PYTORCH DEVELOPER DAY

Farabio: Deep Learning for Biomedical Imaging

PyTorch: Image/Video

Sanzhar Askaruly[1], Nurbolat Aimakov[1], Alisher Iskakov[2], Hyewon Cho[1], Yujin Ahn[1], Myeong Hoon Choi[1], Hyunmo Yang[1], Woonggyu Jung[1]

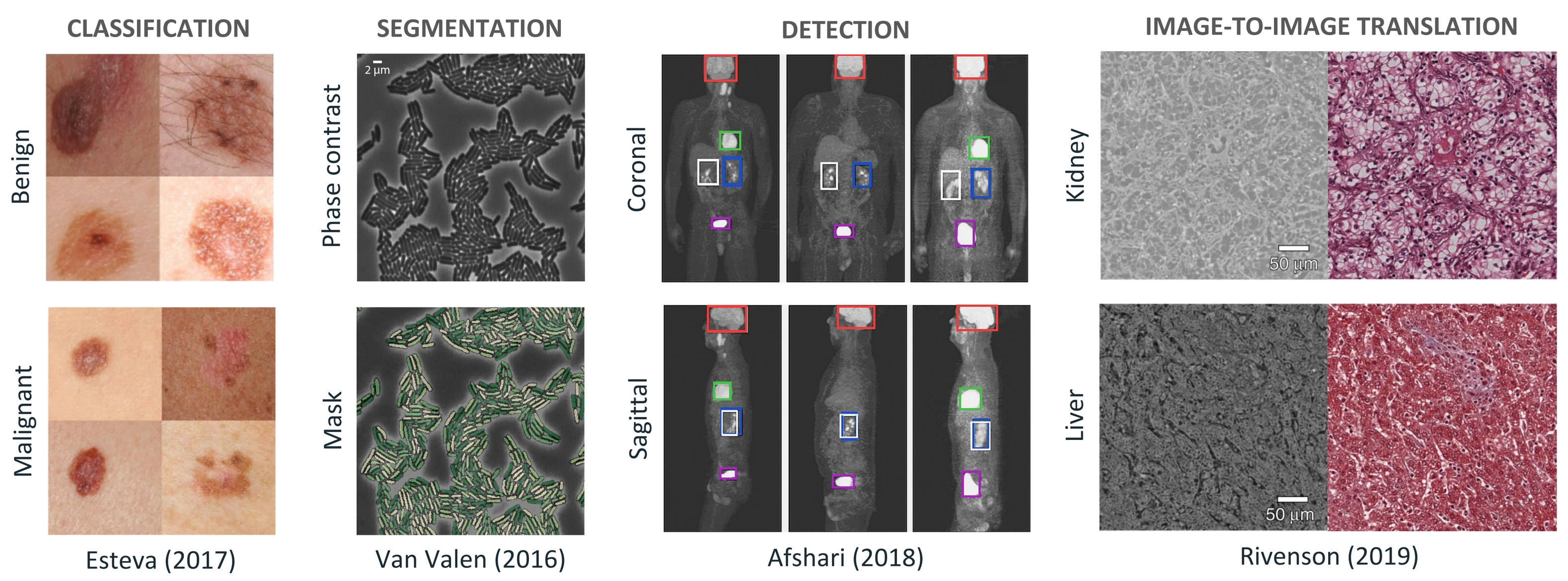
[1]Ulsan National Institute of Science and Technology, [2]Relog.kz

A1

DO NOT COVER

Abstract

Deep learning has transformed many aspects of industrial pipelines recently. Scientists involved in biomedical imaging research are also benefiting from the power of AI to tackle complex challenges (Kim, 2019). Although academic community has widely accepted image processing tools, such as scikit-image, ImageJ, there is still a need for a tool which integrates deep learning into biomedical image analysis. We propose a minimal, but convenient Python package based on PyTorch with common deep learning models, extended by flexible trainers and medical datasets. In this work, we also share theoretical dive in the form of course as well as minimal tutorials to run Android applications, containing models trained with Farabio.



