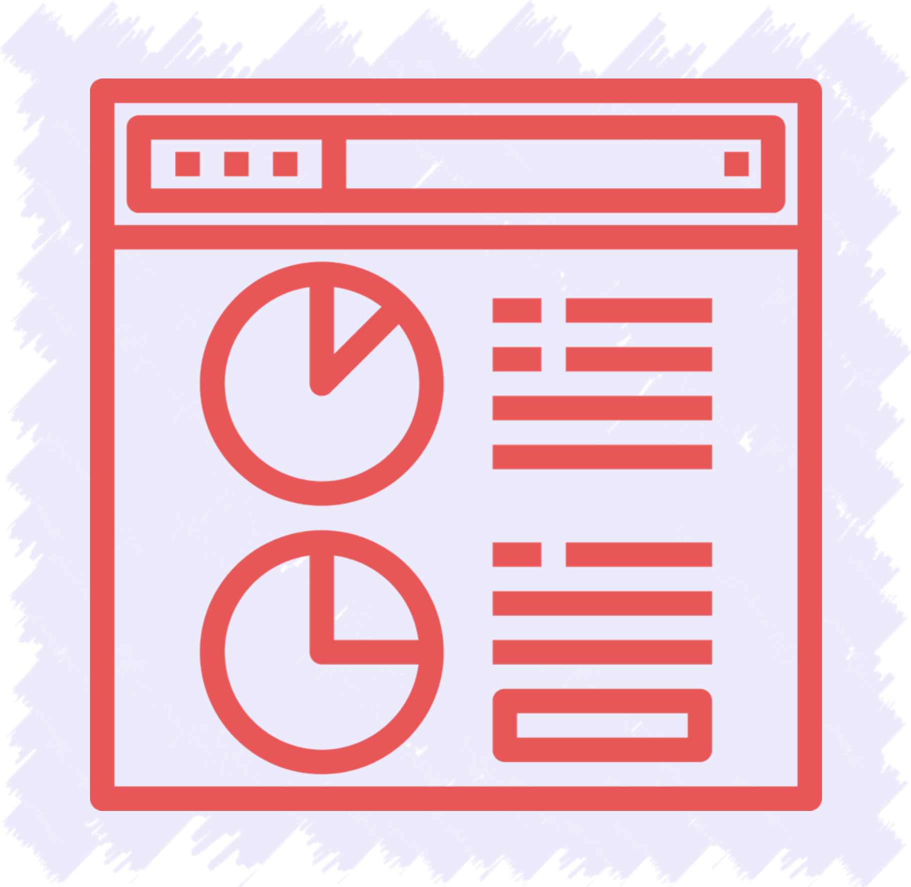
xAI-Proj-B
Bachelor Project
Chair of Explainable Machine Learning (xAI)

