Low-cost GUI-based deep learning deployment solutions

Getting started with napari in phenotyping

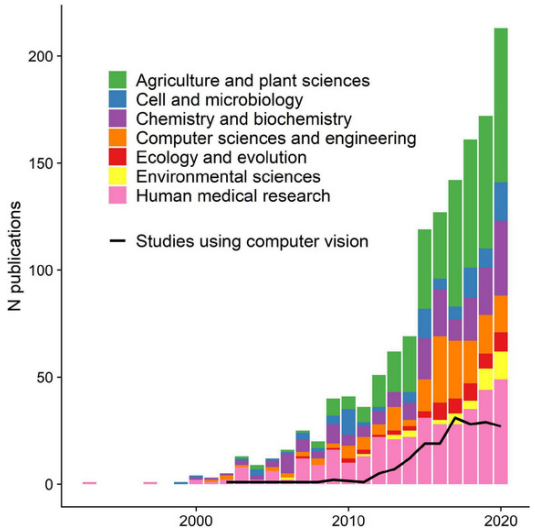
Herearii METUAREA, Engineer INRAe France

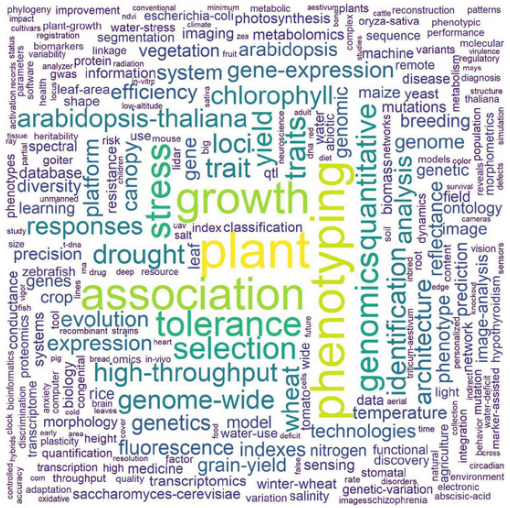
27th June 2023
Institut Agro Rennes-Angers, B202

Table of contents

IPPN : Napari as a tool for phenotyping

Deep learning: increasingly used





Current state of phenomics research (left) and 500 most used keywords from the papers presented in the left panel (right)

