

Theory and Practical Uses of SuRVoS Workbench

Michele C. Darrow, Imanol Luengo, Avery
Pennington, Olly King, Matt Spink, Win Tun, Mark
Basham

SuRVoS Project

University of Nottingham:

- Computer Vision Laboratory

Diamond Light Source:

- B24: Cryo Transmission X-ray Microscopy



The University of
Nottingham



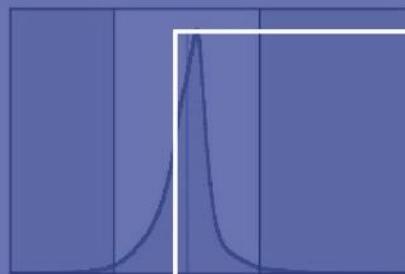
diamond

Contrast

VMin: -6.69 VMax: 2.70
-6.69 2.70

 View Histogram

Default

**Layers****Data**

Data: 100

Super-Regions

SuperVoxels: 100

MegaVoxels: 100

Annotations

Level 6: 100

Predictions

Predictions: 50

[6] Total Variation

Slice Viewer

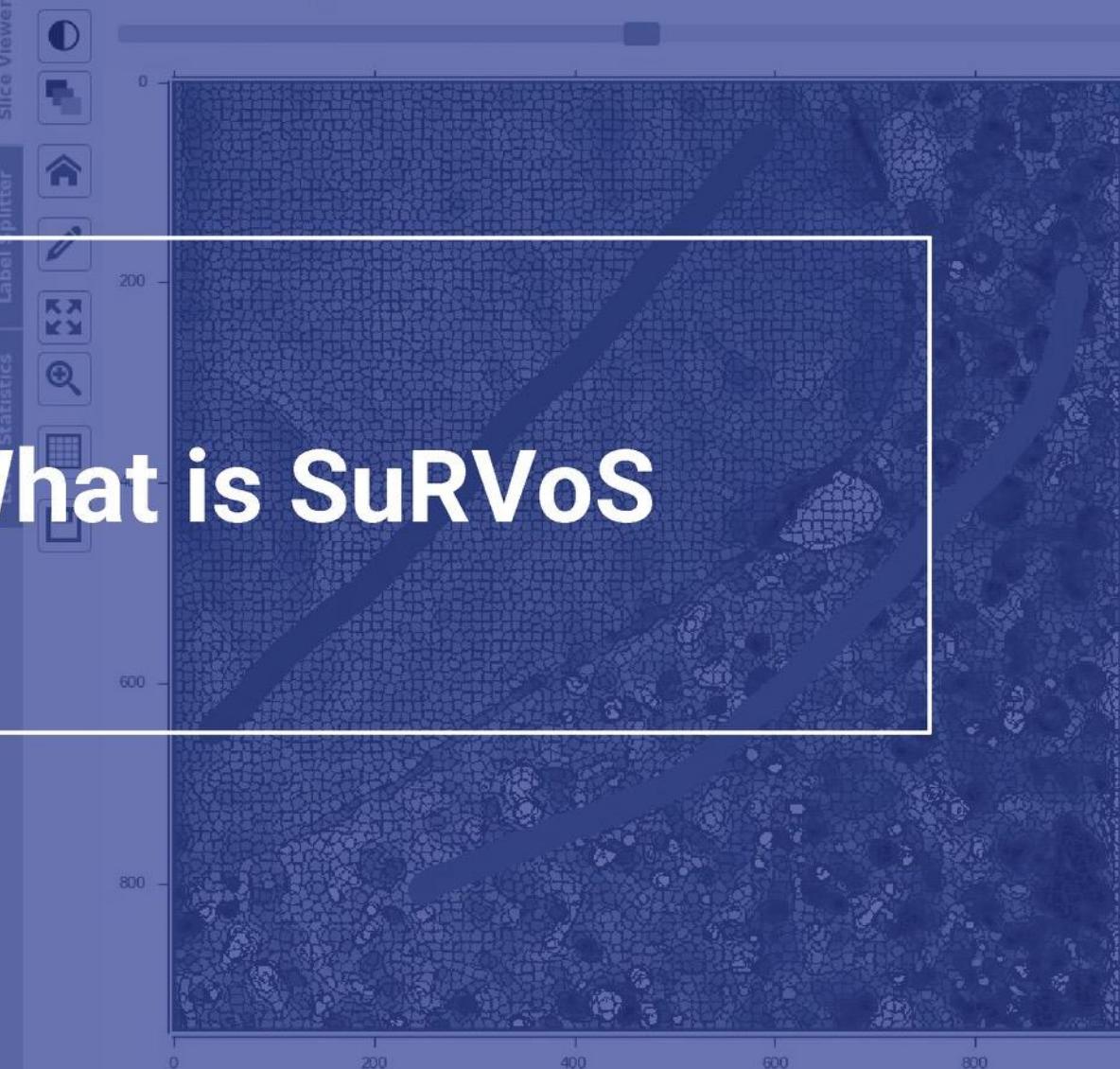
Splitter

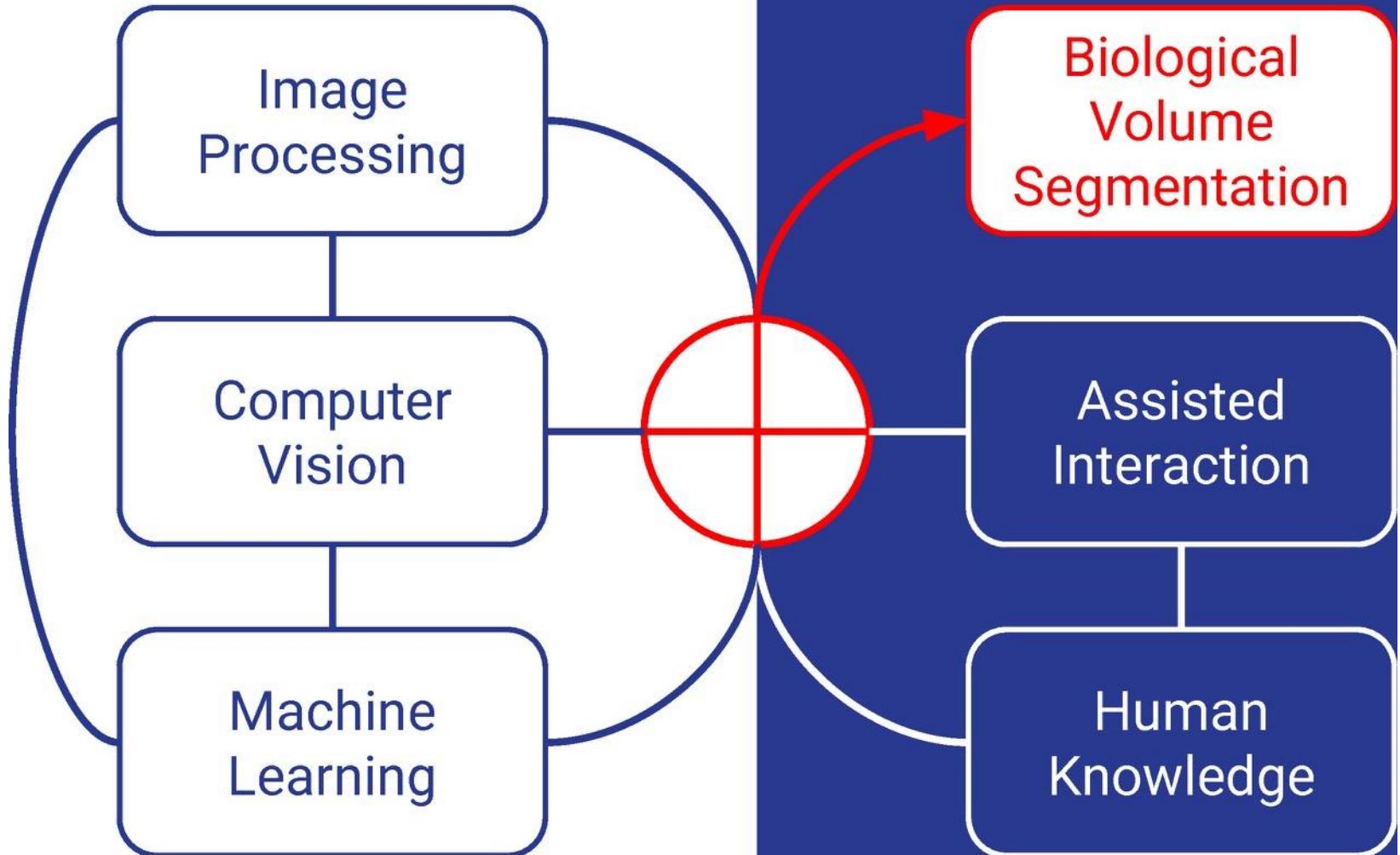
Statistics

Annotations

Predictions

What is SuRVoS

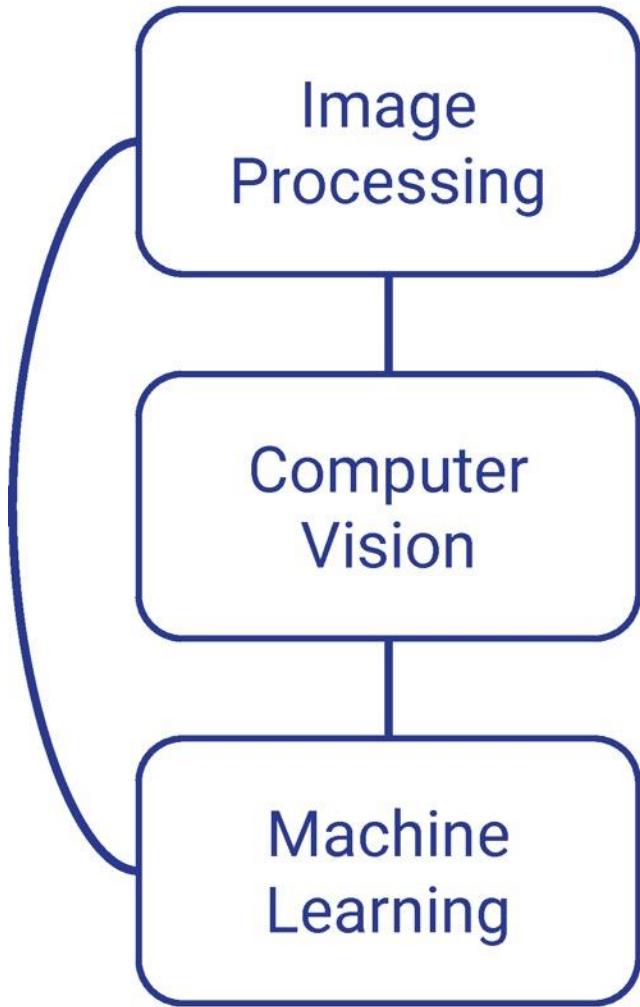




Computational
Techniques

Human Users

Computational Techniques



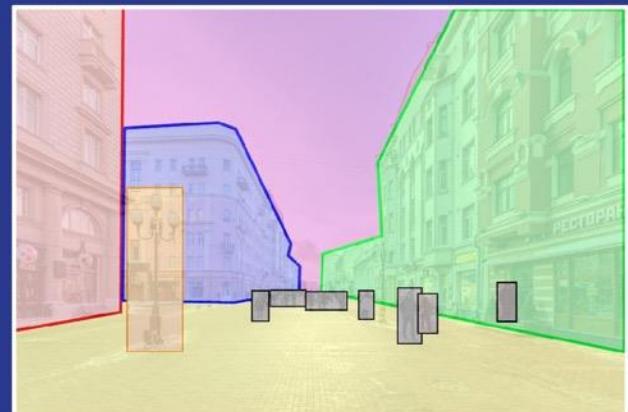
Computer Vision

Digital Image Understanding

Processing

Analysis

*



* Original Image: https://en.wikipedia.org/wiki/Arbat_Street

Computational Techniques

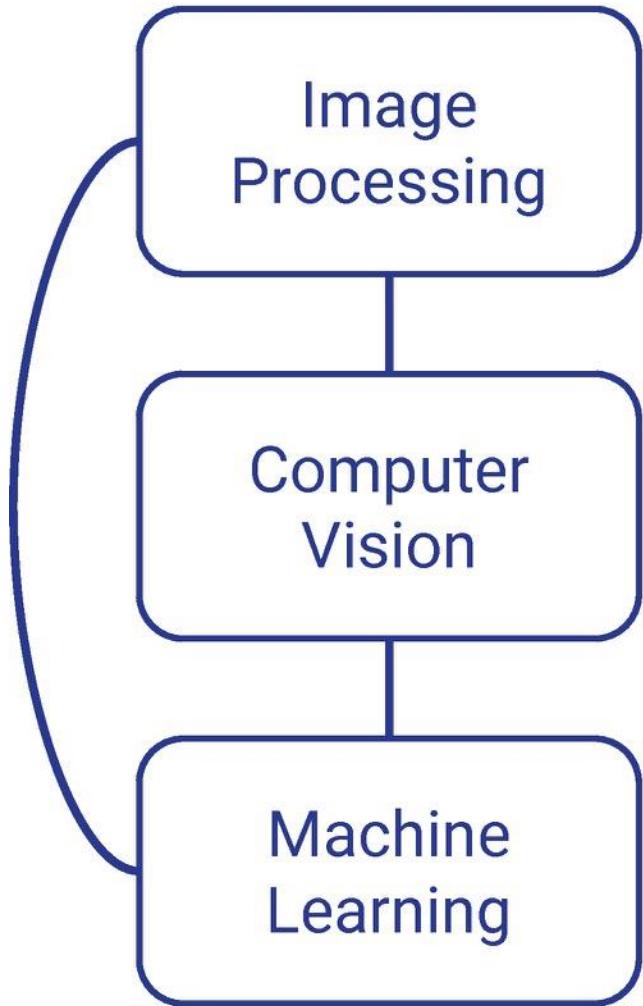


