Presentation on how to install LAUETOOLSNN

Ravi raj purohit PURUSHOTTAM RAJ PUROHIT[1], Jean-Sebastien MICHA[1]

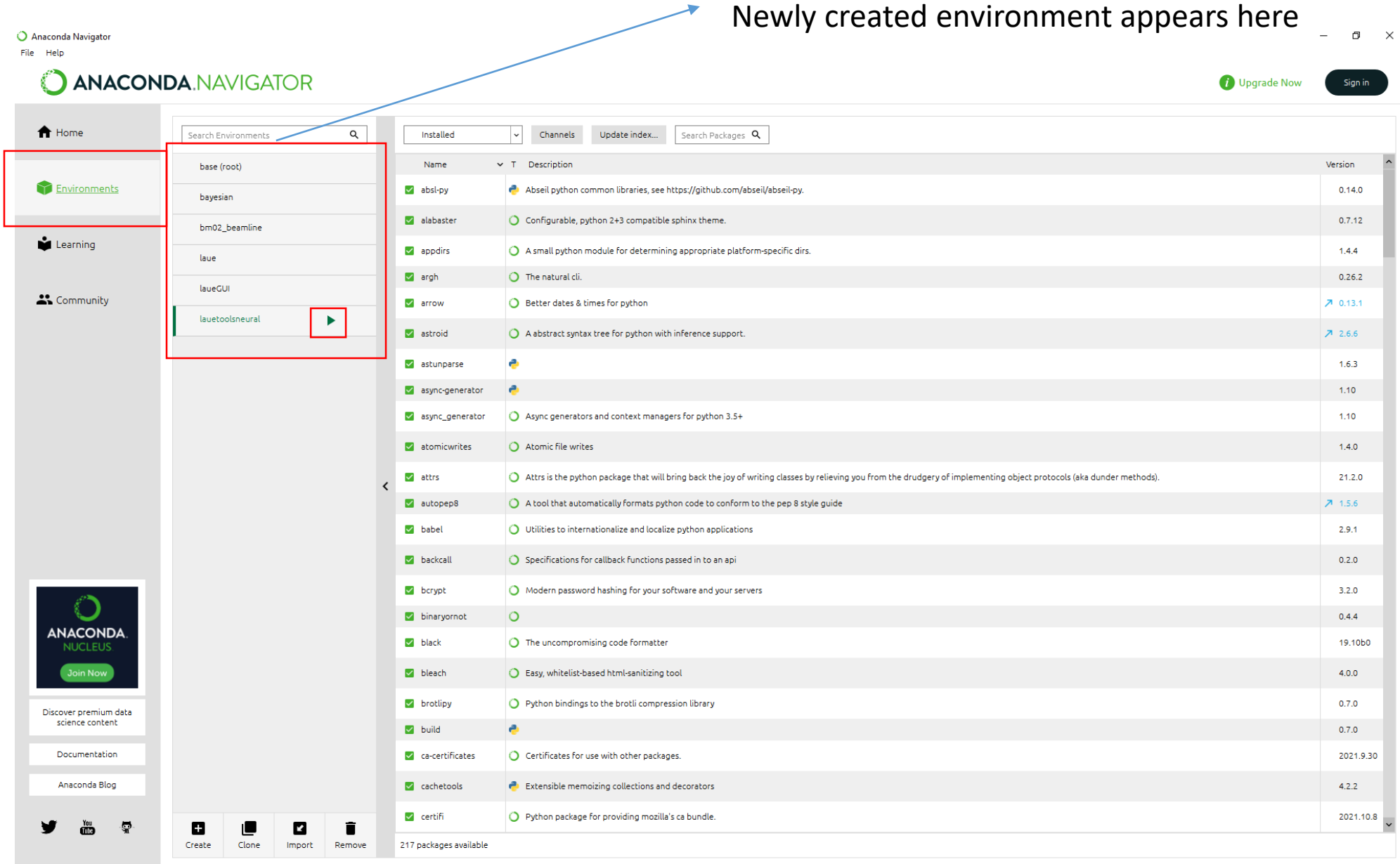
[1]French CRG beamline BM32, ESRF, Grenoble

