

Application Note #47

Applying Cellpose models in arivis Vision4D

How to: install and run predictions with Cellpose

The application-note goal is to guide the user in the correct installation of the Anaconda3 Python package and all the Cellpose module. Cellpose is a cell/nuclei detection algorithm, based on the deep learning approach, primarily on the fluorescence microscopy images.

More information can be gathered from the following articles:

Cellpose: a generalist algorithm for cellular segmentation. C. Stringer, T. Wang, M Michaelos M Pachitariu, Nature Methods, 2021.

Warnings

Vision4D runs the deep learning applications (e.g., Cellpose or Stardist) using external and independent Python libraries and tools produced by third parties.

These tools must be installed by the user under its responsibility, strictly following the instruction on this document.

Arivis has tested the setup protocol on several computers, however, due to the different and not predictable hardware and software configuration of each computer, the results can be different case by case.

Therefore, arivis declines any responsibility concerning the correct tools installation and setup on the user computer.

arivis cannot be blamed about any malfunctioning or failure of the deep learning environment setup.

Arivis will not give technical support on the setup task as well as on any deep learning application.

Both activities are totally on the user charge.

Arivis also declines any responsibility about the scientific results gathered from the deep learning application.

Index

Application Overview

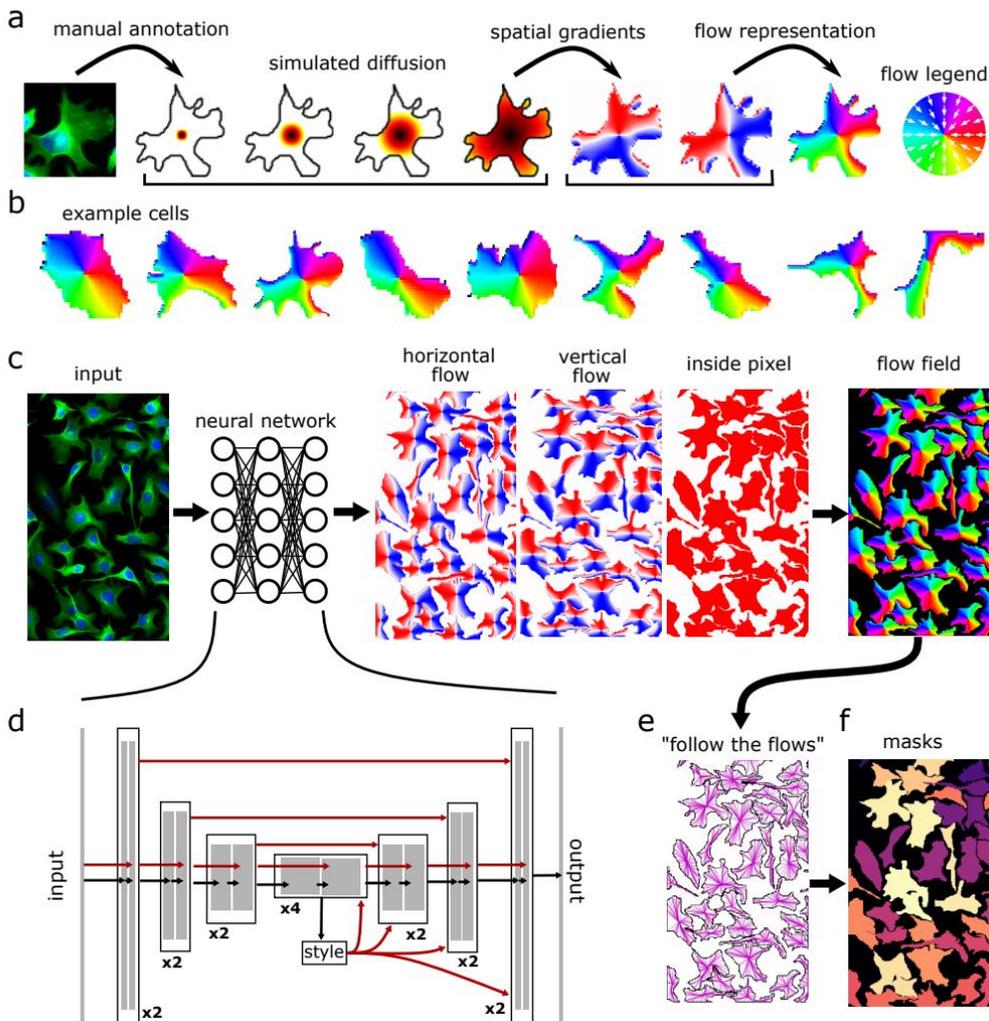
1. Download the Anaconda Package
2. Install the Anaconda package
3. Install the Cellpose modules
4. Vision 4D preferences setup
5. Run the Cellpose predictions in V4D
6. Examples of the Cellpose prediction results
7. Additional operator for the 2D Cellpose Segmentation (Image Segmenter)
8. Troubleshooting the Cellpose script
9. Troubleshooting the Cellpose setup

Application Overview

Cellpose in a Nutshell

Cellpose is a deep learning-based method for 2D and 3D detection of Nuclei and Cytoplasm objects, developed and published by the Carsen Stringer group: [GitHub - MouseLand/cellpose: a generalist algorithm for cellular segmentation](https://github.com/MouseLand/cellpose)

Cellpose uses a cell detection method that predicts the shape representation using the flow-representation of object cell dynamics that is well-suited to approximate and define the complex borders of the cells in the microscopy images.



Application Overview

Why use Cellpose within Vision4D?

arivis Vision4D (V4D) is a modular software for working with multi-channel 2D, 3D and 4D images of almost unlimited size, independent of available RAM. Many imaging systems, such as high speed confocal, light sheet / SPIM and 2-photon microscope systems produce a vast amount of multi-channel data, which V4D handles without constraints.

V4D allows the user to execute complex analysis tasks in automatic or batch mode. It includes sophisticated pre-processing algorithm, multiple segmentation approaches, including the machine learning tools, and powerful data handling. Gigabytes, hundreds of Gigabytes or even Terabytes of data can be quantified by V4D with a single task.

StarDist represents an advanced method to detect roundish objects such as cells and nuclei, especially in crowded fields where the objects are overlapping, but it is limited to these cases. The new frontiers of image analysis in life science need the capability to analyze the complex interactions between biological structures. V4D has the tools to satisfy these requirements. StarDist can be integrated in the V4D analysis workflow and directly contribute to better detect its target structures.

StarDist can be currently executed as a python script but, in the near future, it will be available as a V4D pipeline operator, making its usage even more flexible and powerful.

