FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Computational Imaging Project

Zhengguo Tan and Florian Knoll
Computational Imaging Lab (CIL), Friedrich-Alexander-Universität Erlangen-Nürnberg
Winter Semester 2023/2024

Outline

1 Introduction

Who we are?

- Dr. Zhengguo Tan
  - Postdoc in CIL
- Prof. Dr. Florian Knoll
  - W3 professor of CIL

What is CIL/CIP about?

- Semester period: till 13. April 2024
- **10 ECTS = 300 working hours**
- Prerequisite - You should have taken one of these courses:
  - pattern analysis or pattern recognition
  - magnetic resonance imaging (MRI) 1/2
  - **computational MRI** (given by Prof. Knoll every winter semester)
- **Hands-on**: learning by doing

Effort & Attendance

- 1 project per student
- effort:
  - reading and understanding papers
  - implementing ideas (coding)
  - analyzing results 50 %
  - final presentation (about 20 minutes) 20 %
  - written report (about 10 pages) due in three weeks after your presentation 20 %
- attendance:
  - final presentations: 10 %

Office Hours

- Zhengguo: Tue and Wed 10:00 – 11:30

- Book 30 min slots via StudOn: https://www.studon.fau.de/book5499411.html
- In person: Meet at our office (Room 2.02, Werner-von-Siemens-Str. 61)
- Zoom: https://fau.zoom-x.de/j/6394311813

Mid-Term Presentations

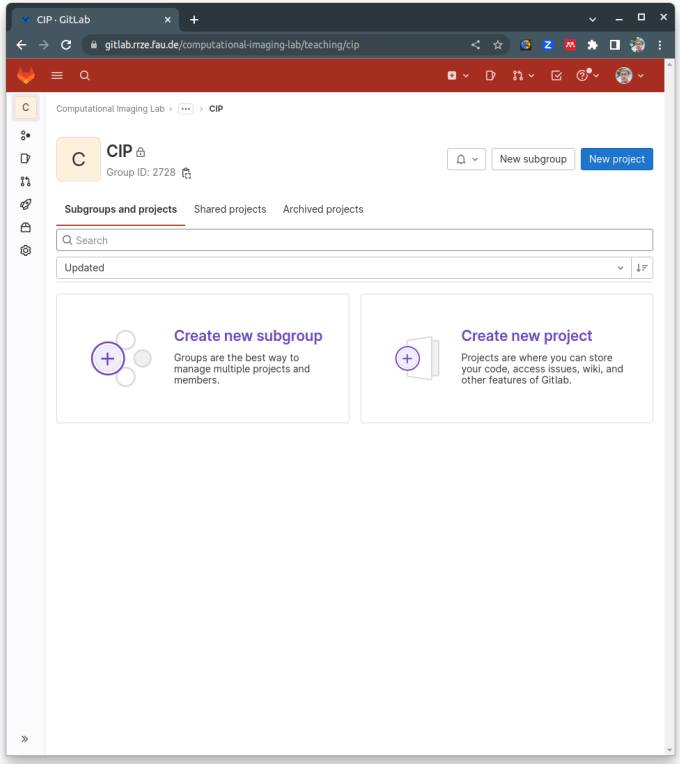
- tentative date: 09.01.2024 starting at 10 AM.
- place: Seminar room 03.17, Werner-von-Siemens-Str. 61
- every one needs to give a 5 to 10 minutes presentation about his/her project.

Final Presentations

- **Registration for the final exam via CAMPO (date: tba)**
- tentative date: 03.2024
- every one needs to give a 20 minutes presentation
- every one needs to attend the others' presentations

Management on Codes, Presentations, and Reports

1. unified environment: `https://gitlab.rrze.fau.de/computational-imaging-lab/teaching/cip/ws2023`
2. please request the FAU GitLab service via IdM-Portal.
3. I will then invite you to the your project repository.



Oral Presentation and Written Report

Presentation

- Format:
  - Motivation and Introduction
  - Theory
  - Methods
  - Results and Discussion
  - Conclusion

Report

- Format:
  - Introduction
  - Theory
  - Methods
  - Results and Discussion
  - Conclusion

- No template for presentation;
- there will be a LaTeX template for the report.

Computing Options

- **HPC @ FAU**:
  - https://hpc.fau.de/
  - requires knowledge on bash script in Linux terminal, anaconda, and python.
  - requires account application, so please let me know soon.
- **JupyterHub @ FAU**:
  - https://hpc.fau.de/systems-services/documentation-instructions/clusters/jupyterhub/
  - this is a new offer from NHR@FAU!
  - GTX1080Ti GPU
- **Google Colab**:
  - usually you can get a Tesla T4 GPU for free.
  - requires knowledge on jupyter notebook (bash script and python).
- **Your own computer**.