Requirements

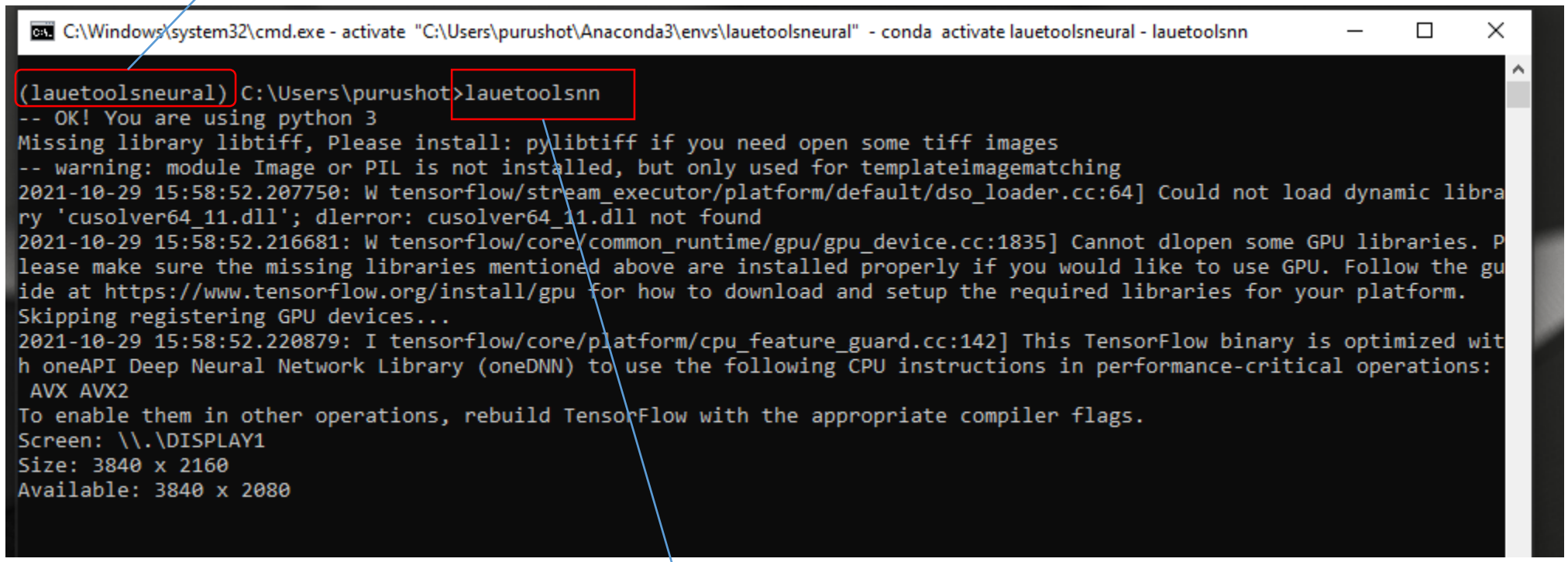
- Anaconda distribution
  - https://www.anaconda.com/products/individual

- Create a new environment
  - conda create –name lauenn python=3.8
  - Lauetoolsnn requires python >3.6