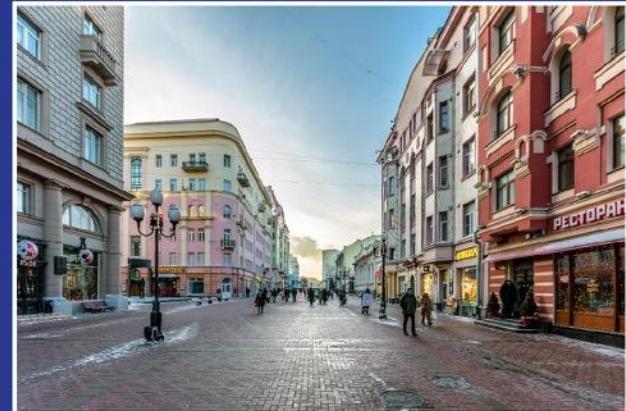
Image Processing

Image Manipulation and Enhancing

Noise Reduction

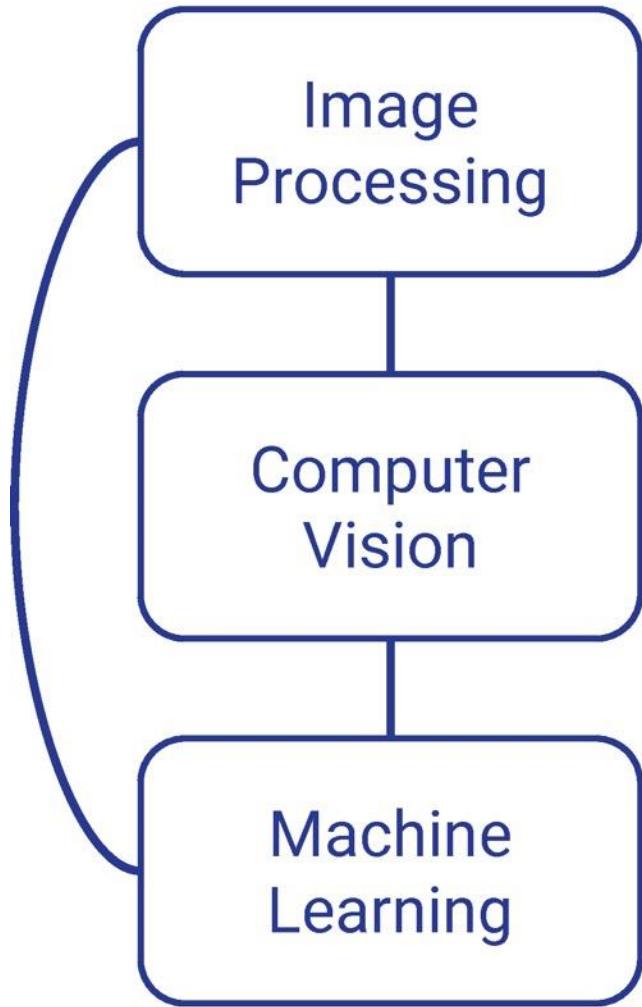
Feature Extraction

*



* Original Image: https://en.wikipedia.org/wiki/Arbat_Street

Computational Techniques



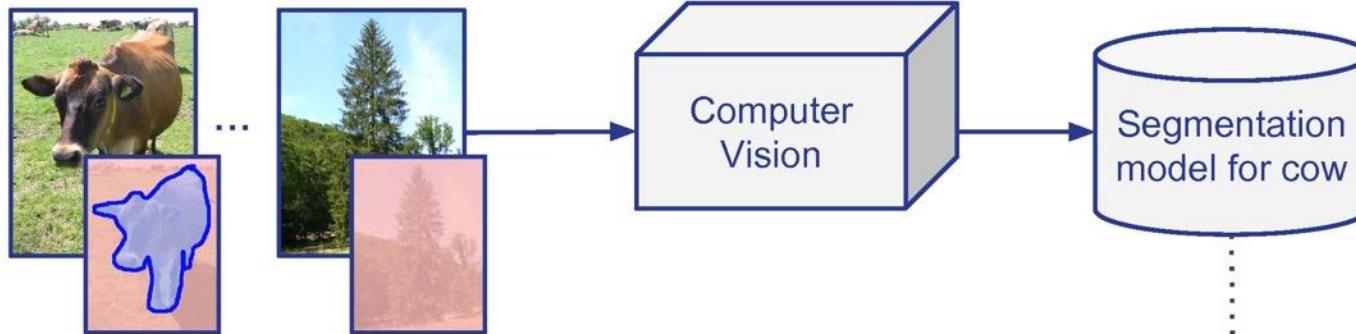
Machine Learning



* Original Image: https://en.wikipedia.org/wiki/Arbat_Street

Automatic Computer Vision

Training: Learning to identify cows



Testing: Predicting if there is a cow in the image



Cow Image: https://commons.wikimedia.org/wiki/Cattle#/media/File:Jersey_cattle_in_Jersey.jpg,
<https://commons.wikimedia.org/wiki/Cattle#/media/File:Braunvieh06.JPG>

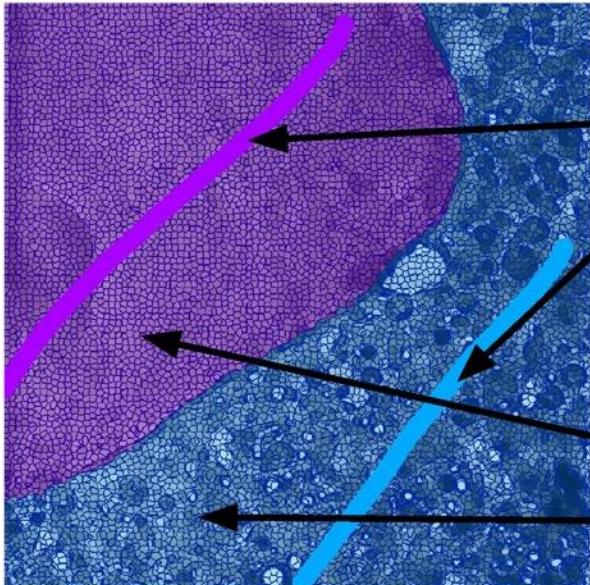
Tree Image: <https://commons.wikimedia.org/wiki/Tree#/media/File:GemeineFichte.jpg>

SuRVoS

Biological Volume Segmentation

Problem

- Different imaging modalities / cell type
- Organelles have different shape / appearance
- **No previous training data is available**



Annotations

Nucleus

Cytoplasm

Predictions

Nucleus

Cytoplasm

SuRVoS

- Assist the user to annotate data.
- Learn to segment with user annotations.