Lürig, Moritz D., et al, 2021

Two plateforms for deep learning models

deepImageJ

DeepImageJ

Impossible to customise the plugin

Google Colaboratory

Google's right to read and use personal data

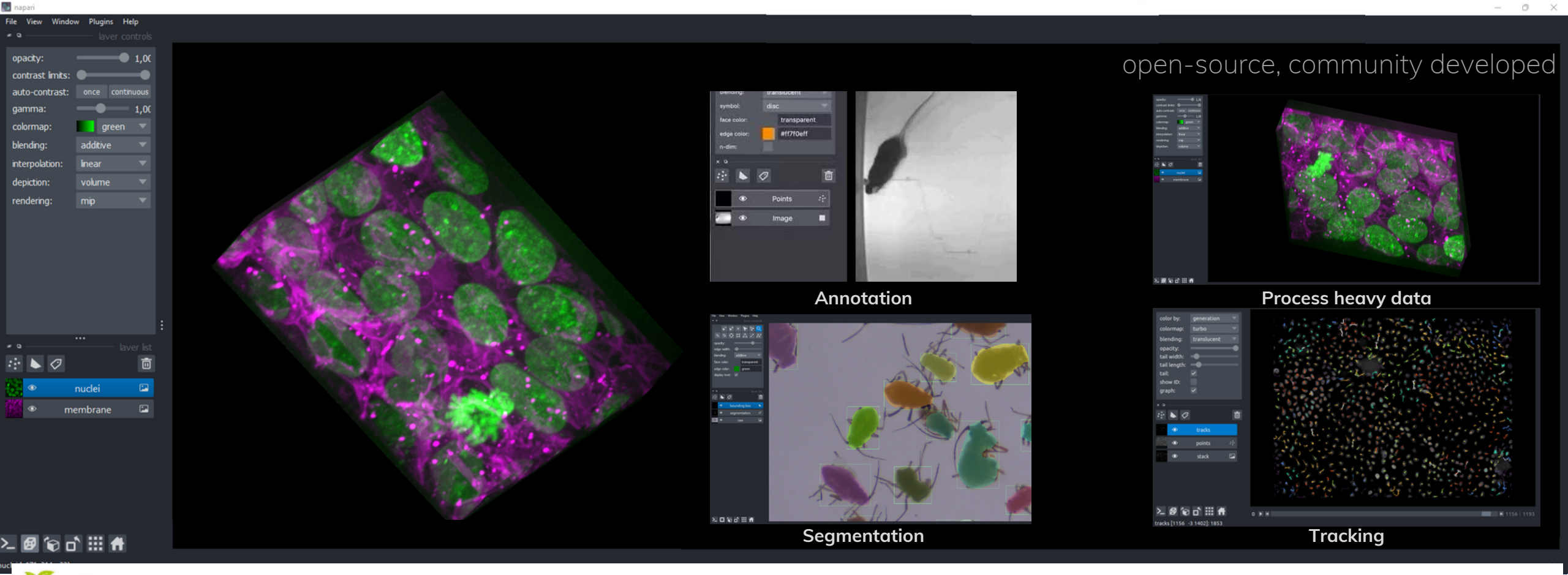
Programming skills required

IPPN : Napari as a tool for phenotyping

napari

n-dimensional data viewer in Python

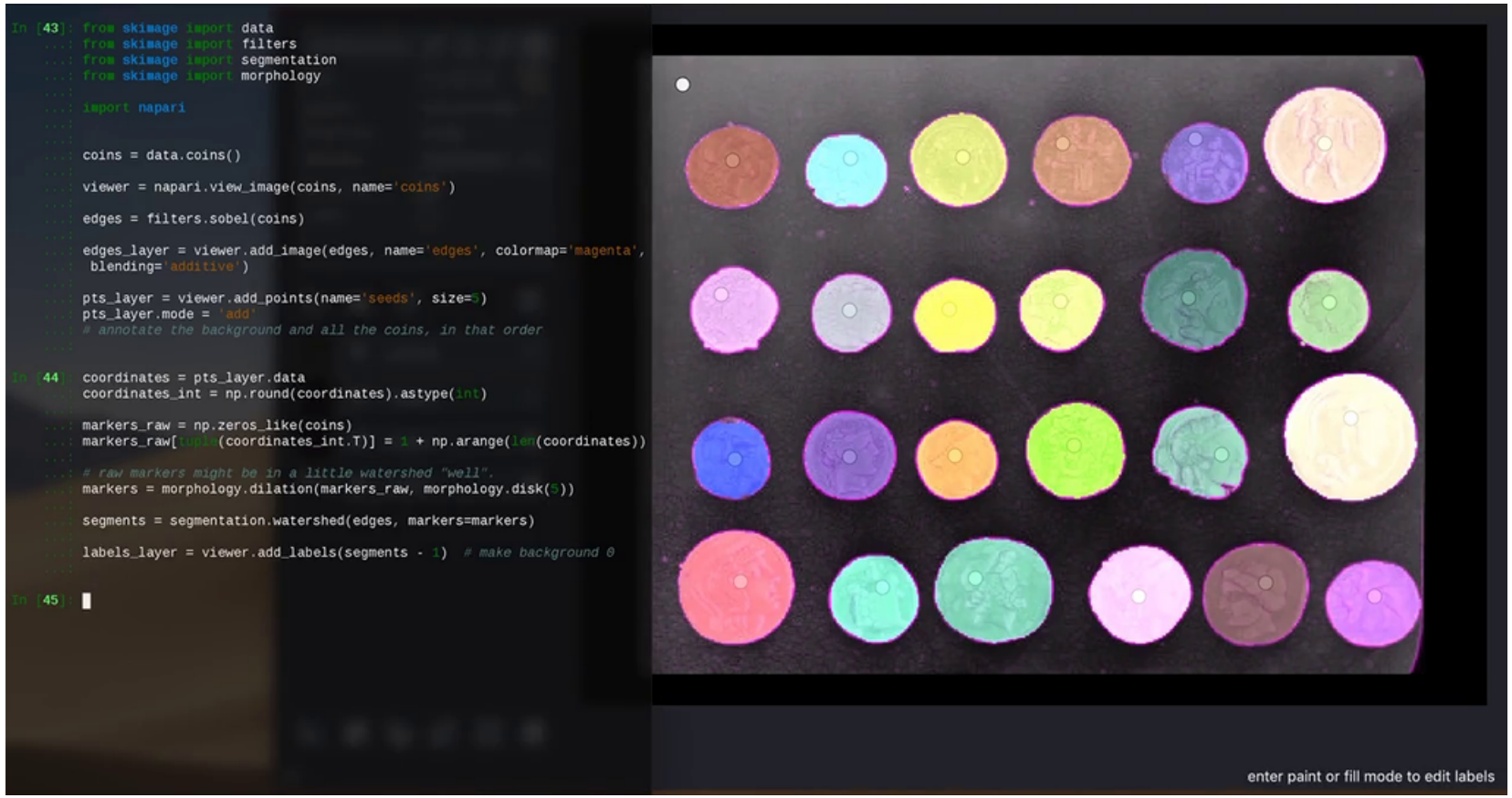
open-source, community developed



IPPN : Napari as a tool for phenotyping

napari