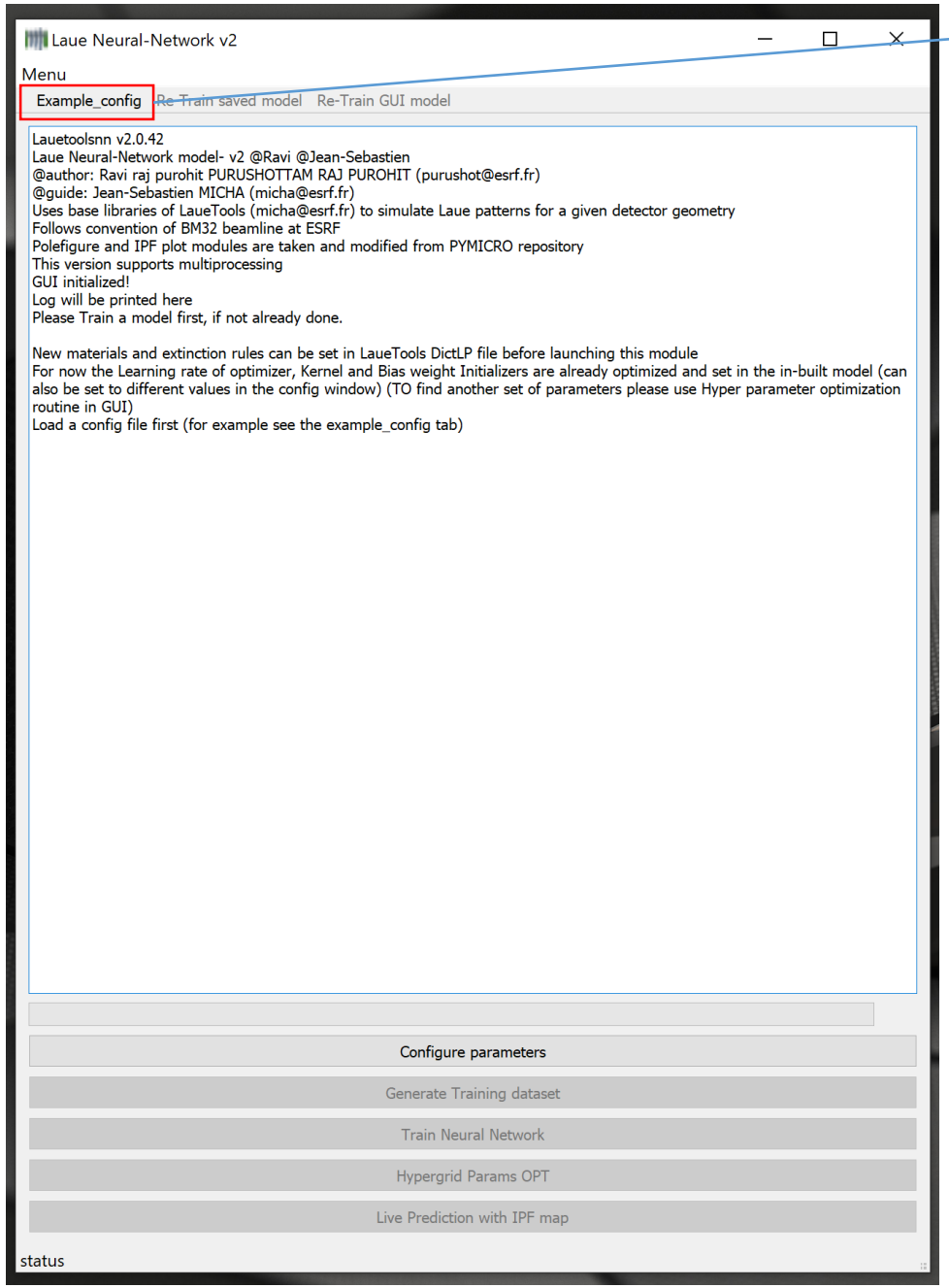
- Activate the created environment and install lauetoolsnn
  - conda activate lauenn
  - pip install lauetoolsnn
    - Includes all required libraries (in case of errors; try the last command with “--user” suffix)

