DenoiSeg Lab - Exercises

Learning Objectives

- ❏ Train DenoiSeg networks using the example notebooks.
- ❏ Understand how to use CSBDeep and DenoiSeg.
- ❏ Use Tensorboard to monitor the training process (while ongoing).
- ❏ Learn how to adapt the examples to use your own data.
- ❏ Bonus: use a trained network from within Fiji

Preamble: Be sure you have followed the installation instructions in the separate document 'DenoiSeg Lab - setup'. If you did so, you should be good to go!

Exercise #0: Start Jupyter

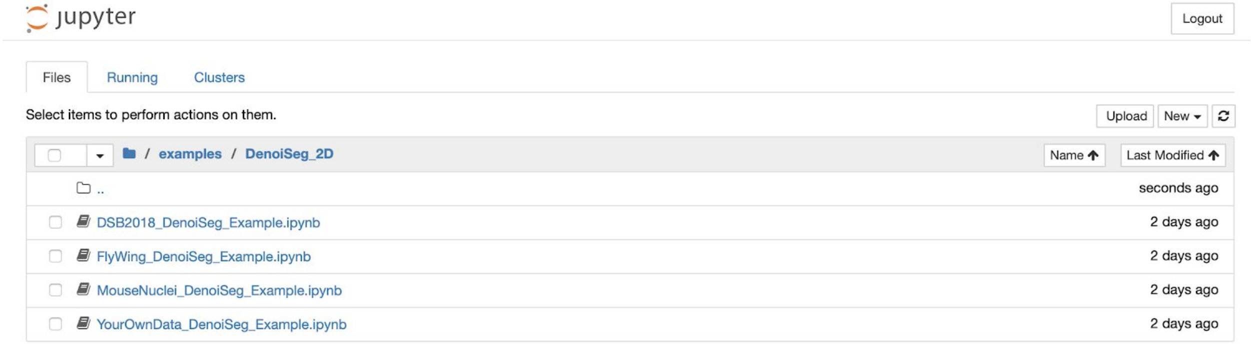
Open a terminal and navigate to the location where you cloned your N2V repository.
Please start the Jupyter server via:

```
$ jupyter notebook
```

This should open a browser and show you all files contained in the directory.

Exercise #1: Train your first DenoiSeg network

The GIT repo 'DenoiSeg' you have previously cloned into '`<some_folder>/DenoiSeg`' contains example notebooks corresponding to three different datasets: 1) DSB 2018 dataset, 2) Flywing dataset and 3) MouseNuclei dataset. Apart from these notebooks, there is an additional notebook to demonstrate how you can use DenoiSeg with your own data. This notebook is called '`YourOwnData_DenoiSeg_Example.ipynb`'.



Select items to perform actions on them.

/ examples / DenoiSeg_2D

..	seconds ago
DSB2018_DenoiSeg_Example.ipynb	2 days ago
FlyWing_DenoiSeg_Example.ipynb	2 days ago
MouseNuclei_DenoiSeg_Example.ipynb	2 days ago
YourOwnData_DenoiSeg_Example.ipynb	2 days ago

We will take a look at the '`YourOwnData_DenoiSeg_Example.ipynb`' notebook which you can use to train DenoiSeg models for your own data. The other notebooks do a similar thing for other datasets mentioned in our paper (https://arxiv.org/abs/2005.02987).

YourOwnData_DenoiSeg_Example.ipynb - This notebook first downloads all data in the folder '`DenoiSeg/examples/DenoiSeg_2D/data/MyData/`'. Inside this directory, there will be three subfolders called '`train`', '`val`' and '`test`'. If you look inside the '`train`' folder, you will find that it contains two more subdirectories called '`raw`' and '`gt`'. The '`raw`' folder contains 1428 raw noisy images and the '`gt`' folder contains only 5 ground truth segmentation annotations corresponding to the first 5 raw images in '`raw`' folder. Similarly, '`val`' folder also contains '`raw`' and '`gt`' subfolders containing 252 raw noisy images and only 2 ground truth segmentation annotations respectively. The '`test`' folder contains 42 unseen raw noisy images which we want to denoise and segment.

Next, the notebook will train your DenoiSeg network. All outputs will be put into the folder '`models`'. While you execute this notebook, see what files will be generated inside the models-folder. As you can see, the example notebooks are containing quite some additional explanations and some cells that have the

purpose of showing you the data you are about to use and some sampled results. In this way you can be sure that the right things are happening. Once you reach the cell that is actually starting the network training, you can go ahead and use Tensorboard (as described in the notebook) to monitor the training process.
If you did not change the default settings you will have around 3-4 hours to play with Tensorboard before the training of your first DenoiSeg network will be done. Enjoy (and please ask questions if you have any)!

After training finishes, you can see the training and validation losses. See that they have converged (Do not go up and down any more and seem to be settled on more or less constant values. If the validation loss looks like it is still going down, it indicates that you may need to train longer by increasing the number of epochs).

Then proceed on with the next few cells which do the prediction on test data and show the denoised and segmented outputs.

At the very bottom you see, that we can even even export a so-called Tensorflow Saved Model'. Such a zip file contains all data and metadata to fully define a Tensorflow model. We will later, in a bonus exercise, use such a saved model to apply a trained DenoiSeg network to raw data from within Fiji.

Exercise #2: What is going on behind the scenes?

Seek to understand what happened behind the scenes while executing the three notebooks you just executed in Exercise 1.

Try to answer the following questions:

1. Where is the training and test data located?
2. How does data have to be stored so that DenoiSeg will find and load it correctly?
3. Where are trained models stored? What models are stored? What is the difference between them?
4. Where in the notebooks is the name of the saved models determined?
5. How can you influence the number of training steps per epoch? What did you use (likely in the interest of time) and what value is suggested?
6. While this is not done in the example notebooks, how would you load an existing DenoiSeg network and continue to train it?
7. How would you do the same thing in such a way that the additionally trained model is stored separately (under a new name)?

Exercise #3 (bonus): Use a trained DenoiSeg network from within Fiji

You can use any `TF_SavedModel.zip` you created at the very bottom of any training notebook.

With a trained and saved network at hand, please open Fiji and follow these sequence of steps:

1. Help > Update…
2. Manage update sites
3. Add update site
4. Give the new site a name, e.g. "CSBDeep", and point it to "https://sites.imagej.net/CSBDeep/"
5. Restart Fiji
6. Open the image that you want to denoise.
7. Click Import > bioimage.io.zip and choose the network ZIP-file you have exported from any of the Noise2Void notebooks.
8. Click 'Predict' and choose the image to be denoised and segmented. Also choose the Axes of prediction input (for 2D images this is likely to be 'YX').
9. Click 'OK' and wait.
10. More details on the the entire process can be found at https://imagej.net/DenoiSeg