Sebastian Doerrich, M.Sc., Prof. Dr. Christian Ledig

xAILab Bamberg

University of Bamberg

Project Format

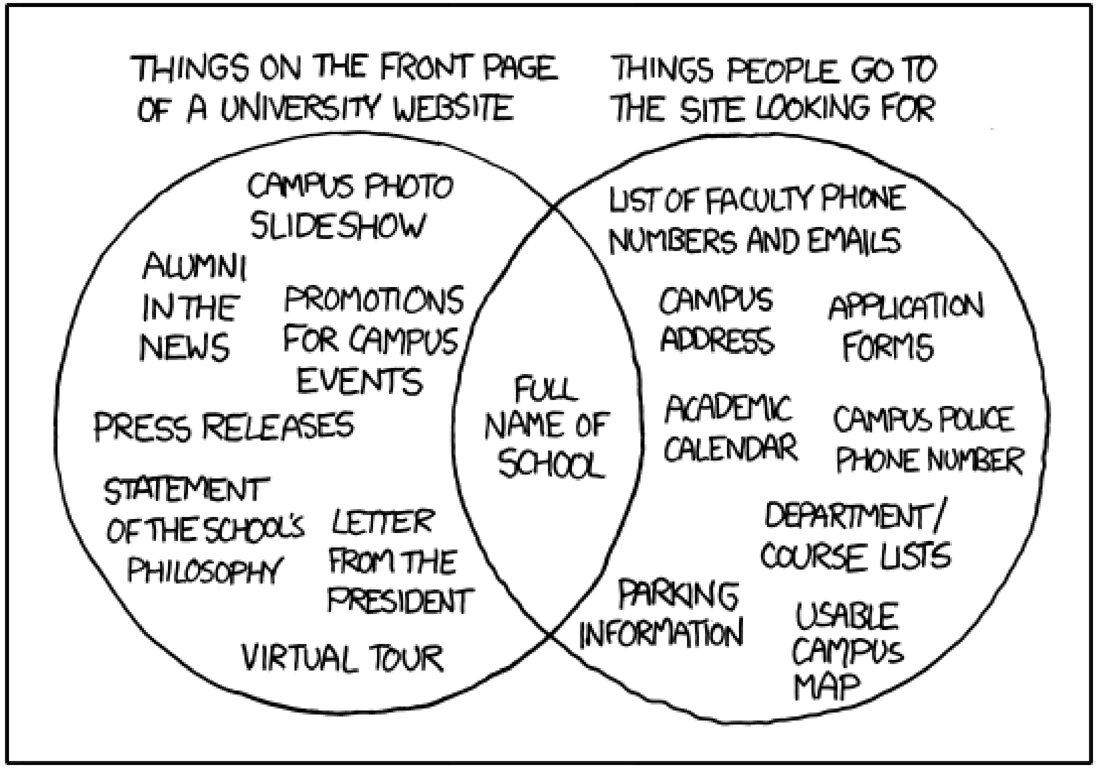


Schedule

21 October	Welcome & Deep Learning Introduction
23 November	Deadline Dataset-Upload 1
16 December	Intermediate Presentations / Pres. + Writ. Advice
21 December	Deadline Dataset-Upload 2
23 December - 06 January	Winter break
20/27 January	Final Presentations (TBD)
08 February	Deadline Final Reports

Time, Location, Contact

- Time: Tuesday, 2 – 6 PM (ct)
- Room: WE5/02.005
- Contact: Sebastian Doerrich (sebastian.doerrich@uni-bamberg.de)



THINGS ON THE FRONT PAGE OF A UNIVERSITY WEBSITE
THINGS PEOPLE GO TO THE SITE LOOKING FOR
CAMPUS PHOTO SLIDESHOW
ALUMNI IN THE NEWS
PROMOTIONS FOR CAMPUS EVENTS
PRESS RELEASES
STATEMENT OF THE SCHOOL'S PHILOSOPHY
LETTER FROM THE PRESIDENT
VIRTUAL TOUR
FULL NAME OF SCHOOL
LIST OF FACULTY PHONE NUMBERS AND EMAILS
CAMPUS ADDRESS
APPLICATION FORMS
ACADEMIC CALENDAR
CAMPUS POLICE PHONE NUMBER
DEPARTMENT/ COURSE LISTS
PARKING INFORMATION
USABLE CAMPUS MAP