NOTE:

Currently, Vision4D can't train Cellpose Deep Neuronal Network model internally. This task is based on math operations that can easily be parallelized. In order to optimize the process and to reduce the time required the support of GPU CUDA is needed.

Therefore, Vision4D will use a pre-trained model generated using other tools.

1. Download the Anaconda Package

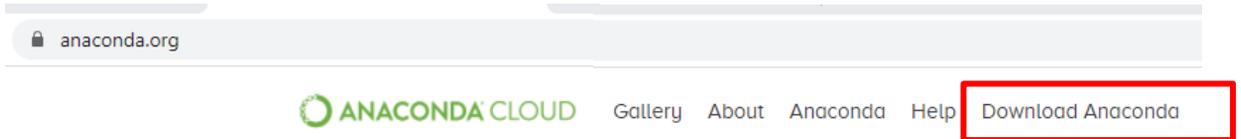
Step 1.1

Open the Anaconda web pages using the following link:

<https://anaconda.org>

Step 1.2

Select the Download Anaconda item.



Step 1.3

Click on **Download**

The Individual Edition is Open source and freely distributable



Individual Edition

Your data science toolkit

With over 20 million users worldwide, the open-source Individual Edition (Distribution) is the easiest way to perform Python/R data science and machine learning on a single machine. Developed for solo practitioners, it is the toolkit that equips you to work with thousands of open-source packages and libraries.



Anaconda Installers

Windows 	MacOS 	Linux 
Python 3.8 64-Bit Graphical Installer (466 MB) 32-Bit Graphical Installer (397 MB)	Python 3.8 64-Bit Graphical Installer (462 MB) 64-Bit Command Line Installer (454 MB)	Python 3.8 64-Bit (x86) Installer (550 MB) 64-Bit (Power8 and Power9) Installer (290 MB)

Click on **64-bit Graphical Installer**

The download task starts.

2. Install the Anaconda package

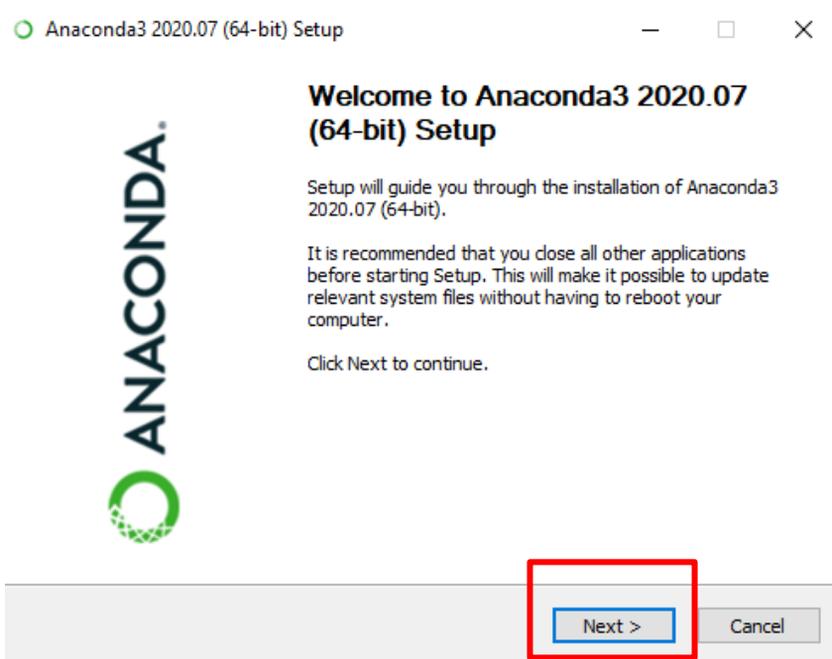
Step 2.1

Open the [Download folder](#) and locate the Anaconda3 setup file.

 **Anaconda3-2020.07-Windows-x86_64.exe**

Step 2.2

Run it and press the [Next](#) button on the setup dialog

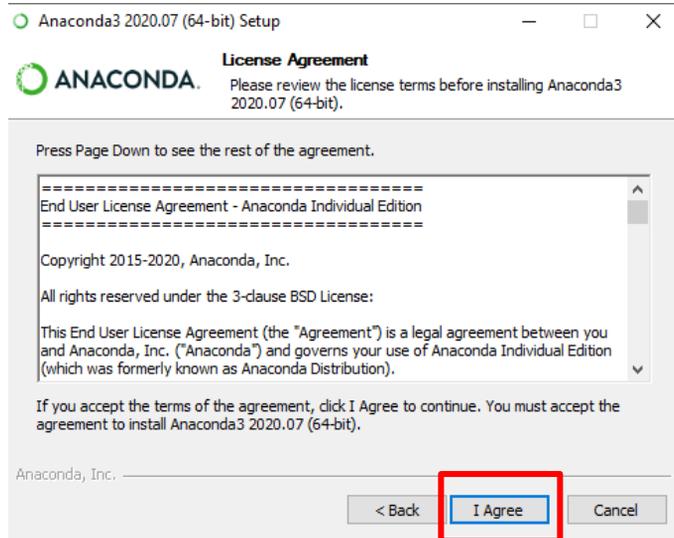


Note :
The Python release must be the 3.8 or higher.

2. Install the Anaconda package

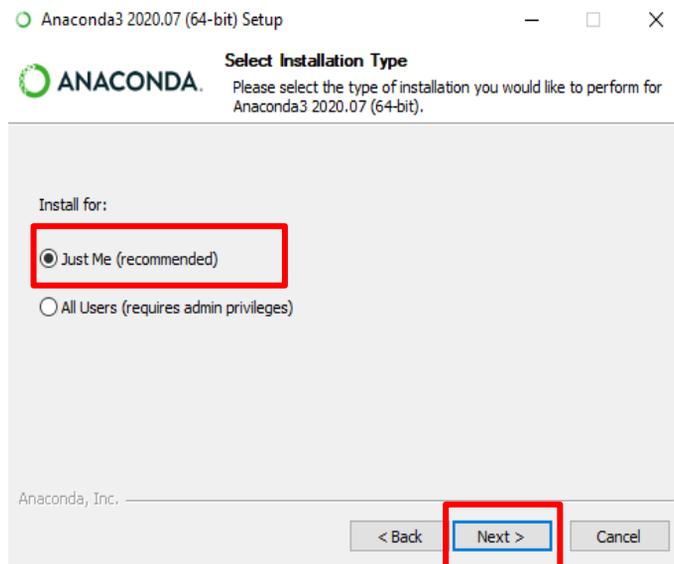
Step 2.3

Click on the „I Agree“ button to accept the License Agreement terms.