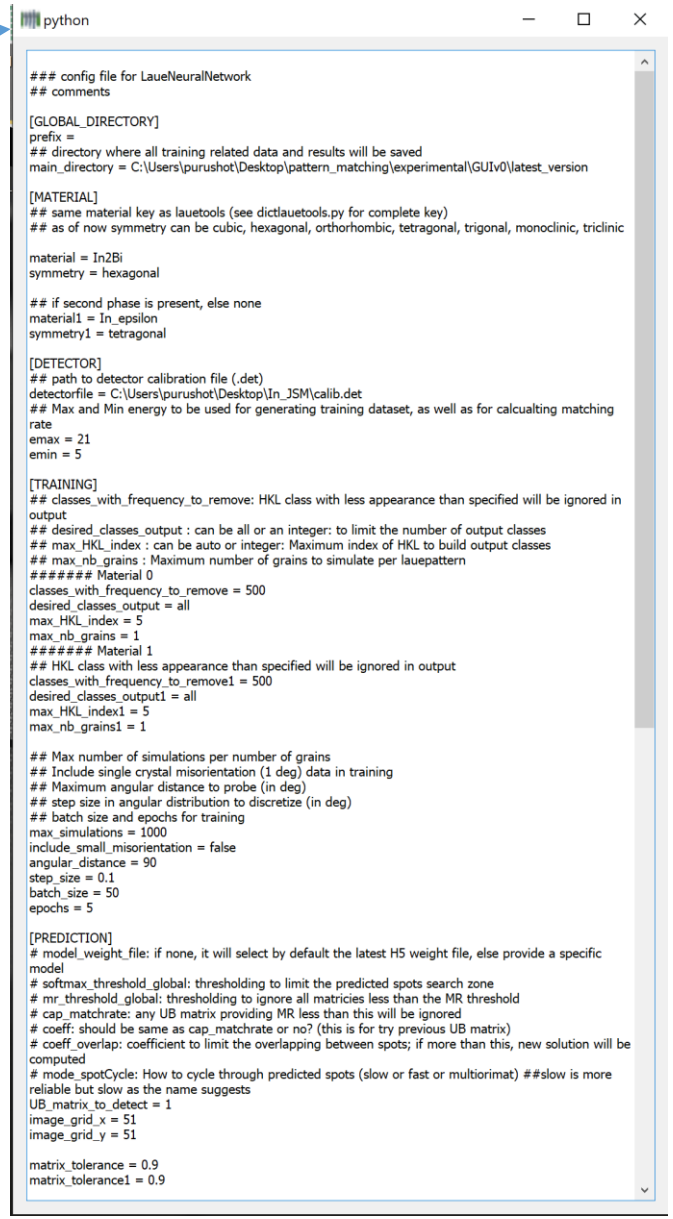

Newly created environment appears here
Anaconda Navigator
File Help
ANACONDA.NAVIGATOR
Upgrade Now
Sign in
Home
Environments
Learning
Community
Search Environments
base (root)
bayesian
bm02_beamline
laue
laueGUI
lauetoolsneural
Installed
Channels
Update index...
Search Packages
Name T Description Version
absl-py Abseil python common libraries, see https://github.com/abseil/abseil-py. 0.14.0
alabaster Configurable, python 2+3 compatible sphinx theme. 0.7.12
appdirs A small python module for determining appropriate platform-specific dirs. 1.4.4
argh The natural cli. 0.26.2
arrow Better dates & times for python 0.13.1
astroid A abstract syntax tree for python with inference support. 2.6.6
astunparse 1.6.3
async-generator 1.10
async_generator Async generators and context managers for python 3.5+ 1.10
atomicwrites Atomic file writes 1.4.0
attrs Attrs is the python package that will bring back the joy of writing classes by relieving you from the drudgery of implementing object protocols (aka dunder methods). 21.2.0
autopep8 A tool that automatically formats python code to conform to the pep 8 style guide 1.5.6
babel Utilities to internationalize and localize python applications 2.9.1
backcall Specifications for callback functions passed in to an api 0.2.0
bcrypt Modern password hashing for your software and your servers 3.2.0
binaryornot 0.4.4
black The uncompromising code formatter 19.10b0
bleach Easy, whitelist-based html-sanitizing tool 4.0.0
brotlipy Python bindings to the brotli compression library 0.7.0
build 0.7.0
ca-certificates Certificates for use with other packages. 2021.9.30
cachetools Extensible memoizing collections and decorators 4.2.2
certifi Python package for providing mozilla's ca bundle. 2021.10.8
ANACONDA. NUCLEUS
Join Now
Discover premium data science content
Documentation
Anaconda Blog
Create
Clone
Import
Remove
217 packages available

Conda environment with lauetoolsnn installed



Once installed, launch the GUI by calling lauetoolsnn in the terminal





Example_config: provides a template to create a config file for automated analysis

Setting in config file

- ### config file for LaueNeuralNetwork
- [GLOBAL_DIRECTORY]
- prefix =
- ## directory where all training related data and results will be saved
- main_directory = C:\Users\purushot\Desktop\pattern_matching\experimental\GUIv0\latest_version
- [MATERIAL]
- ## same material key as lauetools (see dictlauetools.py for complete key)
- ## as of now symmetry can be cubic, hexagonal, orthorhombic, tetragonal, trigonal, monoclinic, triclinic
- material = In2Bi
- symmetry = hexagonal
- ## if second phase is present, else none
- material1 = In_epsilon
- symmetry1 = tetragonal
- [DETECTOR]
- ## path to detector calibration file (.det)
- detectorfile = C:\Users\purushot\Desktop\In_JSM\calib.det
- ## Max and Min energy to be used for generating training dataset, as well as for calcualting matching rate
- emax = 21
- emin = 5

Prefix: specify a string to distinguish the model files or keep blank

If 2 materials present specify them here; in case of only 1 material, keep none for the other