Munroe, R. Xkcd. University Website. https://xkcd.com/773/

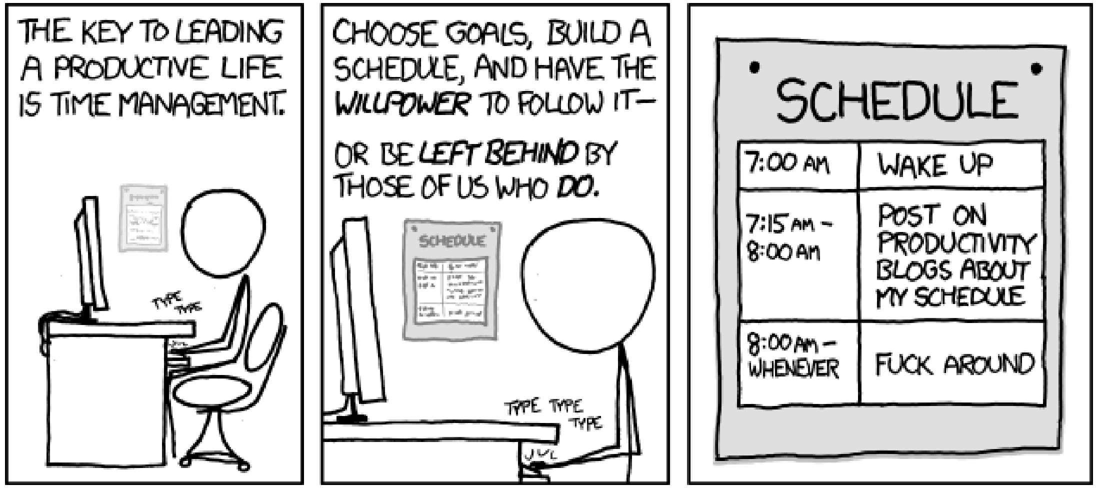
Office Hours

- Opportunity to discuss progress or issues

- Entirely optional

- Generally, every Tuesday from 2 – 6 PM in my office (WE5/04.085) or online

- BUT necessary to schedule an appointment beforehand via E-mail


THE KEY TO LEADING A PRODUCTIVE LIFE IS TIME MANAGEMENT.
TYPE TYPE
CHOOSE GOALS, BUILD A SCHEDULE, AND HAVE THE WILLPOWER TO FOLLOW IT—
OR BE LEFT BEHIND BY THOSE OF US WHO DO.
SCHEDULE
TYPE TYPE TYPE
SCHEDULE
7:00 AM
WAKE UP
7:15 AM - 8:00 AM
POST ON PRODUCTIVITY BLOGS ABOUT MY SCHEDULE
8:00 AM - WHENEVER
FUCK AROUND

Munroe, R. Xkcd. Time Management. https://xkcd.com/874/

Project Description

- *Hands-on experience in deep learning*
- *Train a neural network for an image classification task on ~~CIFAR-10~~ an ImageNet-1k Subset (provided by us in VC)*
- *Explore different architectures, training methodologies and optimization techniques*
- *Assess model performance using evaluation metrics*
- *Evaluate the impact of different data augmentation methods*
- *Acquire and design an entirely new dataset to evaluate model robustness*
- *Present your results*

Learning Objectives

- Agile working in a team environment
  - Team work -> fixed role allocation(?)
  - Organization & coordination on your own
  - Agile development (e.g. Scrum for prototyping)
  - Design thinking
- Writing a report that is scientifically correct and educational
  - Scientific writing in the English language
- Scientific presentations
  - Improve the structure of presentations and your oral presentation skills

Teams

- All teams consist of three students
- In the beginning you can assign roles as following:
  - 1x Git Master
    - Responsible for git tree, branches & merging
    - Remind other members to commit frequently and help with issues
  - 1x Scrum Master
    - Organize regular meetings
    - Update trello board / gitlab issue board / etc. to monitor progress
  - 1x Communication Master
    - Take notes during meetings
    - Organize documentation and keep it clean
    - Responsible for communication with supervisor if required
- But basically, how you work together is up to you!

Teams

- All team members have to contribute to coding, presenting & writing

- Use some platform to keep each other updated such as slack or discord

- Use the version control software **Git** and if you like a project management tool **GitLab** or **Trello** to keep track of your code changes, documentation, etc.

Deliverables

Grading

- Attendance of all in-person meetings and presentations is mandatory
- All team members have to contribute to coding, presenting & writing
- Grades
  - 50% based on the final presentation (25 min + 5 - 10 min questions)
  - 50% based on the report (9 pages without references)
- The presentation and the report have to be in English
- Every participant is graded individually -> Mark the sections to which you contributed in the report

Final Presentation

- 25min presentation of your work + 5-10min questions
- Everybody has to present (~ 7 – 10 min per person)
- General Structure
  1. Introduction (motivation/background, related work, contribution)
  2. Methods (overview, model concept, details and purpose of contribution)
  3. Experiments & Results (used dataset, experimental setup, results)
  4. Discussion (discussion of results, positive & key findings, limitations)
  5. Conclusion (summary, outlook)

- GRADED!

Final Report

- 9 pages without references
- Everybody has to contribute -> Mark the sections to which you contributed (3-4 pages per person in the group)
- Include a link to the GitHub repository which hosts your code
- LaTeX template is available in VC
- Structure
  1. Abstract (motivation, purpose, method, key findings, link to git repository)
  2. Introduction (motivation/background, related work, contribution)
  3. Methods (overview, model concept, details and purpose of contribution)
  4. Experiments & Results (used dataset, experimental setup, results)
  5. Discussion (discussion of results, positive & key findings, limitations)
  6. Conclusion (summary, outlook)
- GRADED!