Package Structure

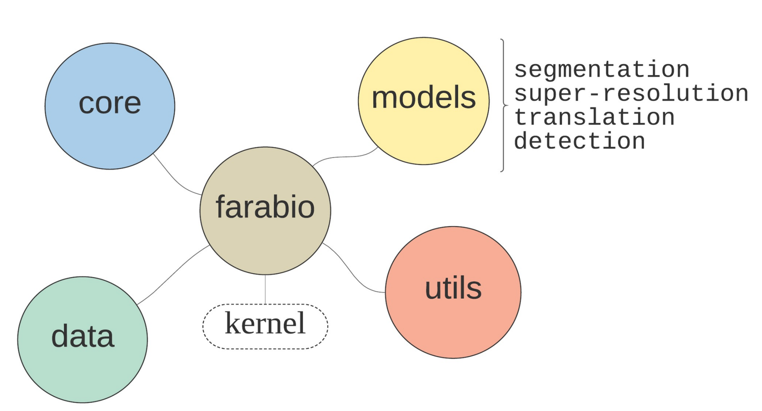


Fig 1. Package structure diagram

farabio package has 4 modules:

- **.core:** base trainer classes
- **.data:** preprocessing, datasets
- **.models:** architectures, trainers
- **.utils:** log, visualization, helpers