Detector settings

Setting in config file

- [TRAINING]
- ## classes_with_frequency_to_remove: HKL class with less appearance than specified will be ignored in output
- ## desired_classes_output : can be all or an integer: to limit the number of output classes
- ## max_HKL_index : can be auto or integer: Maximum index of HKL to build output classes
- ## max_nb_grains : Maximum number of grains to simulate per lauepattern
- ####### Material 0
- classes_with_frequency_to_remove = 500
- desired_classes_output = all
- max_HKL_index = 5
- max_nb_grains = 1
- ####### Material 1
- ## HKL class with less appearance than specified will be ignored in output
- classes_with_frequency_to_remove1 = 500
- desired_classes_output1 = all
- max_HKL_index1 = 5
- max_nb_grains1 = 1
-
- ## Max number of simulations per number of grains
- ## Include single crystal misorientation (1 deg) data in training
- ## Maximum angular distance to probe (in deg)
- ## step size in angular distribution to discretize (in deg)
- ## batch size and epochs for training
- max_simulations = 1000
- include_small_misorientation = false
- angular_distance = 90
- step_size = 0.1
- batch_size = 50
- epochs = 5
-

Settings related to Training of Neural network

If unsure, just modify max_nb_grains and max_nb_grains1 ; i.e. maximum grains expected per material

All the other settings can be kept same

Setting in config file

- [PREDICTION]
- # model_weight_file: if none, it will select by default the latest H5 weight file, else provide a specific model
- # softmax_threshold_global: thresholding to limit the predicted spots search zone
- # mr_threshold_global: thresholding to ignore all matricies less than the MR threshold
- # cap_matchrate: any UB matrix providing MR less than this will be ignored
- # coeff: should be same as cap_matchrate or no? (this is for try previous UB matrix)
- # coeff_overlap: coefficient to limit the overlapping between spots; if more than this, new solution will be computed
- # mode_spotCycle: How to cycle through predicted spots (slow or fast or multiorimat) ##slow is more reliable but slow as the name suggests
- UB_matrix_to_detect = 1
- image_grid_x = 51
- image_grid_y = 51
- matrix_tolerance = 0.9
- matrix_tolerance1 = 0.9
- material0_limit = 1
- material1_limit = 1
- model_weight_file = none
- softmax_threshold_global = 0.85
- mr_threshold_global = 0.80
- cap_matchrate = 0.01
- coeff = 0.3
- coeff_overlap = 0.3
- mode_spotCycle = slow
- ##true for few crystal and prefered texture case, otherwise time consuming; advised for single phase alone
- use_previous = true
-

Settings related to Prediction

if unsure, just modify
Grid_x and grid_y : dimension (steps) of scan area

matrix_tolerance and matrix_tolerance1 ; tolerance angle to search for spots: keep 0.5 if unsure

How many grains to detect per material: material0_limit, material1_limit restricts that
use_previous = true; for trying previously identified rotation matrix

mode_spotCycle: graphmode, beamtime mode are the fastest; subsequently on can use also slow mode for more thorough search.

All the other settings can be kept same

Setting in config file

- [EXPERIMENT]
- experiment_directory = C:\Users\purushot\Desktop\In_JSM\ech875_ROI01
- experiment_file_prefix = ech875_ROI01_
- [PEAKSEARCH]
- intensity_threshold = 90
- boxsize = 15
- fit_peaks_gaussian = 1
- FitPixelDev = 15
- NumberMaxofFits = 3000

- [STRAINCALCULATION]
- strain_compute = true
- tolerance_strain_refinement = 0.7,0.6,0.5,0.4,0.3,0.2
- tolerance_strain_refinement1 = 0.7,0.6,0.5,0.4,0.3,0.2

- [POSTPROCESS]
- hkls_subsets = [1,1,0],[1,0,0],[1,1,1]
- [DEVELOPMENT]
- # could be 1 or 2 / none in case of single phase
- material_phase_always_present = 1
- write_MTEX_file = true
- material0_lauegroup = 9
- material1_lauegroup = 5

Settings related to Experiment and misc

material_phase_always_present = 1; forces the neural network to predict always the first grain from material 1; leads to nice plots

MTEX output for plotting PF and IPFs ; specifiy the proper laue group to get proper colors

Predefined uniform orientation distribution for all symmetries (used during training)

- It is possible to define particular orientations too (in case of preferred texture) → however this will lead to bias in model