Evaluation Criteria – Presentation

- Formalities
    - Timing
    - Appropriate references
    - Key info on slides
    - Avoidable technical problems
- Expertise/Content
    - Structure of Content
    - Appropriate Scope & Complexity
    - Appropriate use of technical terms
    - Educational for audience
    - Knowledgeable Q&A
- Presentation Technique
    - Layout of Slides
    - Quality of Media
    - Posture/eye contact w/ audience
    - Clear communication/voice
    - Respectful/non-defensive

Evaluation Criteria – Report

- Formalities
  - Appropriate length
  - Appropriate references (inter)
  - Appropriate references (intra)
  - Key info present
- Content
  - Abstract & Introduction
  - Main Body
  - Discussion & Summary
  - Technical understanding
  - Educational & clear to follow
- Presentation Technique
  - Logic/Structure
  - Appropriate use of tech. terms
  - Quality of layout and graphics
  - Readability/Language

Requirements
Code / GitHub repository

- The code has to be reproducible

- The README has to be self-explanatory
  - introduce and motivate your work
  - explain how it works
  - write a small tutorial
  - contact information

- Examples
  - Example 1, Example 2

Dataset Creation

- We will provide a dataset containing 10 classes of "indoor" specific objects
    - *coffee mug, wooden spoon, notebook, teapot, soup bowl, remote control, computer keyboard, mouse, toilet tissue, binder*
- You will design and create your own "hard" test set
    - Place objects in outdoor contexts (e.g., coffee mug on the grass)
    - Be creative!
    - Have fun!
- **Goal**: Measure the robustness of your algorithm

Dataset Creation – Classes

coffee_mug

notebook

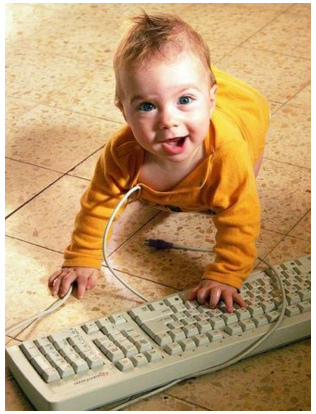
remote_control

soup_bowl

teapot

wooden_spoon

computer_keyboard



mouse

binder

toilet_tissue

Dataset Creation – Requirements

- Each student has to acquire, for each class
  - 30 images (30 imgs x 10 cls = 300 imgs)
  - [bonus] 60 images (600 imgs)

- Note
  - You must provide .jpg images
  - Use different objects (e.g., different mugs, etc)
  - Change orientation

