Application Note #35
StarDist: A Deep Learning Application

How to run StarDist within arivis Vision4D

The goal of this application note is to guide the user in working with StarDist, the well-established deep learning application focused on cell / nuclei detection, within the arivis Vision4D (V4D) environment. A pre-defined StarDist neuronal network training is required. At the time of writing only a 2D training is supported by V4D (3D coming soon).

More information about StarDist can be gathered from the following articles:

- *Star-convex Polyhedra for 3D Object Detection and Segmentation in Microscopy*
- *Cell Detection with Star-convex Polygons*

Please note that the recommended Anaconda Python package and all StarDist modules (Python release) must already be installed on your workstation and fully tested.

Application Note Purpose

StarDist is reported to run well in multiple open source environments such as Fiji/ImageJ, Qupath or Python (Using Python editor like Jupyter Notebook, PyCham and Spider)

But what are the benefits of integrating StarDist directly inside of your arivis Vision4D imaging software? In this document, we will highlight the advantages of integrating open source analysis tools such as StarDist directly in Vision4D.

Preliminary Remarks

Vision4D is able to run deep learning applications such as StarDist using external and arivisindependent Python libraries and tools produced by third parties.
These tools must be installed by the user under his or her own responsibility, strictly following the instructions in this document. arivis has tested the setup protocol on several computers, however, due to the different and not predictable hardware and software configurations of any given computer system, the results may vary on a case by case basis. Therefore, arivis declines any responsibility concerning the correct tools, installation and setup on the individual user's workstation. arivis cannot be made responsible for 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. Furthermore, arivis also declines any responsibility about the validity of the scientific results gathered from the deep learning application.

NOTE:
QuPath **is cross-platform, user-friendly open source software for digital pathology and whole slide image analysis, written using JavaFX.**
ImageJ **is public domain software for processing and analyzing scientific images, with many derivatives and variants, including ImageJ2, Fiji, and *others*.**
Jupyter Notebook, PyCham and Spider **are Python editors**

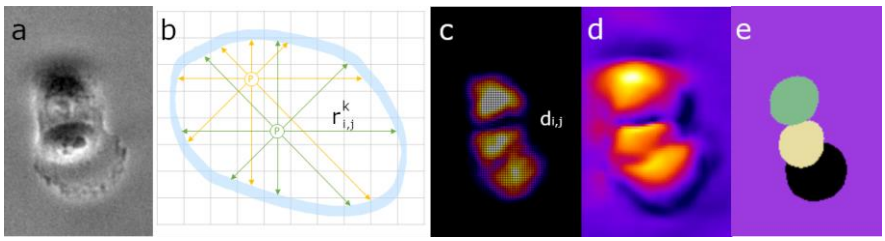
Application Overview

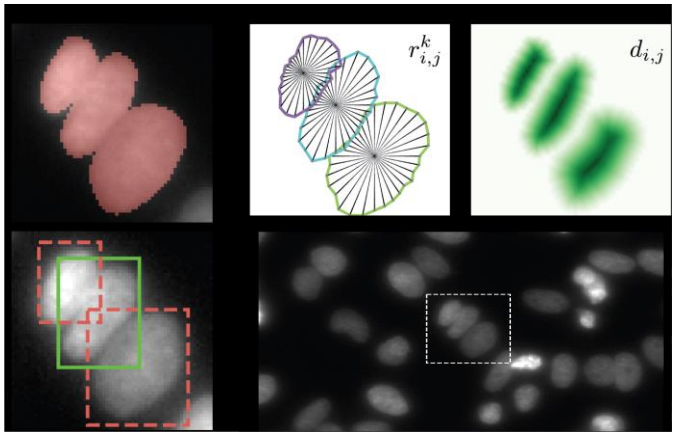
StarDist in a Nutshell

StarDist is a deep learning-based method for 2D and 3D nucleus detection, developed & published by Martin Weigert and Uwe Schmidt: github.com/stardist/

StarDist uses a cell detection method that predicts a shape representation with star-convex polygons that is well-suited to approximate the typically roundish shapes of cell nuclei in microscopy images.

The 3D shape of a single object (cell nucleus) is described using a star-convex polyhedron instead of polygons.