Assisted
Interaction

Human
Knowledge

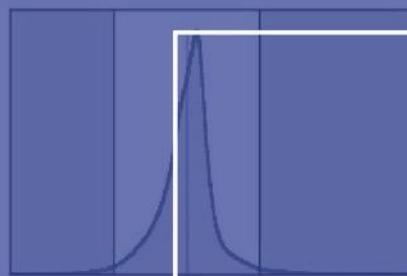
Human Users

Contrast

VMin -6.69 -2.78
VMax -6.69 2.70 8.46

 View Histogram

Default

**Layers****Data**

Data: 100

Super-Regions

SuperVoxels: 100

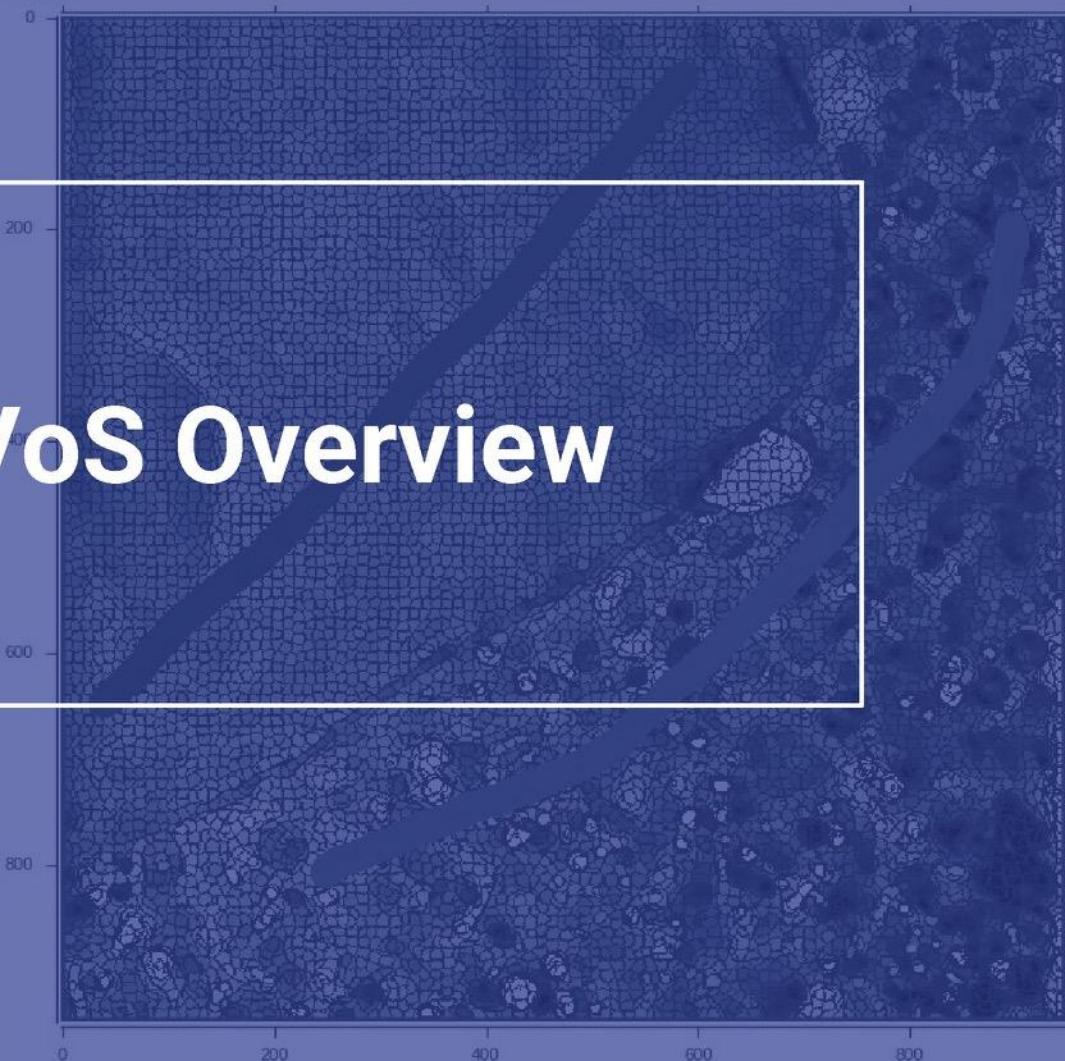
MegaVoxels: 100

Annotations

Level 6: 100

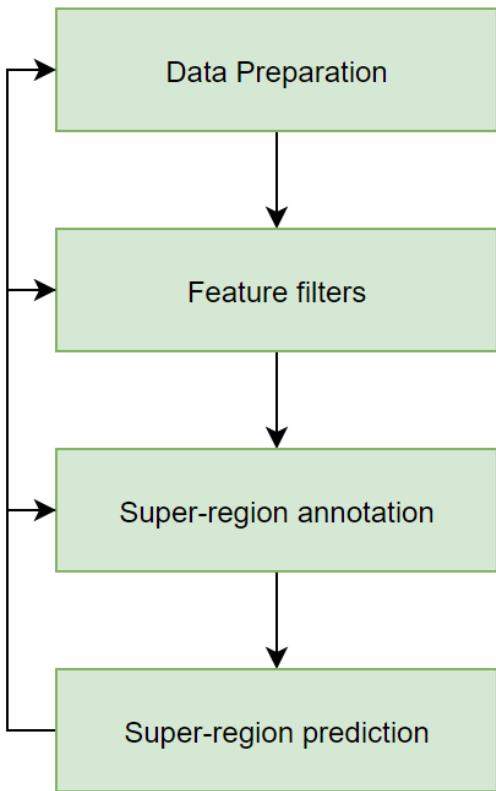
Predictions

Predictions: 50

[6] Total Variation

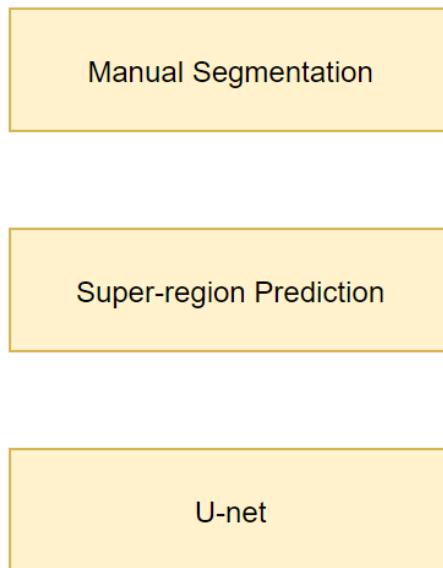
SuRVoS Overview

Annotation



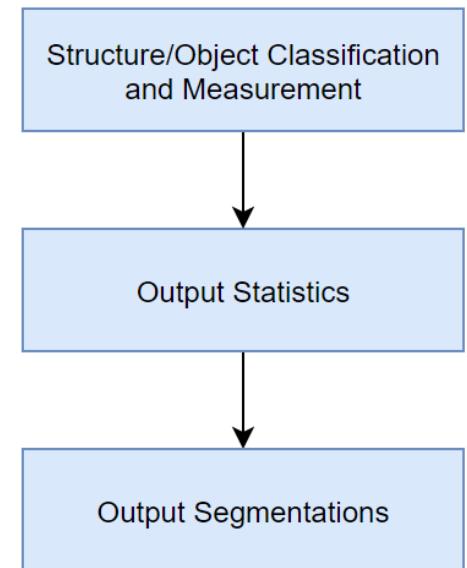
An iterative process is followed allowing the data to be explored and the features of interest determined, generally working from coarse structures to finer structures.

Segmentation strategies



Different approaches to producing a final segmentation.

Post-processing



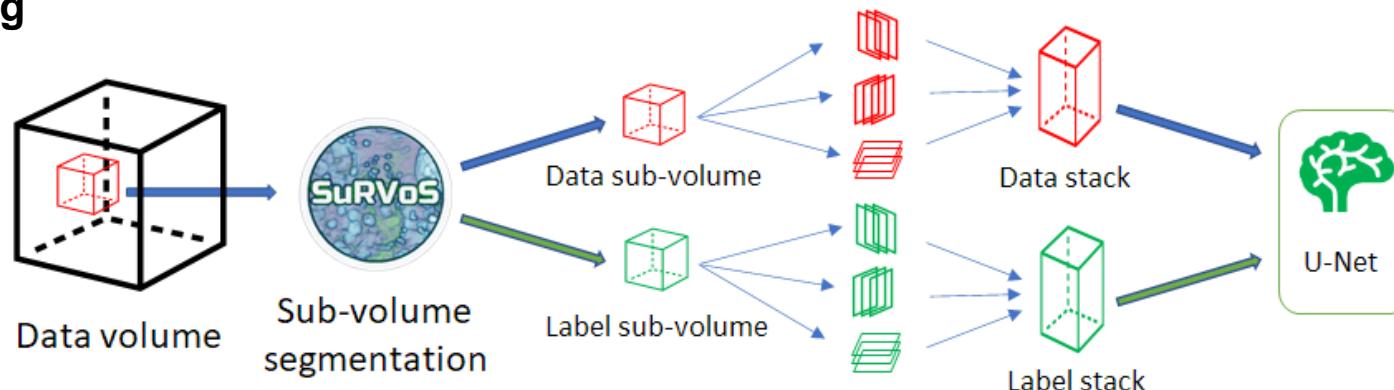
Once a final segmentation is obtained, extracting final structure/object and producing appropriate measurements and statistics, and outputting the segmentations for further visualization.

U-Net/CNN Prediction

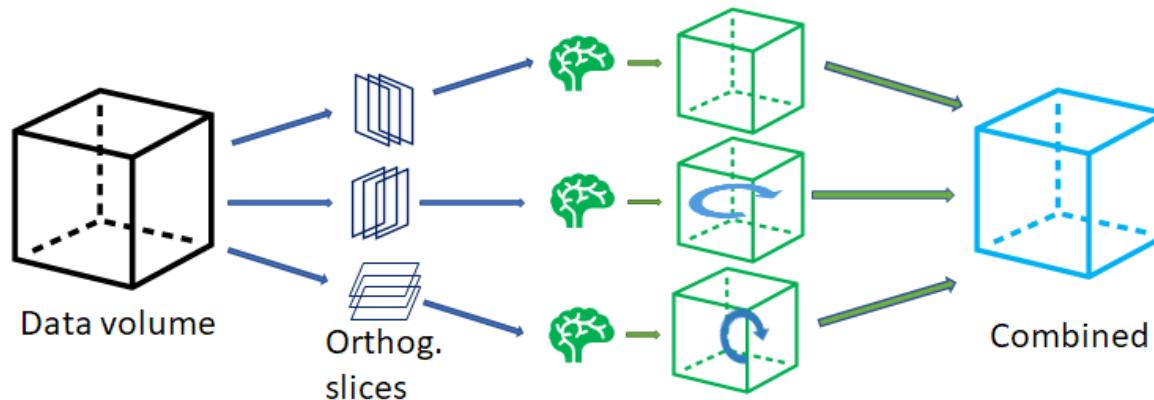
- CNN-based segmentation models, particularly the U-net, are a state-of-the-art segmentation model when trained with sufficient, high-quality annotation (and when predicting on a suitable volume).
- The U-net is provided as a plugin in the Pipelines panel.
- The u-net can be trained on a volume e.g. a 128x128x128 cube of precisely annotated data.
- By using SuRVoS supervoxel annotation and the super-region segmentation pipeline, it is possible to annotate a detailed volume of biological image data and export it as training data for a U-net model.