Computing Environments

- **Anaconda** → conda
  - flexible
  - reproducible
  - learning material:
    https://conda.io/projects/conda/en/latest/user-guide/index.html
- **Jupyter Notebook**
  - learning material:
    https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/
- IDE: Spyder, Visual Studio Code, PyCharm

Questions?

Self Introduction

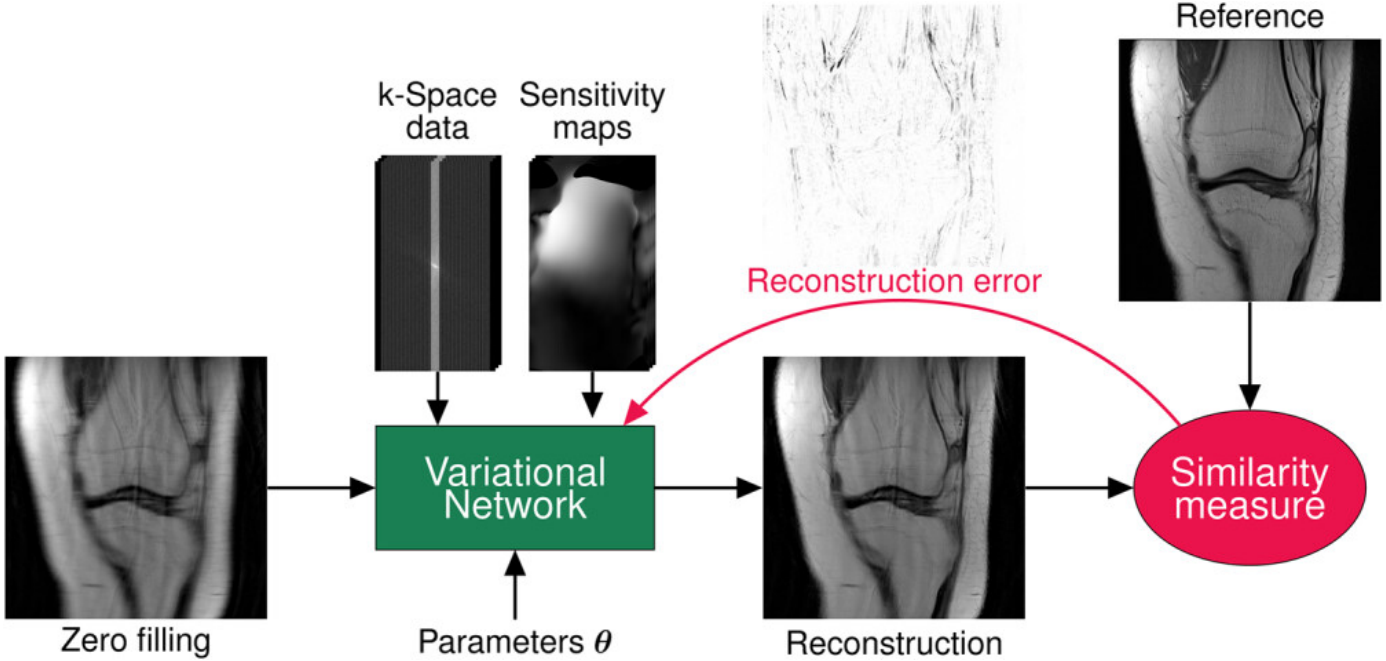
- who I am?
- study program / semester / courses
- what you want to learn/do in the CIL/CIP?

2 Projects

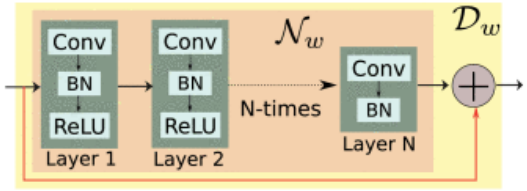
VarNet vs. MoDL

- Articles:
  - Hammernik K, Klatzer T, Kobler E, Recht MP, Sodickson DK, Pock T, Knoll F. Learning a variational network for reconstruction of accelerated MRI data. *Magn Reson Med* (2018). doi: 10.1002/mrm.26977
  - Aggarwal HK, Mani MP, Jacob M. MoDL: Model-based deep learning architecture for inverse problems. *IEEE Trans Med Imaging* (2019). doi: 10.1109/TMI.2018.2865356
- Basic code & data: `https:github.com/ZhengguoTan/MoDL_PyTorch`
- Suggested computing option: HPC
- Tasks:
  - Run both Varnet and MoDL based on the github repository;
  - Use fastmri dataset instead;
  - Change the UNet in VarNet and the ResNet in MoDL to a transformer.

