



Napari as a tool for deep learning project management

# NAPARI

Herearri METUAREA, Engineer, Université d'Angers, France

David ROUSSEAU, Full professor, Université d'Angers, France

5th February 2024

Virtual meeting



**ESRF**  
The European Synchrotron



**INRAe**

**IRHS**  
Institut de Recherche en  
Horticulture et Semences



**université  
angers**



Expert



PRO S2 (Weighted Loss) Unet\_segmentation\_notebook.ipynb ☆

Fichier Modifier Affichage Insérer Exécution Outils Aide Dernière modification effectuée le 27 juillet

+ Code + Texte

▼ Split train data into train and validation

```
[ ] from sklearn.model_selection import train_test_split
x_train, x_val, y_train, y_val = train_test_split(X_train, Y_train, test_size=0.2, random_state=42) #text_
[ ] print('Train data shape is:',x_train.shape)

print('validation data shape is:',x_val.shape)

Train data shape is: (32, 256, 256, 3)
validation data shape is: (8, 256, 256, 3)

[ ] # Fiting the model
results = model.fit(x_train, y_train,
                     validation_data=(x_val,y_val),
                     batch_size=2, epochs=50,
                     callbacks=[early_stop,Model_check])

Epoch 11/50
16/16 [=====] - ETA: 0s - loss: 5.5026e-04 - dice_coefficient: 0.2432
Epoch 11: val_loss improved from 0.00069 to 0.00065, saving model to /content/gdrive/My Drive/data/best_mo
16/16 [=====] - 1s 44ms/step - loss: 5.5026e-04 - dice_coefficient: 0.2432 - val_
Epoch 12/50
```



Biologist



deepImageJ



NIS-element





Expert



CO PRO S2 (Weighted Loss) Unet\_segmentation\_notebook.ipynb ☆

Fichier Modifier Affichage Insérer Exécution Outils Aide Dernière modification effectuée le 27 juillet

+ Code + Texte

Split train data into train and validation

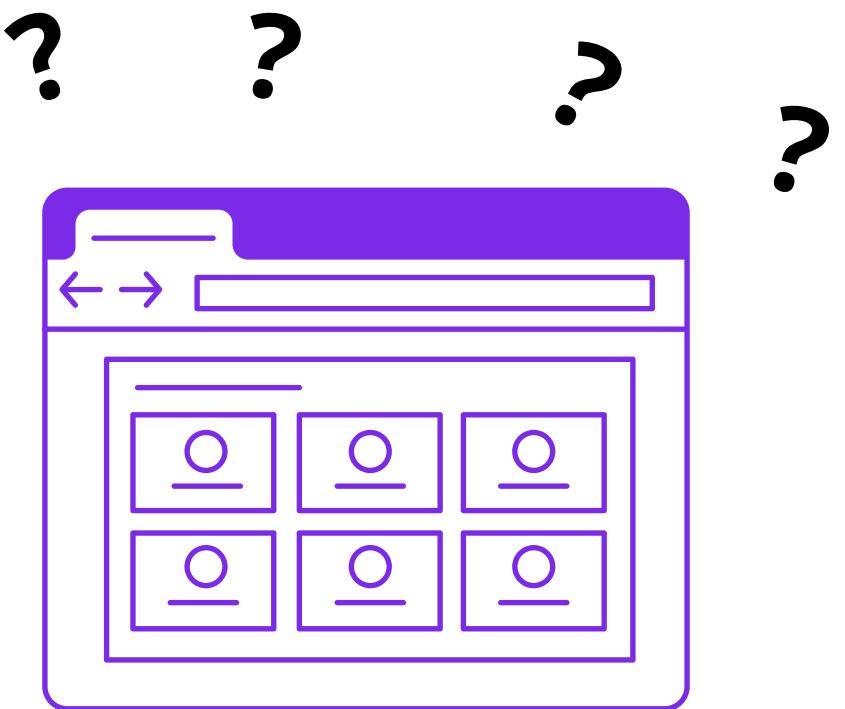
```
[ ] from sklearn.model_selection import train_test_split
x_train, x_val, y_train, y_val = train_test_split(X_train, Y_train, test_size=0.2, random_state=42) #text_
[ ] print('Train data shape is:',x_train.shape)

print('validation data shape is:',x_val.shape)

Train data shape is: (32, 256, 256, 3)
validation data shape is: (8, 256, 256, 3)

[ ] # Fiting the model
results = model.fit(x_train, y_train,
                     validation_data=(x_val,y_val),
                     batch_size=2, epochs=50,
                     callbacks=[early_stop,Model_check])

Epoch 11/50
16/16 [=====] - ETA: 0s - loss: 5.5026e-04 - dice_coefficient: 0.2432
Epoch 11: val_loss improved from 0.00069 to 0.00065, saving model to /content/gdrive/My Drive/data/best_mo
16/16 [=====] - 1s 44ms/step - loss: 5.5026e-04 - dice_coefficient: 0.2432 - val_
```



interface that  
satisfies  
everyone



Biologist





Expert



PRO Fichier Modifier Affichage Insérer Exécution Outils Aide Dernière modification effectuée le 27 juillet

+ Code + Texte

▼ Split train data into train and validation

```
[ ] from sklearn.model_selection import train_test_split
x_train, x_val, y_train, y_val = train_test_split(X_train, Y_train, test_size=0.2, random_state=42) #text_
[ ] print('Train data shape is:',x_train.shape)

print('validation data shape is:',x_val.shape)

Train data shape is: (32, 256, 256, 3)
validation data shape is: (8, 256, 256, 3)

[ ] # Fiting the model
results = model.fit(x_train, y_train,
                     validation_data=(x_val,y_val),
                     batch_size=2, epochs=50,
                     callbacks=[early_stop,Model_check])

Epoch 11/50
16/16 [=====] - ETA: 0s - loss: 5.5026e-04 - dice_coefficient: 0.2432
Epoch 11: val_loss improved from 0.00069 to 0.00065, saving model to /content/gdrive/My Drive/data/best_mo
16/16 [=====] - 1s 44ms/step - loss: 5.5026e-04 - dice_coefficient: 0.2432 - val_
Epoch 12/50
```



