## Tutorial how to analyse data

* Installation
  + Install Python
  + Install GitHub
    - Download the NanoObjectDetection package
  + Install Anaconda
  + Open Anaconda Prompt and create environment out of spyder-env.yml (<https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html>)
  + Open Spyder, which is automatically installed by the environment
    - Add in preference (symbol: wrench), IPython Console, Startup, “Run a file” the startup.py from the package. So all required packages are loaded at the beginning
    - Add the Githubfolder (and maybe subfolder) to the PythonPath Manager
  + You might want to use a fast PC with RAM (8 GB is very low, 16 OK, 32 super and more is needed for very large images or number of frames)
* Copy Data
  + Copy the TestData on your hard drive (or your mars folder)
  + Don’t work on the GitHub Folder
* Create the evaluation file
  + Run nd.Start.NewEvaluation() in console
    - Choose 1 as single tif file
    - Select the copied data
    - Choose a folder to save it in
    - Select setup 3 “Olympus\_10x\_0-25\_plan\_n\_Olympus\_corpus\_Basler\_AC4096-40um\_camera “ (the number might change!) for all three given examples
    - Choose Fiber “2 - 1250b3”(the number might change)
    - Wavelength 532 and Framerate 450 fps and 1.9 ms (for all three test data)
    - Temperature control: recommended (1)
      * 22.4°C (9nm), 21°C (20nm) and 22.3 °C (30nm)
      * Solvent is water (without “”)
    - Choose 1 to enable a recommended choice of help functions
      * Choose 2 if you know the best image diameter from previous experiments with identical hardware
  + Prepare the python script
    - Go to the folder that is written in the console and open auswertung.py
  + path of parameter file
    - Change the ParameterJsonFile variable to the absolute path of the created parameter file
  + If the SNR of the data is poor, consider switching “EnhanceSNR” to 1, in the parameter.json file
* Make evaluation
  + #%% check if the python version and the library are good
    - See if the logger produces errors or warnings
  + #%% read in the raw data to numpy
    - Data is read in and follow the instructions regarding the saturation
  + #%% choose ROI (Region of Interest) if wanted
    - Here it is possible to not use the full image in terms of x and y, and also of frames. This can be changed in the json at the key “ROI”. Don’t forget to “apply” it if needed
  + #%% standard image preprocessing
  + #%% help with the parameters for finding objects
    - Insert the smallest expected hydrodynamic diameter. The physical diameter is a good choice here if known.
* Create 3D Tiff stack out of list of 2D files
  + Call nd.handle\_data.SaveTifSeriesAsStack(r"\\mars\testfolder ", CreateSubFolder=True). Keep in mind that you give the image list containing folder
  + handle\_data.SaveTifSeriesAsStack\_MainDirectory can do the previous command for all folders inside the given folder