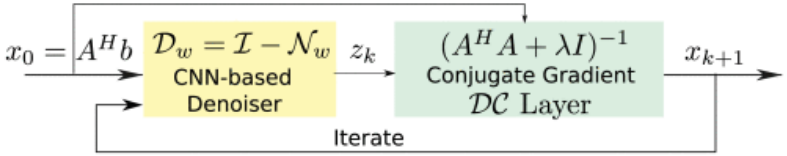
VarNet: with fully-sampled images available; supervised learning



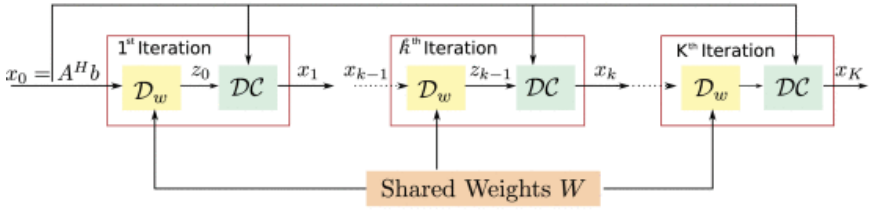
MoDL: with fully-sampled images available; supervised learning; shared weights



(a) The Residual learning based denoiser



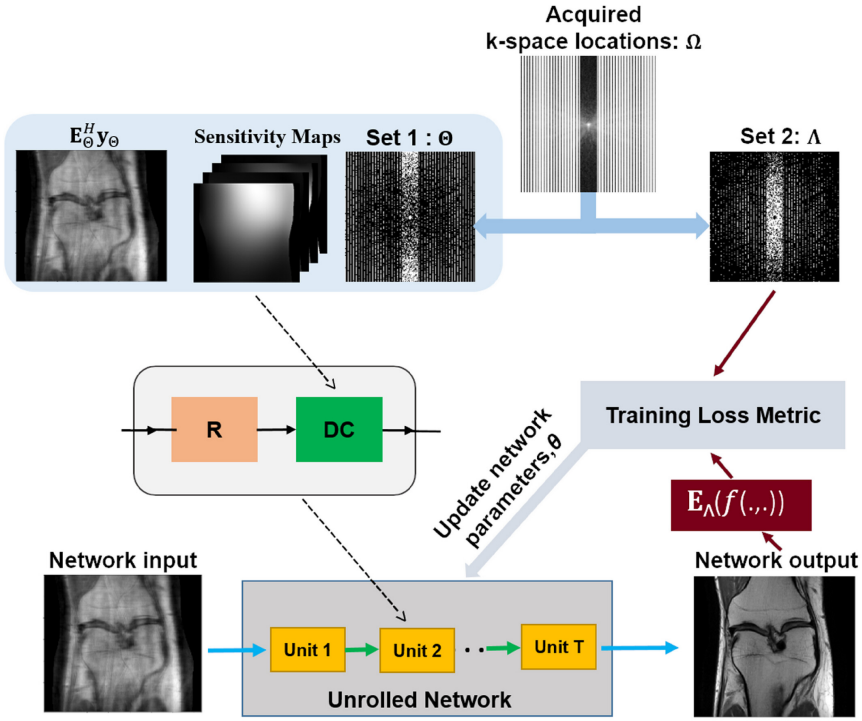
(b) Proposed Model-based Deep Learning (MoDL) architecture



Self-supervised learning via data undersampling (SSDU)

- Article:
  - Yaman B, Hosseini SAH, Moeller S, Ellermann J, Uğurbil K, Akçakaya M. Self-supervised learning of physics-guided reconstruction neural networks without fully sampled reference data. *Magn Reson Med* (2020). doi: 10.1002/mrm.28378
- Basic code: https://github.com/byaman14/SSDU, which was implemented in tensorflow
- Suggested computing option: HPC
- Tasks:
  - Understand the main concept of the SSDU architecture: ResNet;
  - Reproduce the existing implementation;
  - Apply different types of SSDU masks;
  - Analyze training and testing results;
  - Change the ResNet to a transformer.

SSDU: without fully-sampled images; self-supervised learning



Notes

- Every 2 or 3 students has the same project;
- Students with the same project are allowed to discuss with each other, but not to copy & paste;
- Presentations and Reports must be done individually and will be graded individually.

3 Homework before our next meeting

Homework

- read the slides and papers again;
- think about which project you want to work on;
- read the articles (and codes) related to your project;
- start to work on the project.

- **please sign up for the office hours:**
  https://www.studon.fau.de/book5115803.html
- Meet again next week (10 AM, 21.11.2023; Seminar Room 03.17)

Let's get started ...

- Thank you for your interest and attention!
- However, attention is not all you need - you also need to accomplish the project.