napari



Biologist



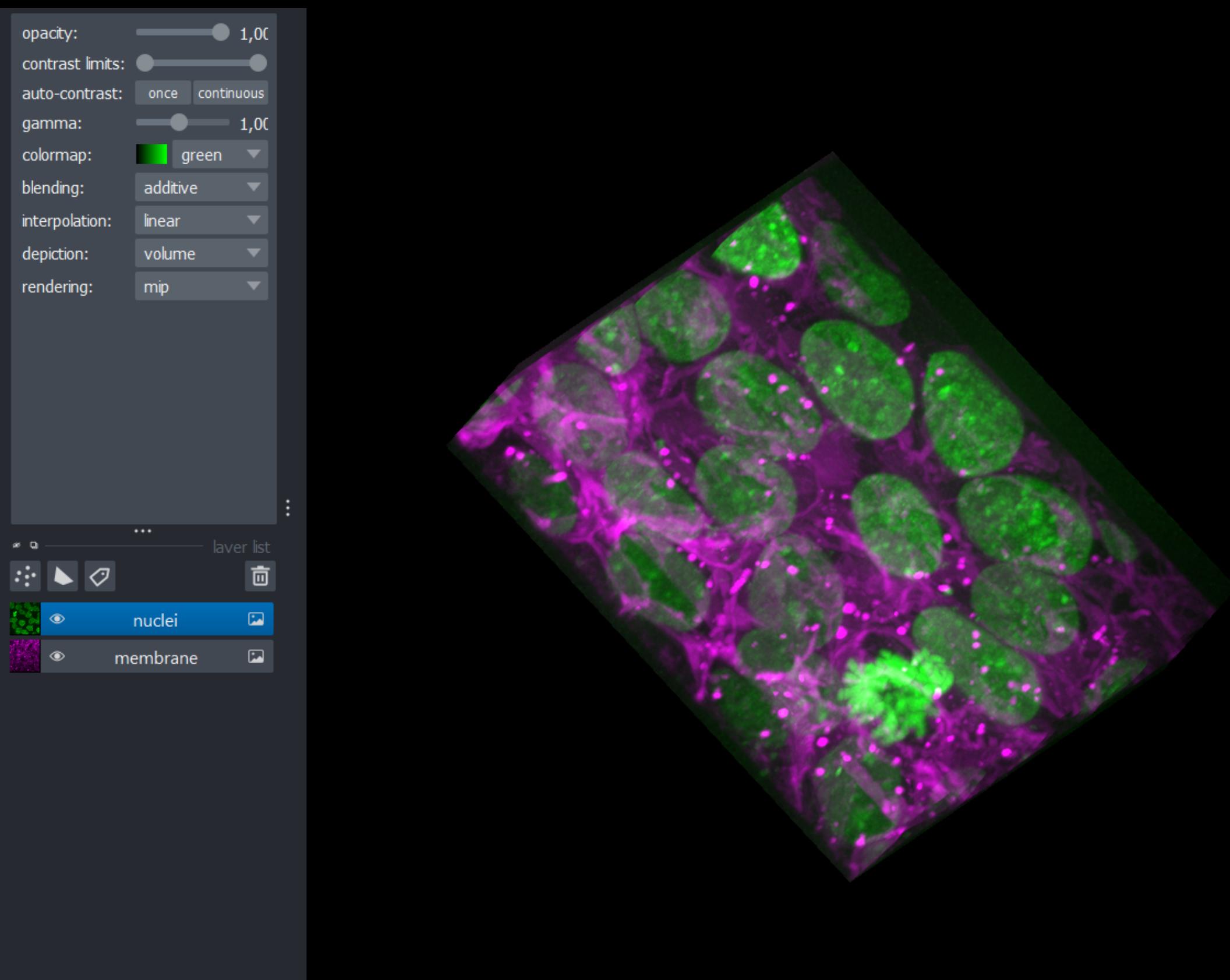
Nikon



NIS-element



napari

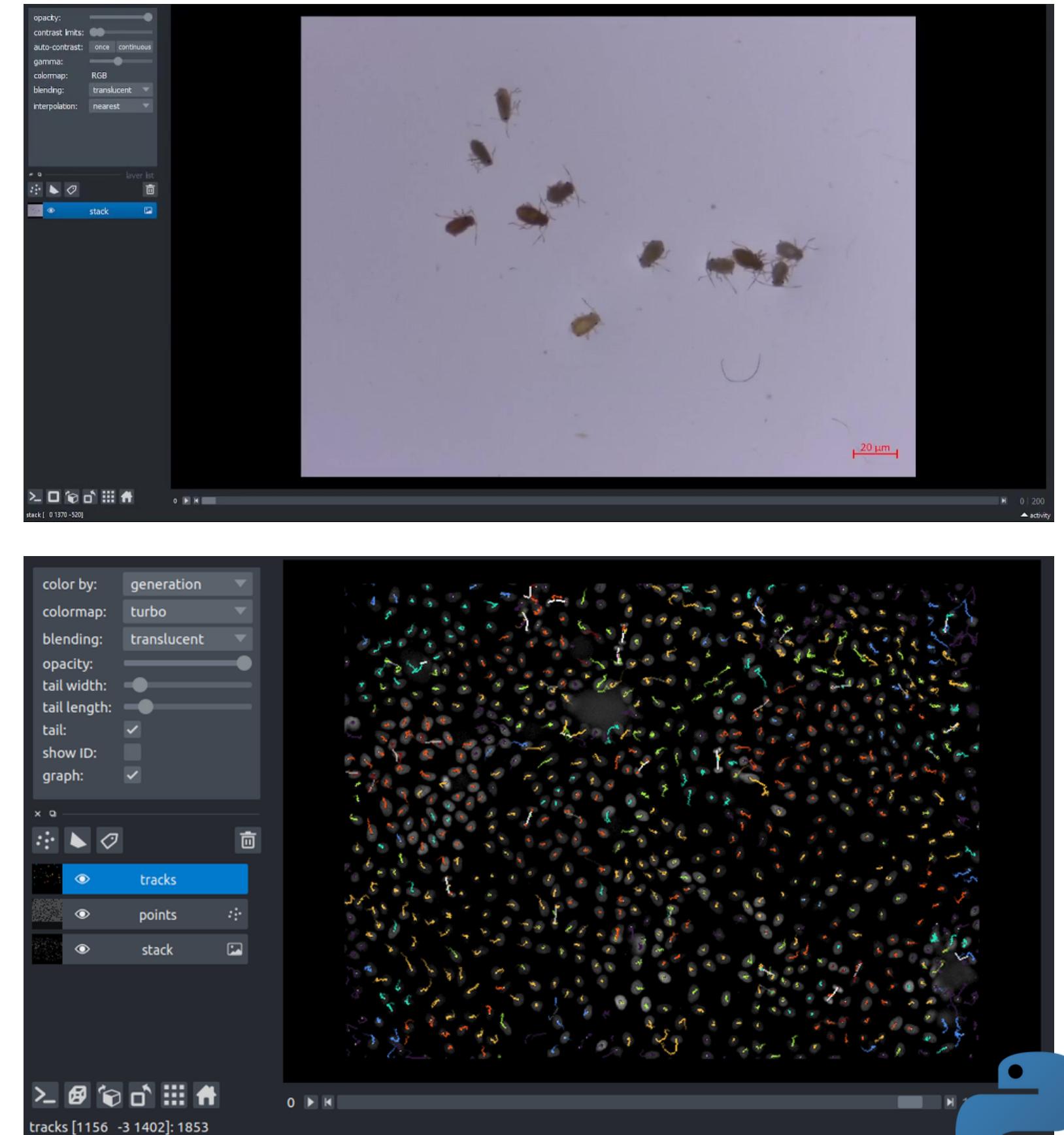
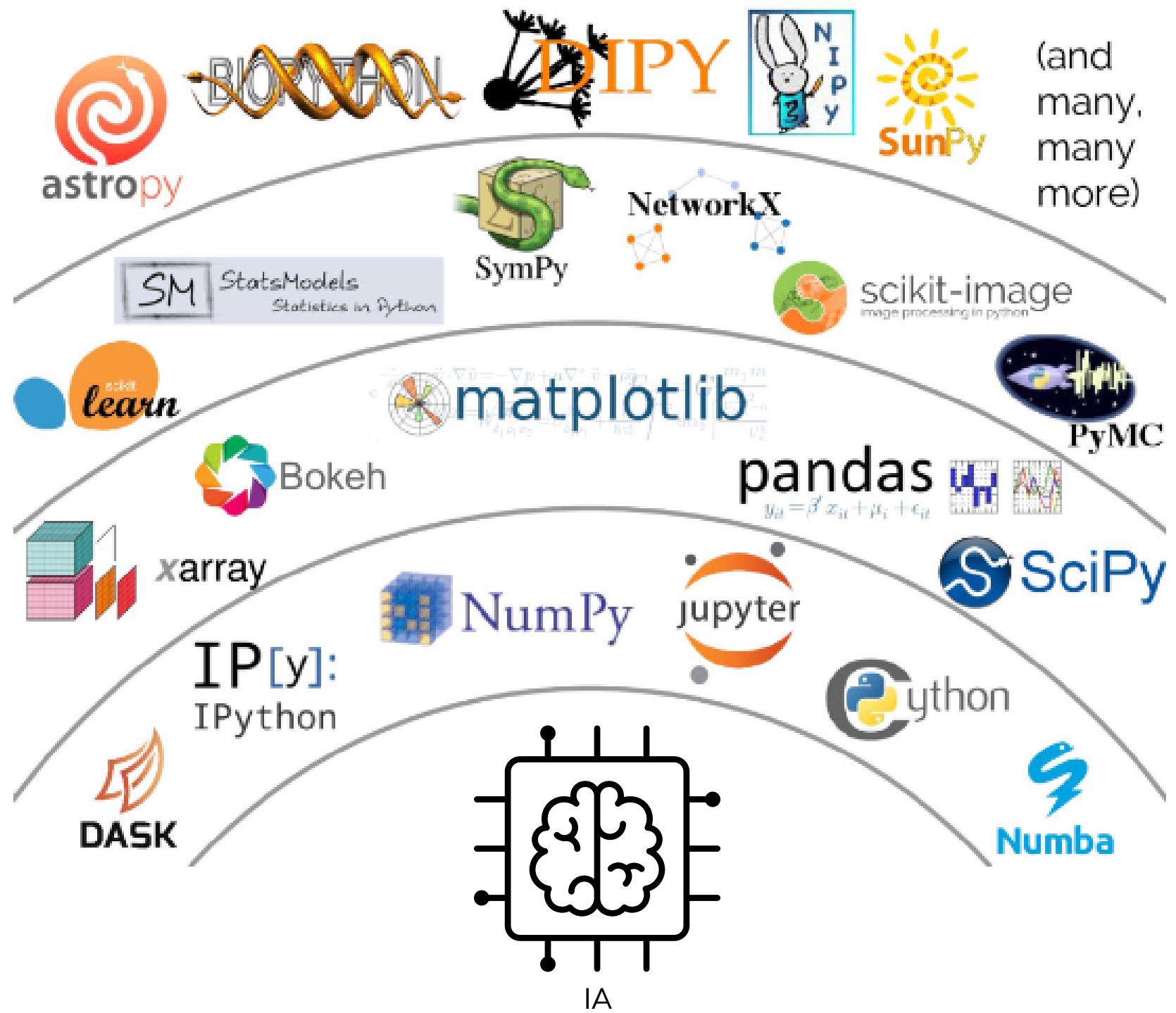


Help scientist to access Python's scientific ecosystem, with no prior coding experience