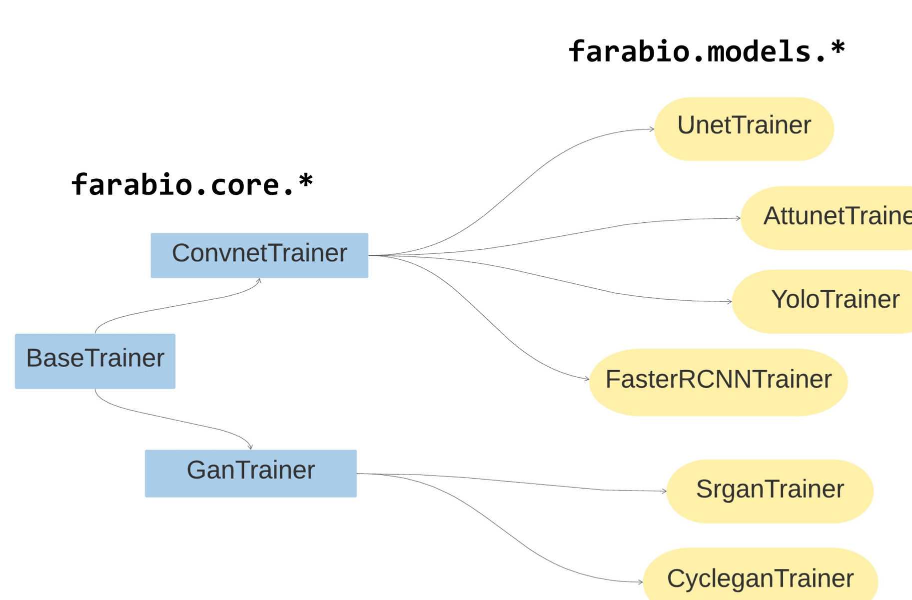


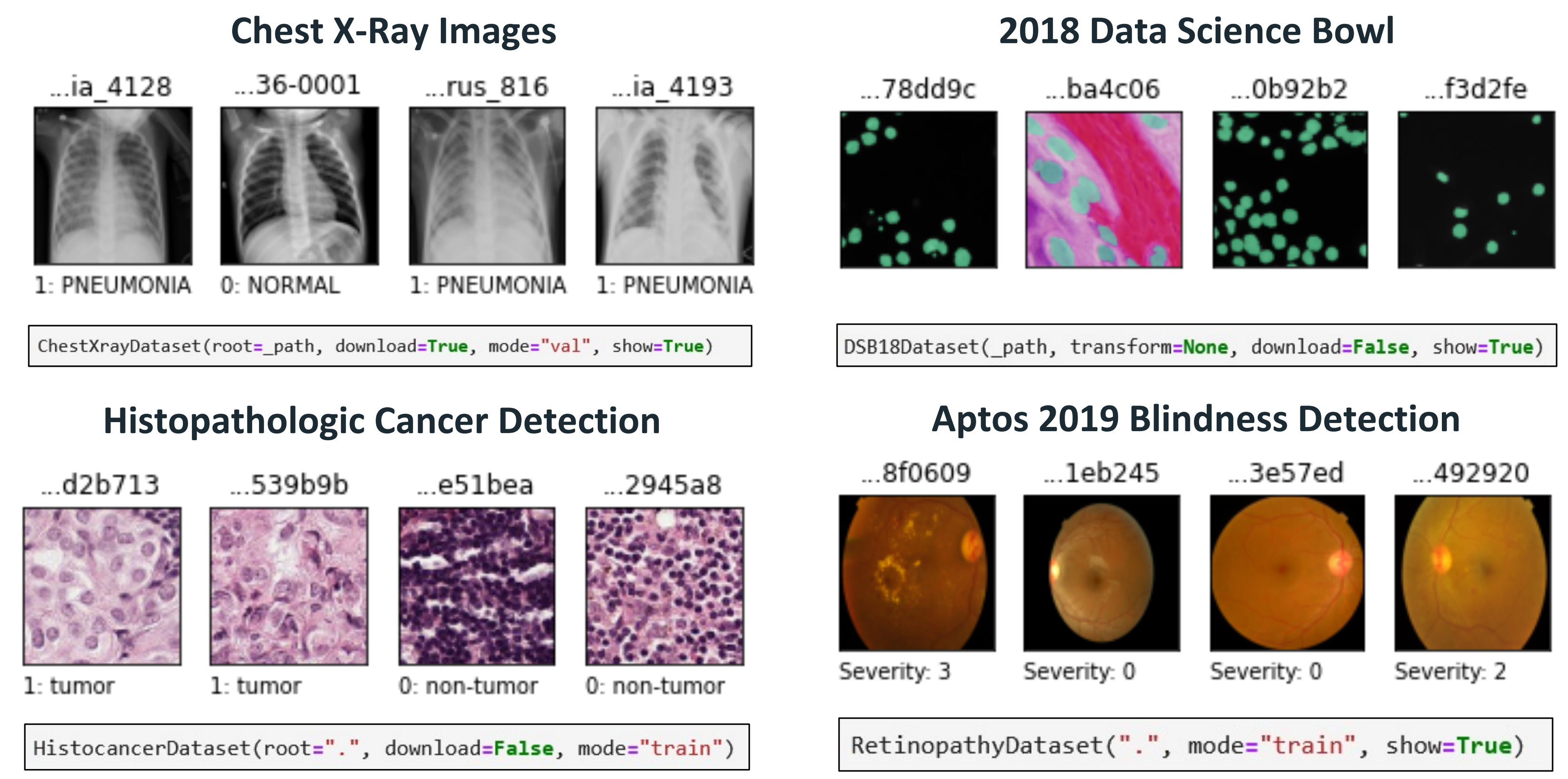
Fig 2. Inheritance diagram of trainers

Every deep learning model trainer derives from the **core** module. Based on the type, it has either **ConvnetTrainer** or **GanTrainer** as its parent.

farabio incorporates and modifies common deep learning models from open-source code released on GitHub *(see the docs)*