U-Net: Training and Predicting in Multiple Dimensions

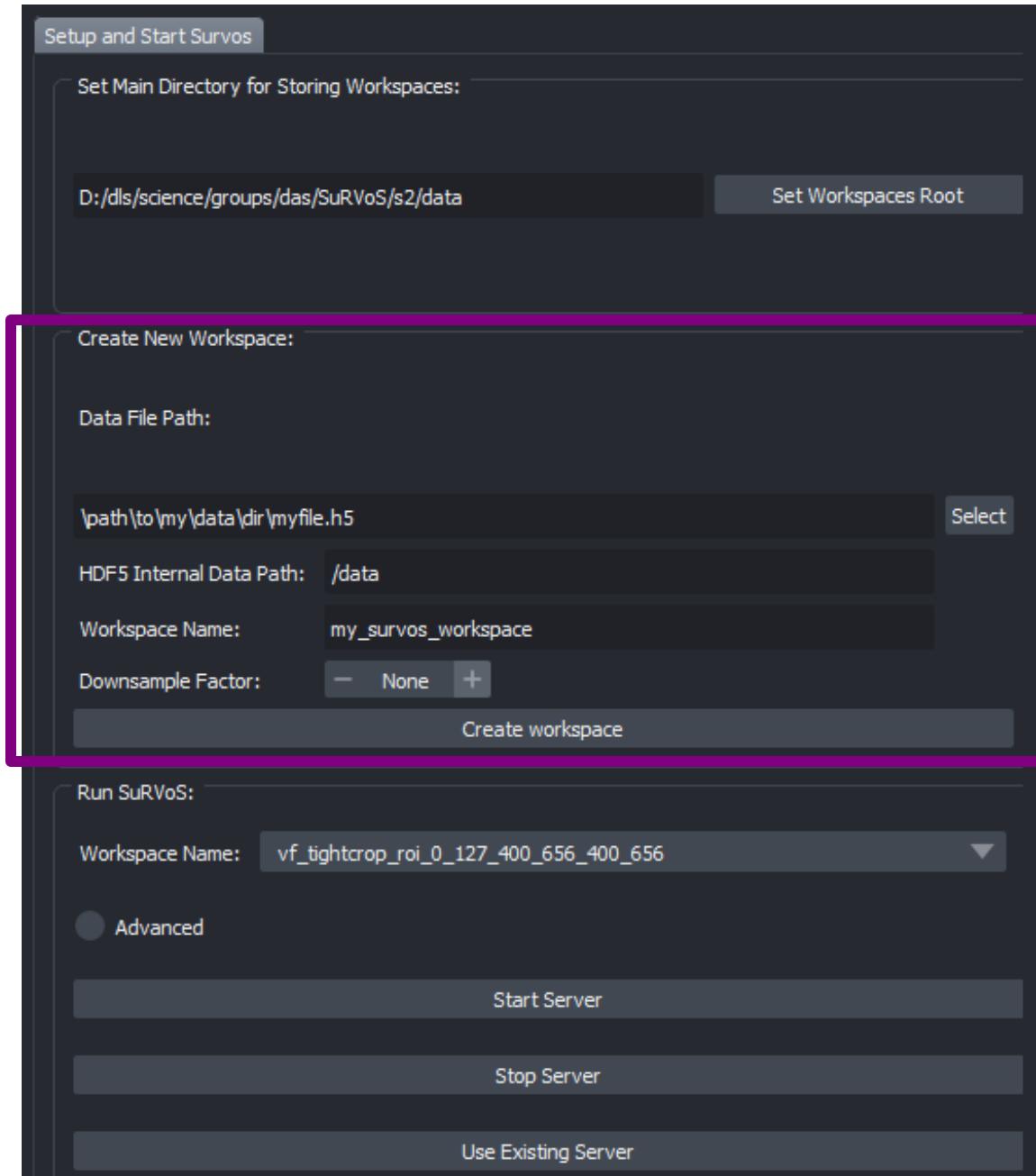
Training



Prediction



Running SuRVoS



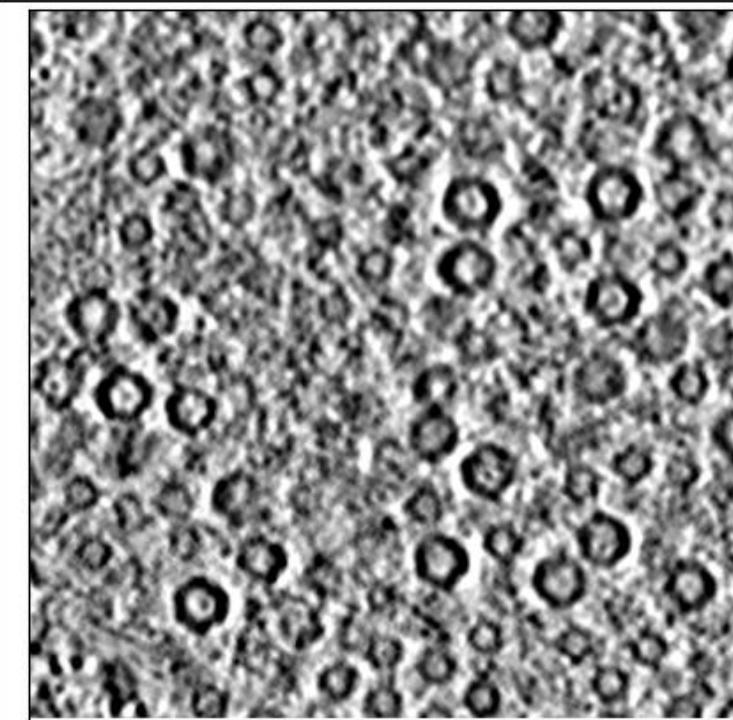
1. For a new image, under Data File Path click Select.
2. (Next slide) Use the ROI selection panel to select an ROI
3. Give the workspace a name
4. Click 'Create Workspace'

Preview Dataset

0

81

162



Input Dataset:

C:/datasets/huntd2_test.tif



Internal HDF5 data path:

None selected

Select Region of Interest:

Drag a box in the image window or type manually

Axis	Start Value:	End Value:	Apply ROI
------	--------------	------------	-----------

x:	0	512	Reset ROI
----	---	-----	-----------

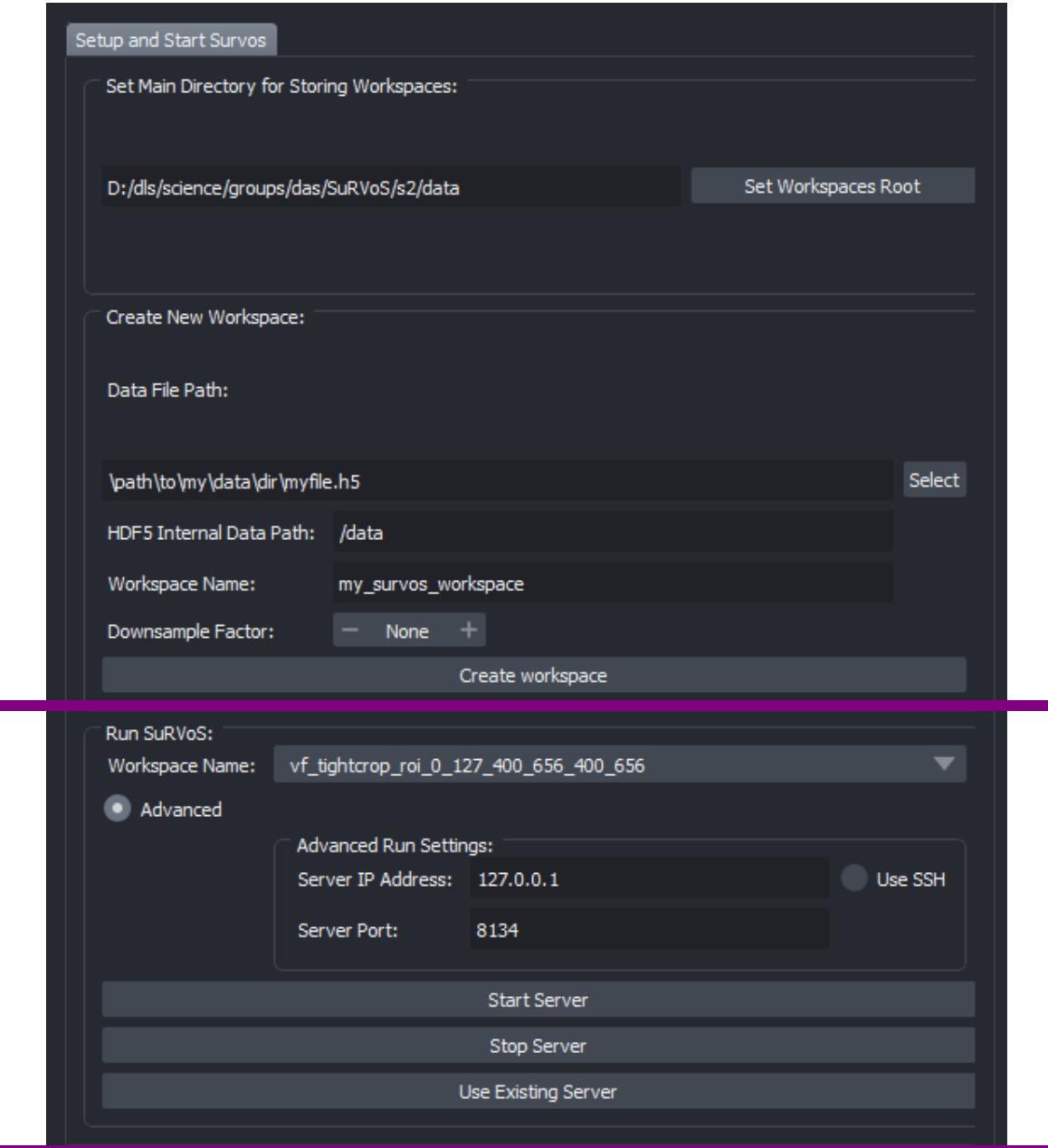
y:	0	512	
----	---	-----	--

z:	0	162	
----	---	-----	--

Downsample Factor: None Estimated datasize (MB): 169.87

This window is used to select the Region of Interest and crop your image to that Region.