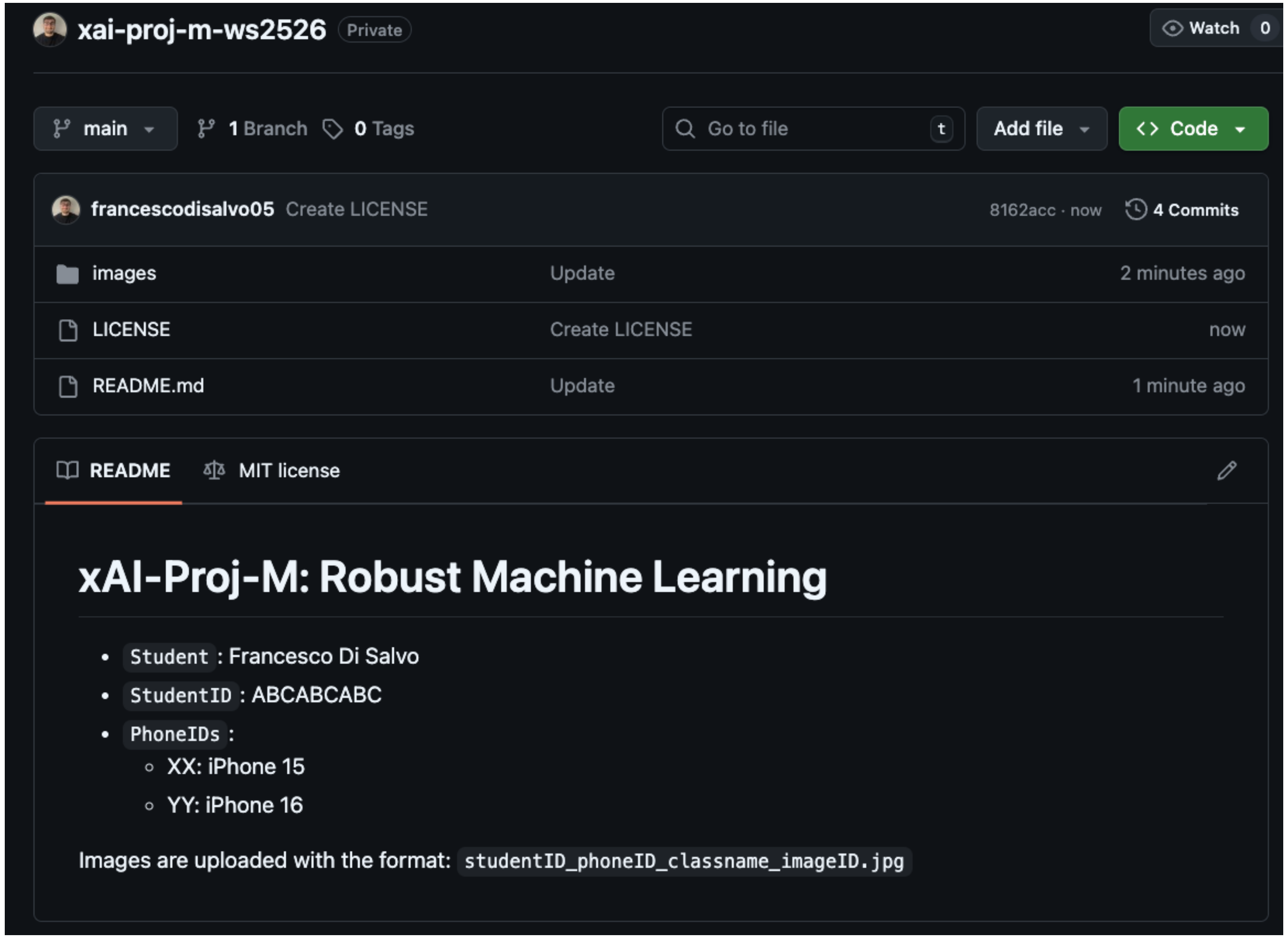
Dataset Creation – Uploading

- Where to upload
  - Each student has to upload their images into the public GitHub repository of their group
  - Consider adding an MIT License if you want others (e.g., us at the xAI Chair) to use these images in future scientific publications.

- How
  - Each student will be assigned a unique ID
  - You will upload the images in the format: `studentID_phoneID_classname_imageID.jpg`

- When
  - Two deadlines with a *minimum* number of images each (150)
  - Deadline 1: 23.11.2025
  - Deadline 2: 21.12.2025

Dataset Creation – Uploading Example



images	Update	2 minutes ago
LICENSE	Create LICENSE	now
README.md	Update	1 minute ago

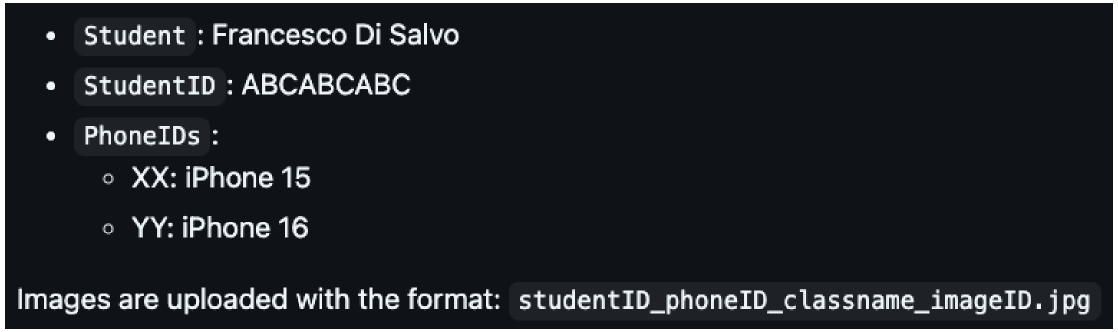
xAI-Proj-M: Robust Machine Learning

- `Student` : Francesco Di Salvo
- `StudentID` : ABCABCABC
- `PhoneIDs` :
  - XX: iPhone 15
  - YY: iPhone 16