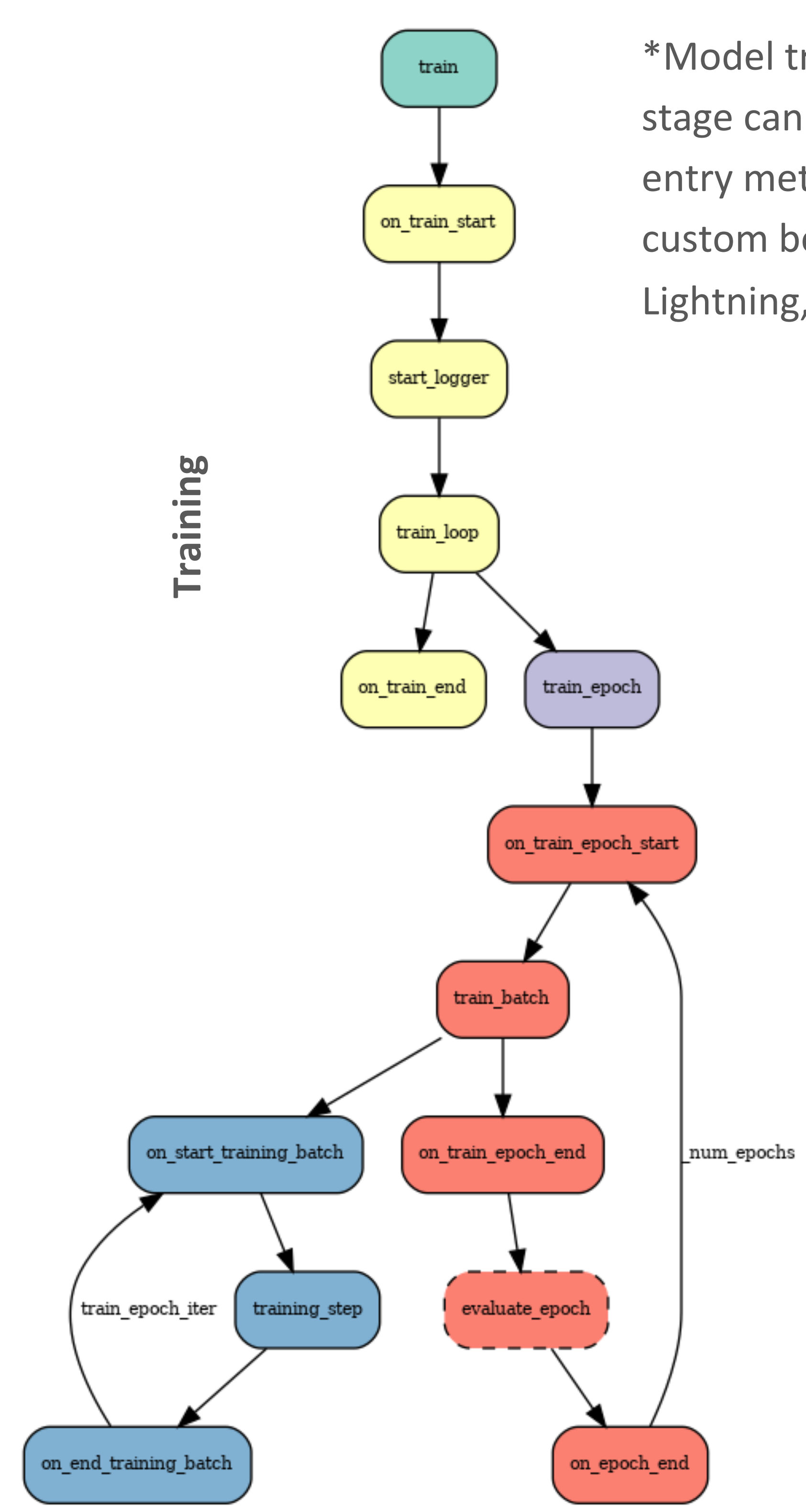
Biomedical Datasets

Several publicly available biomedical datasets which are retrieved from ISBI, MICCAI and Kaggle can be loaded from **farabio.data.biodatasets** module:



Lifecycle of Trainers

*Model trainers have predefined lifecycles. Each stage can be accessed from hooks, i.e., a specialized entry methods. In this way, users can override a custom behavior within loops, similar to PyTorch Lightning, but simpler *(see the docs for details)*.