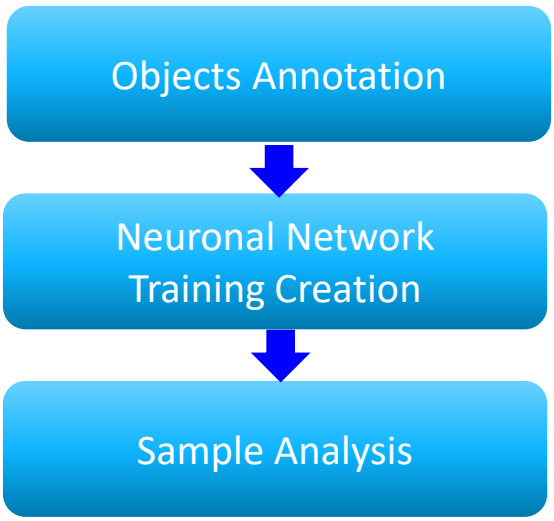




Application Overview

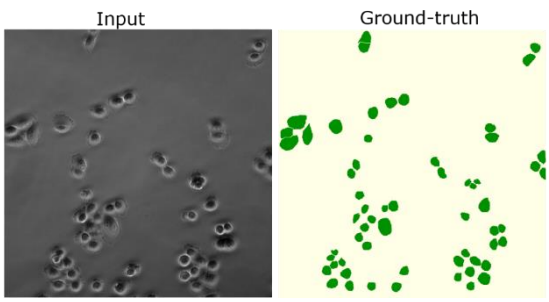
How does it work?

The StarDist workflow is based on three main steps:



a. Objects Annotation

This task consists in manually drawing the objects shape over a set of representative images (2D or 3D). The reference objects should describe all their possible variation within the reference samples. The annotations are then used to create a binary masked image (Ground-truth). Both the annotations and the related binary masks are used afterward by the training task to build the Neuronal Network (training)



Application Overview

How does it work? (continued)

A. Objects Annotation (continued)

The annotation task is a manual activity and therefore requires a lot of time to be performed. The correct number of annotations must be estimated in advance in order to get a reliable training. During the training the annotations can be increased if required. The annotations range starts from a minimum number of 200 samples, spread over 10 - 20 different images, up to thousand (or even tens of thousand) in the more complex cases.

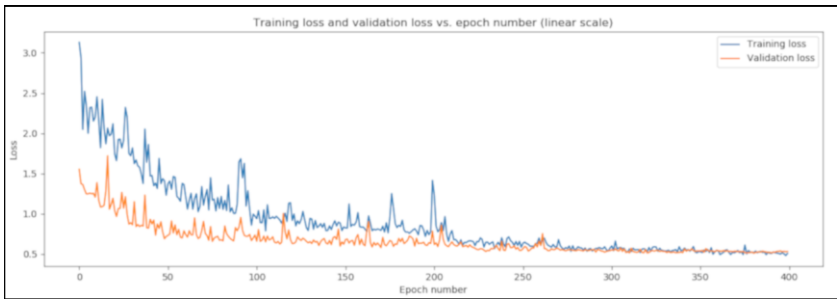
In order to increase the number of samples without the need to acquire new images, it is also possible to re-use the already existing images after applying some operations like rotations, random flips or intensity shifts of the original.

b. Neuronal Network Training Creation

This task takes either the sample images or the related masked images to build the Neuronal Network. The training is a loop in which, in any cycle, 2 parameters are computed: the training loss and the validation loss.

The progress of training can be evaluated by comparing the training loss with the validation loss. During training, both values should decrease before reaching the minimal value, which should not change significantly with further cycles. Comparing the validation loss development with the training loss can give insights into the model's performance. Decreasing of both training and validation loss indicates that training is still necessary. If the validation loss suddenly increases again, while the training loss decreases towards zero, it means that the network is overfitting to the training data.

The training loss and the validation loss vs the number of cycles.