Running SuRVoS



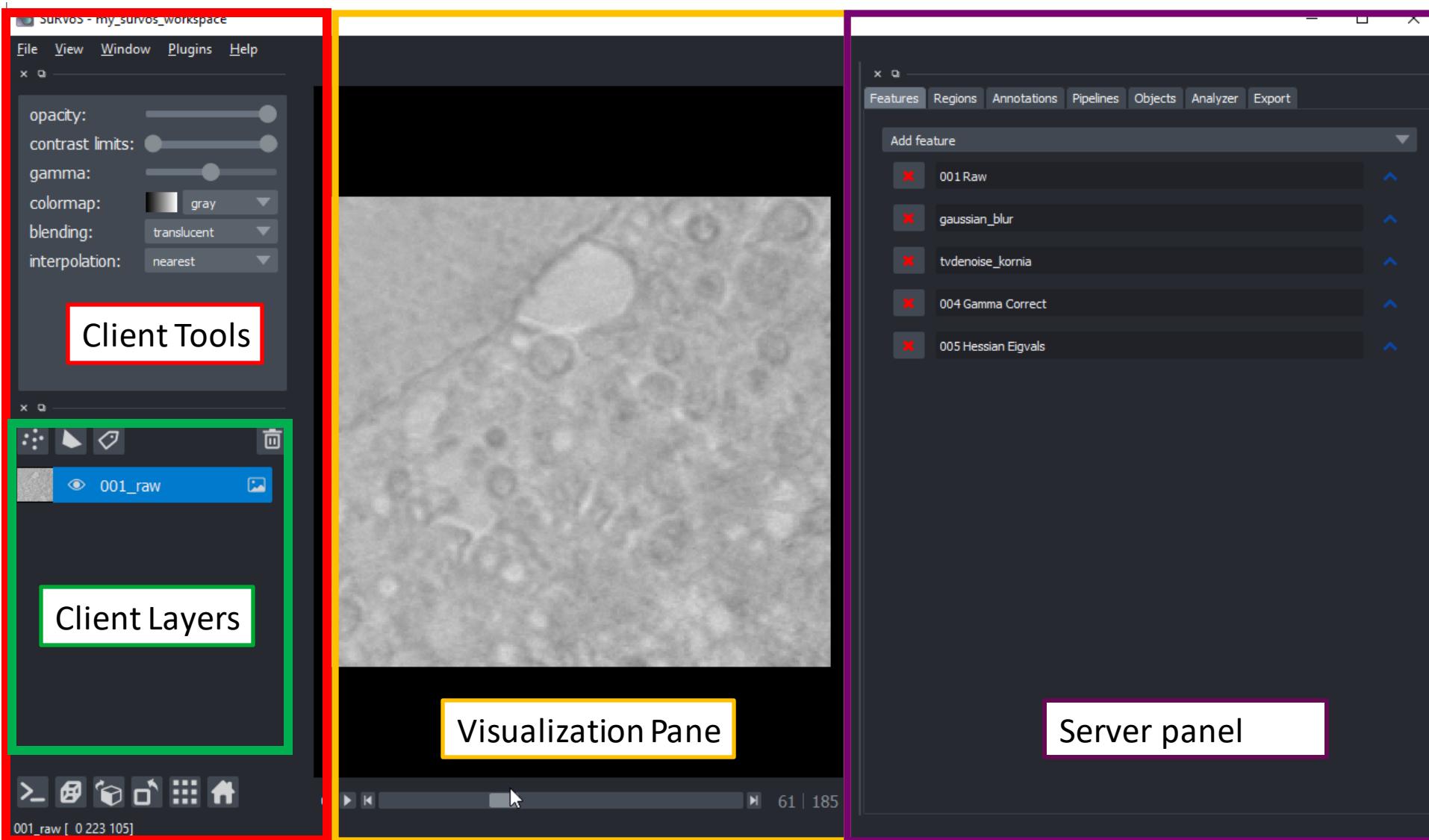
1. Select workspace name from dropdown (will be filled in automatically if workspace has just been created).

2. Click advanced. Check that SSH settings are off (in most cases).

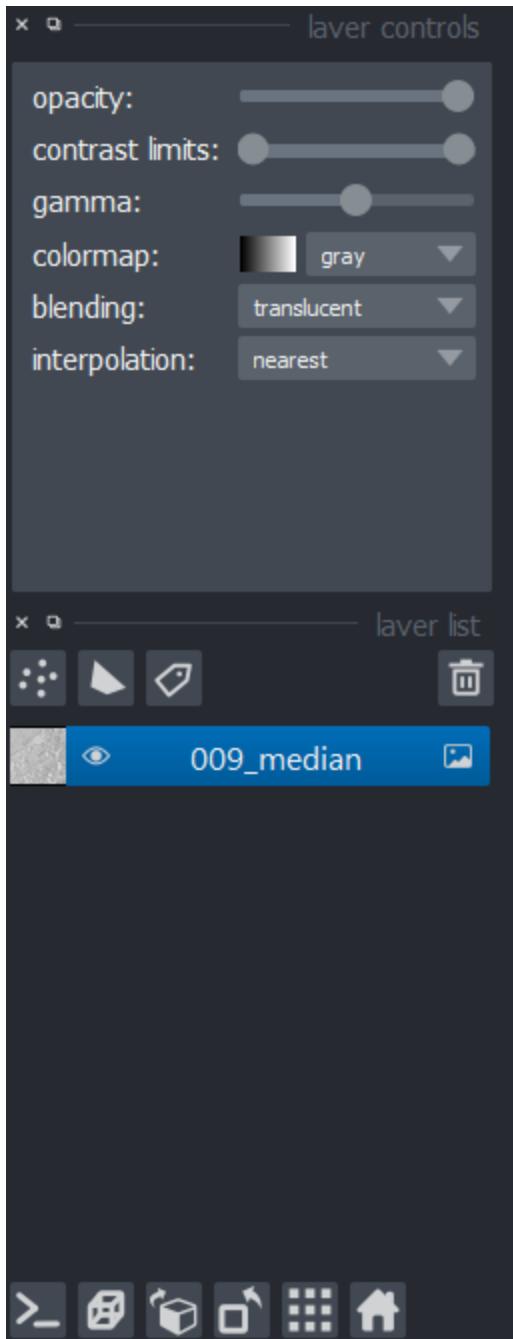
3. Click ‘Start Server’

If an existing program is on a given port then there will be an error. Change the port number and try again.

Orientation to SuRVoS



Layer controls and Layers in GUI



The left side of SuRVoS is the Napari volumetric data viewer. More information can be found at napari.org/docs.

SuRVoS supports 3 Napari layer types

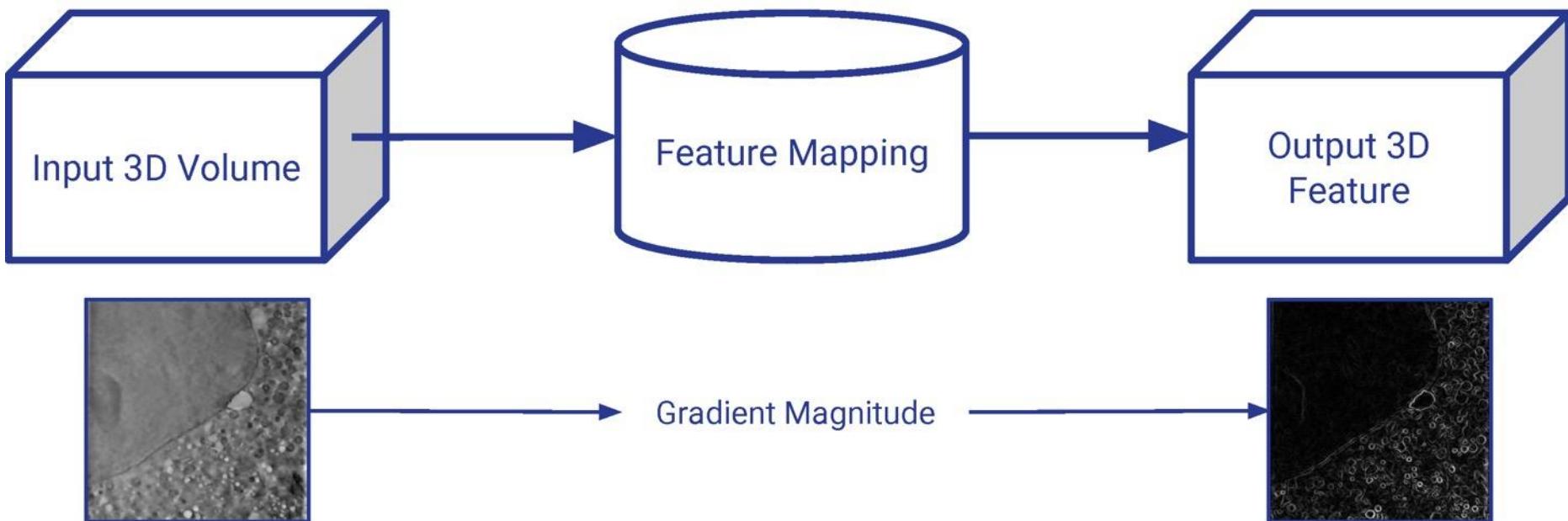
1. Images (with a mountain and sun picture icon) correspond to SuRVoS Features.
2. Labels (With a label icon) correspond to SuRVoS Annotations
3. Points (Points icon) are supported in SuRVoS Objects

Each layer type has a specific set of controls. Left are the controls for images, including opacity, contrast limits, gamma and colormap settings.

The bottom row of icons supports 3d viewing of The image volume (the cube icon)

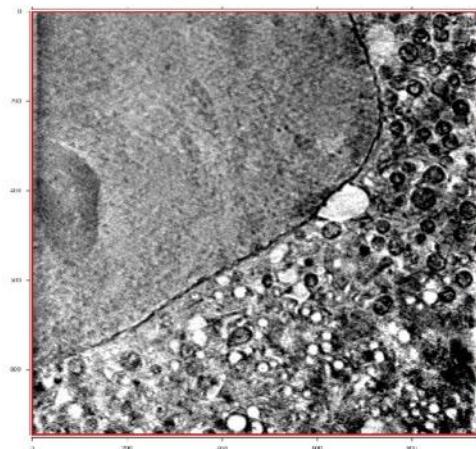
Data Preprocessing

- Every preprocessing method outputs a *feature channel*
- *Feature channels* are obtained by modifying each pixel according to a function applied to their neighbourhood.
- *Feature channels* are volumes of the same size as the input volume
- *Feature channels* can be visualized inside **SuRVoS**