Combining interactive annotation and segmentation algorithms



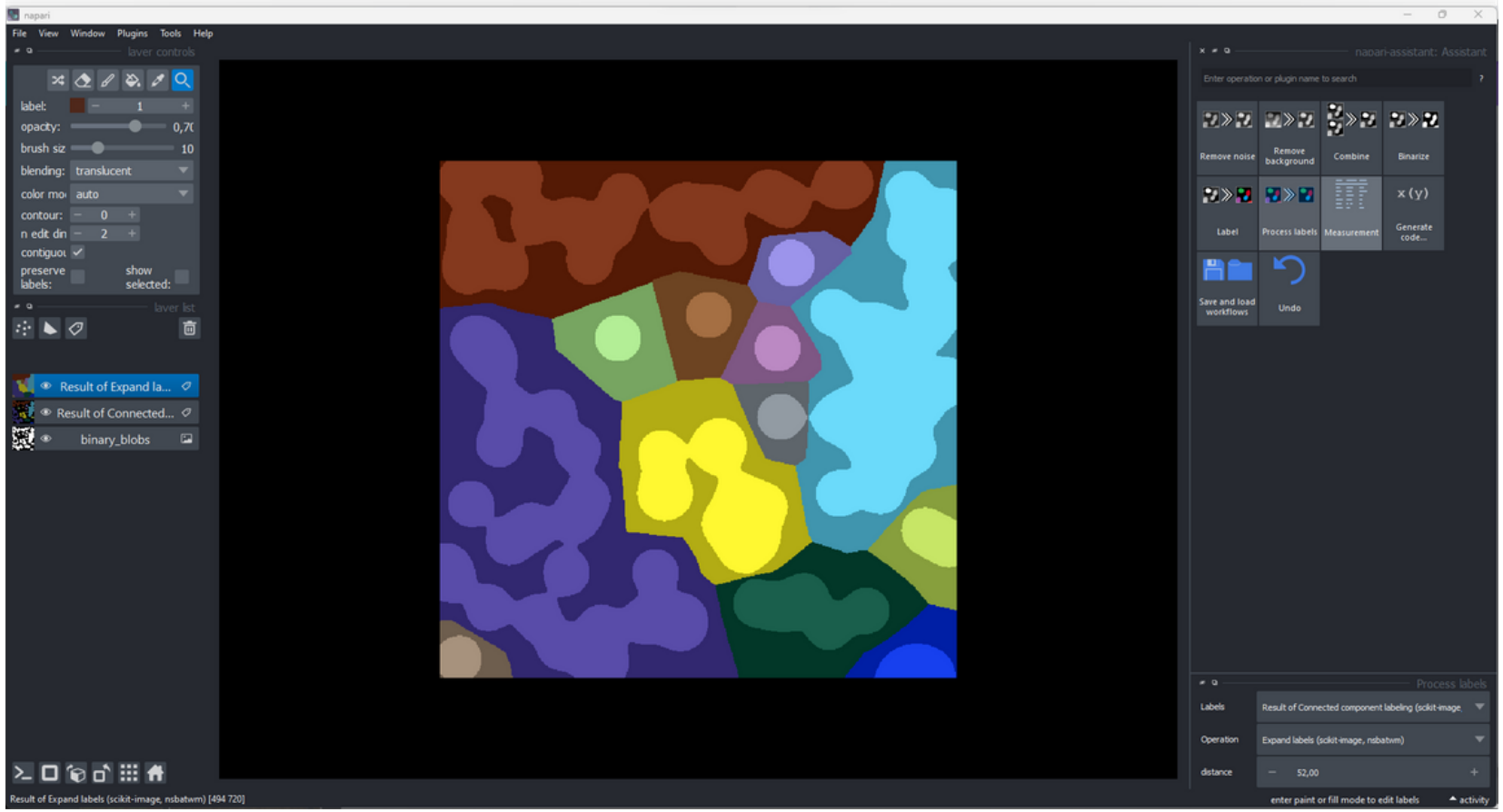
IPPN : Napari as a tool for phenotyping

napari

Combining interactive annotation and segmentation algorithms

Annotate label and process labels

Plugin: napari_pyclesperanto_assitant



IPPN : Napari as a tool for phenotyping

napari

Combining interactive annotation and segmentation algorithms

Annotate label and process labels

Deep-Learning (denoising, cell+nuclei segmentation)

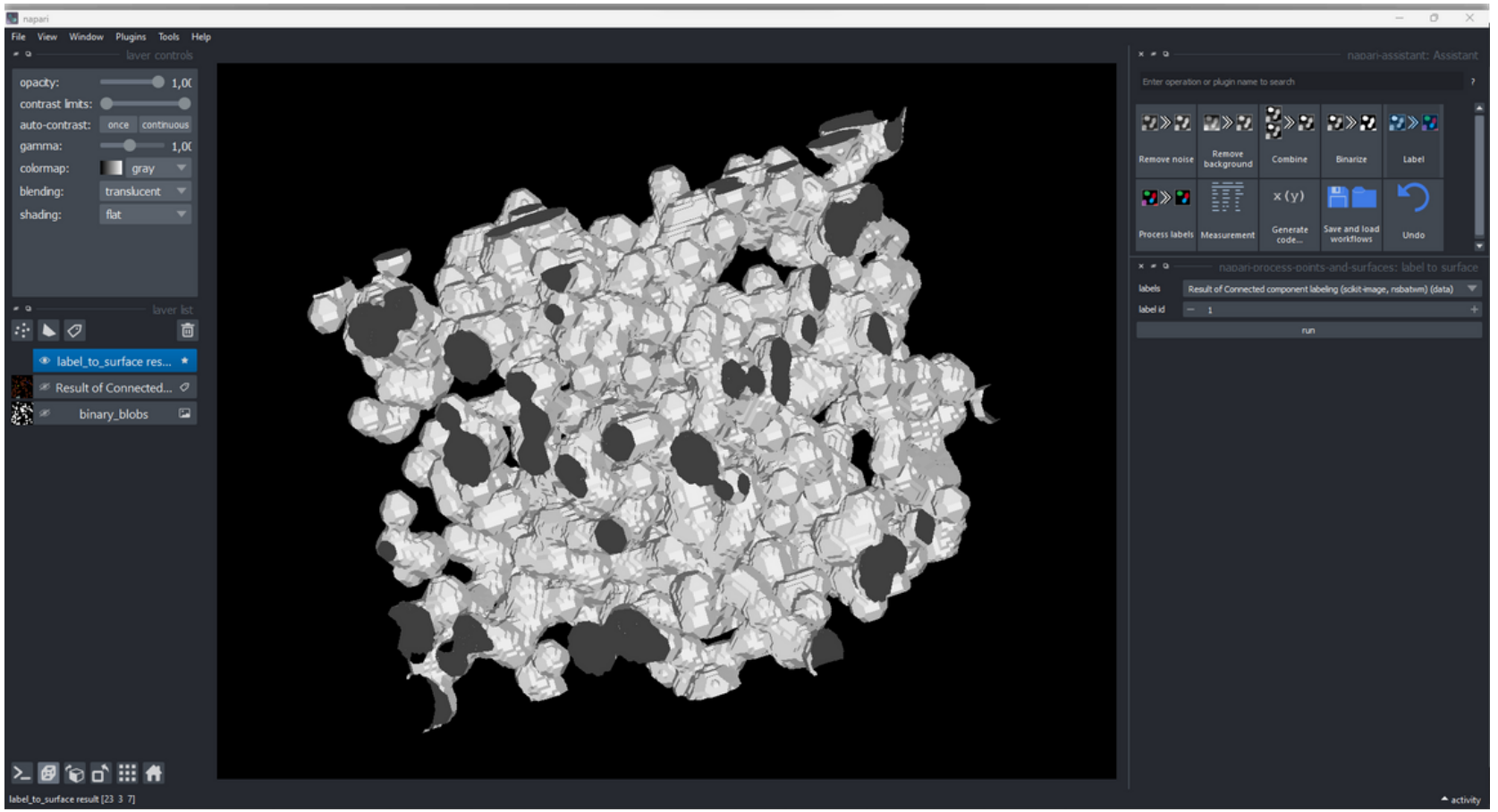
Plugin: napari-n2v

IPPN : Napari as a tool for phenotyping

napari

- Combining interactive annotation and segmentation algorithms
- Annotate label and process labels
- Deep-Learning (denoising, cell+nuclei segmentation)
- Surface extraction & analysis

Plugin: napari-process-points-and-surfaces



IPPN : Napari as a tool for phenotyping

napari

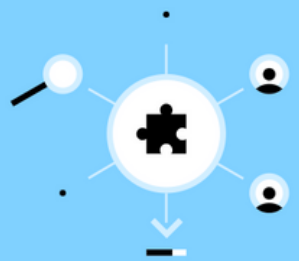
326 plugins

napari **hub**

Plugins

Collections

Discover, install, and share napari plugins



- Discover plugins that solve your image analysis challenges
- Learn how to install into napari

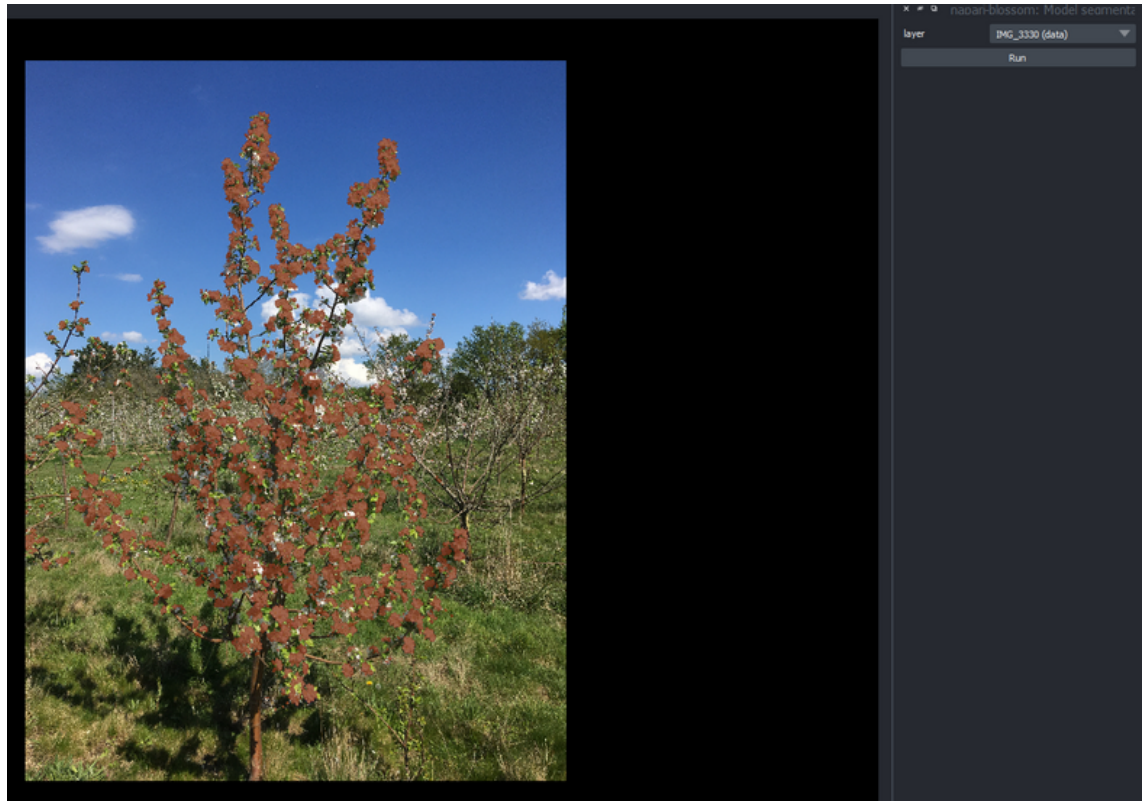
Share your image analysis tools with napari's growing community

Search for a plugin by keyword or author

IPPN : Napari as a tool for phenotyping

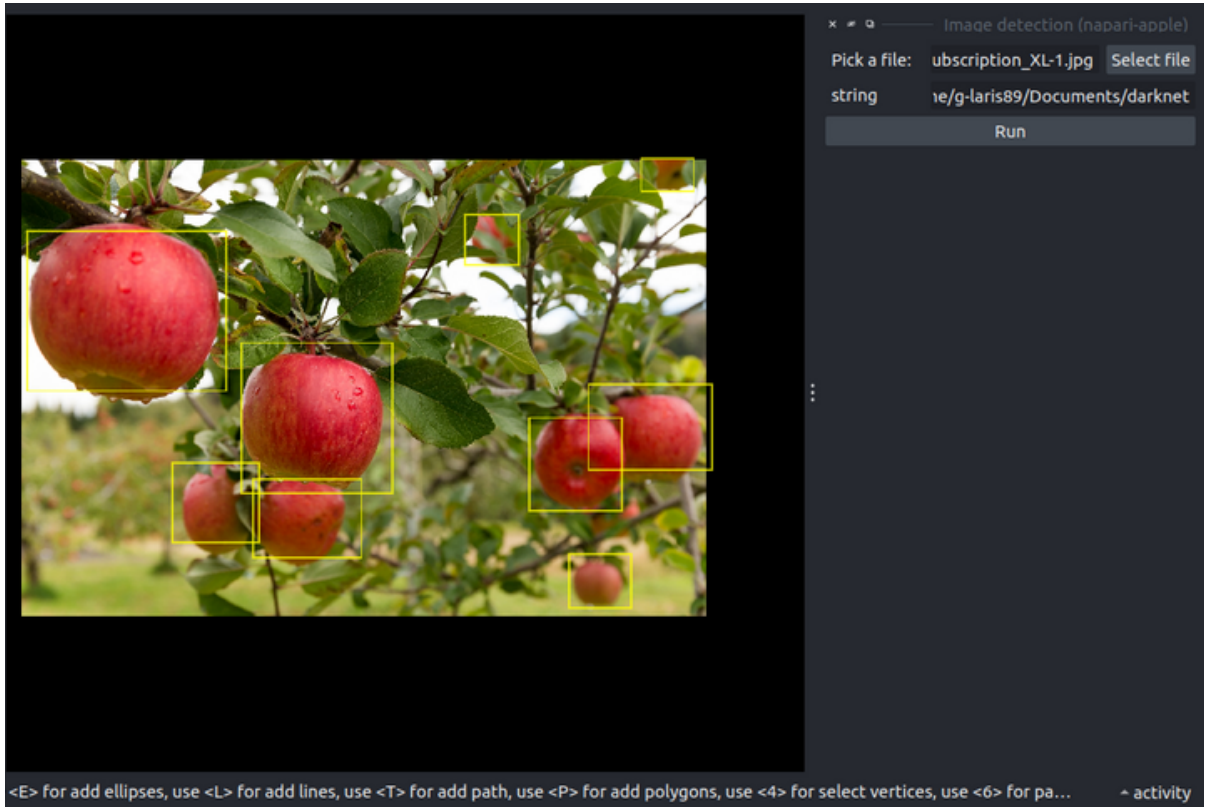
napari

Plugin: napari-blossom



Detection of apple flowering

Plugin: napari-apple



Detection of apple

IPPN : Napari as a tool for phenotyping

napari

Which tools can be included in a plugin ?

Reader

Add instructions for special input data

Match a set of incomplete ground truths to an image sequence

RGB image sequence:

Ground truth image sequence:

Widget

Add instructions for processing data with a user interface

Applying a deep learning model on RGB image sequence

RGB image sequence:

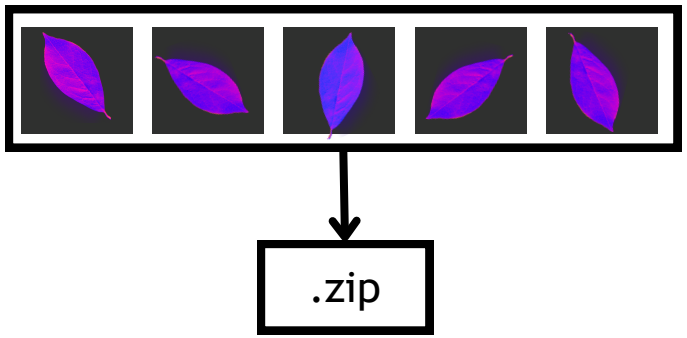
Mask image sequence:

Writer

Add instructions for special ouput data

Save image sequence in compressed file

Mask image sequence:



Plugin: workshop-demo

IPPN : Napari as a tool for phenotyping

napari

Which tools can be included in a plugin ?

Reader

Add instructions for special input data

Match a set of incomplete ground truths to an image sequence

RGB image sequence:

Ground truth image sequence:

Widget

Add instructions for processing data with a user interface

Applying a deep learning model on RGB image sequence

RGB image sequence:

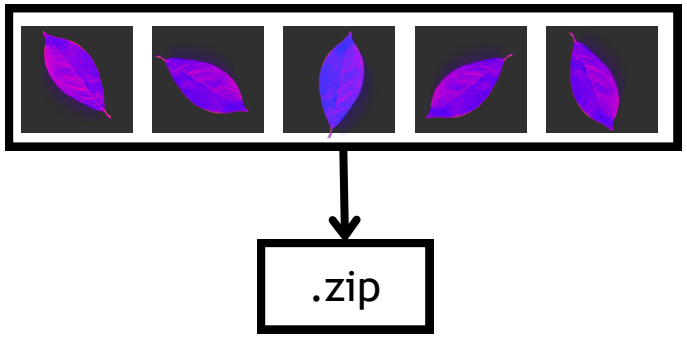
Mask image sequence:

Writer

Add instructions for special ouput data

Save image sequence in compressed file

Mask image sequence:



Plugin: workshop-demo

IPPN : Napari as a tool for phenotyping

napari

- Create your plugin package
- Import the codes into the widget.py file
- Connect widget code to napari
- Add dependencies in metadata
- Add some test
- Deploy

How to design a plugin and a widget ?

Generate minimal napari plugin repository

`cookiecutter https://github.com/napari/cookiecutter-napari-plugin`

```
full_name [Napari Developer]: guest-0000
email [yourname@example.com]: guest-0000@gmail.com
github_username_or_organization [githubuser]: guest-0000_pizalliolo
plugin_name [napari-foobar]: napari-thresholds
Select github_repository_url:
1 - https://github.com/guest-0000_pizalliolo/napari-thresholds
2 - provide later
Choose from 1, 2 [1]:
module_name [napari_thresholds]: napari_thresholds
display_name [napari FooBar]: Thresholds
short_description [A simple plugin to use with napari]: Several thresholds available
include_reader_plugin [y]: n
include_writer_plugin [y]: n
include_sample_data_plugin [y]: n
include_dock_widget_plugin [y]: y
use_git_tags_for_versioning [n]: n
install_precommit [n]: n
Select license:
1 - BSD-3
2 - MIT
3 - Mozilla Public License 2.0
4 - Apache Software License 2.0
5 - GNU LGPL v3.0
6 - GNU GPL v3.0
Choose from 1, 2, 3, 4, 5, 6 (1, 2, 3, 4, 5, 6) [1]: 1
```

IPPN : Napari as a tool for phenotyping

napari

- Create your plugin package
- **Import the codes into the widget.py file**
- Connect widget code to napari
- Add dependencies in metadata
- Add some test
- Deploy

How to design a plugin and a widget ?

Write your code into a function and adapt to napari convention

`napari.types`

`ImageData`

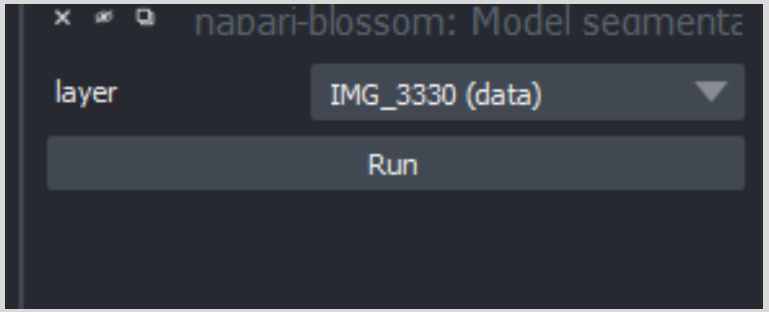
`code`

`napari.types`

`LabelsData`

`widget.py`

Use magicgui library to create user interface



IPPN : Napari as a tool for phenotyping

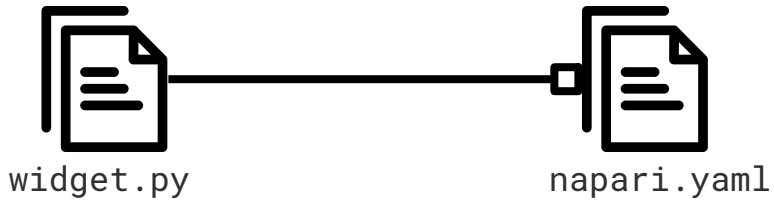
napari

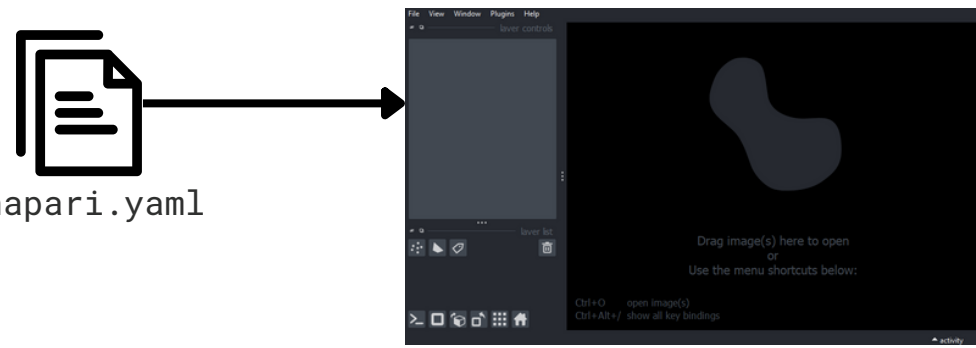
How to design a plugin and a widget ?

- Create your plugin package
- Import the codes into the widget.py file
- Connect widget code to napari
- Add dependencies in metadata
- Add some test
- Deploy

```yaml
name: napari-thresholds
display_name: Thresholds
contributions:
  commands:
    - id: napari-thresholds.my_widget #must be unique !
      python_name: napari_thresholds._widget:threshold_f
      title: Thresholds
  widgets:
    - command: napari-thresholds.my_widget #identity backend
      display_name: Thresholds
```

napari.yaml





IPPN : Napari as a tool for phenotyping

napari

- Create your plugin package
- Import the codes into the widget.py file
- Connect widget code to napari
- **Add dependencies in metadata**
- Add some test
- Deploy

How to design a plugin and a widget ?

```
[options]
packages = find:
install_requires =
    numpy
    magicgui
    qtpy
    scikit-image
    napari

python_requires = >=3.8
include_package_data = True
package_dir =
    =src

# add your package requirements here

[options.packages.find]
where = src

[options.entry_points]
napari.manifest =
    napari-thresholds = napari_thresholds:napari.yaml
```