Application Overview

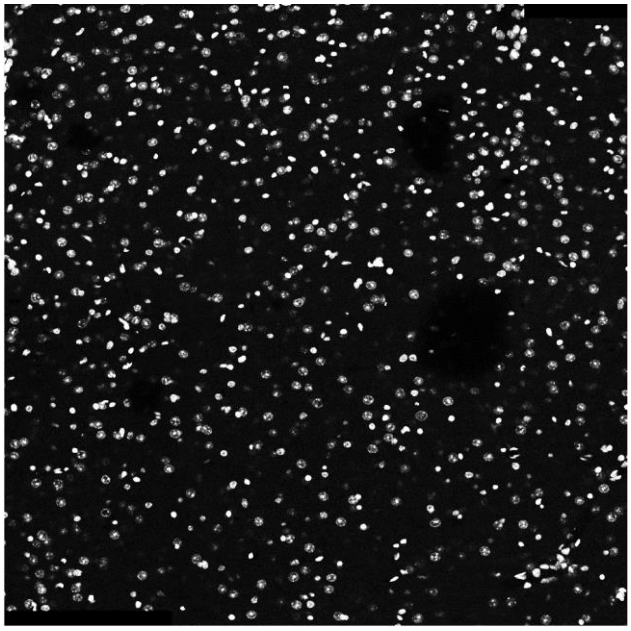
How does it work? (continued)

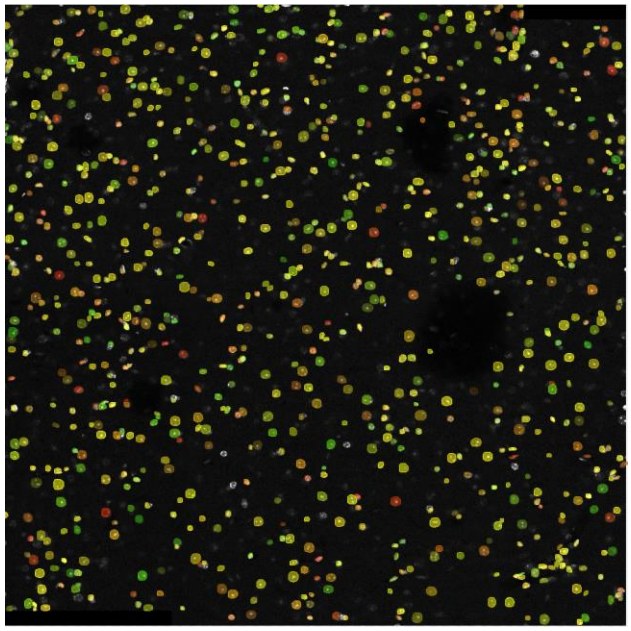
b. Neuronal Network training creation (continued)

Basically, the training is based on math operations. These operations are repetitive and time consuming and can easily be parallelized.
The usage of GPU resources improves the training performance in reducing the total time. Working with the CPU only, a complex training can take 7 to 10 days of work, while using the GPU the total time is reduced to mere hours (10 to 12).

c. Dataset Analysis

Once the Neuronal Network is ready, it can be used to analyze the samples.



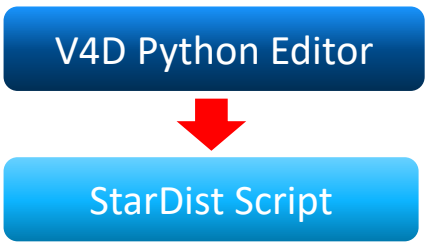


Application Overview

Why should I use StarDist 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 dataset 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 iterations 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 cannot internally generate the Neuronal Network model. 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.
The script imports a pre-trained model from a defined folder.

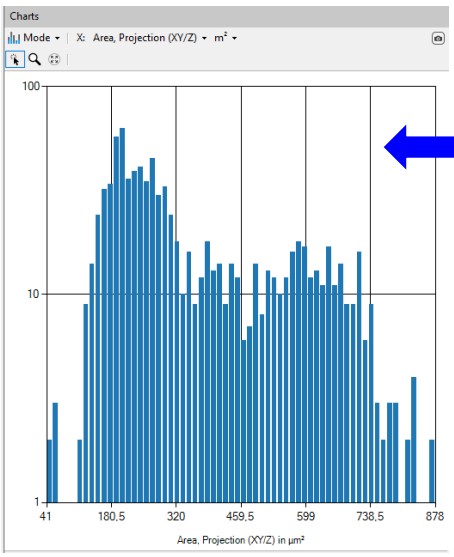
Application Overview

Why should I use StarDist within V4D? (continued)