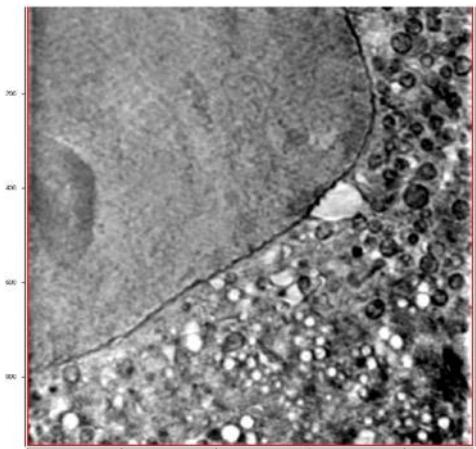


Data Preprocessing

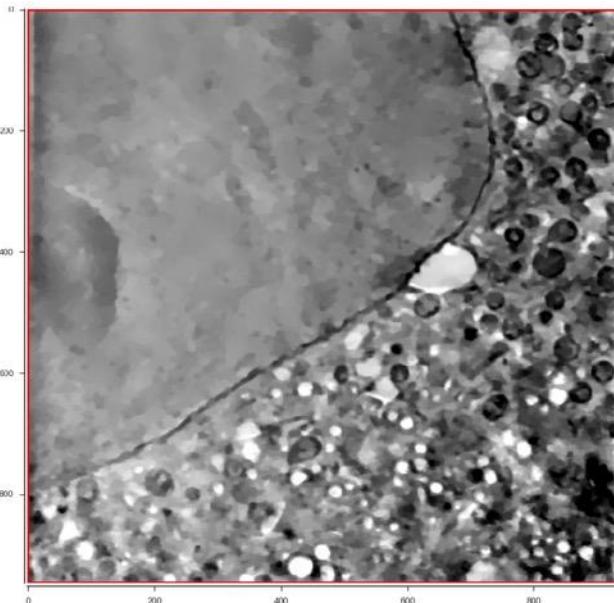
Denoising



Original Image



Gaussian Smooth

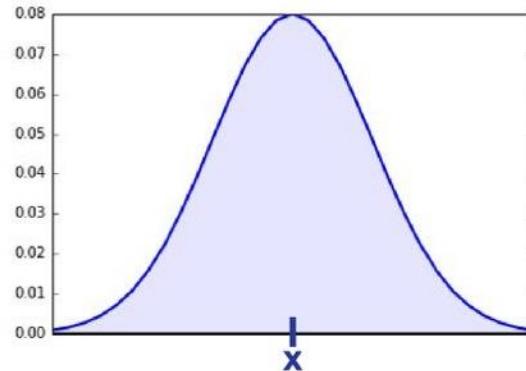


Total Variation

- Over-smooth
- Preserve Strong Edges
- Easier to identify objects

Data Preprocessing

Gaussian filters

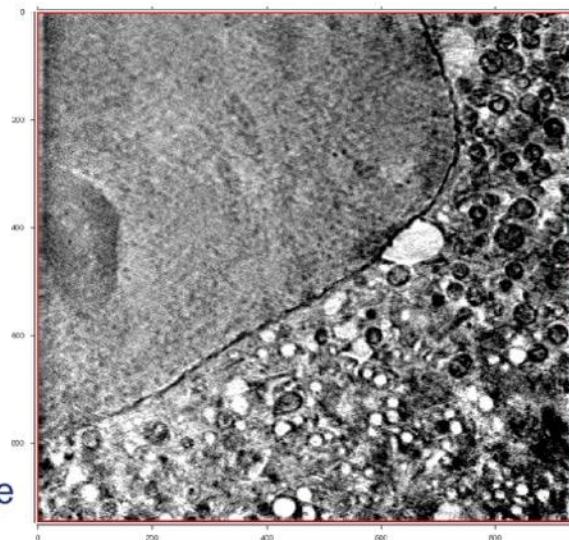


1D Gaussian:

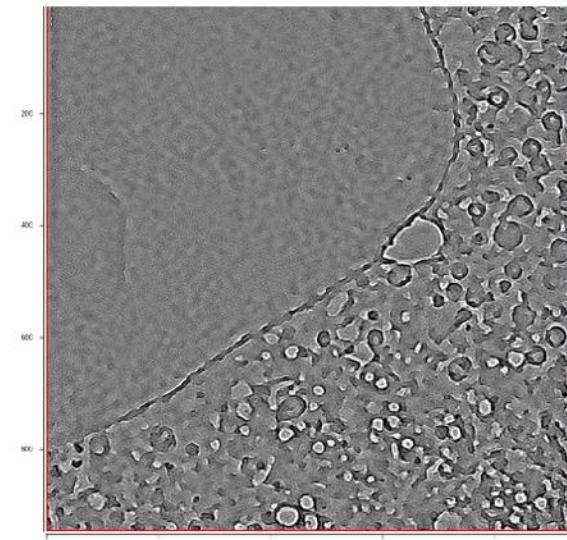
Pixels near the center have more importance.

- A Gaussian neighbourhood of size $N \times N \times N$ centered on every pixel
- Better data fidelity.

Mean Subtraction

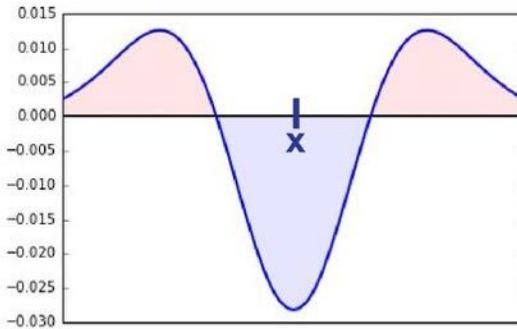


Original Image



Data Preprocessing

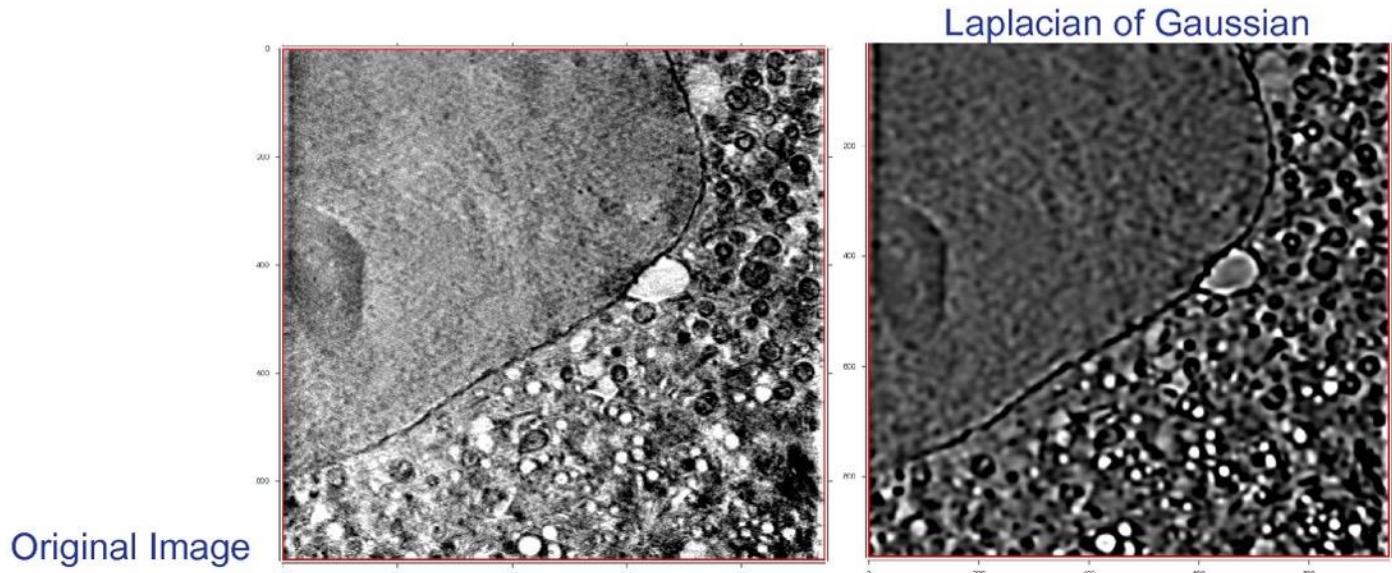
Edge filters



1D Laplacian of Gaussian:

Intensity near the center is subtracted to the surroundings:
 $(x = \text{red} - \text{blue})$.

- A Laplacian neighbourhood of size $N \times N \times N$ centered
- Identify objects brighter or darker than their surroundings.

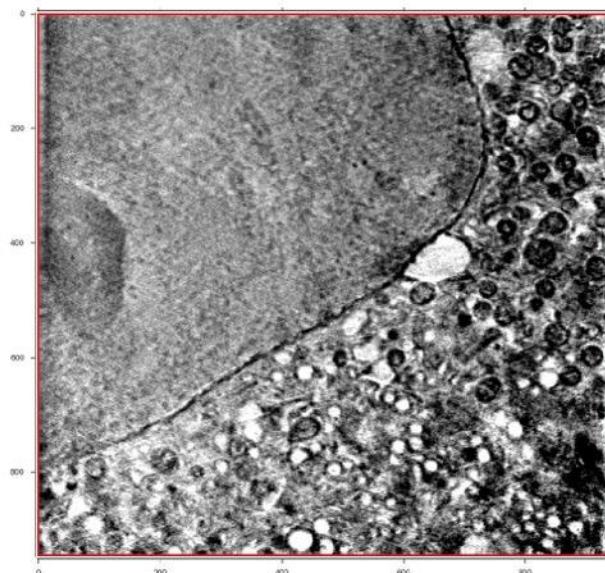


Data Preprocessing

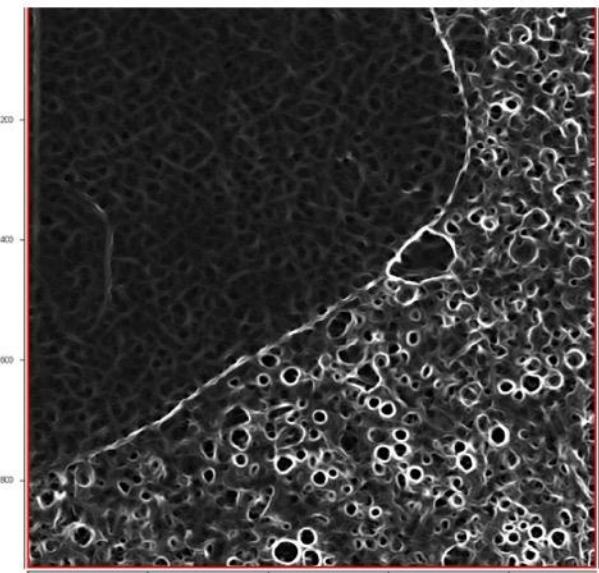
Blob filters

Projects the data to analyze its main axis of variance

- Hessian Eigenvalues: texture
- Structure Tensor Eigenvalues: structure



Original Image



Largest Eigenvalue of the
Hessian Matrix

Available Feature Filters

- Raw
 - Threshold
 - Invert
- Denoising
 - Gaussian Filter
 - Total Variation Filter
 - Median
- Gaussian Features
 - Gaussian Normalization
 - Gaussian Centre
 - Spatial Gradient
- Blob Detection
 - Difference of Gaussian
 - Laplacian of Gaussian
 - Determinant of Structure Tensor
- Texture and Structure
 - Hessian Eigenvalues

Available Filter and Feature Algorithms

- Raw
Binary selection of data

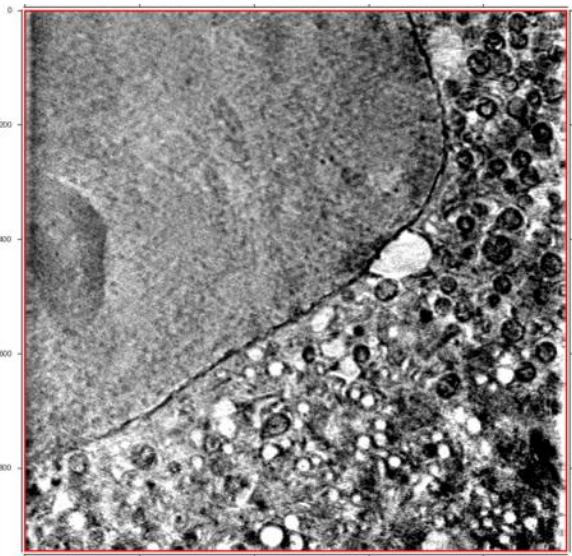
- Denoising
Removes noise

- Gaussian Features
Uses info from neighboring voxels
based on a Gaussian neighborhood