Synergies between the two projects

- VarNet & MoDL: Supervised learning; public source on PyTorch
- SSDU: Self-supervised learning; public source on tensorflow

→ I decide to ask all of you to work on one project: **VarNet vs. MoDL vs. SSDU**

VarNet vs. MoDL vs. SSDU Project Tasks

1. run the VarNet and MoDL on `https:github.com/ZhengguoTan/MoDL_PyTorch` (PyTorch)
2. run the SSDU on `https://github.com/byaman14/SSDU` (TensorFlow)
3. implement a Python script to convert 172 fastmri datasets to the VarNet format
4. implement SSDU on the project "MoDL_PyTorch" (PyTorch)
5. compare VarNet, MoDL and SSDU using the same datasets

⋆. change the U-Net architecture to a transformer

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

4 HPC

Getting Started [1]

1. Activate your HPC account via IdM FAU

2. Your accounts will be valid till the end of April 2024

3. Connecting to HPC systems, ssh USERNAME@CLUSTERNAME.nhr.fau.de
   - The cluster you need is tinyx
   - For IDE, I suggest use the SSH connection in VS Code;
   - For terminal in Windows, I suggest use MobaXterm.

4. Working with data
   - $HOME: standard home directory at login, available under /home/hpc (50 GB)
   - $WORK: general-purpose work directory (500 GB)

5. Data transfer
   - Under Linux and Mac systems, use scp or rsync

[1] https://hpc.fau.de/systems-services/documentation-instructions/getting-started/

Available Hardware in tinyx

- Use the command sinfo on the respective cluster frontend node to get the current status of the cluster nodes (idle, mixed, allocated)

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
rtx3080      up 1-00:00:00      1  down* tg080
rtx3080      up 1-00:00:00      1    mix tg081
rtx3080      up 1-00:00:00      5  alloc tg[082-086]
v100         up 1-00:00:00      1    mix tg071
v100         up 1-00:00:00      2  alloc tg[073-074]
v100         up 1-00:00:00      1   idle tg072
a100         up 1-00:00:00      3    mix tg[091-092,097]
a100         up 1-00:00:00      5  alloc tg[090,093-096]
work*        up 1-00:00:00      1  down* tg080
work*        up 1-00:00:00      2    mix tg081,tg06a
work*        up 1-00:00:00      6  alloc tg[082-086],tg06b
work*        up 1-00:00:00     10   idle tg[060-069]
```

Available Software

- General:
  1. The majority of software is provided by RRZE via the modules system
  2. The available modules can be listed via module avail
  3. Modules can be loaded via module load <modulename>
  4. The loaded modules can be displayed via module list
- TODO [2]:
  1. Load the Python module: module load python/3.8-anaconda

[2] https://hpc.fau.de/systems-services/documentation-instructions/special-applications-and-tips-tricks/python-and-jupyter/

Python & Jupyter: Installing Packages [IMPORTANT]

1. It is recommended to build packages using an interactive job on the target cluster to make sure all hardware can be used properly.
2. Make sure to load modules that might be needed by your python code (e.g. CUDA for gpu support)
3. Set the following lines in the file ~/.bashrc:

```
export http_proxy=http://proxy:80
export https_proxy=http://proxy:80
```

JupyterHub

- webpage based interface
- requires no local software installation
- good for interactive debugging and visualization

- login: `https://hub.hpc.fau.de/jupyter/` with your HPC username and password

Conda

1. standard way:
   - create a new conda environment: conda create -n varnet python=3.9

2. create a conda environment modl based on "requirements.txt":
   - copy / git clone the MoDL repository [3] to your $WORK directory
   - cd to the model directory
   - conda create –name <env> –file <this file>

[3] https://github.com/ZhengguoTan/MoDL_PyTorch

Jobs

1. Interactive jobs:
   - e.g. salloc –gres=gpu:a100:1 –partition=a100 –time=04:00:00
2. sbatch jobs:
   - sbatch
3. monitor jobs:
   - squeue
4. after the job is finished or if the job fails, check the output file.

fastmri data

- fastmri: https://fastmri.med.nyu.edu/
- data available on HPC: /home/janus/iwbi-cip-datasets/shared/fastMRI
- link 50 datasets to your WORK directory:

```
$ cd $WORK
$ mkdir cip_ws2023_data
$ cd cip_ws2023_data
$ ln -snf /home/janus/iwbi-cip-datasets/shared/fastMRI/brain/
  multicoil_train/file_brain_AXT2_210_6001* .
```

HPC@FAU Resources

- Please don't forget the great instructions: `https://hpc.fau.de/`
- For user training slides: `https://hpc.fau.de/systems-services/hpc-user-training/`

TODO

Tasks for today:

- install the conda env "modl" on HPC;
- install the conda env "ssdu" on HPC;

Tasks to be done within 7 days:

- user jupyter notebooks to check data/dataset.hdf5
  - What keys are stored inside?
  - What is the meaning of each key?
  - Are you able to make a zero-filled reconstruction?
- be able to run VarNet, MoDL and SSDU.

Question to you:

- to meet together here every Tuesday?
- to use offer hours such that you and I meet individually?