Application examples

1. Nuclei Quantification

Nuclei Detection (Segments)

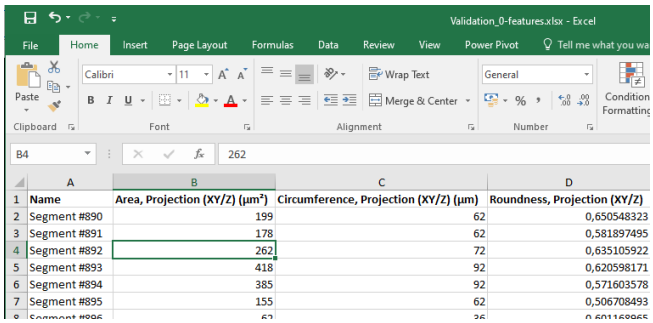


Distribution of the Nuclei area

Feature Columns... | Go to | Summary | Im/Export...

Name	Area, Projection (XY/Z) (µm²)	Circumference, Projection ...	Roundness, Projection (XY/Z)
Segment #890	199,000	62,000	0,651
Segment #891	178,000	62,000	0,582
Segment #892	262,000	72,000	0,635
Segment #893	418,000	92,000	0,621
Segment #894	385,000	92,000	0,572
Segment #895	155,000	62,000	0,507
Segment #896	62,000	36,000	0,601

Nuclei features (e.g. Area, Roundness, Circumference)



	A	B	C	D
1	Name	Area, Projection (XY/Z) (µm²)	Circumference, Projection (XY/Z) (µm)	Roundness, Projection (XY/Z)
2	Segment #890	199	62	0,650548323
3	Segment #891	178	62	0,581897495
4	Segment #892	262	72	0,635105922
5	Segment #893	418	92	0,620598171
6	Segment #894	385	92	0,571603578
7	Segment #895	155	62	0,506708493