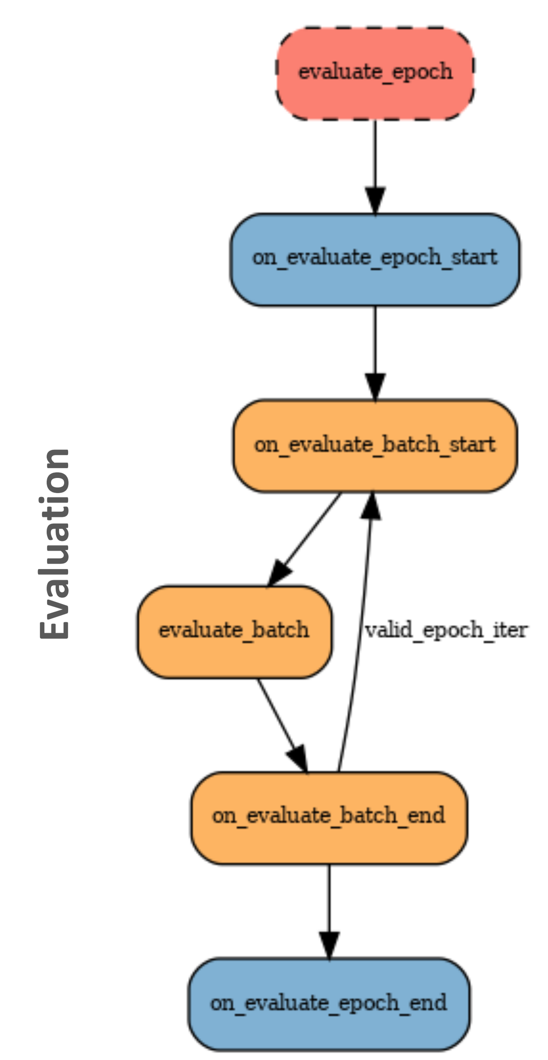
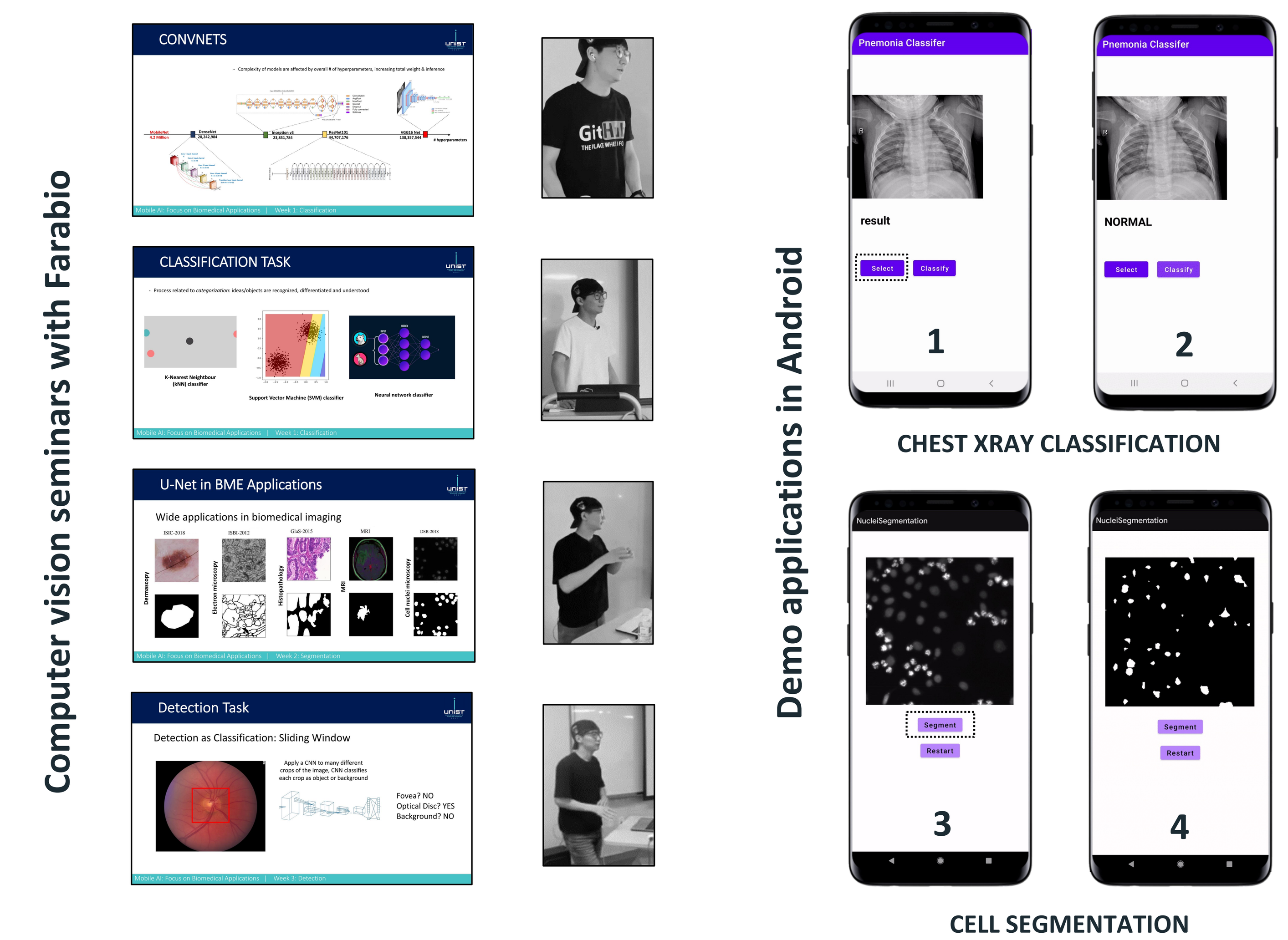