Images are uploaded with the format: `studentID_phoneID_classname_imageID.jpg`

Dataset Creation – Uploading Example

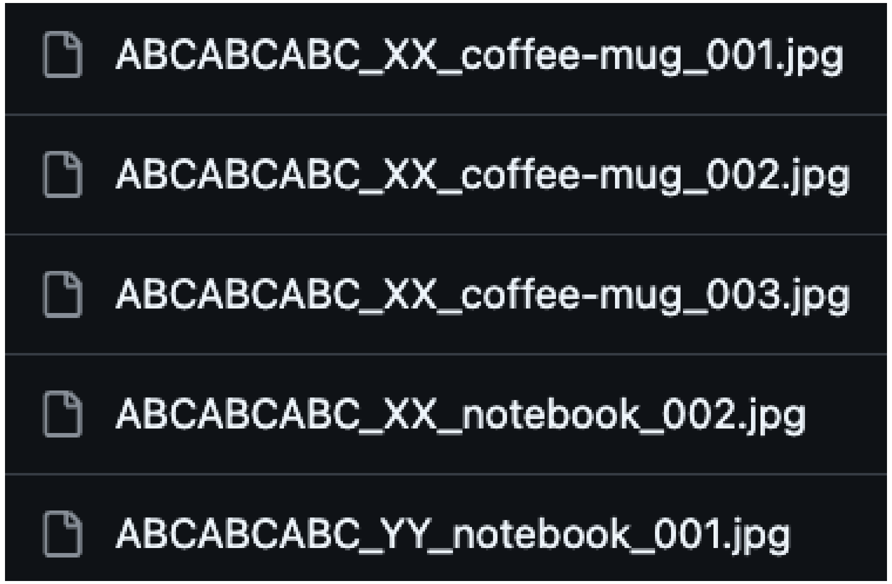
- Use "-" for multi-word classes
  - e.g., coffee-mug



- `Student` : Francesco Di Salvo
- `StudentID` : ABCABCABC
- `PhoneIDs` :
  - XX: iPhone 15
  - YY: iPhone 16

Images are uploaded with the format: `studentID_phoneID_classname_imageID.jpg`

- ImageID serves as class-counter
  - phoneID doesn't affect that



ABCABCABC_XX_coffee-mug_001.jpg

ABCABCABC_XX_coffee-mug_002.jpg

ABCABCABC_XX_coffee-mug_003.jpg

ABCABCABC_XX_notebook_002.jpg

ABCABCABC_YY_notebook_001.jpg

Resources

Scrum

- Product development framework
- Used to manage and organize complex development processes
- Describes roles, events, artifacts, rules for the development process
- Heart of Scrum is the Sprint (time frame of some weeks)
  - Incremental creation of a potentially launchable product
  - Steps involve: planning, daily updates, development work, Sprint review and Sprint retrospective
- Tutorials: tutorialspoint, atlassian

Git / GitLab

- Git: Version control system
  - Allows the collaboration on projects
  - Tutorials: the simple guide to git, w3schools

Git Logo © Jason Long, CC BY 3.0. Accessed: 04/07/2025

- GitLab: Project management tool around git
  - Definition of milestones and issues
  - (Visual) tracking of issues
  - File sharing
  - Documentation via the Wiki
  - Communication
  - Tutorial: official GitLab tutorial, w3schools,

GitLab Logo. Accessed: 04/07/2025

Coding

- Programming language – Python
  - Tutorials: Official Python Tutorial, W3 Schools Python
  - **Conda**: Open-source package and environment management (mini/ana)
- Recommended IDE – PyCharm
  - JetBrains offers free educational licenses for students
  - Offers split-windows, in-built debugger, version control system, ...
- Alternative IDE – Visual Studio Code
  - Free but needs some more configuration
  - Can be used for other programming languages as well
- Project Design Recommendations
  - Academics: You’re Doing Open Source Wrong:
  - The Good Research Code Handbook

Deep Learning – What do we need?

- Underlying Framework: PyTorch (Learn the Basics / Introduction to PyTorch)
- Datasets: Torchvision, Huggingface, etc.
- Dataloader: PyTorch (Tutorials: official [1, 2], wandb, mlm)
- Neural Networks / Models: Torchvision, timm [1, 2, 3], Huggingface
- Optimizer & Learning Rate Scheduler: timm [1, 2, 3], torch
- Loss Functions: torch.nn (Tutorials: neptune.ai, DigitalOcean, mlm)
- Evaluation Metrics & statistical Tests: neptune.ai, Paper
- Training & Evaluation Loop Tracker: wandb

Computation: Google Colab

Advantages

- Free compute resources including GPUs
- No infrastructure setup, works out of the box
- Uses Jupyter Notebooks

- Tutorials:
  - Towards Data Science
  - GeeksforGeeks
  - PyTorch Tutorial

Disadvantages

- Google account required
- Repetitive tasks in every new session (e.g. install dependencies)
- No persistent storage, all files are removed when session is restarted
- Google drive has a maximum free space of 15GB
- No live editing
- Notebooks can run up to 12 hours a day

Computation – Your Own

Not this!

But this!

Photo by Jens Mahnke on Pexels

Photo by John Pentalcurin on Pexels

Questions

Homework

Homework - General

- Organize yourselves in groups and let me know your team name, members and associated roles (if you have dedicated ones)

- Refresh **or gain** knowledge by working through (the provided) tutorials
  - Python, PyTorch/Timm/Huggingface, Git (& Gitlab), Google Colab, LaTeX, Scrum

Homework – Project Planning

- Create a schedule for your project
  - Purpose
    - Ensure you stay on track throughout the project
    - Break a complex task into manageable steps
    - Clarify responsibilities if working in teams
  - What your schedule should include
    - **Timeline:** with weekly or bi-weekly milestones
    - **Phases:** Python introduction, literature review, implementation, evaluation, augmentation, report writing
    - **Specific deliverables:** per phase (e.g., "Train baseline CNN by Week 5")
    - **Time allocation:** for reading, coding, testing, and documentation

Homework – Project Planning

- Key Milestones to plan for
  - Getting familiar with all tools, i.e., Python, PyTorch, Git, GitLab, etc. (e.g. week 1–2)
    - Work through the provided Tutorials and take notes
  - Literature review on architectures and optimization techniques (e.g. week 2–3)
    - What architectures are there for classification (ResNet-18, VGG-16, etc.)? Benefits/Disadvantages? Why did you choose the one or the other?
    - What optimizers and learning rate schedulers exist? Benefits/Disadvantages? Why did you choose the one or the other?
  - Own dataset collection (e.g. week 2-15)
  - Baseline model implementation and training (e.g. week 4–5)
    - Dataset Implementation (CIFAR10 + ImageNet-1k Subset [provided in VC])
      Specifics about the datasets (nr of samples, train/val/test splits, are they balanced or not, etc.)
    - Dataloader, Model, Optimizer and LR scheduler, train and eval loop, loss, metrics
  - Model tuning and performance evaluation (e.g. week 6–7)
    - Hyperparameter optimization (e.g. learning rate, batch size, etc.)
    - Evaluation of your model for at least three random seeds -> more reliable evaluation
  - Data augmentation experiments (e.g. week 8–9)
    - What data augmentation techniques are there? Benefits/Disadvantages?
    - How can these be integrated during training or even evaluation?
  - Evaluation against own dataset (e.g. week 10-11)
    - What is the performance on the collectively created dataset?
  - Report writing & presentation prep (e.g. Week 12-13+)

This is really important! If you do this considerate, you will have a much easier time to do the presentation and write the report!

Homework – Project Planning

- Tips for Success
  - Leave buffer time for troubleshooting
  - Document progress weekly
  - Set goals for group meetings

- IMPORTANT!
  I want you to first work closely together so everybody gets to experience literature review, learns about deep learning architectures and optimization techniques.
  Afterward you can also parallelize the work (e.g. one works on evaluation metrics, while someone else works on data augmentation)

Homework – Deliverables

- In two weeks (until November 4) I want you to create 1+ pages containing information about:
  - What tutorials you have done and what you did and learn there
  - Show which tools you want to use for your project and how you set these up (e.g. screenshots of your GitLab, Trello boards, Google Colab, etc.)
  - Show your role assignment among the team members or how you intend to proceed with this
  - Present your project schedule and what thoughts/ideas/issues you had while designing it
    - Which Milestones do you have?
    - How do you keep track of the information you gathered during the tutorials and literature review?
    - How do you keep each other in the team informed?
- ...and share them in the VC course with the other groups