Multi-dimensional data viewer in Python  
open-source, community-developed

# Napari

Process large data and run Deep Learning (DL) model

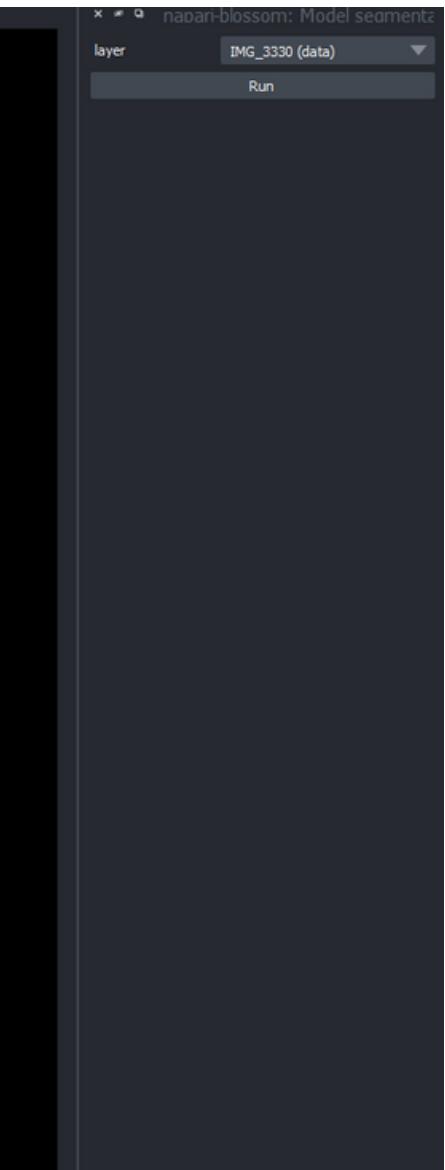


# Deep learning in Napari

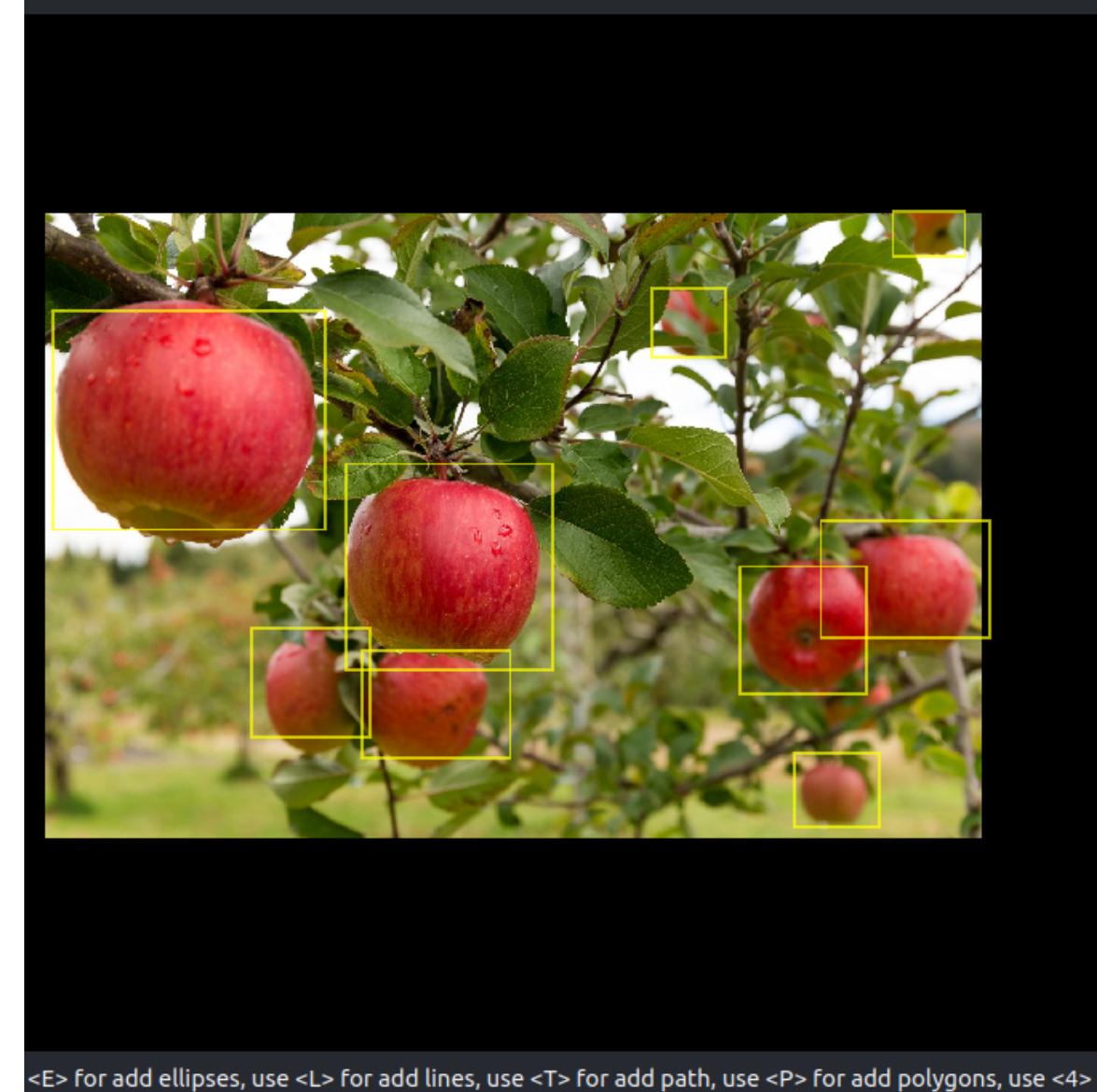
Apple flower detection



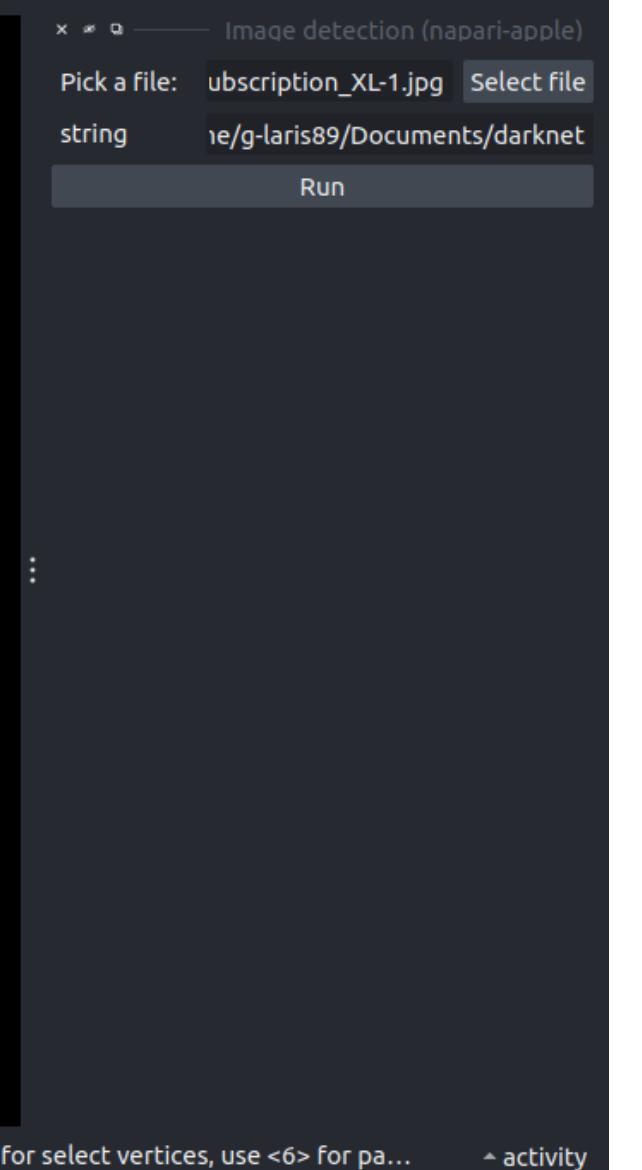
Plugin: napari-blossom



Apple detection in orchard

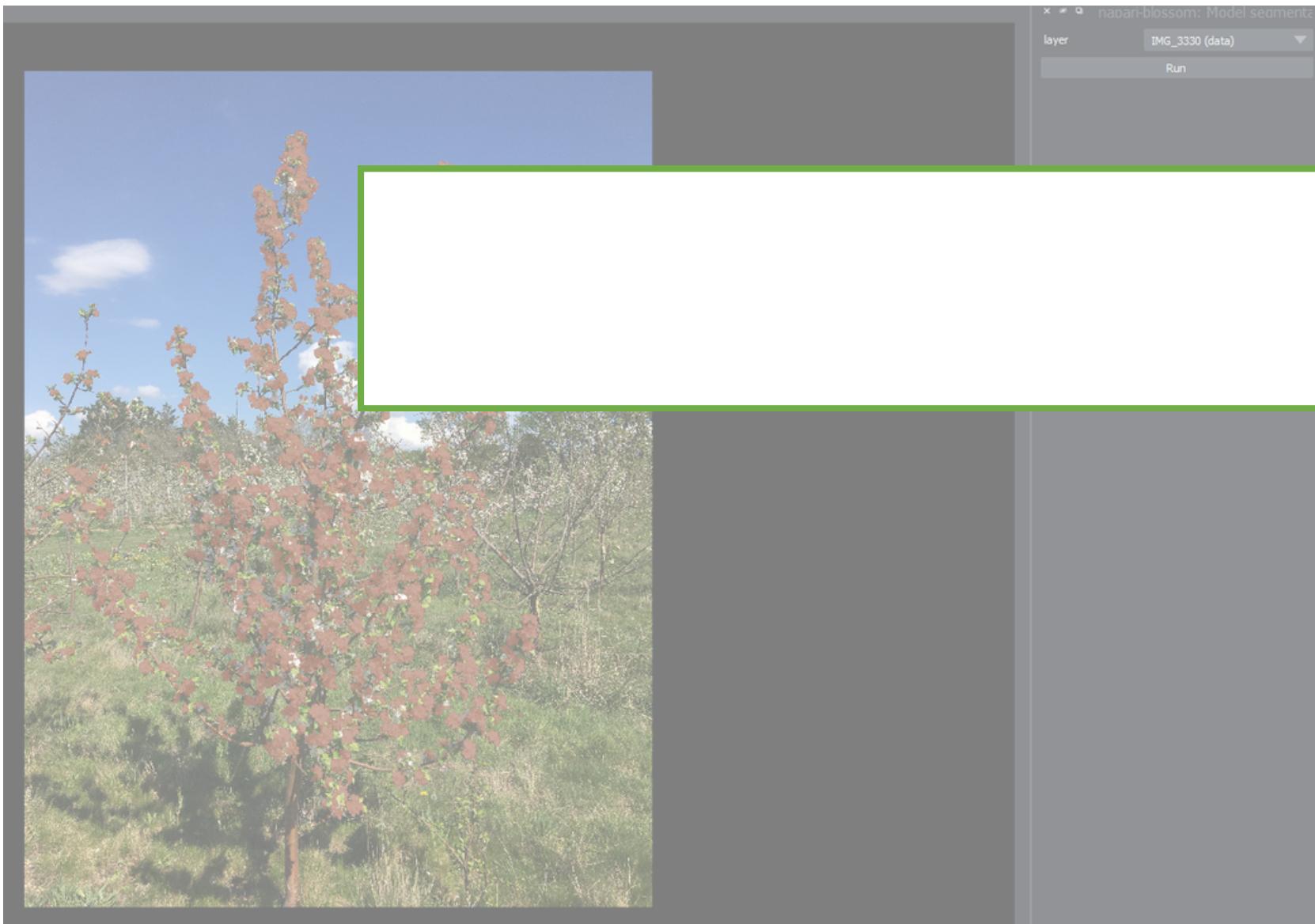


Plugin: napari-apple



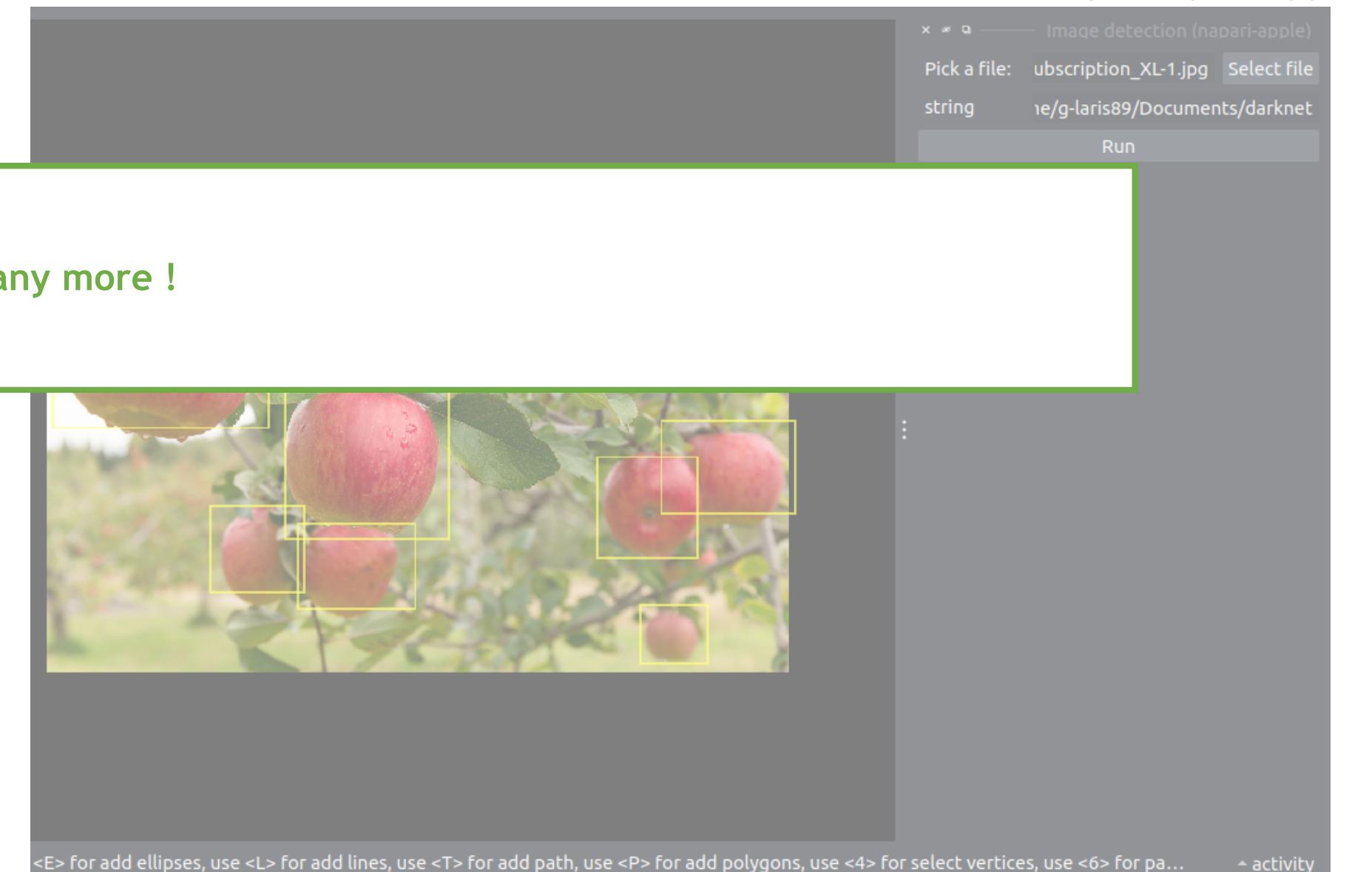
# Deep learning in Napari

Apple flower detection



Plugin: napari-blossom

Apple detection in orchard



And many more !

# Topic of this workshop

## Objective

Create a napari plugin from image processing code including DL model

## What we will do

Review a DL code

Create a napari plugin in local

Integrate DL model (tensorflow) in plugin

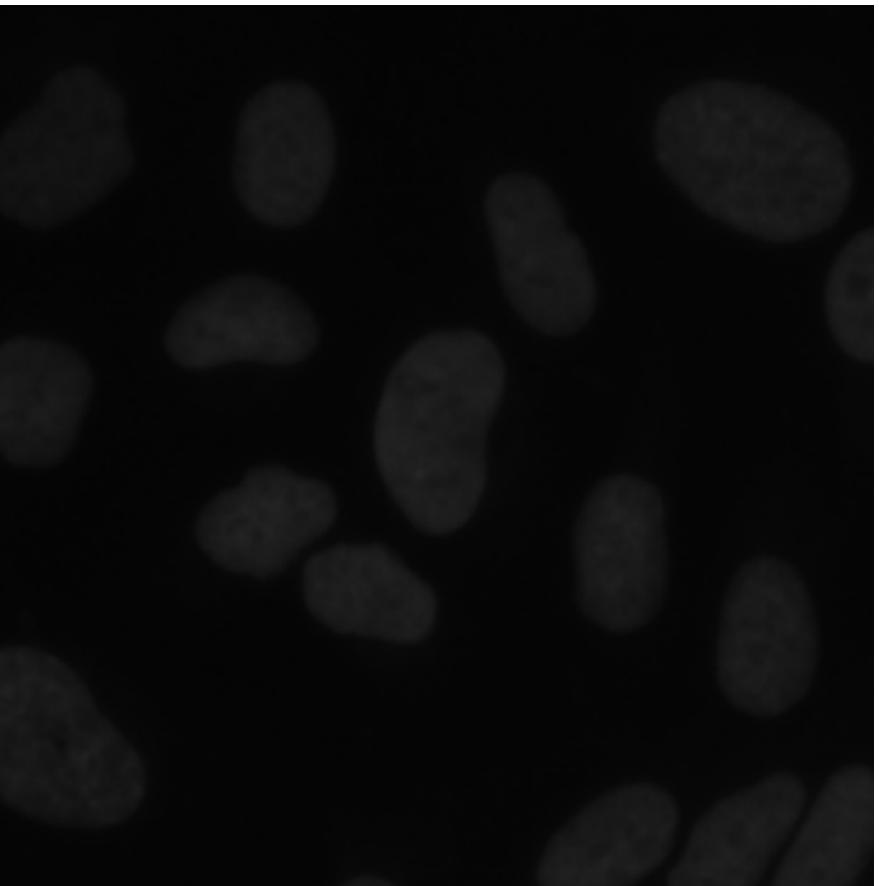
## What we will not do

Teach Python programming and the base of deep learning (requirements)

Deploy the plugin in napari-hub platform (napari library plugin)

# Practical session

Let's suppose we have



```
In [69]: # INPUT
# Write model ESRF_Seg_Hands_on_best_model.h5 path
model_path_ = "/home/g-laris89/DEEP-NAPARI/Exercise-4_My first widget/model/ESRF_Seg_Hands_on_best_model.h5"
input_ = "/home/g-laris89/DEEP-NAPARI/Exercise-2_Notebook_Code/images/03f583ec5018739f4abb9b3b4a580ac43bd933c4337ad8877aa18l

model_New = tf.keras.models.load_model(model_path_,custom_objects={'dice_coefficient': dice_coefficient})

_, IMG_HEIGHT, IMG_WIDTH, IMG_CHANNELS = list(model_New.input.shape)

img_ = imread(input_)

nbr_image = len(img_.shape)
if nbr_image==3:
    img_ = img_[:, :, :IMG_CHANNELS]
    X = np.zeros((1, IMG_HEIGHT, IMG_WIDTH, IMG_CHANNELS), dtype=np.uint8)
    h_or, w_or, _ = img_.shape
    ORIGIN = np.zeros((1, h_or, w_or, IMG_CHANNELS), dtype=np.uint8)
    ORIGIN[0] = img_
    img = resize(img_, (IMG_HEIGHT, IMG_WIDTH), mode='constant', preserve_range=True)
    X[0] = img
elif nbr_image==4:
    img_ = img_[:, :, :, :IMG_CHANNELS]
    X = np.zeros((img_.shape[0], IMG_HEIGHT, IMG_WIDTH, IMG_CHANNELS), dtype=np.uint8)
    shape_h, shape_w = [], []
    for i in range(img_.shape[0]):
        print(img_[i,...].shape)
        shape_h.append(img_[i,...].shape[0])
        shape_w.append(img_[i,...].shape[1])
        img = resize(img_[i,...], (IMG_HEIGHT, IMG_WIDTH), mode='constant', preserve_range=True)
        X[i] = img

    ORIGIN = np.zeros((img_.shape[0],np.max(shape_h),np.max(shape_w),IMG_CHANNELS), dtype=np.uint8)
    for i in range(img_.shape[0]):
        ORIGIN[i,...][:shape_h[i], :shape_w[i],:] = img_[i,...]

preds_test = model_New.predict(X, verbose=1)
# we apply a threshold on predicted mask (probability mask) to convert it to a binary mask.
preds_test_opt = (preds_test > 0.5).astype(np.uint8)

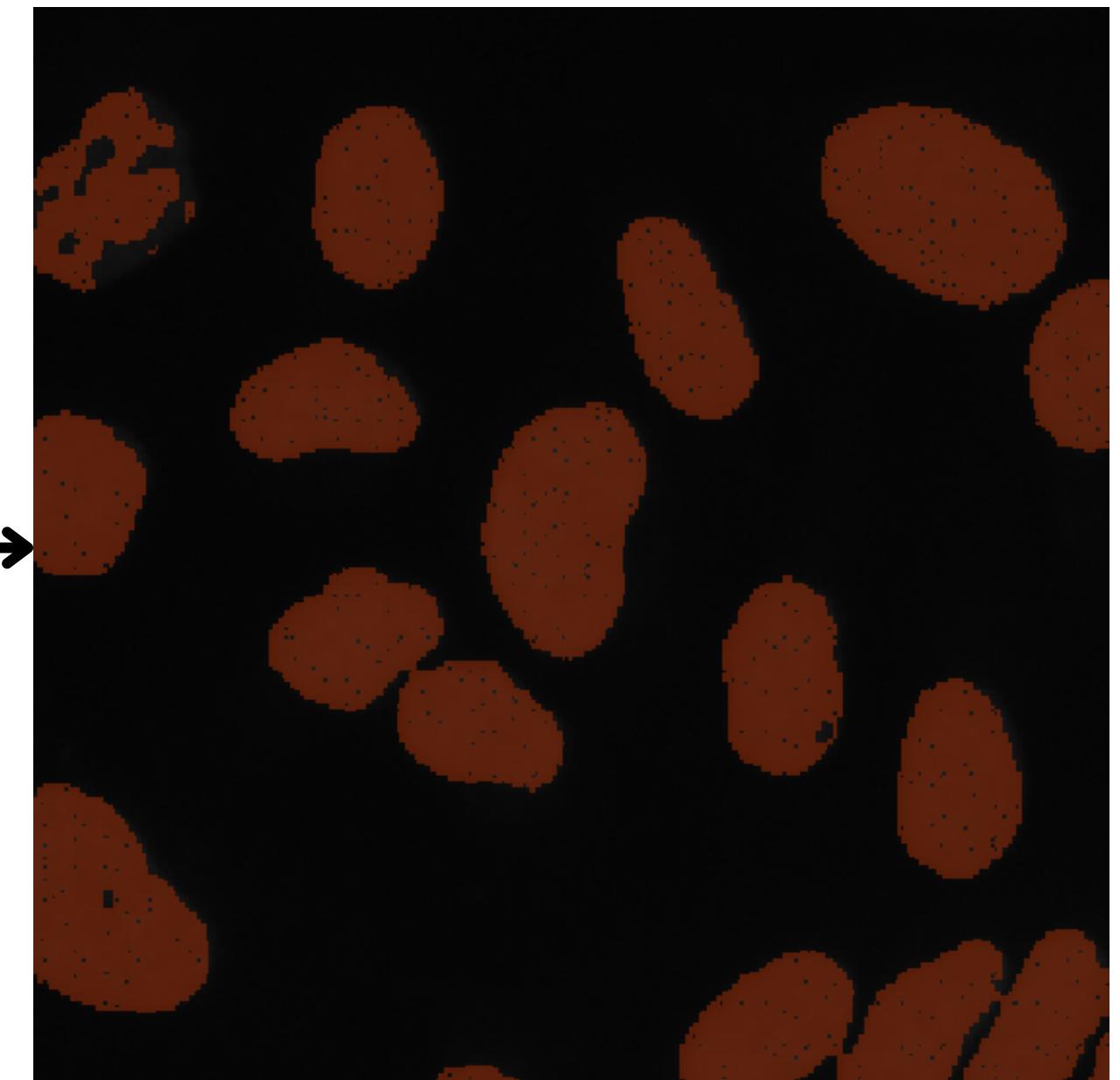
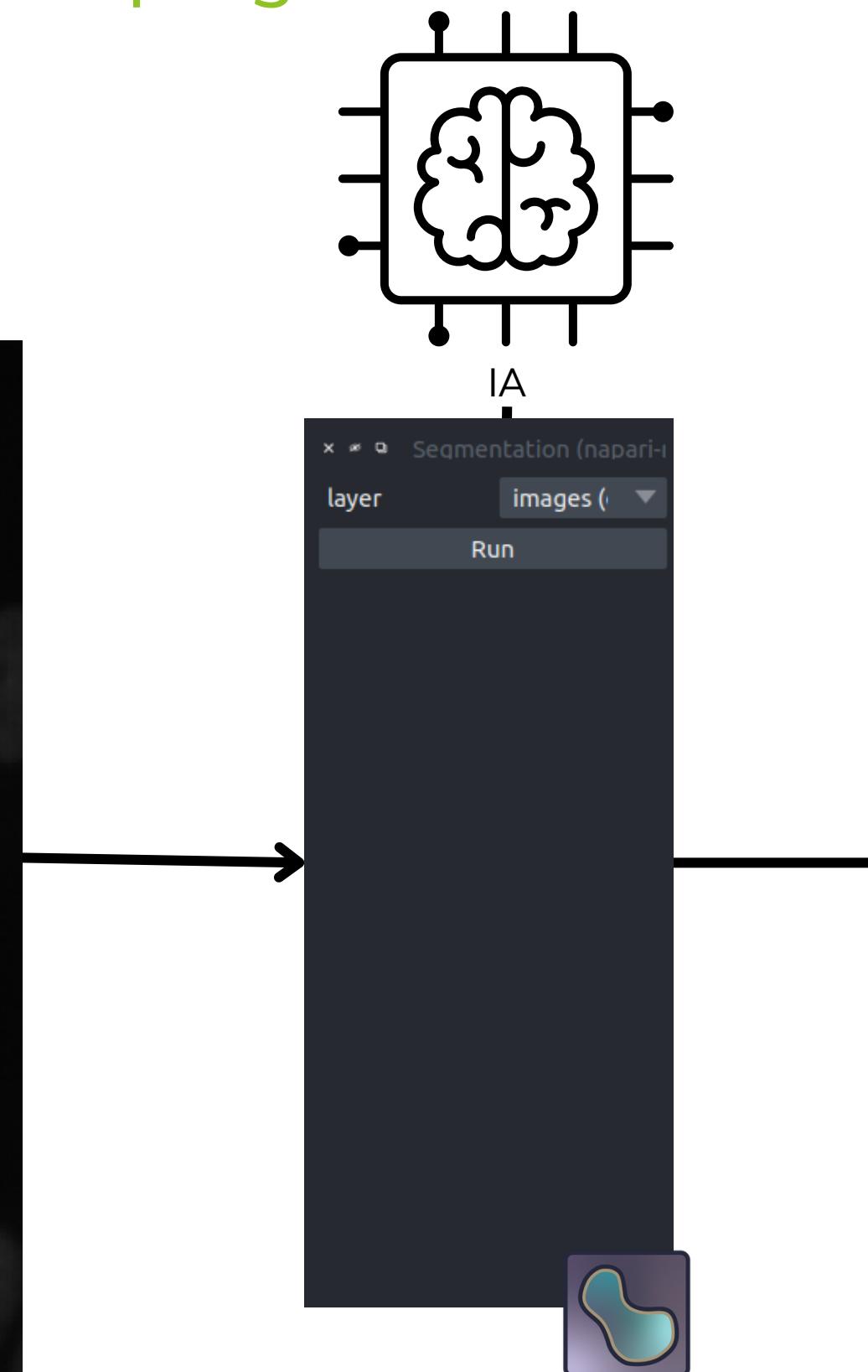
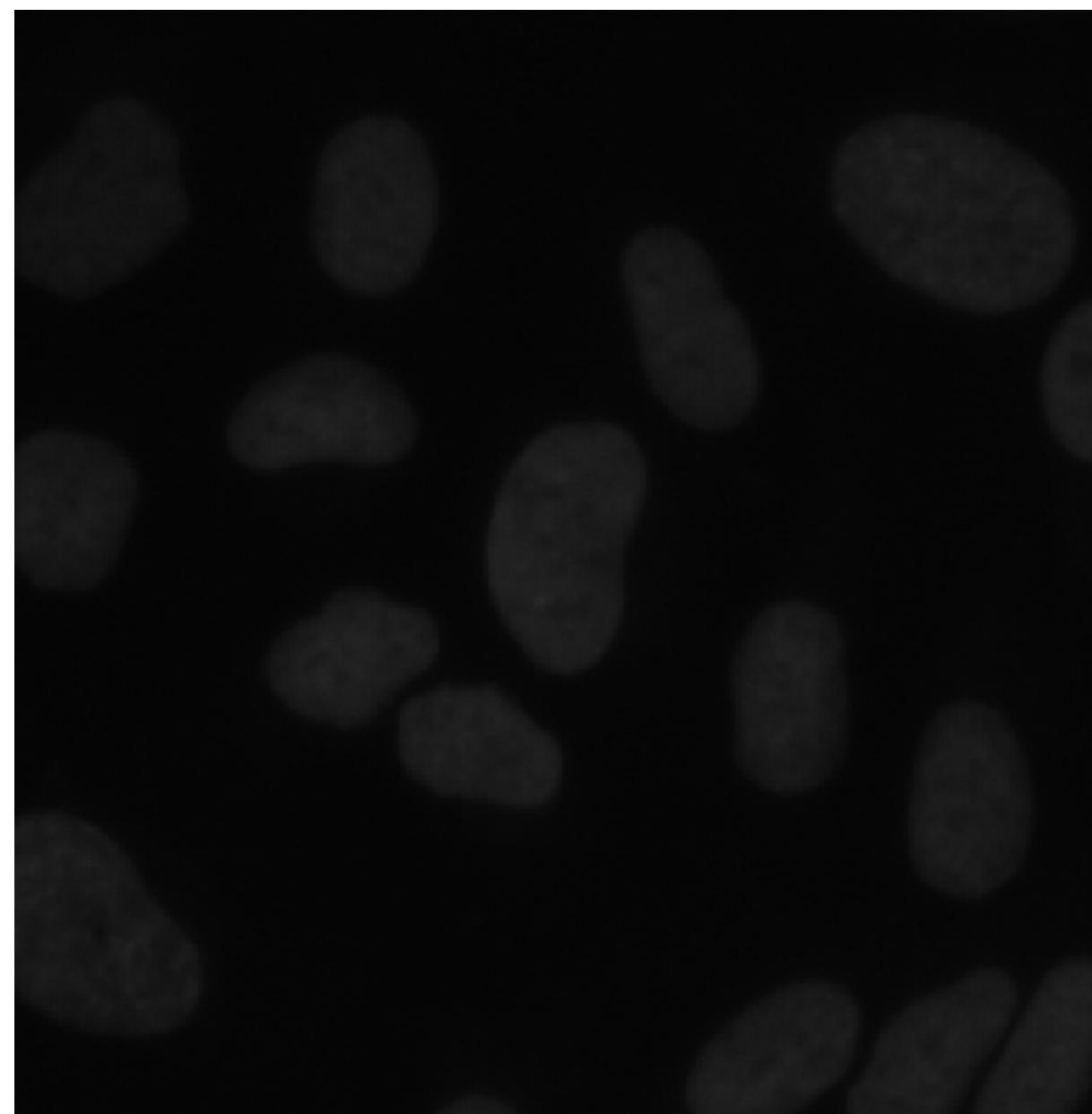
if nbr_image==3:
    result_ = np.squeeze(preds_test_opt[0, :, :, 0])
    output_ = np.zeros((1,h_or,w_or), dtype=np.uint8)
    res_ = resize(result_, (h_or, w_or), mode='constant', preserve_range=True)
    output_[0] = res_
elif nbr_image==4:
    img_ = img_[:, :, :, :IMG_CHANNELS]
    result_ = np.squeeze(preds_test_opt[:, :, :, 0])
    output_ = np.zeros((img_.shape[0],np.max(shape_h),np.max(shape_w)), dtype=np.uint8)
    for i in range(img_.shape[0]):
        res_ = resize(result_[i], (shape_h[i], shape_w[i]), mode='constant', preserve_range=True)
        output_[i,...][:shape_h[i], :shape_w[i]] = res_
```

1/1 [=====] - 0s 290ms/step



# Practical session

Deploy DL model in Napari plugin



# Practical session

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



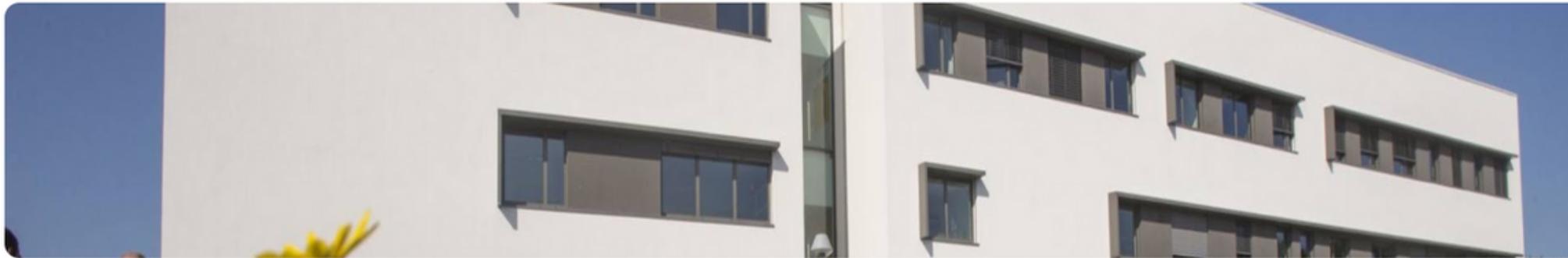
The screenshot shows the GitHub repository page for "hereariim / ESRF-NAPARI". The repository is public and contains several files and commits:

- hereariim.pptx**: Contains commits for "Exercise-1\_Getting started", "Exercise-2\_Notebook\_Code", "Exercise-3\_My first plugin", "Exercise-4\_My first widget", "Extra-Deploy\_in\_builtin", "images-credit", "ESRF-DEEPNAPARI.pdf", and "README.md".
- README**: A file containing the text: "ESRF NAPARI : Napari as a tool for deep learning project management".

The repository has 1 branch, 0 tags, and 8 commits. It has 0 stars, 1 watching, and 0 forks. The repository is described as having "No description, website, or topics provided." It also lists "Readme", "Activity", "0 stars", "1 watching", and "0 forks". There are no releases or packages published. The languages used are Jupyter Notebook (91.4%) and Python (8.6%). Suggested workflows include Django.

# To go further in this exercise

Tutorial on the presentation, design of a plugin... and even more



**ImHorPhen Bio imaging research group**

@imhorphenbioimagingresearch · 365 subscribers · 175 videos

ImHorPhen is a bioimaging, research group, headed by Prof. David Rousseau, located in An... >

[www6.angers-nantes.inrae.fr/irhs/Recherche/Imagerie-pour-l-Horticulture-et-le...](http://www6.angers-nantes.inrae.fr/irhs/Recherche/Imagerie-pour-l-Horticulture-et-le...) and 1 more link

[Subscribe](#)

Home Videos Playlists Community

Latest Popular Oldest

**Deepaas & Napari hand in hand** 23:57  
116 views · 1 year ago

**Creating and modifying a Napari widget** 19:48  
32 views · 2 months ago

**Getting started with Napari in Phenotyping** 12:52  
17 views · 2 months ago

**NAPARI Plugin for deep segmentation and manual correction** 35:42  
196 views · 1 year ago

**Installer NAPARI et faire son premier plugin** 13:14  
140 views · 1 year ago

**Easy shallow learning with Ilastik** 16:06  
464 views · 2 years ago

**Panorama des solutions "user-friendly" de deep learning en bioimagerie** 14:30  
38 views · 1 year ago

**Installation de Cellpose** 42:53  
753 views · 1 year ago



# Conclusion



What we learnt ?

What we learnt ?

What we learnt ?

What we learnt ?

We learnt a method to use easily your DL model with GUI aka Napari



We learnt a method to use easily your DL model with GUI aka Napari



I hope this method we let you deploy your DL model to help experimental scientist

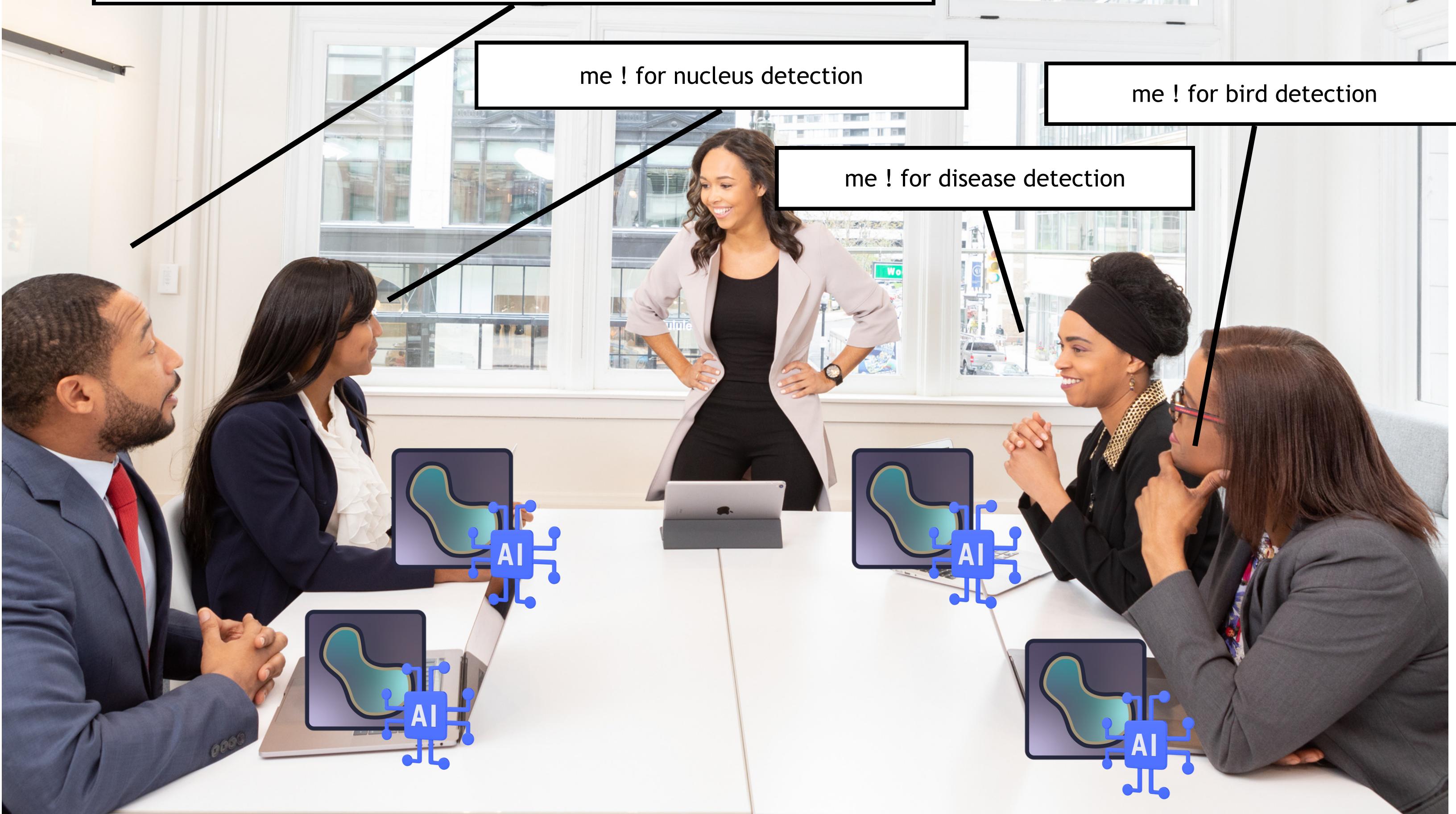


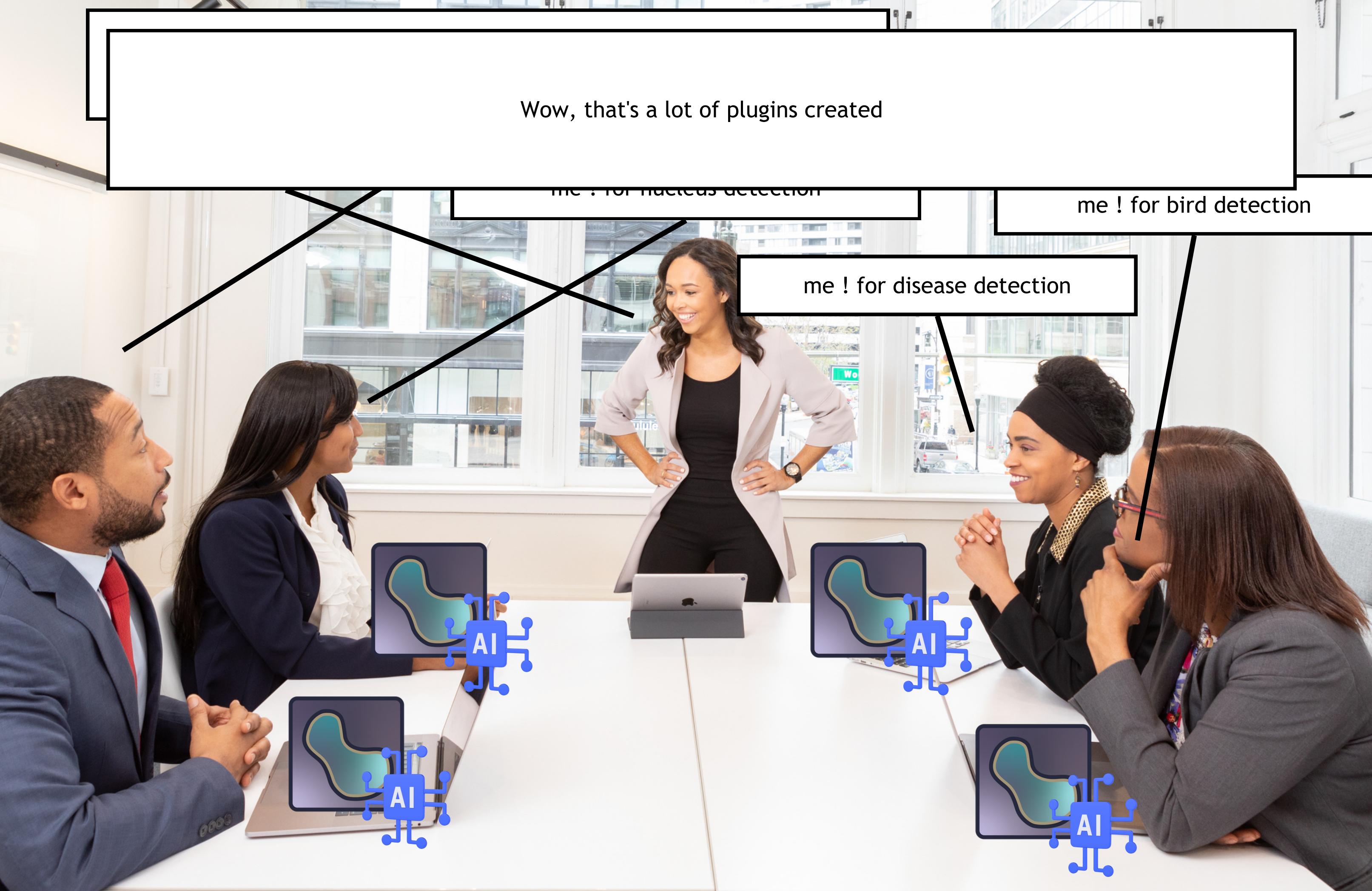
Thanks ! I will deploy my model for Cell detection

me ! for nucleus detection

me ! for bird detection

me ! for disease detection



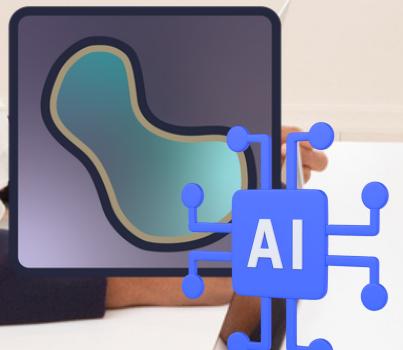


Wow, that's a lot of plugins created

me ! for nucleus detection

me ! for bird detection

me ! for disease detection

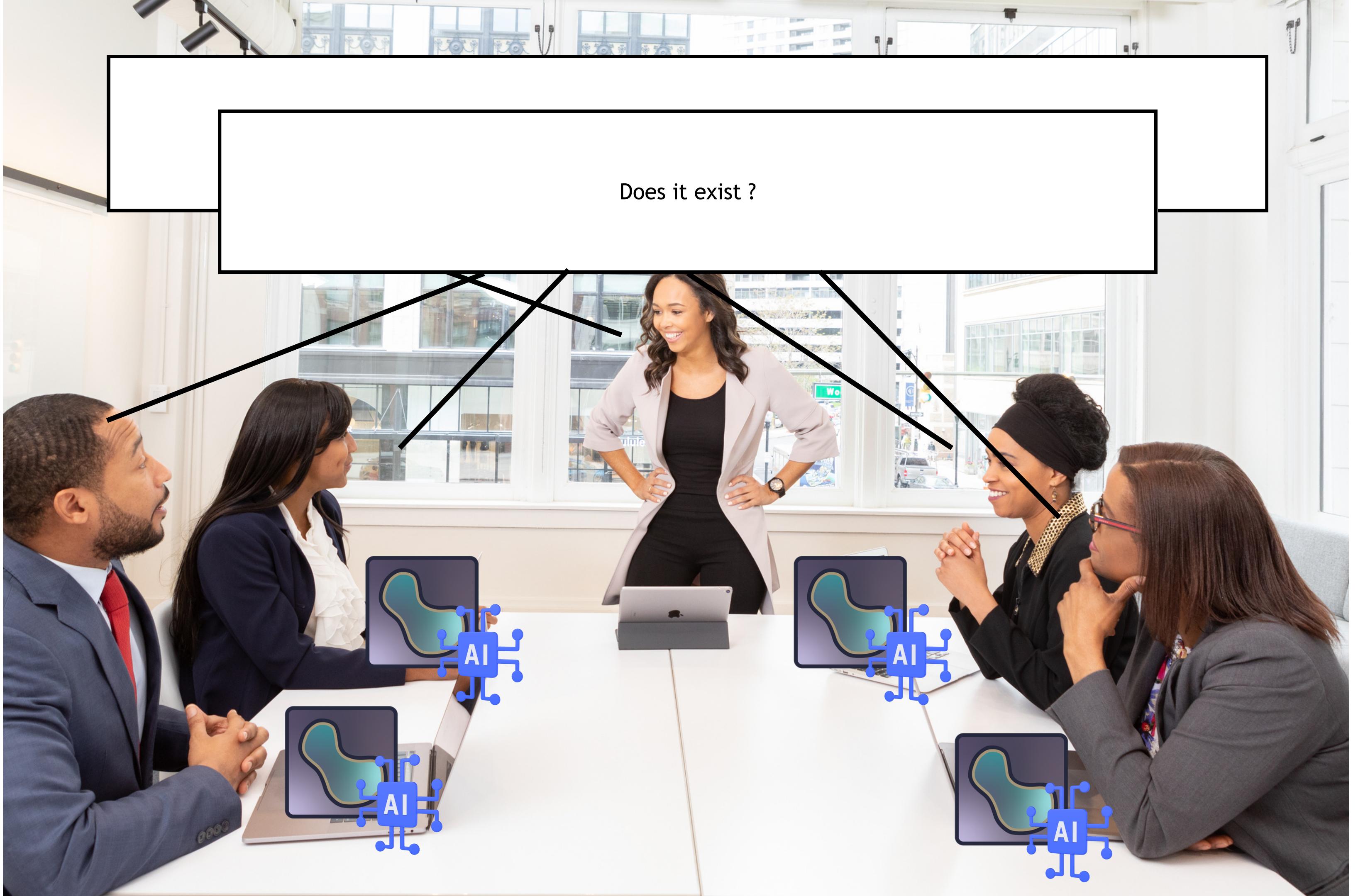


Imagine, we have a single plugin to read all your DL model.



No need to create a set of plugins and your DL model remains confidential





Does it exist ?



Does it exist ?

Sure !!!!!



napari hub Plugins



## Manini MAchiNe INference & correction

manini



MANINI

An user-friendly plugin that enables to annotate images from a pre-trained model (segmentation, classification, detection) given by an user.

[Herearii Metuarea](#)

[View project data](#)

Description Activity NEW!

license BSD-3-Clause pypi v0.0.6 python 3.8 | 3.9 | 3.10 tests passing codecov 9% napari hub manini

Manini (MAchiNe INference & Correction) is thought as a tool to boost the collaborative contribution of end-users to the assessment of deep learning model during their testing phase. It is a user-Friendly plugin that enables to manually correct the result of an inference of deep learning model by an end-user. The plugin covers the following informational tasks: segmentation, classification and object detection.

### White paper

Herearii Metuarea, David Rousseau. Toward more collaborative deep learning project management in nplant