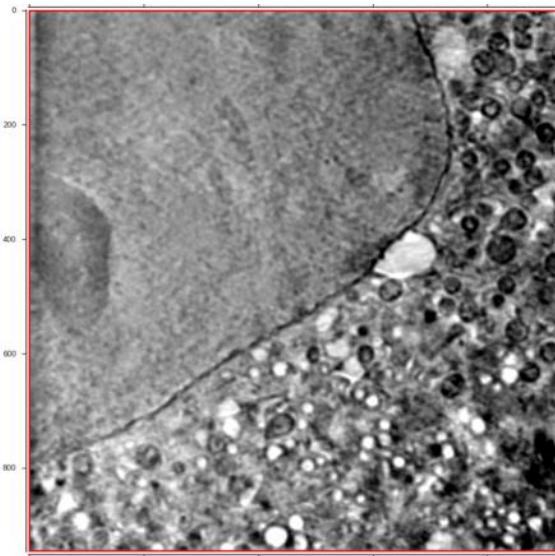
- Blob Detection
Uses info from neighboring voxels
based on a Lapcian neighborhood

- Texture and Structure
Highlights textural differences

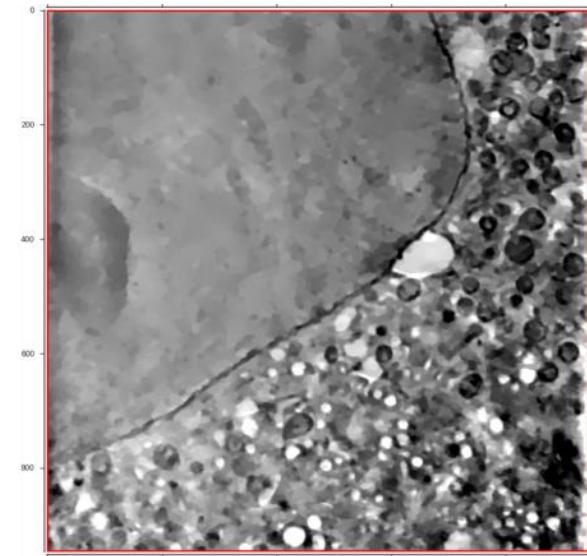
Available Filter Algorithms



(a) Raw SIRT reconstruction



(b) Gaussian Smooth



(c) Total Variation

Recommend

Sigma: 2.0 (range 1-5)

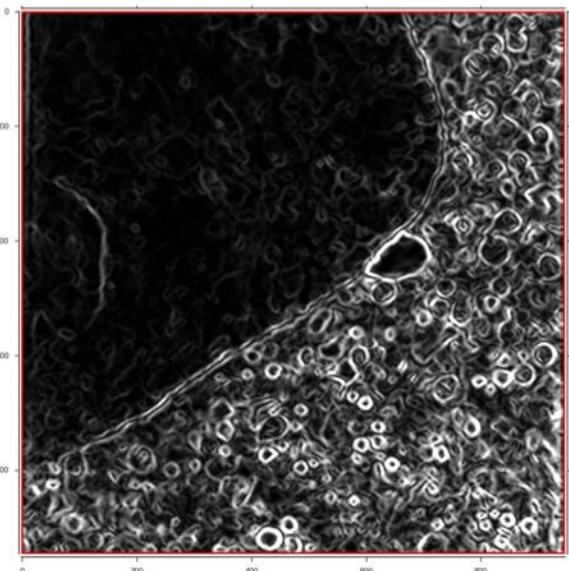
Recommend

Lambda: 10 (range 1-15)
(lower more denoising)

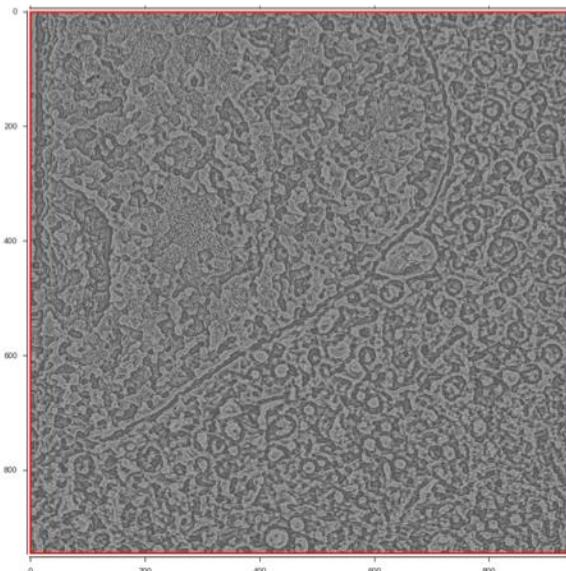
Note: Recommend to use Gaussian filter for supervoxel and megavoxel calculations, total variation for filter and feature calculations.

Maxiter: 100 (range 50-500)
(more iterations more denoising)

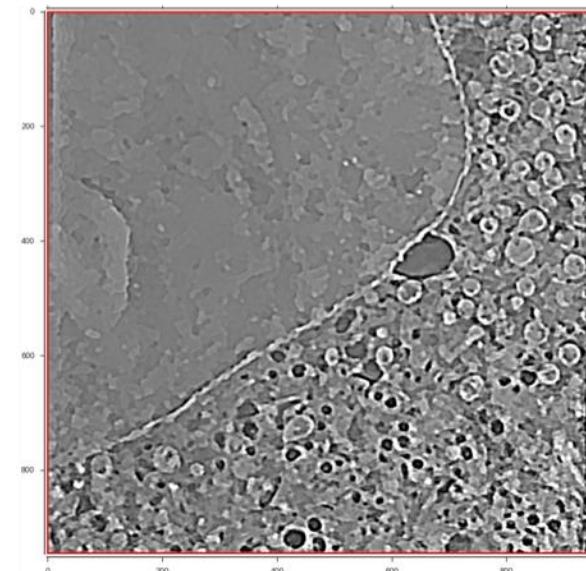
Selected Feature Filters



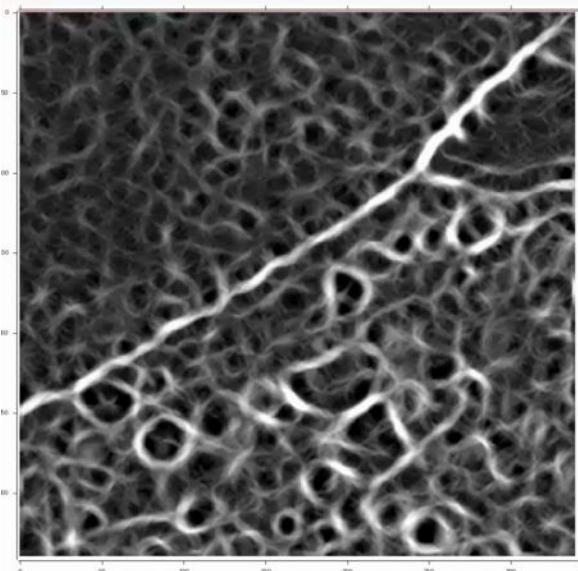
(d) Gradient Magnitude



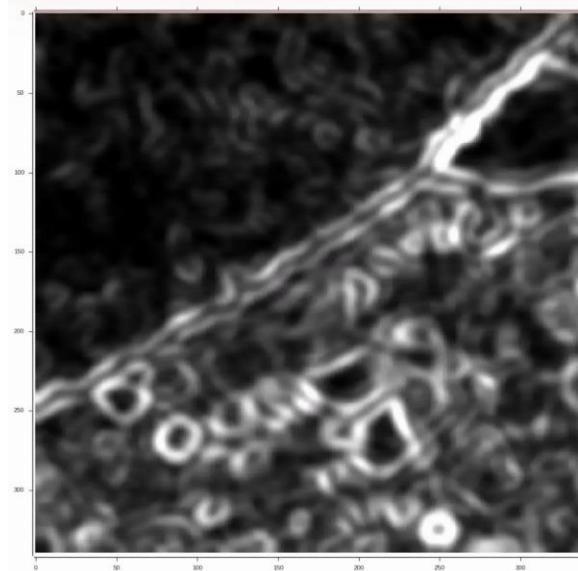
(e) Gaussian Local Normalization



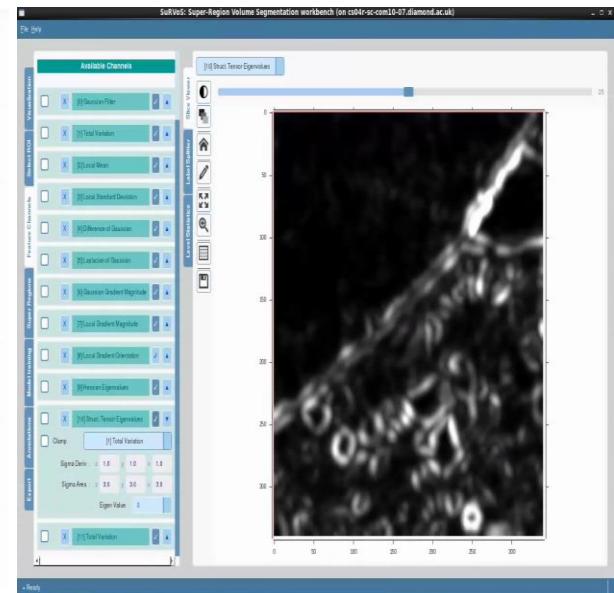
(f) Laplacian of Gaussian



(g) Hessian Eigenvalues



(h) Local Gradient Magnitude



(i) Structure Tensor Eigenvalues

The Feature Panel

The screenshot shows the 'Add feature' panel with a list of seven features: 001 Raw, 002 Gaussian Blur, 003 Tvdenoise, 004 Median, 005 Threshold, 006 Gaussian Norm, and 007 Hessian Eigenvalues. Below this, a 'Clamp' section is expanded for '002 Gaussian Blur', displaying input fields for sigma (3.0), z (3.0), y (3.0), and x (3.0), along with 'Compute' and 'View' buttons. At the bottom, there are 'Save workflow' and 'Run workflow' buttons, and a file selection field labeled 'Click to select File'.