Step 2.4

Select the „Just Me“ option.



Step 2.5

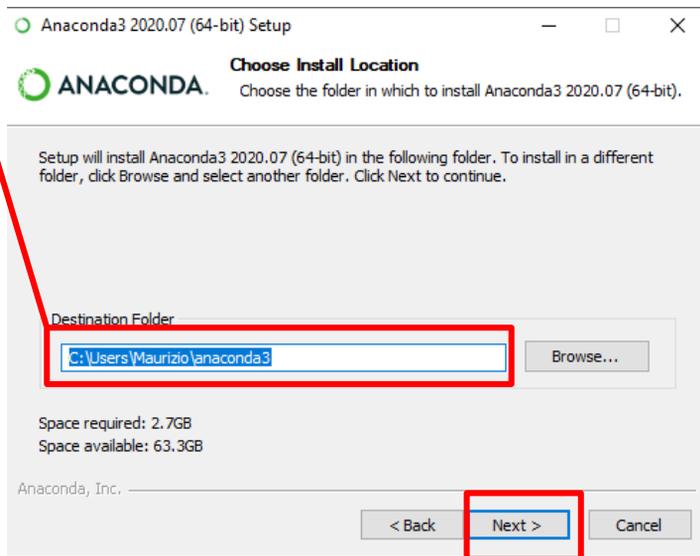
Press the „Next“ button to complete the installation.

2. Install the Anaconda package

Step 2.6

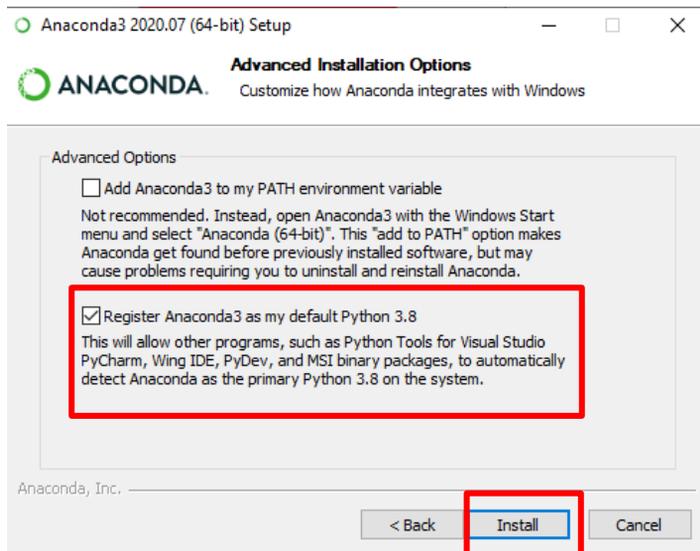
Leave the default install path. Usually, it is located under the user's directory

C:\Users\xxxxxx\Anaconda3



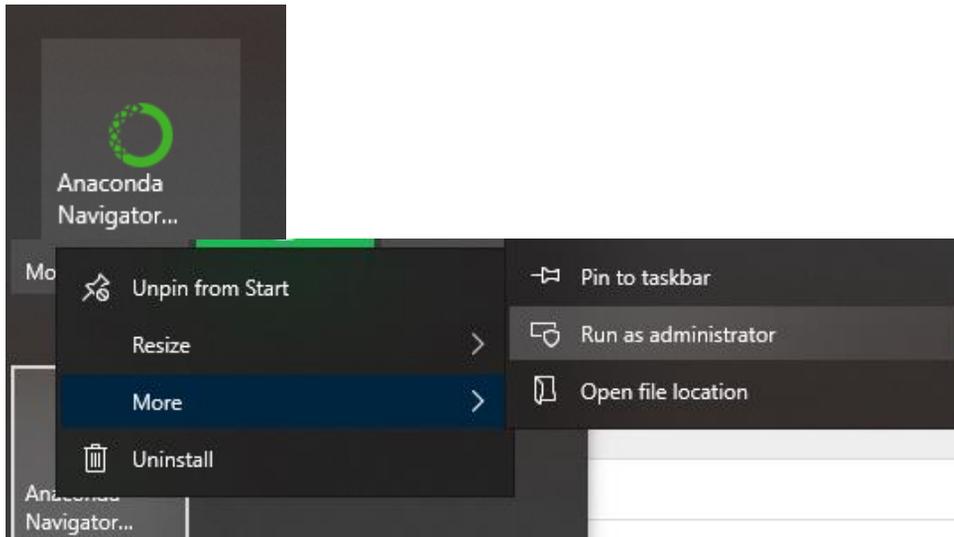
Step 2.7

Enable the option «Register Anaconda3 as my default Python 3.8» package. Press «***Install***» to start the installation.

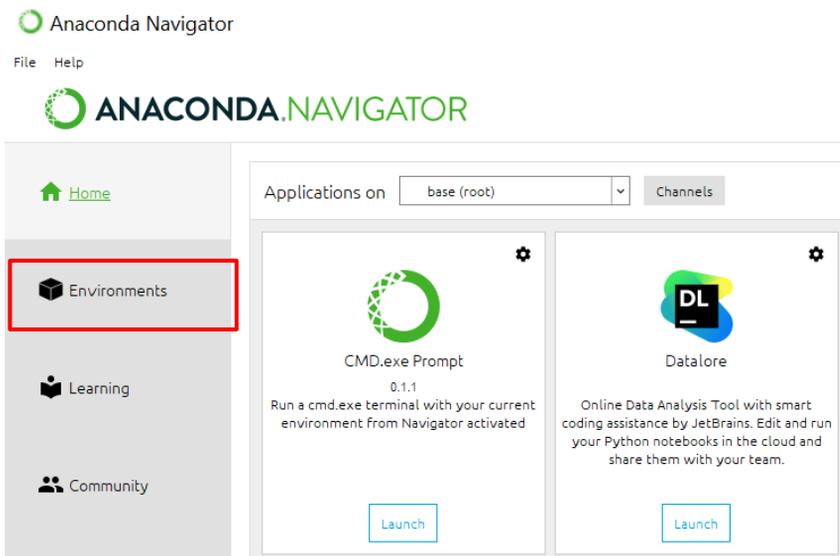


3. Install the Cellpose modules

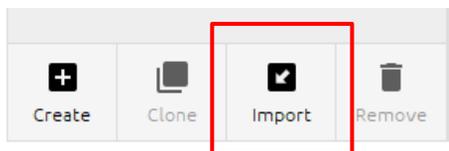
Step 3.1 Run the Anaconda Navigator in the Administrator account:



Step 3.2 Select the Environments tab:



Step 3.3 Select Import the Environment on the bottom :

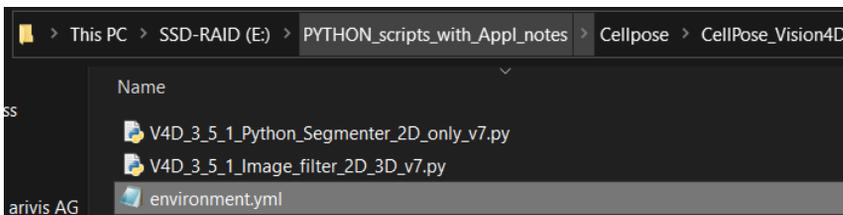
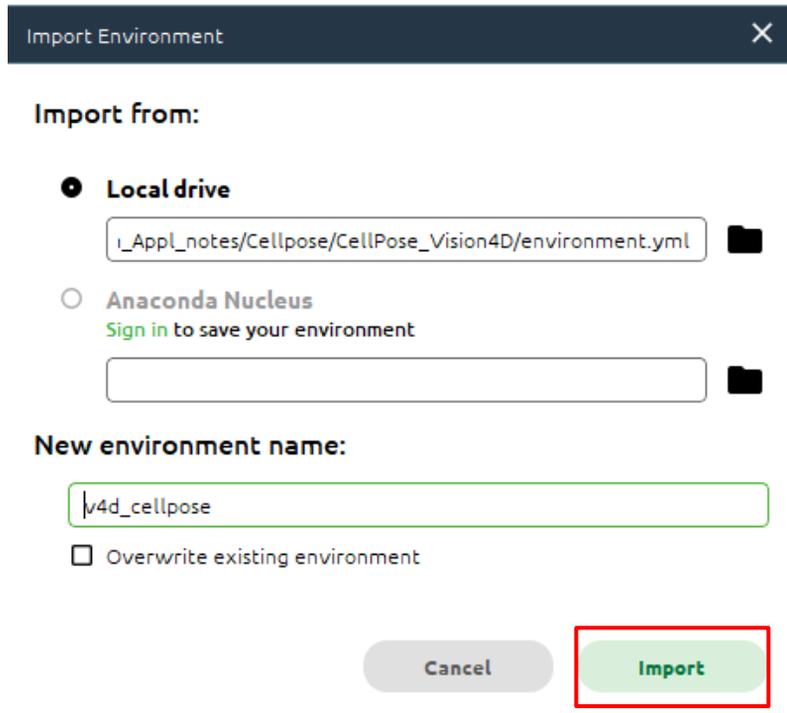


Note :

To install the Cellpose modules, the computer must be connected to internet.

3. Install the Cellpose modules

Step 3.4 In the Dialog box, give a name to your new Cellpose environment and select the downloaded *environment.yml* file and press Import:



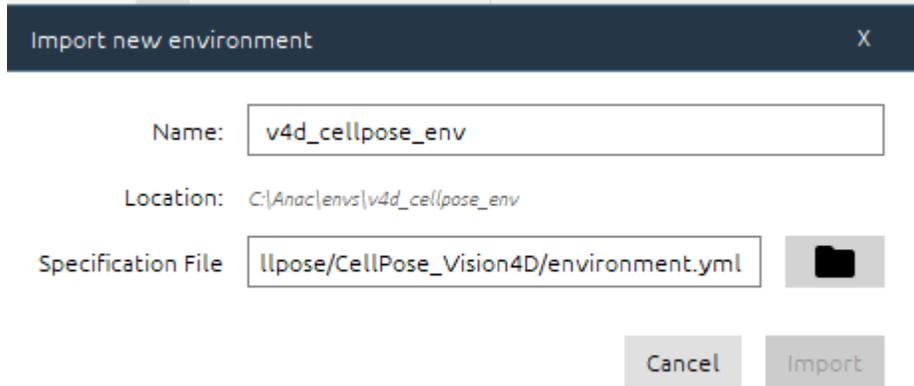
Note :

To install the Cellpose modules, the computer must be connected to internet.

3. Install the Cellpose modules

Step 3.3 Continued

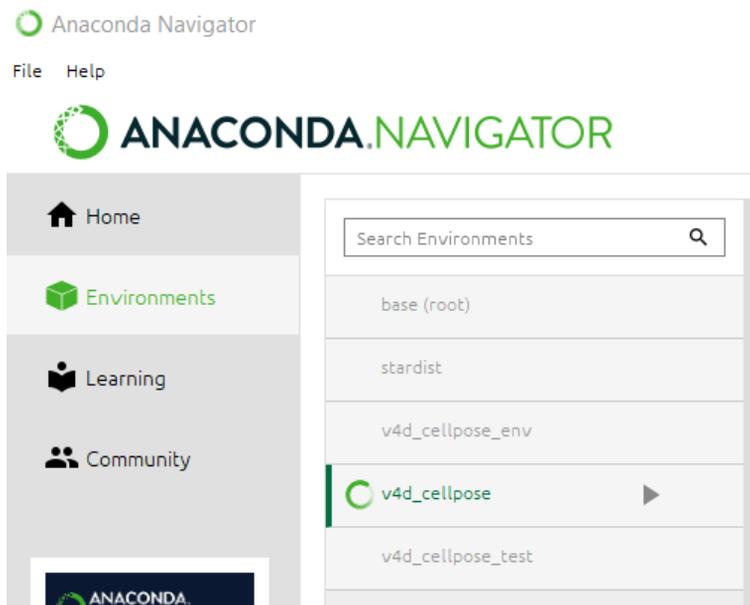
This process can take 10-15 min depending the computer. If the Import button is not green, the given environment likely already exists:



You can monitor the progress on the bottom of the window:



The existing environments and the currently importing one can be checked here:



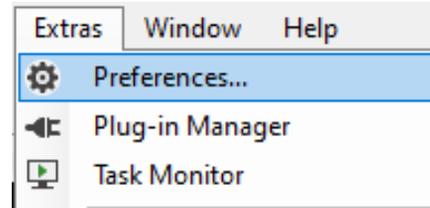
Note :

To install the Cellpose modules, the computer must be connected to internet. In case of problems with the environment setup, please update to the latest Anaconda Navigator release. Please also check the Troubleshooting section of this Application Note.

4. Vision 4D preferences setup

Step 4.1

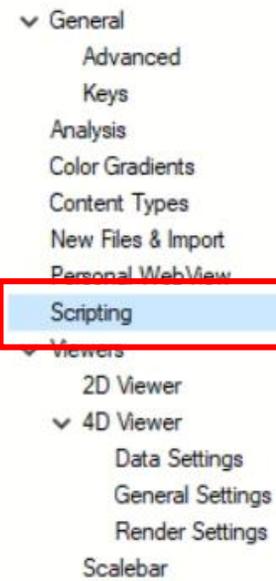
Run Vision4D (3.4 and next release) select the **Preferences** item from the **Extras** menu.



Step 4.2

On the left panel, click on the „**Scripting**“ item.

Preferences - Scripting



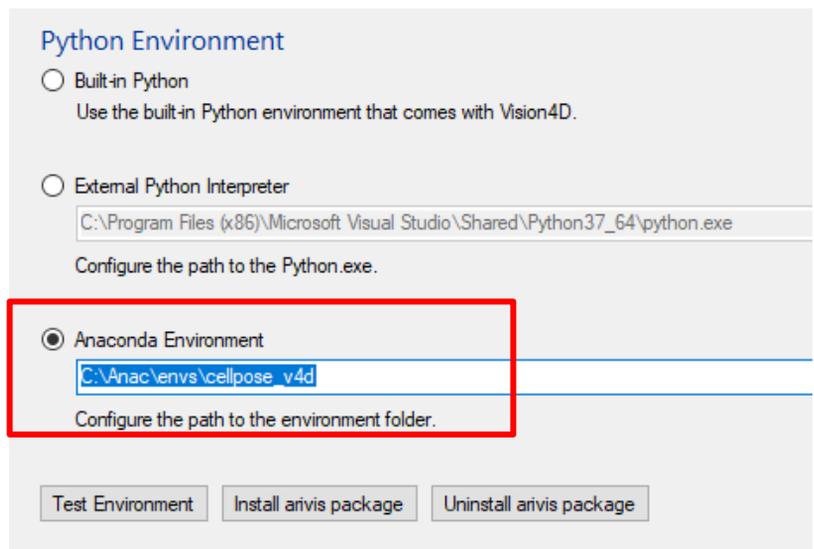
Step 4.3

Enable the “**Anaconda Environment**” option. Browse the Anaconda3 installation folder and select the **scellpose_v4d** environment previously created.

By default, the new environments are stored under the `\envs` folder located in the Anaconda3 installation folder

e.g.

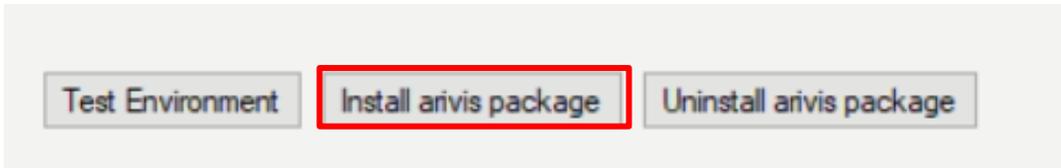
`C:\users\xxxxxx\Anaconda3\envs\scellpose_v4d`



4. V4D preferences setup

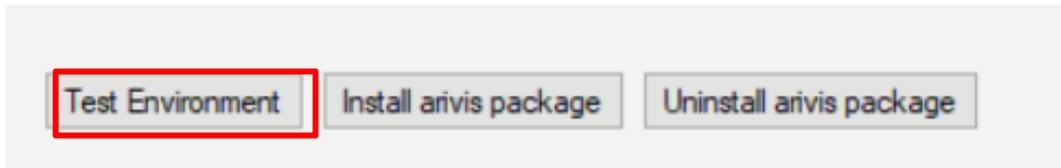
Step 4.4

Install the arivis package

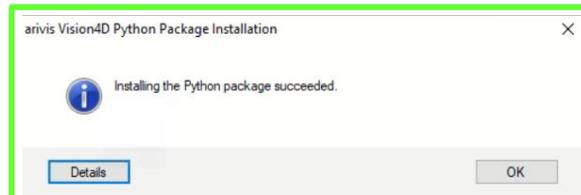
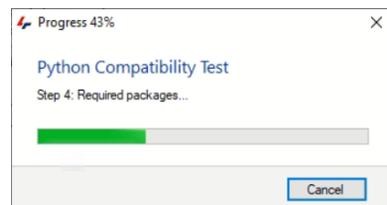


Step 4.5

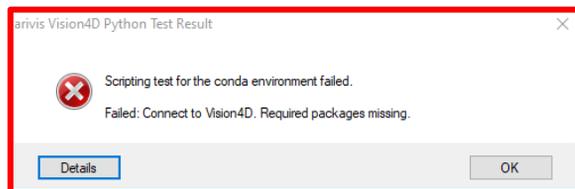
Run the Compatibility test



Test completed successfully



Test failed



Note :

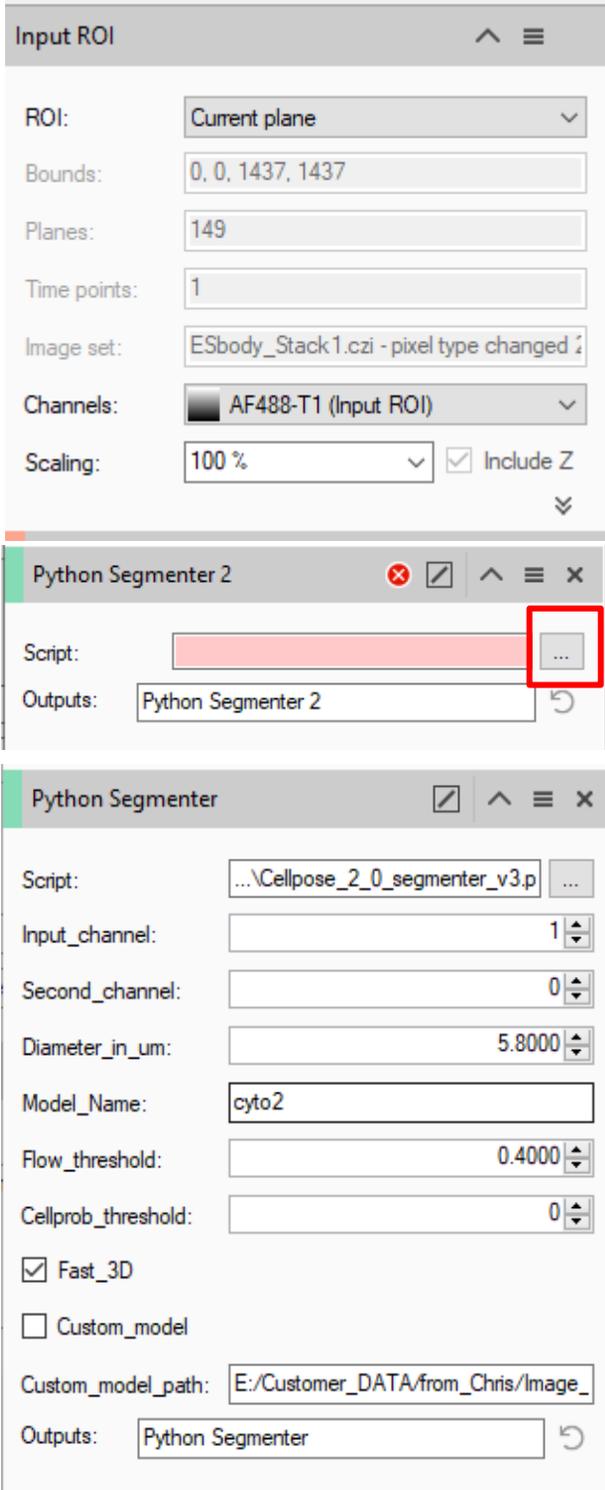
If the left above error message is issued, try again to install the arivis python package.

Note :

Apply the settings and close the preferences panel.

5. Run the Cellpose predictions in V4D

In order to run the Cellpose, a dataset must be opened in V4D. We suggest a simple 3D stack to test the compatibility of the Python environment and adjust the diameter of the objects.



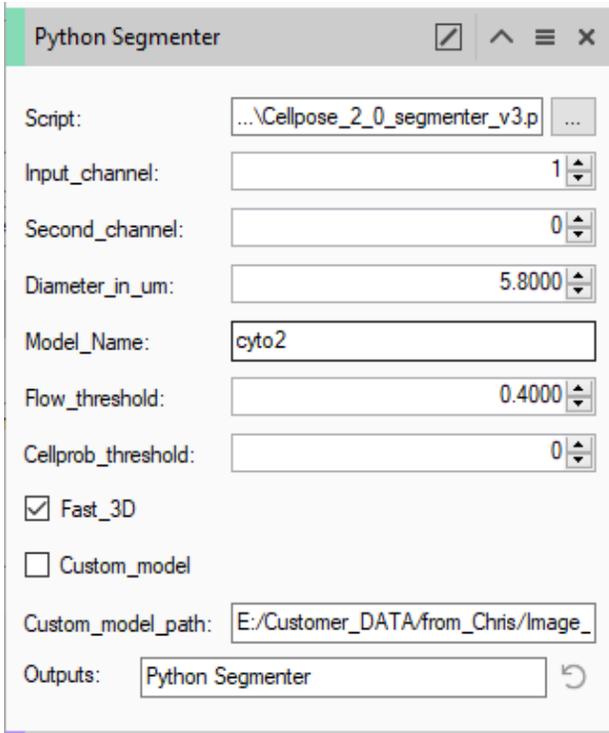
NOTE: On the first Cellpose run, the models will be downloaded onto your default user folder. The computer has to be the connection to Internet.

TIP: beware of the channels imported in the ROI. In the Python image Filter the channel count starts from 1, based on the ROI import and not the raw data.

Select the Python Image Filter operator and use the  button to upload the Cellpose operator script.

Input_channel: channel to run the predictions on.
Second_scannel: additional channel to run the predictions (i.e., with the nuclei). Keep to 0, if not used.
Diameter_in_mkm: diameter of the objects.
Model_Name: select between 'cyto', 'cyto2' or 'nuclei'. *Note: avoid typos!*
Flow_threshold: probability threshold for the prediction results.
Cellprob_threshold: threshold for merging the predicted object masks in 3D between the planes.

5. Run the Cellpose predictions in V4D



The screenshot shows the 'Python Segementer' window with the following settings:

- Script: ...\\Cellpose_2_0_segementer_v3.p
- Input_channel: 1
- Second_channel: 0
- Diameter_in_um: 5.8000
- Model_Name: cyto2
- Flow_threshold: 0.4000
- Cellprob_threshold: 0
- Fast_3D
- Custom_model
- Custom_model_path: E:\\Customer_DATA\\from_Chris\\Image_
- Outputs: Python Segementer

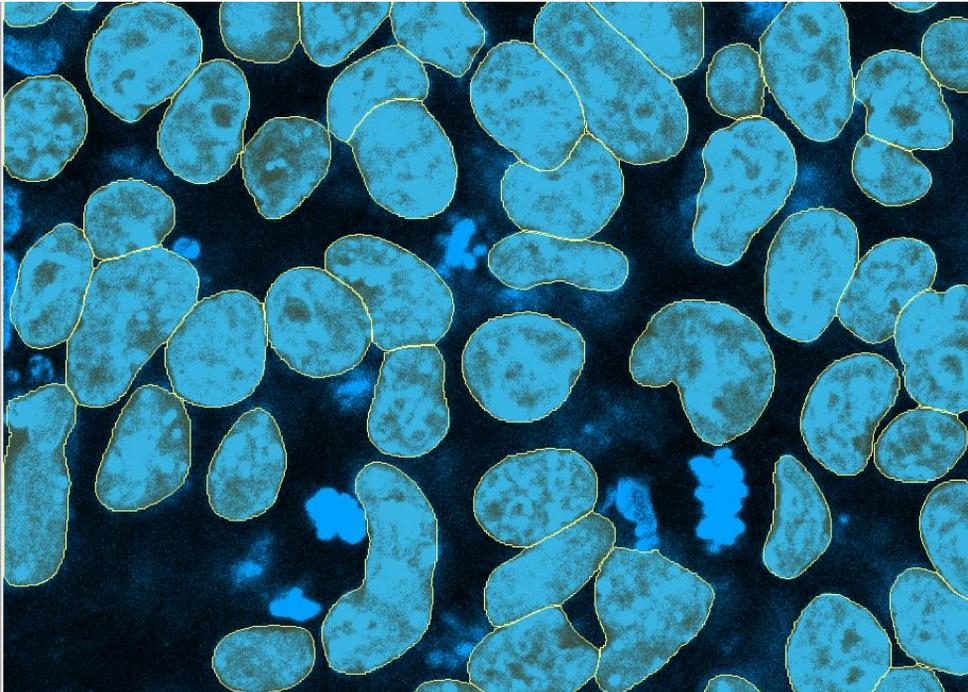
Continued

Fast_3D: if false, the model will be run plane-wise, and the resulting prediction maps will be combined into the 3D object as a 3D stack

Fast_3D: if true, the predictions will be created from all three image stack axes and the result averaged

Custom_model: a custom-trained model will be used instead of the general models

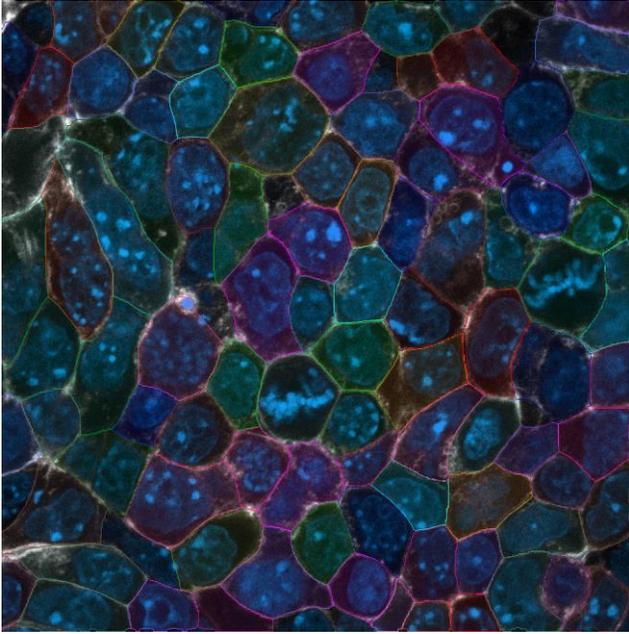
Custom_model_path: path to the trained model. Please use the forward slashes.



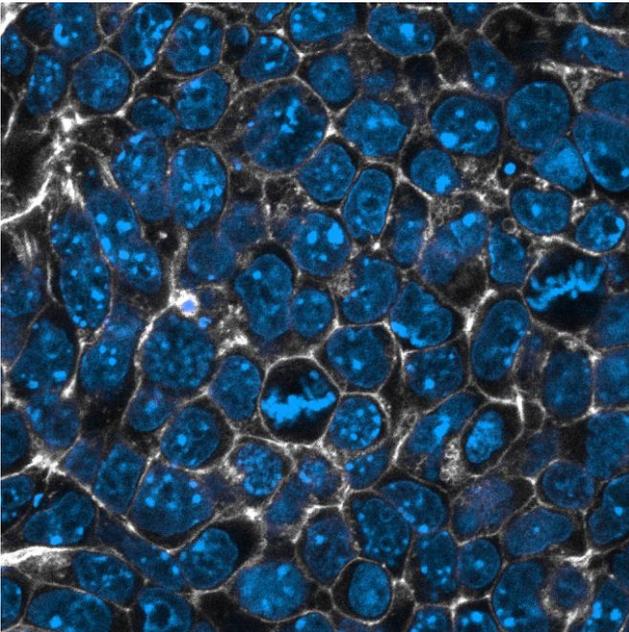
The objects created with the Cellpose Python Image Filter and the Labeled Image Segementer can be classified, tracked, filtered, grouped, used for the compartmentalization analysis within the same or separate Vision4D pipeline.

6. Examples of the Cellpose prediction results

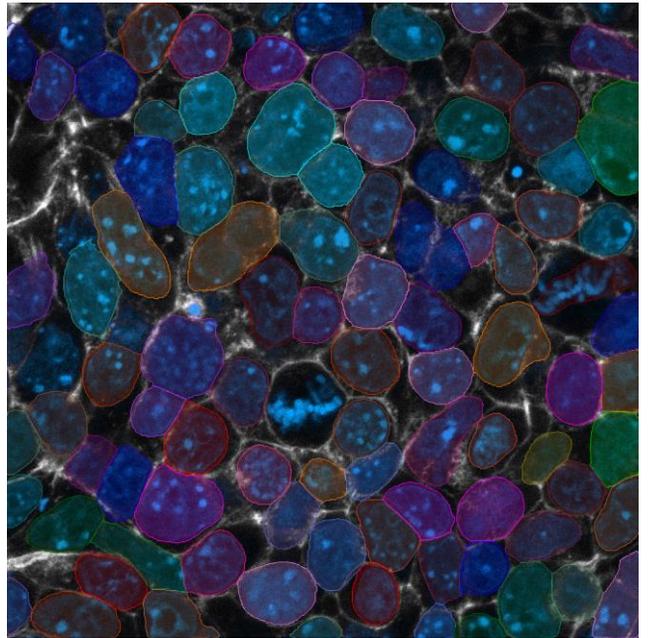
cyto2 model on the cytoplasm staining:



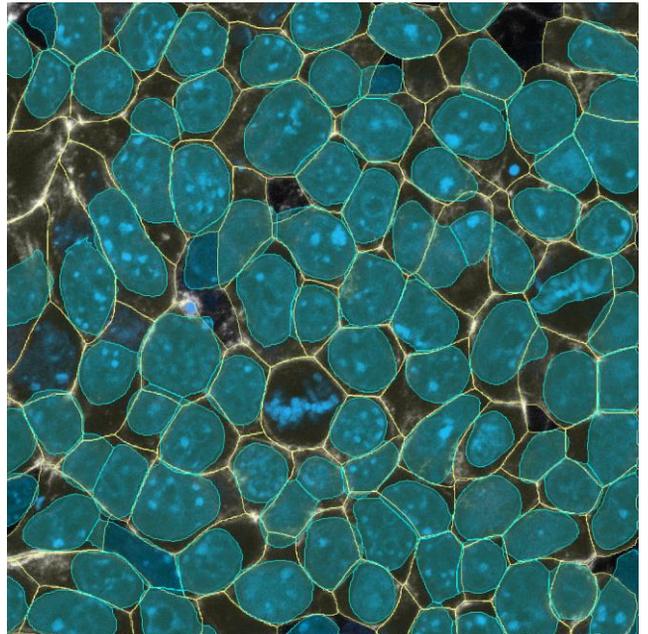
Raw data:



cyto2 model on the nuclei staining:



Overlay of the Nuclei and Cytoplasm objects on the raw data:



The images used in this application is a courtesy from Dr. Masahiro Narimatsu (Dr. Wrana's lab), Lunenfeld-Tanenbaum Research Institute.

8. Troubleshooting the Cellpose operator-script

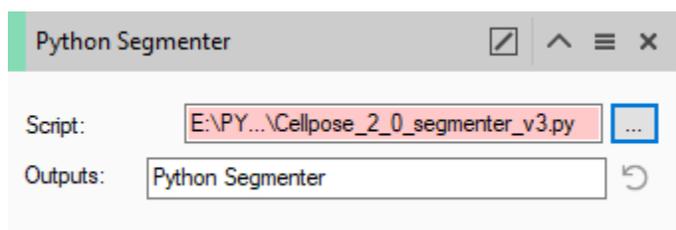
- If no labeled masks were created or if there is a corresponding error message, please check the Diameter_in_mkm value and adjust it.
- If an error message regarding the image size mismatch arises, please check the selection of the channels in the ROI Import and in the Python Image Segmenter operator.

```
2022-01-31 22:47:45,601 [22] STDERR:WARNING: no mask pixels found
```

- Currently, there is no tiling strategy implemented in these scripts, since that would introduce artifacts on the borders of the tiles. Cellpose has a certain limit to the stack size. The error message would look like below:

```
2022-02-02 09:49:48,752 [2] STDERR: Remote error message: Array dimensions exceeded supported range. Exception has been thrown by the target of an invocation.
```

- In case the environment is not set up correctly, the first UI in the Python Segmenter will remain red:



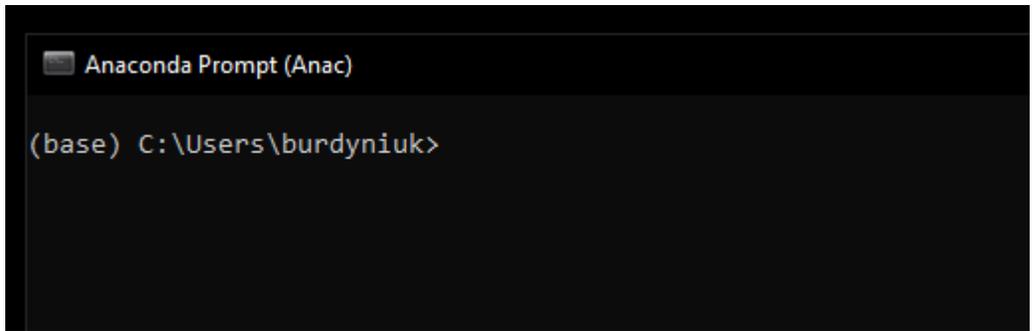
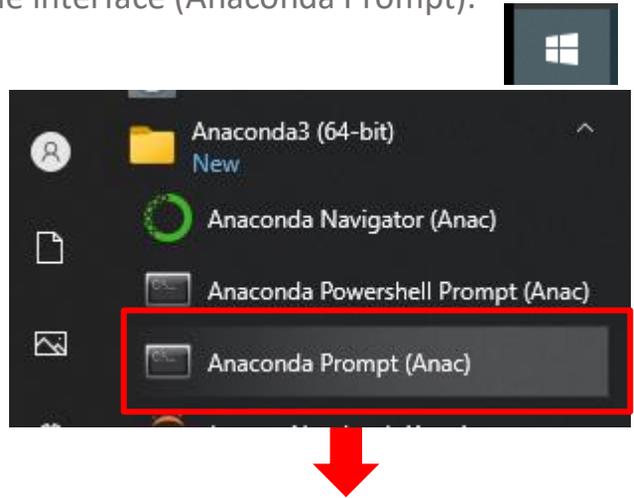
9. Troubleshooting the Cellpose setup

Cellpose environment can be created directly in the Anaconda prompt.

Step 1. Open the Anaconda console interface (Anaconda Prompt).
Click on the Window icon

Step 1.1

Click on the Window icon and select the *Anaconda Prompt* item.

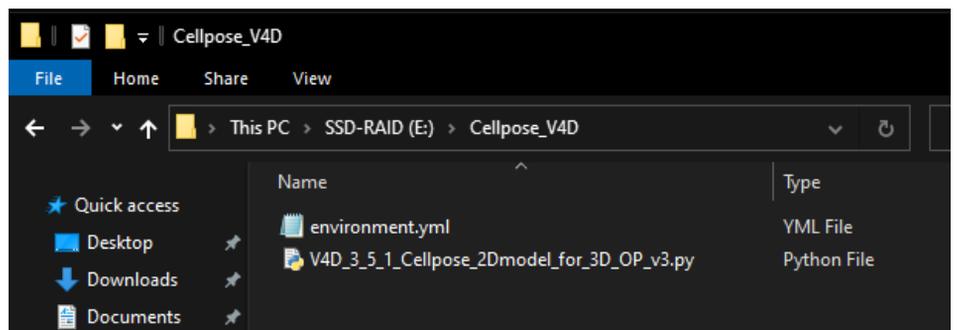


Step 2

Create the Cellpose environment

Step 2.1

Download the *environment.yml* file (as well as the Cellpose Operator script) to your computer:



Note :

To install the Cellpose modules, the computer must be connected to internet.

9. Troubleshooting the Cellpose setup

Step 2.2

Change directory to the other disk, if necessary:

```
Anaconda Prompt (Anac)
(base) C:\Users\burdyniuk>e:
(base) E:\>
```

Step 2.3

Change directory to the *environment.yml* file:

```
Anaconda Prompt (Anac)
(base) C:\Users\burdyniuk>e:
(base) E:\>cd E:\Cellpose_V4D
(base) E:\Cellpose_V4D>_
```

Step 2.4

digit `conda env create -f environment.yml` and press *return*

```
(base) E:\Cellpose_V4D>conda env create -f environment.yml_
```

Step 2.5

Activate the new environment (*cellpose_v4d*)

digit «*cellpose_v4d*» and press *return*.

The Cellpose environment name is now shown at the beginning of each row.

```
(base) E:\Cellpose_V4D>conda activate cellpose_v4d
(cellpose_v4d) E:\Cellpose_V4D>_
```

Note: Cellpose 2.0 allows training the custom models for 2D data. To install the Cellpose GUI to perform custom training and/or to create the Cellpose environments for GPU-computing, please follow the official Cellpose documentation at [Installation — cellpose 0.7.2 documentation](#)

To enable connecting the new environment to Vision4D, please install pywin32 module:

```
Administrator: Anaconda Powershell Prompt (Anac)
(base) PS C:\Windows\system32> conda activate cellpose
(cellpose) PS C:\Windows\system32> conda install pywin32_
```



Contact the arivis application support to receive additional technical details about the topic described in the document.

*arivis AG, Am Kabutzenhof 21,
18057 Rostock, Germany*

Email : support@arivis.com