Fig 3. Diagrams of ConvnetTrainer lifecycle in training and evaluation loops

Mobile AI: Focus on biomedical applications

Farabio package was used to support computer vision seminars for biomedical engineers. It also integrates mobile-friendly models to deploy on mobile devices.



Future Works

In perspective, the vision of **farabio** package is to become an important player of PyTorch ecosystem, with research and academic community of biomedical engineering adopting it widely for different deep learning purposes. This is only possible with robust architectural foundations, rich dataset extensions and openness to new ideas, collaborations, which we aspire to cultivate.

Acknowledgements

This work has been inspired during the research at Translational Biophotonics Lab, UNIST and was supported with computational resources of the laboratory.

References

Afshari S, BenTaieb A, Hamarneh G. (2018), Automatic localization of normal active organs in 3D PET scans, *Comput Med Imaging Graph.*, Dec; 70:111-118

Dutta, S. (2020), A 2020 Guide to Deep Learning for Medical Imaging and the Healthcare Industry, retrieved from https://nanonets.com/blog/deep-learning-for-medical-imaging/

Esteva, A., Kuprel, B., Novoa, R. et al. (2017), Dermatologist-level classification of skin cancer with deep neural networks, *Nature* 542, 115–118.

Kim, M., Yun, J., Cho, Y., Shin, K., Jang, R., Bae, H. J., & Kim, N. (2019), Deep Learning in Medical Imagin, *Neurospine*, 16(4), 657–668.

Rivenson, Y., Liu, T., Wei, Z. et al. (2019), PhaseStain: the digital staining of label-free quantitative phase microscopy images using deep learning, *Light Sci Appl* 8, 23

Van Valen DA, Kudo T, Lane KM, Macklin DN, Quach NT, et al. (2016), Deep Learning Automates the Quantitative Analysis of Individual Cells in Live-Cell Imaging Experiments, *PLOS Computational Biology*, 12(11)

Links

Email: s.askaruly@gmail.com (San)

Git: @tuttelikz

Take a photo to learn more: