



DEEP NAPARI

Getting started with napari in phenotyping

Herearri METUAREA, Engineer University of Angers, France

15th November 2023

Belambra clubs & hôtels, BAS-MADRAGUE



INRAe

IRHS
Institut de Recherche en
Horticulture et Semences



**université
angers**



For this session:

- Short introduction to Napari
- Getting started with Napari



Short introduction to Napari

Objective:

Getting to know Napari

Getting to know Napari plugin

Getting to know Napari widget

Table of contents

Context

Napari

Plugins

Widget

IPPN : Napari as a tool for phenotyping

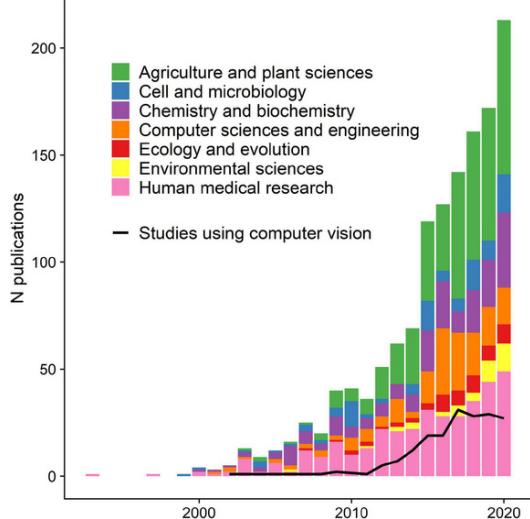
Context

Napari

Plugins

Widget

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 platforms for deep learning models



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

Context

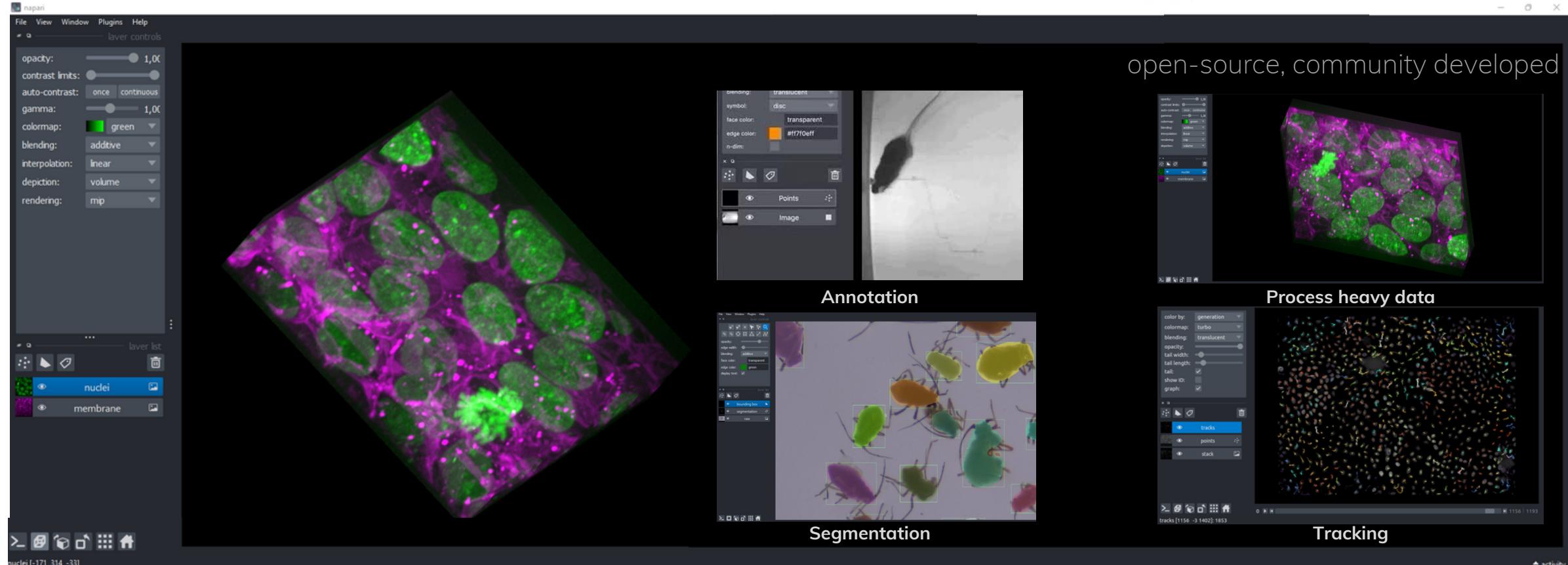
Napari

Plugins

Widget



napari



INRAe



Institut de Recherche en
Horticulture et Semences



IPPN : Napari as a tool for phenotyping

Context

Napari

Plugins

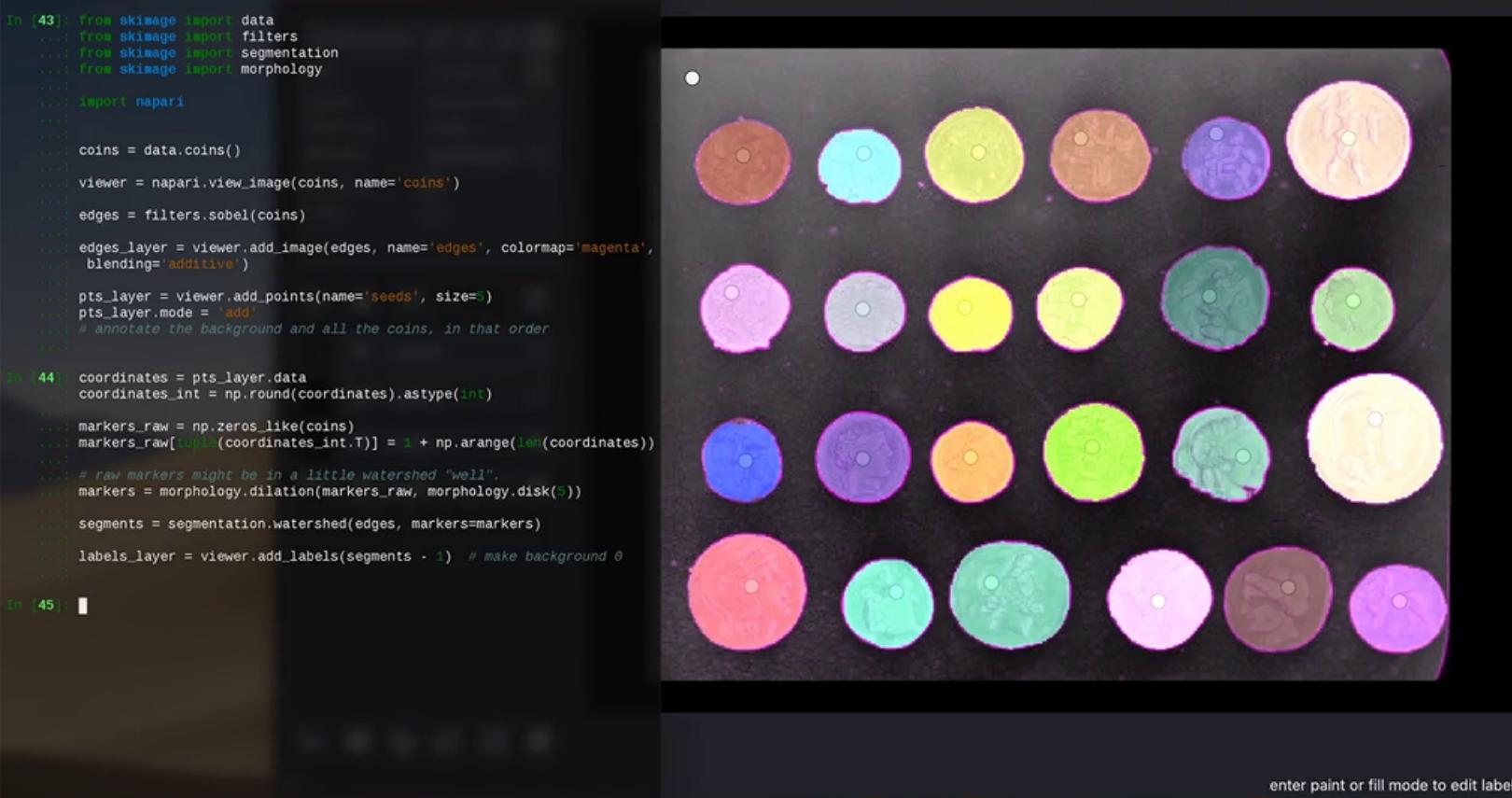
Widget



napari

Combining interactive annotation and segmentation algorithms

```
In [43]: from skimage import data
... from skimage import filters
... from skimage import segmentation
... from skimage import morphology
...
... import napari
...
...
coins = data.coins()
viewer = napari.view_image(coins, name='coins')
edges = filters.sobel(coins)
edges_layer = viewer.add_image(edges, name='edges', colormap='magenta',
blending='additive')
pts_layer = viewer.add_points(name='seeds', size=5)
pts_layer.mode = 'add'
# annotate the background and all the coins, in that order
...
In [44]: coordinates = pts_layer.data
coordinates_int = np.round(coordinates).astype(int)
...
markers_raw = np.zeros_like(coins)
markers_raw[tuple(coordinates_int.T)] = 1 + np.arange(len(coordinates))
...
# raw markers might be in a little watershed "well".
markers = morphology.dilation(markers_raw, morphology.disk(5))
...
segments = segmentation.watershed(edges, markers=markers)
...
labels_layer = viewer.add_labels(segments - 1) # make background 0
...
In [45]:
```





IPPN : Napari as a tool for phenotyping

Context

Napari

Plugins

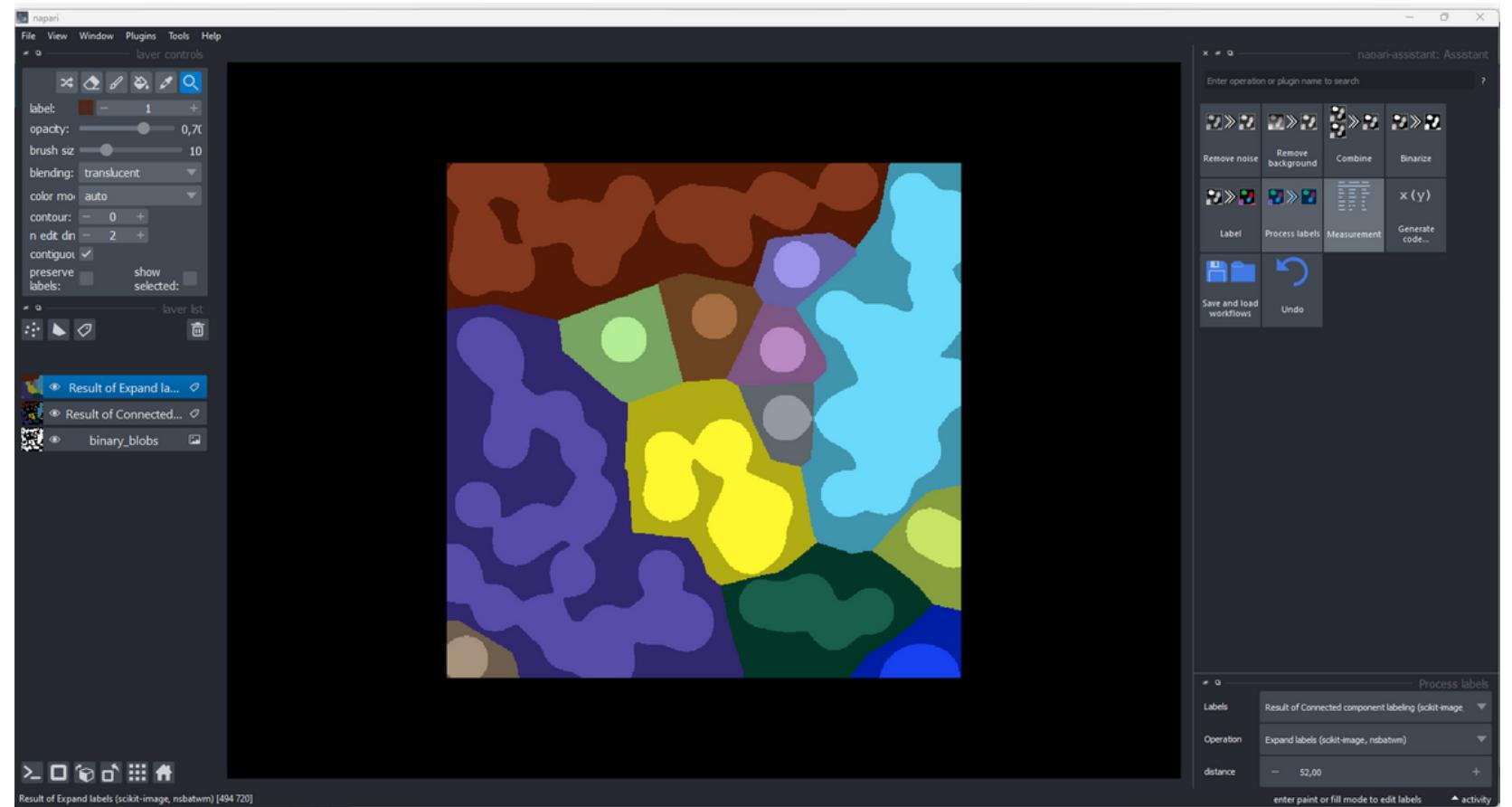
Widget



napari

Combining interactive annotation and segmentation algorithms

Annotate label and process labels



Plugin: napari_pyclesperanto_assistant



INRAe

IRHS
Institut de Recherche en
Horticulture et Semences



**université
angers**

IPPN : Napari as a tool for phenotyping

Context

Napari

Plugins

Widget



napari

Combining interactive annotation and segmentation algorithms

Annotate label and process labels

Deep-Learning (denoising, cell+nuclei segmentation)



IPPN : Napari as a tool for phenotyping

Context

Napari

Plugins

Widget

Plugin: napari-process-points-and-surfaces



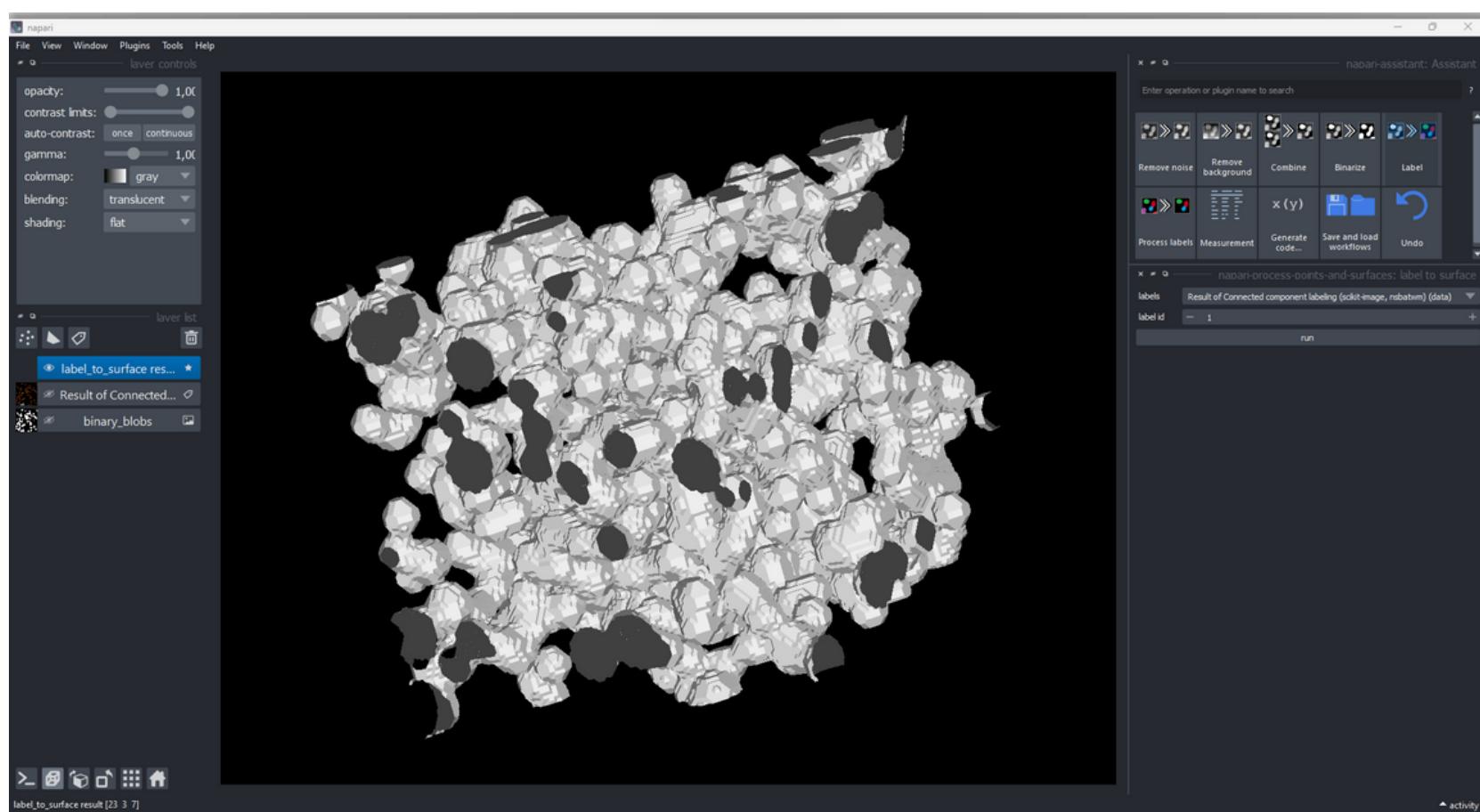
napari

Combining interactive annotation and segmentation algorithms

Annotate label and process labels

Deep-Learning (denoising, cell+nuclei segmentation)

Surface extraction & analysis



INRAe

IRHS
Institut de Recherche en
Horticulture et Semences



**université
angers**

Context

Napari

Plugins

Widget

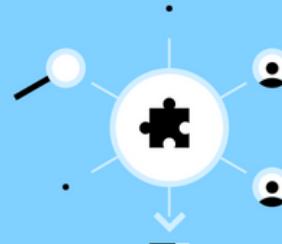


napari

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

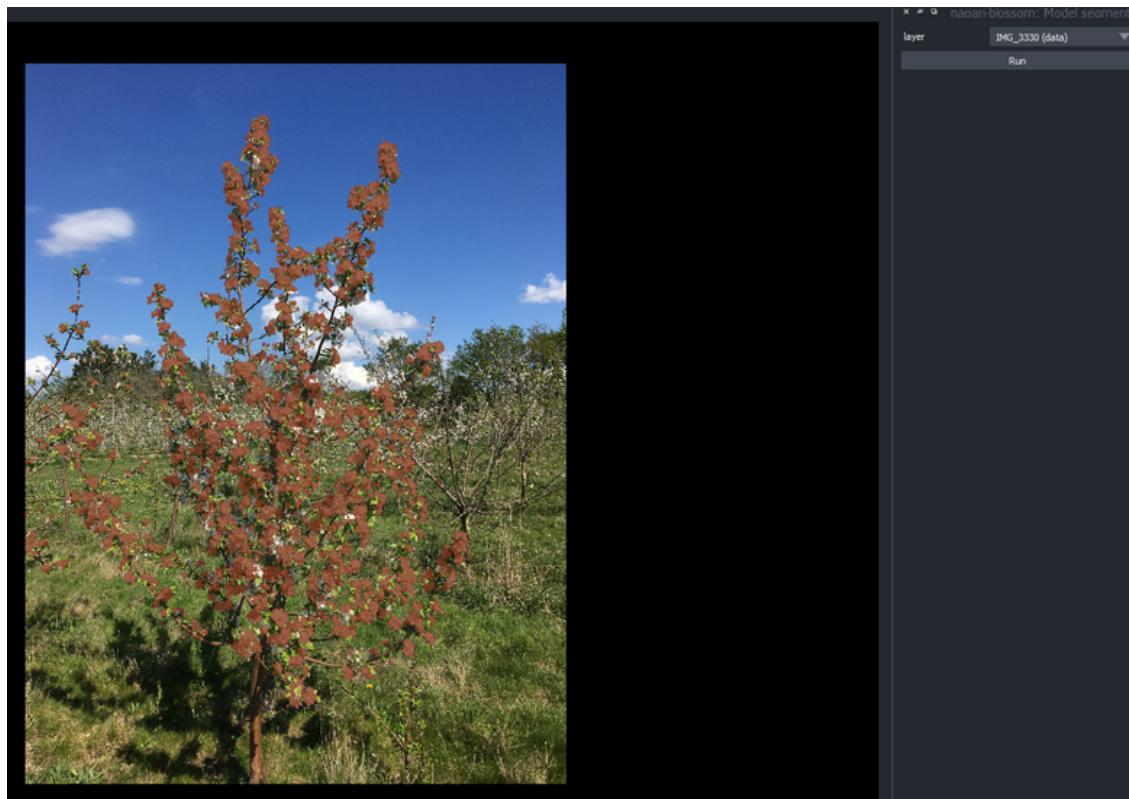
Search for a plugin by keyword or author 