</> 🌐 ⚙️

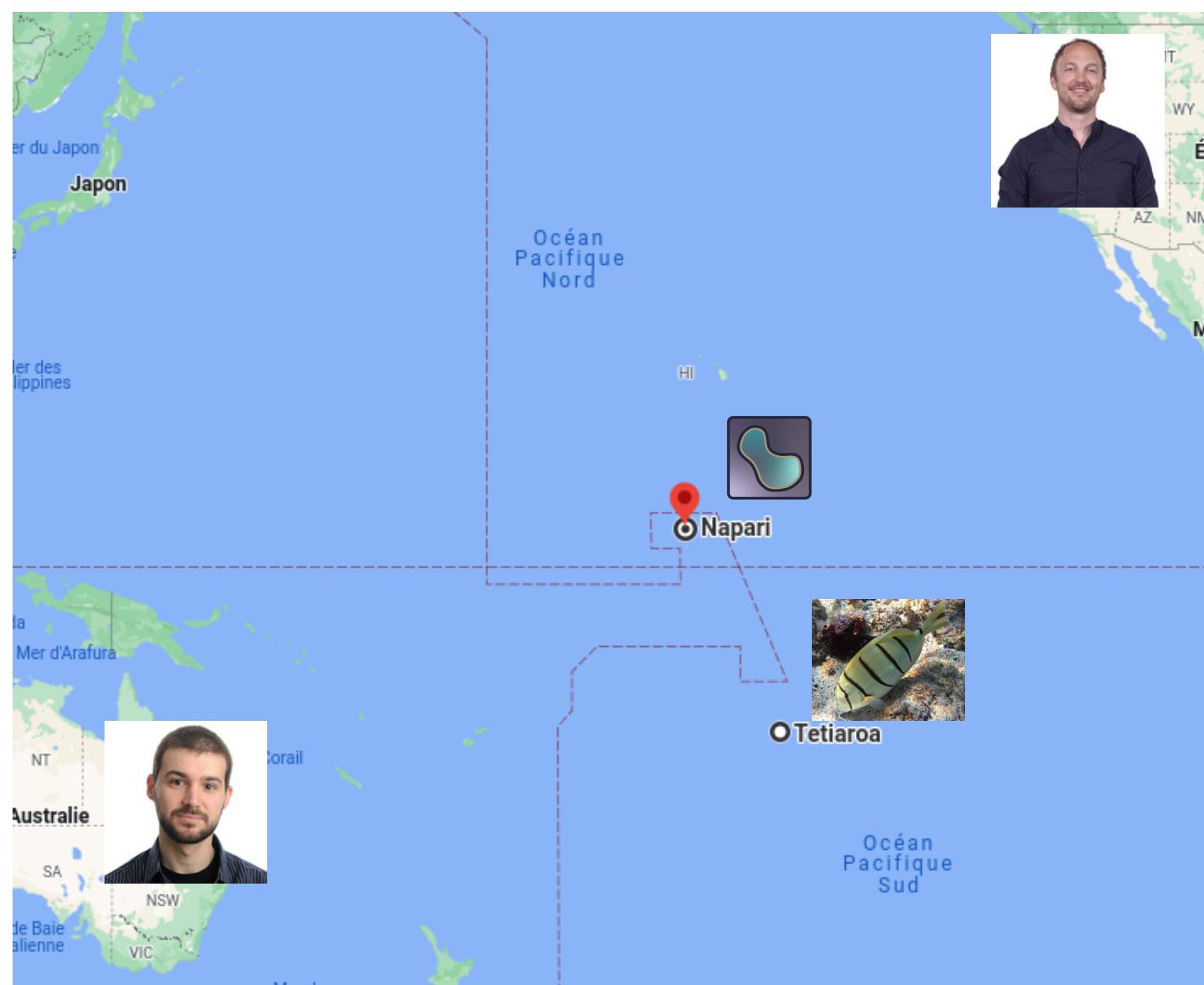
White paper

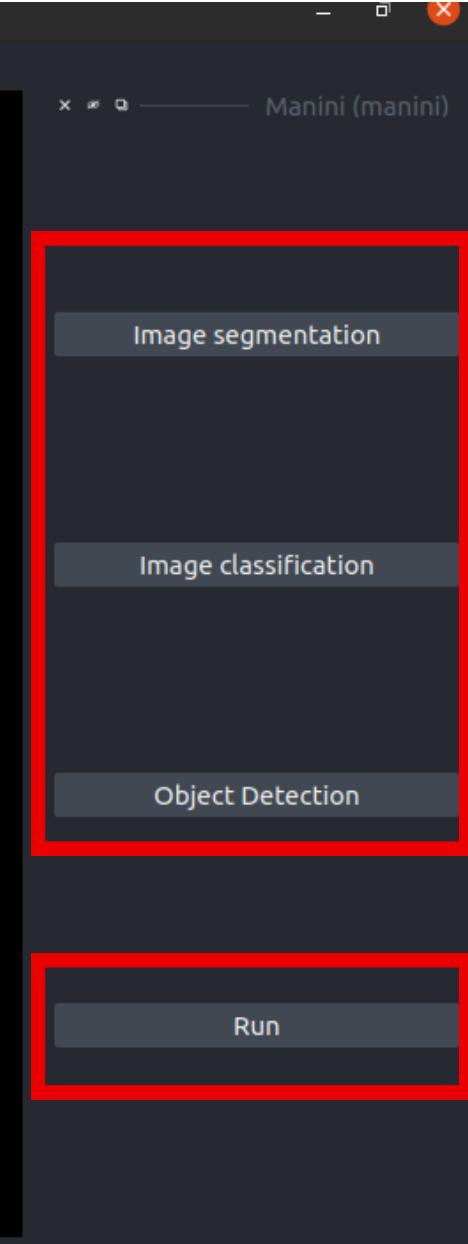
Installation

Description

License

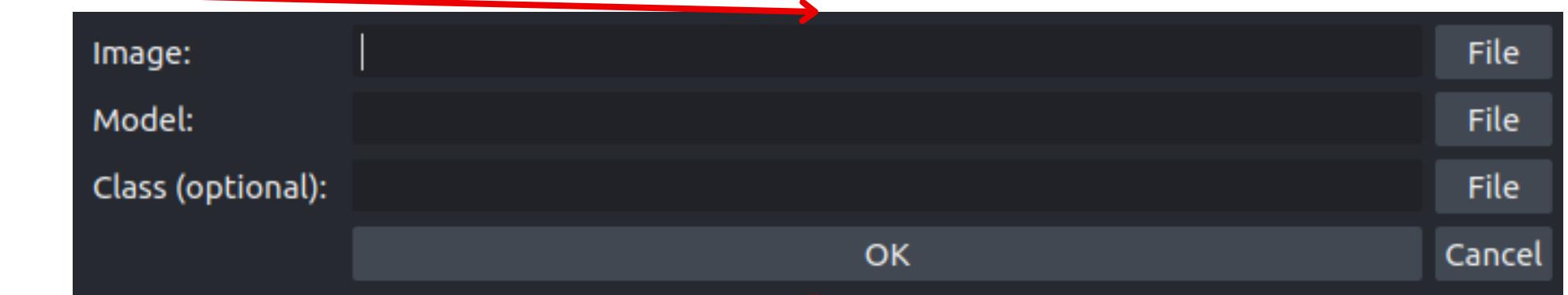
Issues





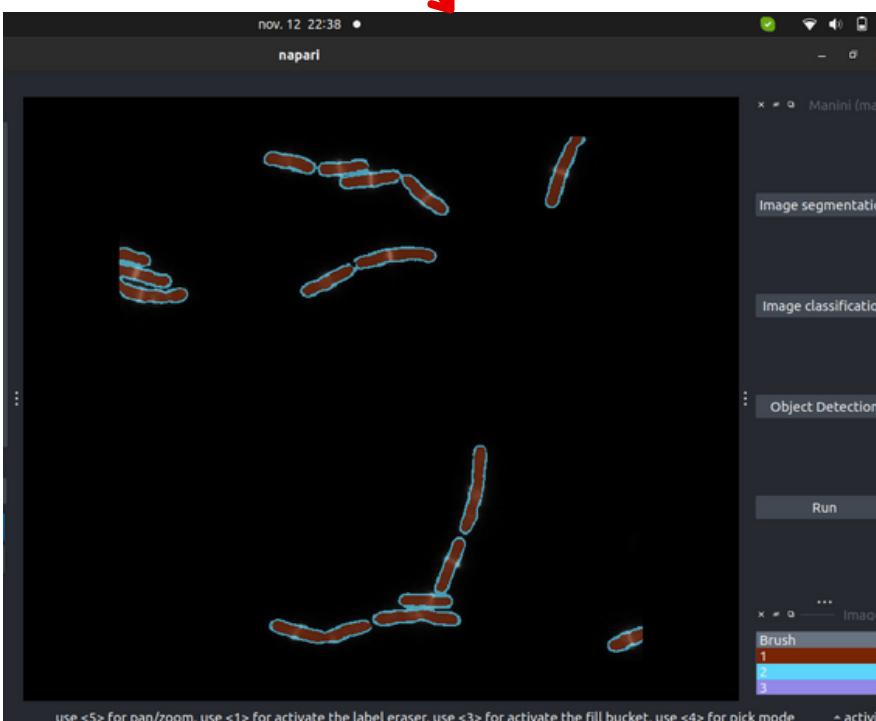
## 1. Choose the type of Deep learning task

## 2. Import your data (Image, Model, Class (optional))

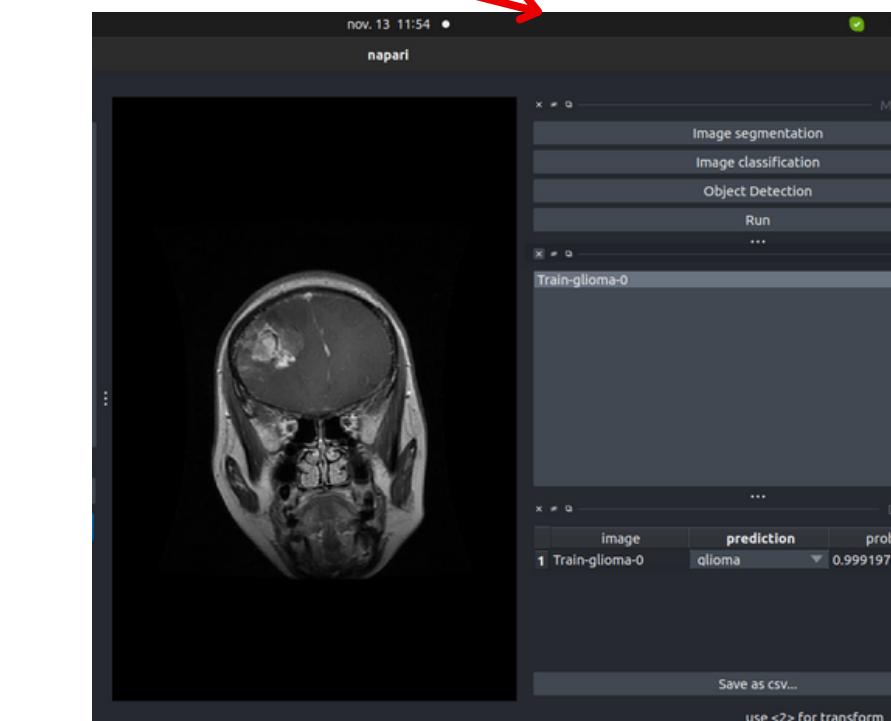


## 3. Run

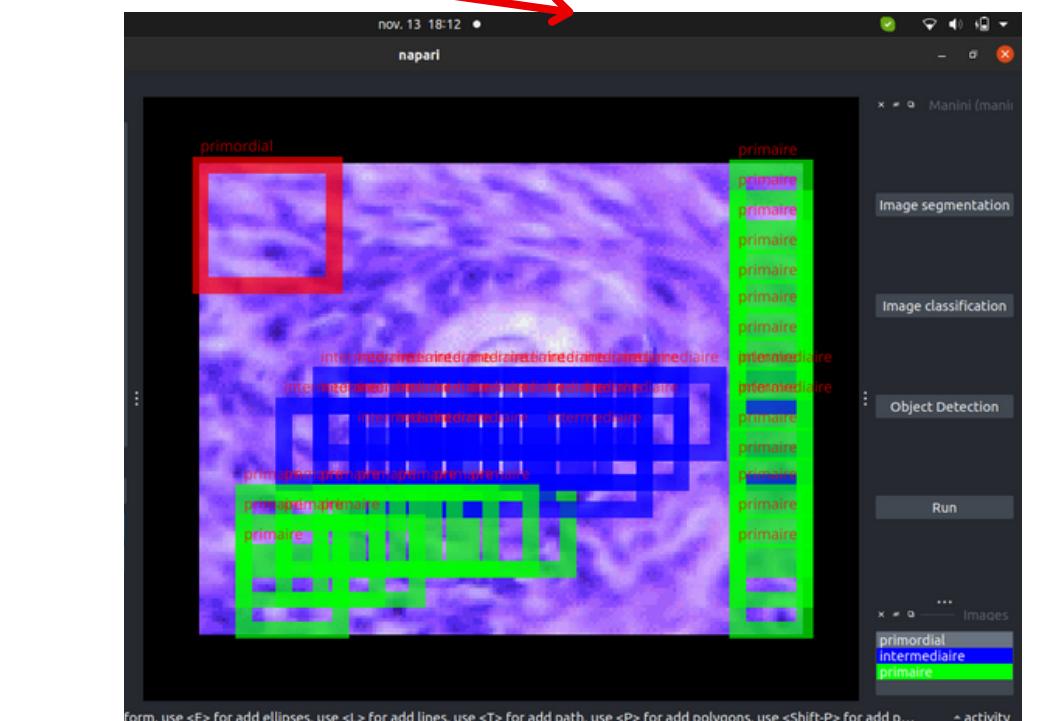
## 4. Correct your data



Segmentation



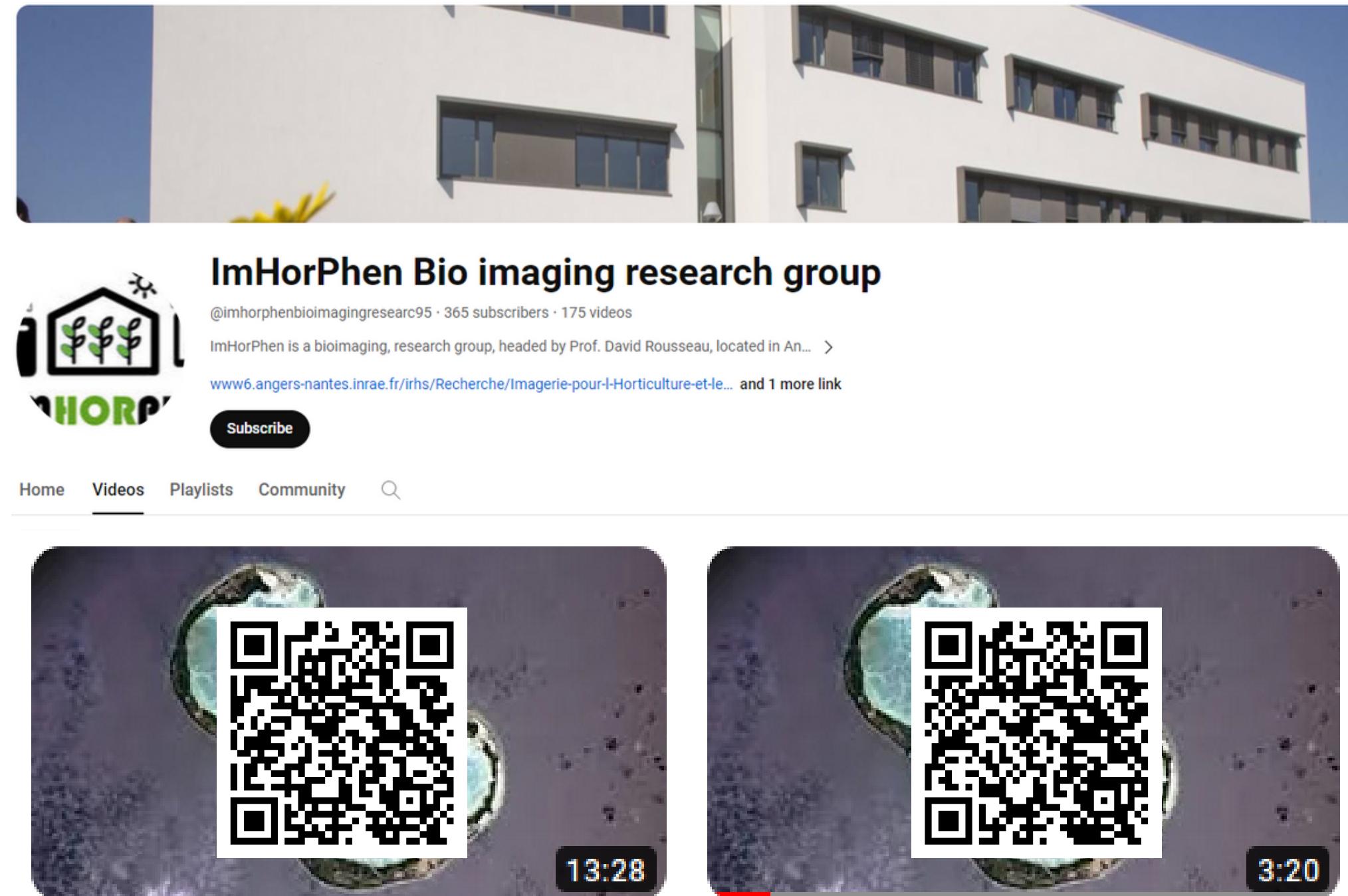
Classification



Detection

# To know more about it

## Manini presentation



**MANINI Napari Plugin Part 2**

Aucune vue • il y a 3 heures

**MANINI Napari Plugin Part 1**

3 vues • il y a 1 jour





Thanks you for your attention



**INRAe**