Determine dependencies

Determine the repository containing codes

Determine napari-thresholds is napari plugin

setup.cfg

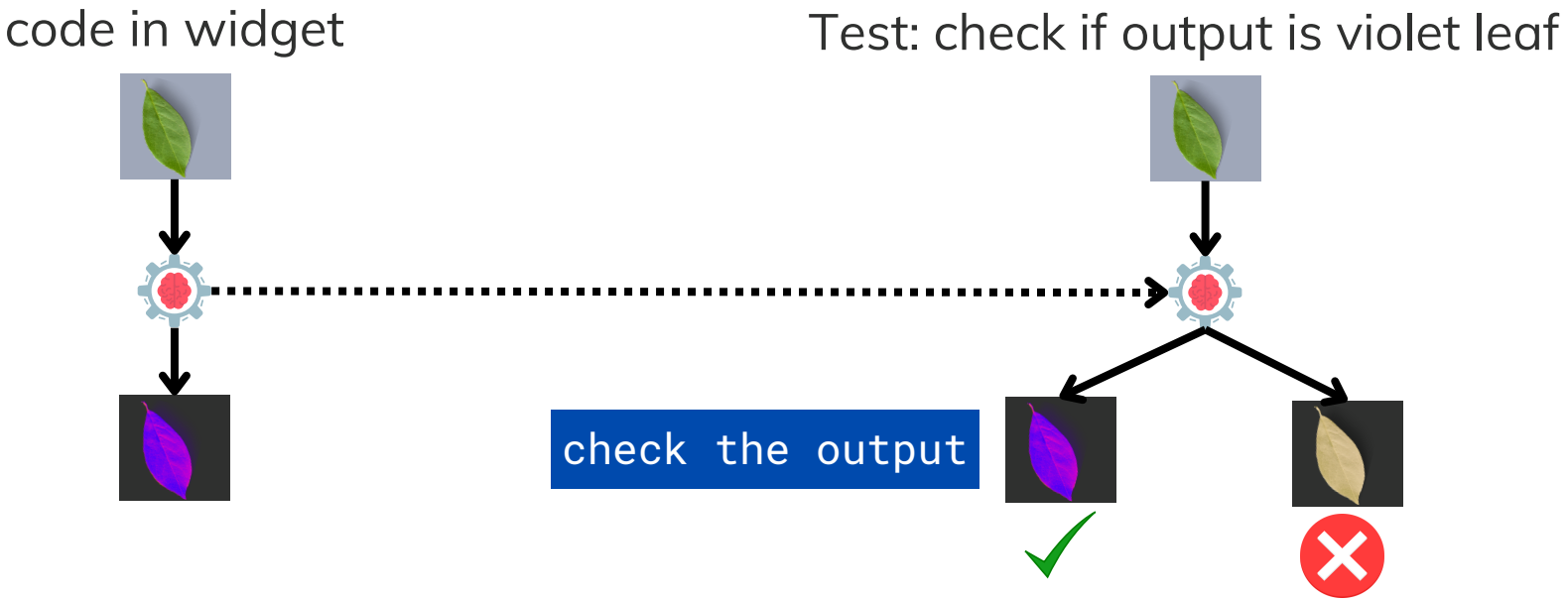
IPPN : Napari as a tool for phenotyping

napari

- Create your plugin package
- Import the codes into the widget.py file
- Connect widget code to napari
- Add dependencies in metadata
- **Add some test**
- Deploy

How to design a plugin and a widget ?

Instruction to be sure the widget works well whatever the change made



test_widget.py

IPPN : Napari as a tool for phenotyping

napari

- Create your plugin package
- Import the codes into the widget.py file
- Connect widget code to napari
- Add dependencies in metadata
- Add some test
- **Deploy**

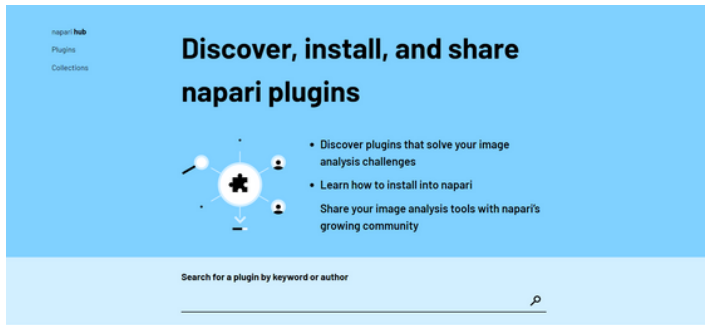
How to design a plugin and a widget ?

Requirements:

python Package Index PyPI

GitHub

1. Add napari project in GitHub (public access)
2. Generate API token
3. Add API token in GitHub as secret key
4. Create a build in napari-thresholds folder
5. Upload package to the PyPI



Exercise

napari

Go to this page: `https://github.com/hereariim/IPPN_napari`