IPPN : Napari as a tool for phenotyping

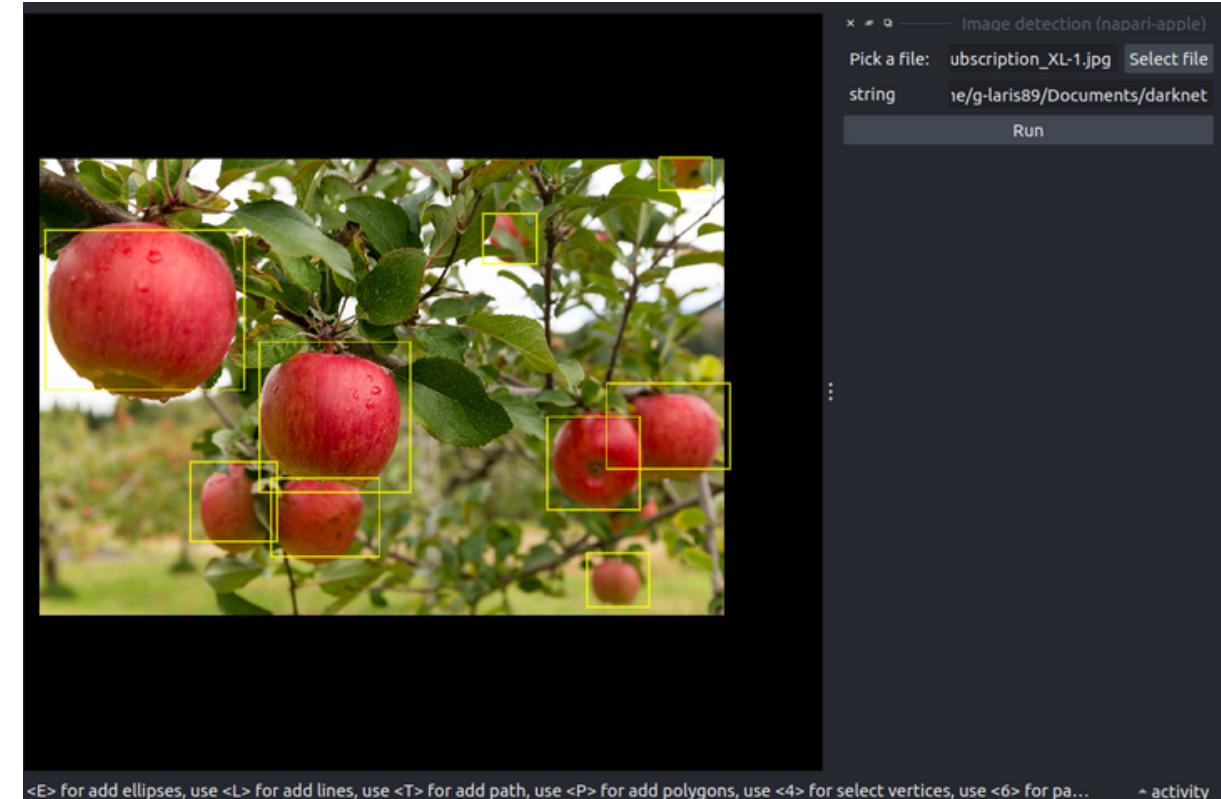
[Context](#)[Napari](#)[Plugins](#)[Widget](#)

napari

Plugin: napari-blossom



Detection of apple flowering

**INRAe**

Detection of apple

Context

Napari

Plugins

Widget



napari

Which tools can be included in a plugin ?