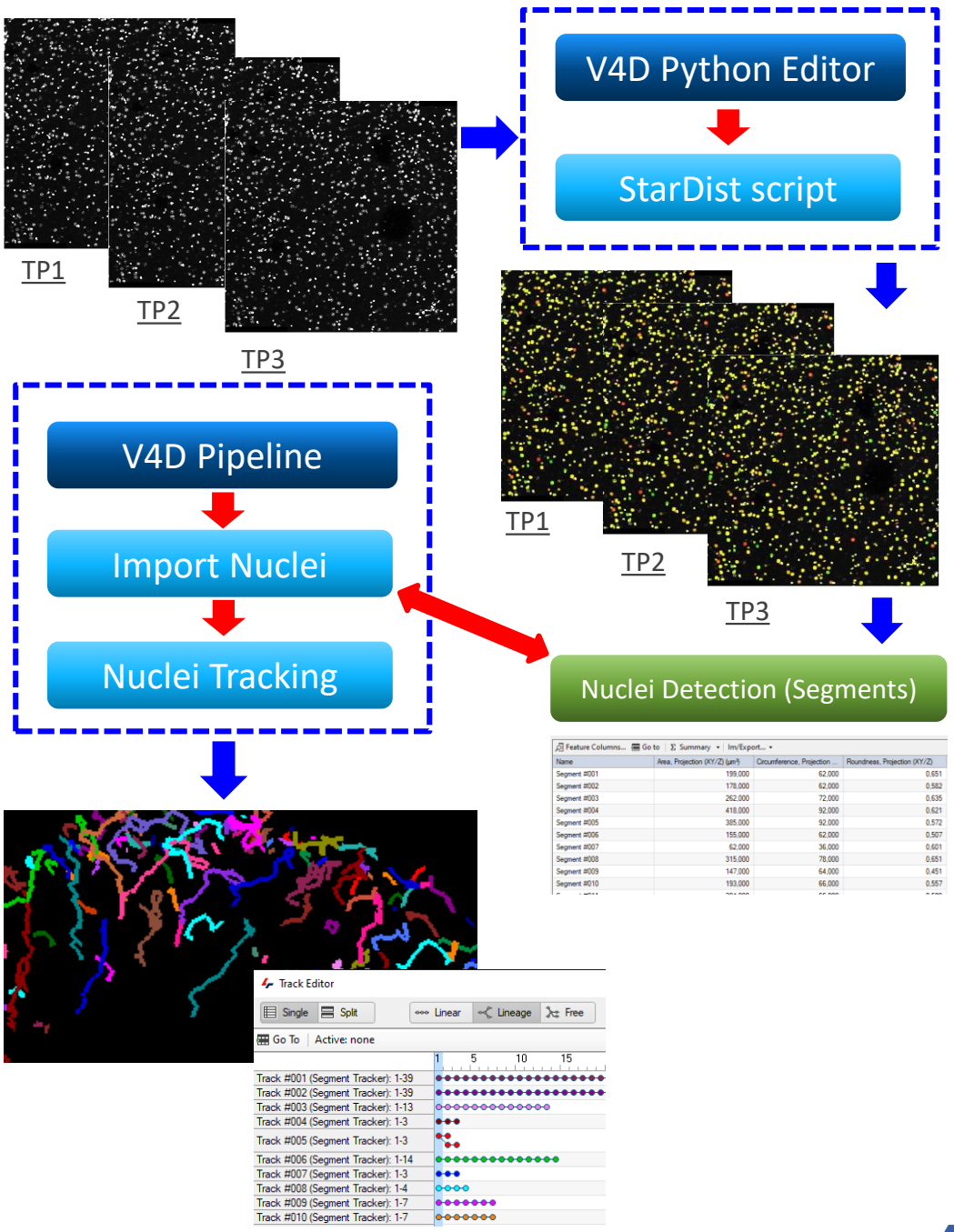
Nuclei features export to Excel

Application Overview

Why should I use StarDist within V4D? (continued)

Application examples:

2. Nuclei Tracking

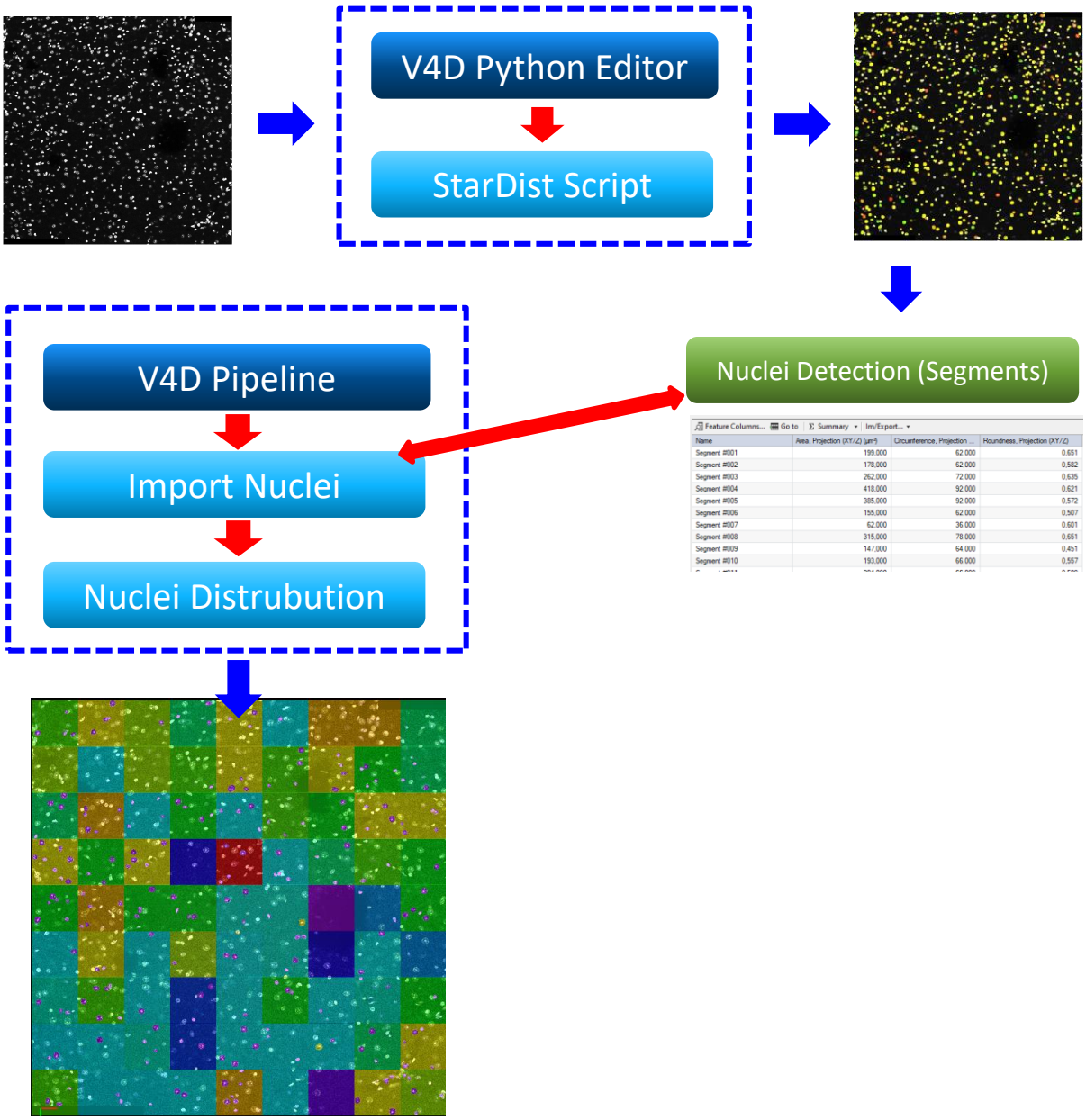


Application Overview

Why should I use StarDist within V4D? (continued)

Application examples:

3. Nuclei Distribution (density map)



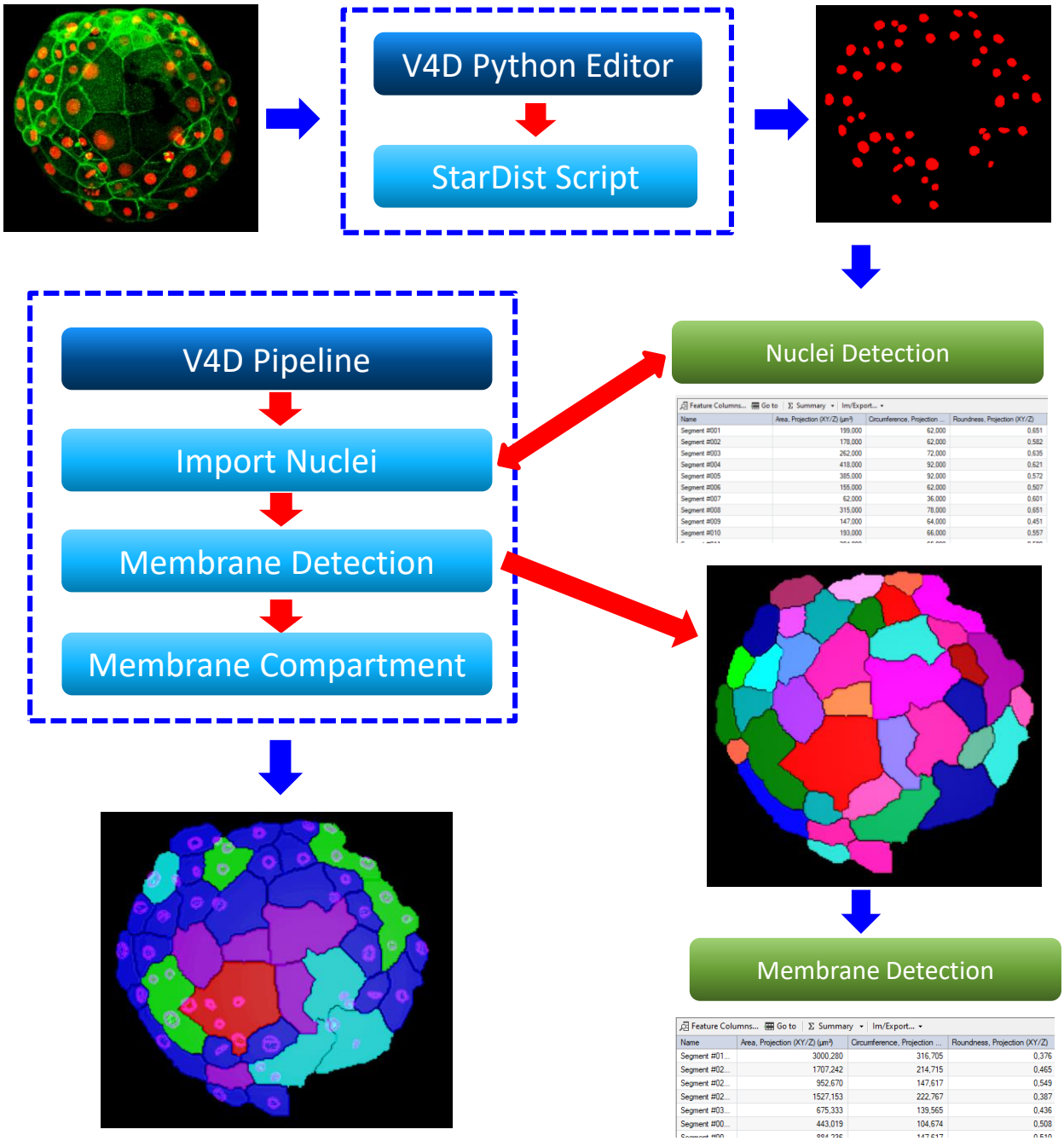
Heatmap (Nuclei density distribution)

Application Overview

Why should I use StarDist within V4D? (continued)

Application examples:

4. Cells / Nuclei Compartmentalization



Application Overview

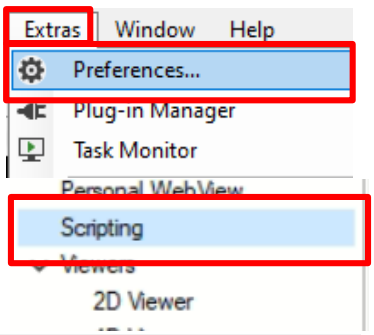
How to get StarDist working with V4D?

The StarDist python package is required to get it to work with V4D and must be added to an existing python 3.x enviroment. We tested & strongly recommend the Anaconda 3.x python package for the scope of this application.
Once the StarDist package has been correctly set up, V4D must also be configured accordingly by changing the scripting preferences.

Application Note #29 and #30 describe how to install the Anaconda3 environment and the StarDist package. Please refer to these for more details.

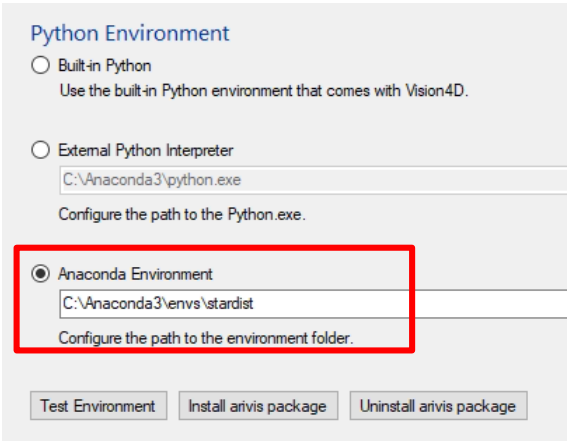
Set up V4D preferences

1. Start Vision4D (min. 3.4+ releases) and select the ***Preferences*** item from the ***Extras*** menu.

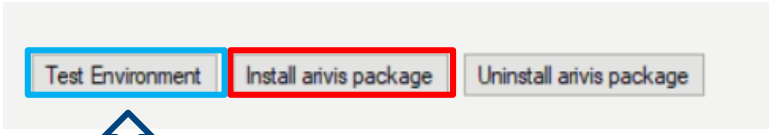


2. On the left panel, click on the ***Scripting*** item.

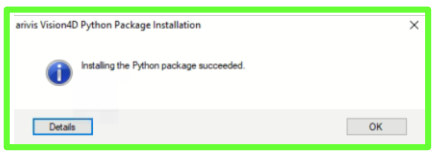
3. Enable the **Anaconda Enviroment**: Browse the “Anaconda3” installation folder and select the ***StarDist*** environment previously created.
By default, the new enviroments are stored under the **\envs** folder located in the Anaconda installation folder
e.g. C:\Anaconda3\envs\stardist



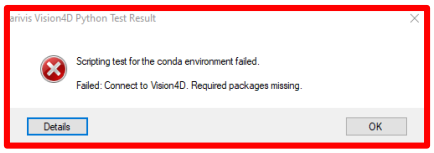
4. Run “Install arivis package” and »Test Environment» for compatibility:



Test completed successfully



Test failed



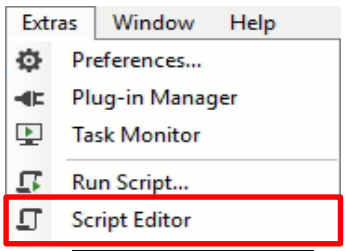
Application Overview

How to run the StarDist application?

arivis provides a free python script to run the StarDist algorithm inside of V4D. The script allows to set the active channel on which the algorithm will be applied as well as the time points (all or the active one) and the Z planes (full stack or a range of planes). A new channel is created to store the labeled objects found by StarDist.

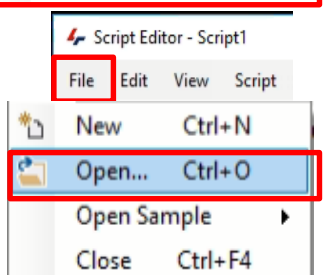
Load the script

1. Open Python Script Editor.
From the «***Extra***» menu, select the «***Script Editor***» item.



2. Open the ***Stardist_2D.py***.

3. Set the ***Stardist_2D.py*** parameters.



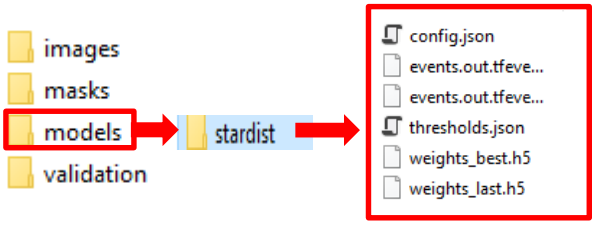

```
INPUT_CHANNEL = 0     # <---- Count start from 0 (Ch#1 == 0)
# name of the new channel (skeleton storage)
OUTPUT_CH_NAME = "Stardist "        #  Net Skel
```

INPUT_CHANNEL : Set the channel on which Stardist will be applied. ***Count starts from 0***
OUTPUT_CH_NAME : Set the name of the result channel

```
MODEL2D_PATH = "D:/Arivis_Dataset/Carlo Stardist/2021-05-10/2D/2021-05-03/models"
MODEL_NAME = "stardist"
```

MODEL2D_PATH: **Set the full path on which the pre-trained 2D model is stored**
MODEL_NAME : **Set the name of the 2D model**.

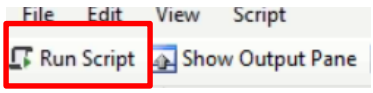
The pre-trained models are stored in a folder structure as shown here:




```
CURRENT_TIME_POINT = True
FIRST_PLANE = -1     # -1 == BOTTOM
LAST_PLANE = -1      # -1 TOP
```

CURRENT_TIME_POINT: **True** == Only the active time point will be processed.
False == All available time points will be processed.
FIRST_PLANE : Set the bottom plane from which the process is applied.
LAST_PLANE : Set the top plane from which the process is applied.
NOTE: If both are set to -1, the full stack is used

4. Click "***Run Script***" (F5)



A startup package including the python script, the technical instructions and the test image is available on request.
Contact your arivis local area sales manager to get more information about how to get the python script mentioned here.

Contact the arivis application support to receive additional technical details about the topic described in the application note, or how to adapt the application workflow to your requirements.

"The quantitative analysis of the images represents the art of transforming a visual sensation into its schematic and discrete form allowing its univocal description, classification and mathematical and logical interpretation of its spatial and temporal components"

arivis AG, Am Kabutzenhof 21,
18057 Rostock, Germany

Email : support@arivis.com