Reader

Widget

Writer

Add instructions for special input data

Add instructions for processing data
with a user interfaceAdd instructions for special output
data

Match a set of incomplete ground truths to
an image sequence

Applying a deep learning model on RGB
image sequence

Save image sequence in compressed file

RGB image sequence:



RGB image sequence:



Mask image sequence:



Ground truth image sequence:



Mask image sequence:



n: workshop-demo

Context

Napari

Plugins

Widget



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

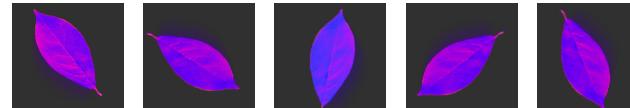
Add instructions for processing data with a user interface

Applying a deep learning model on RGB image sequence

RGB image sequence:



Mask image sequence:



Writer

Add instructions for special output data

Save image sequence in compressed file

Mask image sequence:



.zip



INRAe



n: workshop-demo
université angers

Context

Napari

Plugins

Widget



napari

Create your plugin package

Import the codes into the
widget.py fileConnect widget code to
napariAdd 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_pizalliol
plugin_name [napari-foobar]: napari-thresholds
Select github_repository_url:
1 - https://github.com/guest-0000_pizalliol/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
```

Context

Napari

Plugins

Widget



napari

Create your plugin package

Import the codes into the
widget.py fileConnect widget code to
napariAdd 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

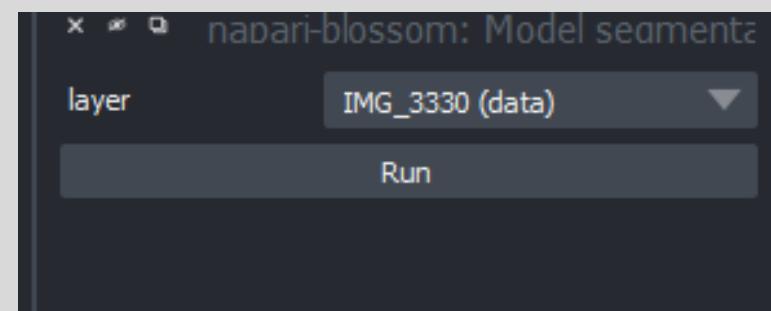
code

napari.types

LabelsData



Use magicgui library to create
user interface



Context

Napari

Plugins

Widget



napari

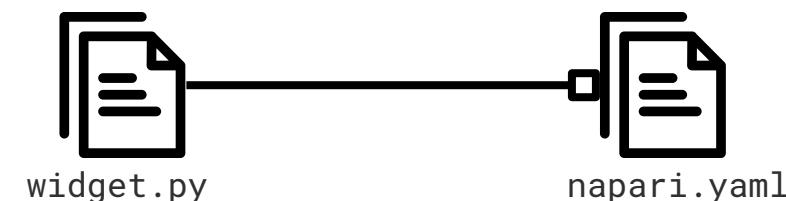
How to design a plugin and a widget ?

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**

```
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

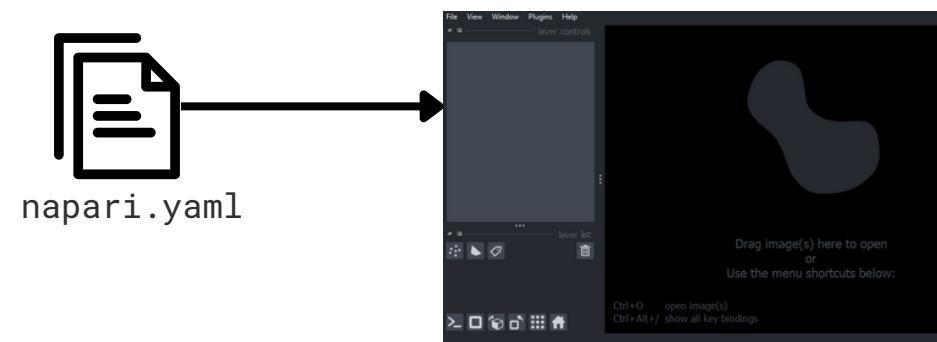


widget.py

napari.yaml



napari.yaml



Context

Napari

Plugins

Widget



napari

How to design a plugin and a widget ?

Create your plugin package

Import the codes into the
widget.py fileConnect widget code to
napariAdd dependencies in
metadata

Add some test

Deploy



```
[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 codesDetermine napari-thresholds is
napari plugin

Context

Napari

Plugins

Widget



napari

Create your plugin package

Import the codes into the
widget.py fileConnect widget code to
napariAdd dependencies in
metadata

Add some test

Deploy

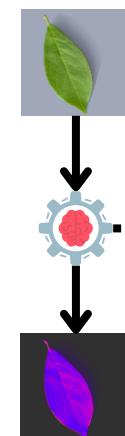
How to design a plugin and a widget ?

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

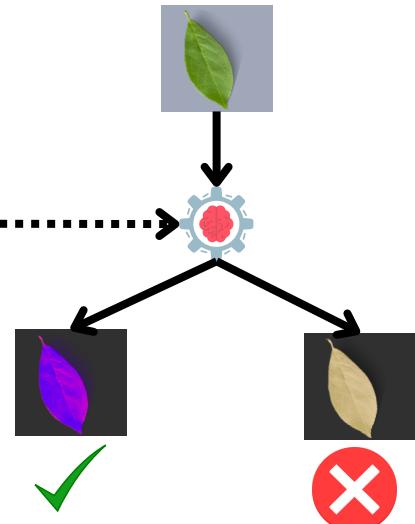
code in widget



test_widget.py



check the output



Test: check if output is violet leaf



[Context](#)[Napari](#)[Plugins](#)[Widget](#)

napari

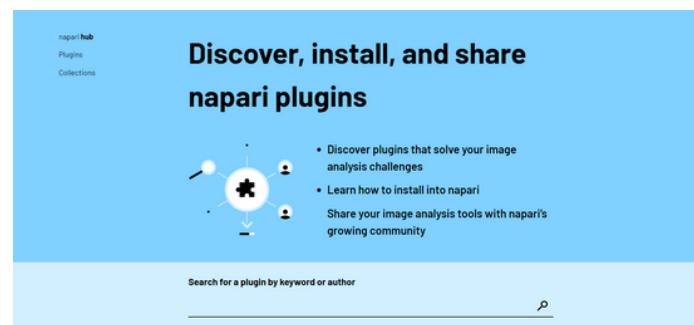
[Create your plugin package](#)[Import the codes into the
widget.py file](#)[Connect widget code to
napari](#)[Add dependencies in
metadata](#)[Add some test](#)[Deploy](#)

How to design a plugin and a widget ?

Requirements:



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

**INRAe**
Institut de Recherche en
Horticulture et Semences
université
angers



Exercise : Getting started with Napari

Context

Napari

Plugins

Widget



napari

Go to this page: <https://github.com/hereariim/DEEP-NAPARI>

hereariim / DEEP-NAPARI

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

DEEP-NAPARI Public

main · 1 branch · 0 tags

Go to file Add file Code

herearilim readme update 4581128 · 9 minutes ago 4 commits

- Exercise-1,_Getting started import all data 20 minutes ago
- Exercise-2,_Getting started plugin import all data 20 minutes ago
- Exercise-3,_My first plugin import all data 20 minutes ago
- Exercise-4,_My first widget import all data 20 minutes ago
- Exercise-5,_Do it yourself import all data 20 minutes ago
- Extra-Deploy_in_builtin import all data 20 minutes ago
- images-credit logo update 10 minutes ago
- Introduction.pdf import all data 20 minutes ago
- README.md readme update 9 minutes ago

About

This is a training tutorial to learn how to integrate deep learning model into napari plugin from scratch

Readme Activity 0 stars 1 watching 0 forks

Releases

No releases published Create a new release

Packages

No packages published Publish your first package

Languages

Jupyter Notebook 90.4% Python 9.6%

Suggested Workflows

Based on your tech stack

DEEP NAPARI : Napari as a tool for phenotyping

The diagram shows the following logos at the top of the page:

- Imabio
- CNRS
- Inserm
- FRANCE-BIOIMAGING
- GDR FONDÉS
- INSTITUT FRESNEL



INRAe





Thanks you for your attention