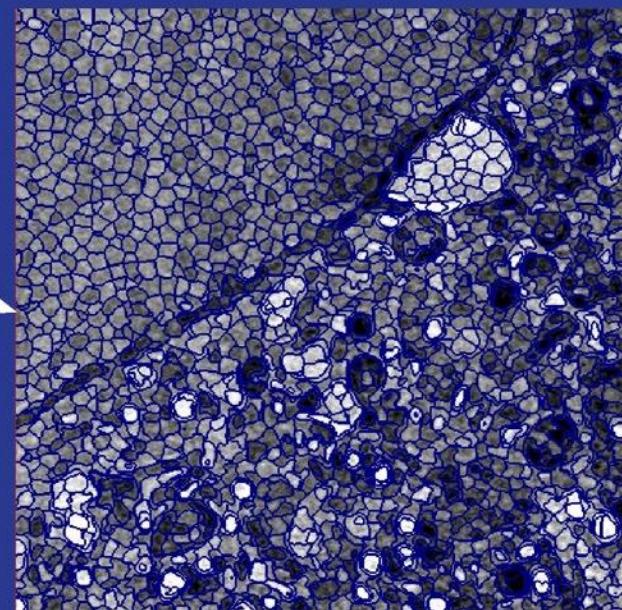
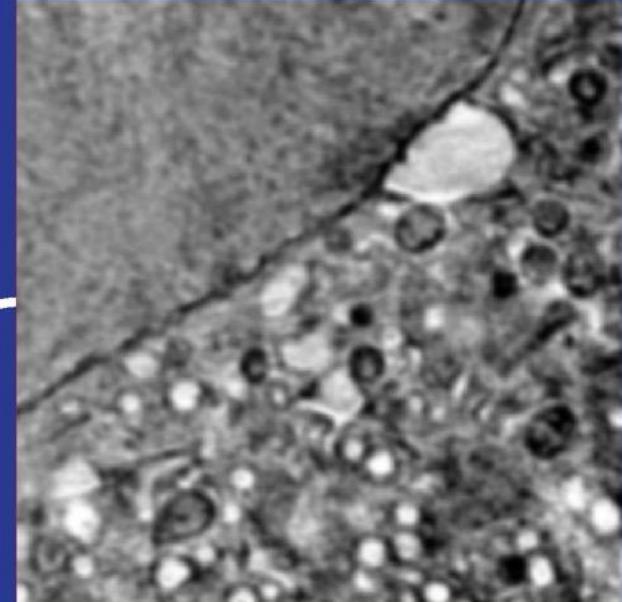
Select feature in dropdown
Click on name, or arrow to see details
Change input and algorithm values if needed
Click compute to run

Note: For filters that use them, coordinate order is Z, Y, X

Data Representation

Represent data in coherent regions

- Voxels
- SuperVoxels
- MegaVoxels



Data Representation

SMURFS



Original Image →

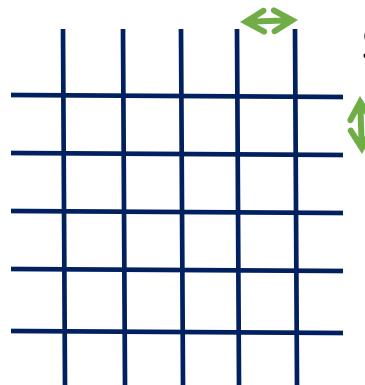
Assign to each pixel the mean
color of all the pixels that belong
to that superpixel

→ Reconstructed Image
Only 200 superpixels

Super Region Parameters: Supervoxels

SuperVoxels:

- SP shape
- Spacing
- Compactness

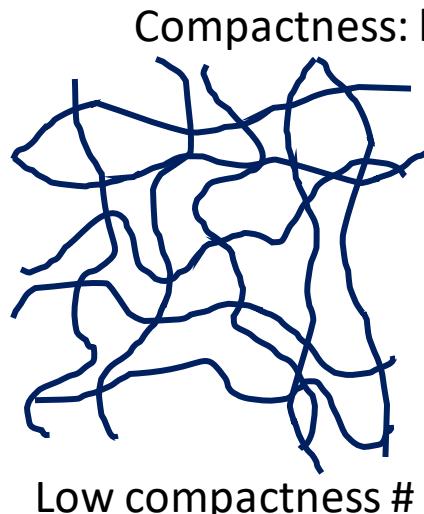


SP Shape: i.e. # of voxels to include in supervoxel grid

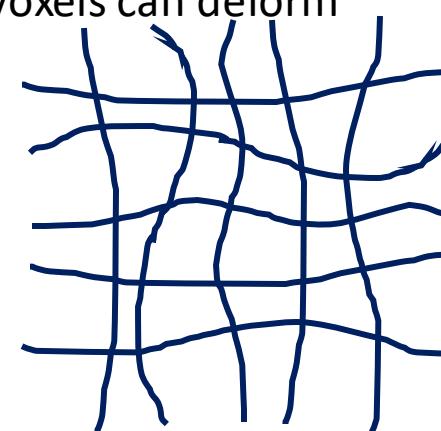
Note: order is Z, Y, X



Spacing: Importance of boundaries in each direction.



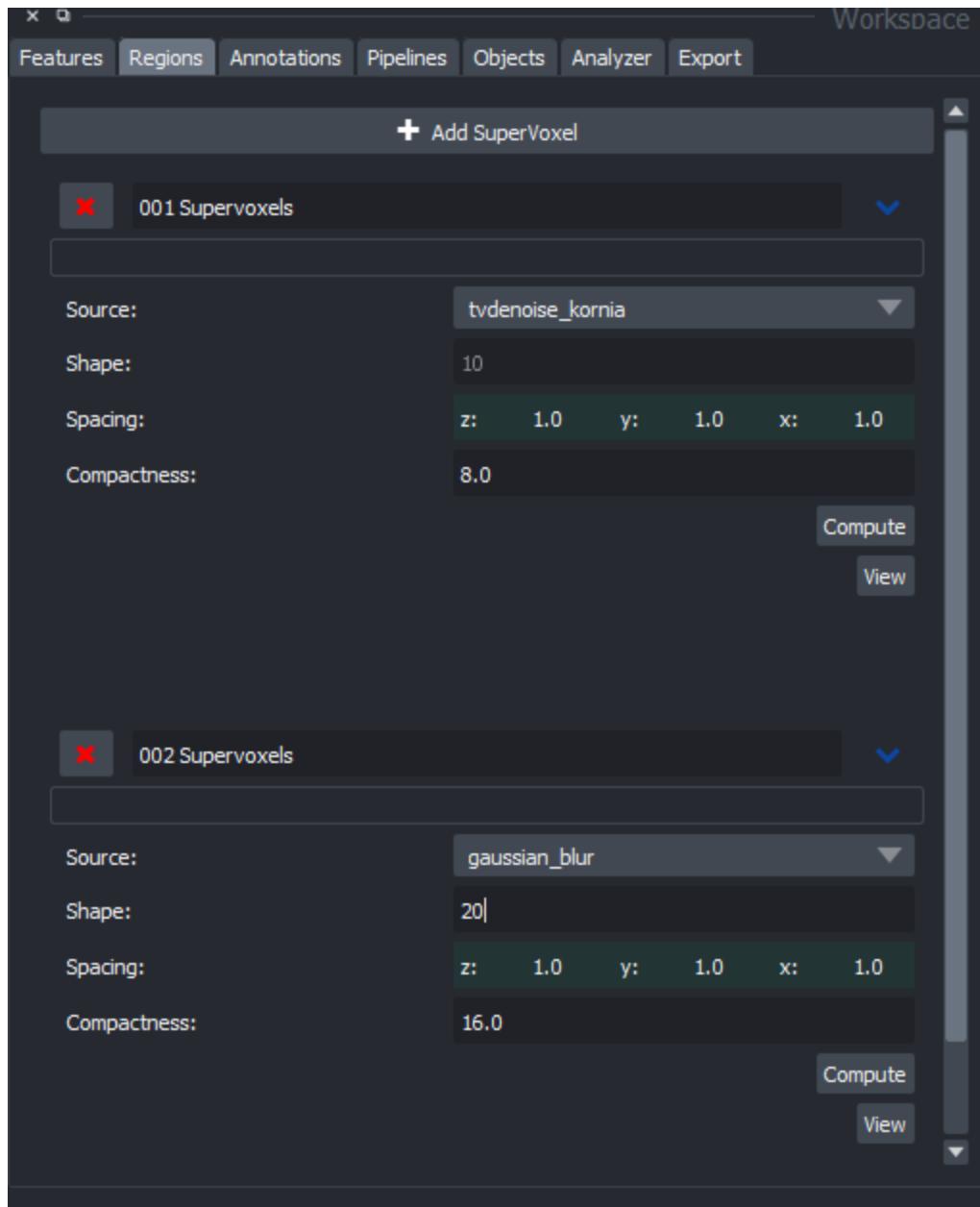
Low compactness #



High compactness #

Compactness: how much supervoxels can deform

Super Region Parameters: Recommendations



SuperVoxels:

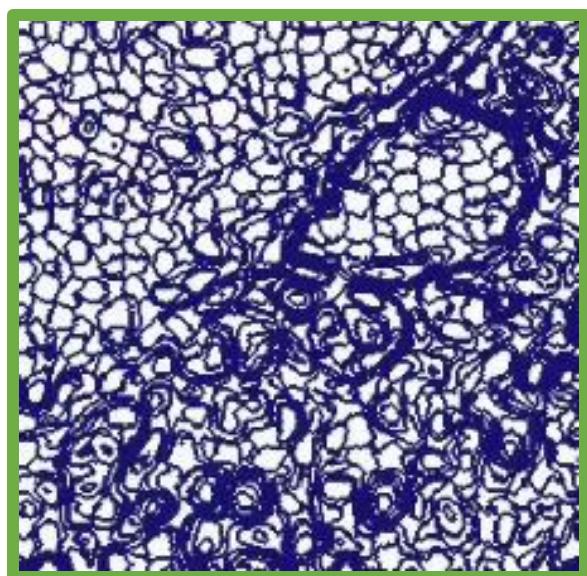
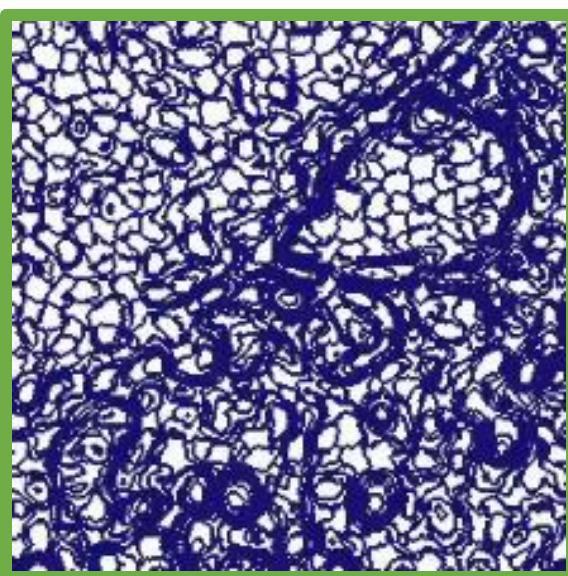
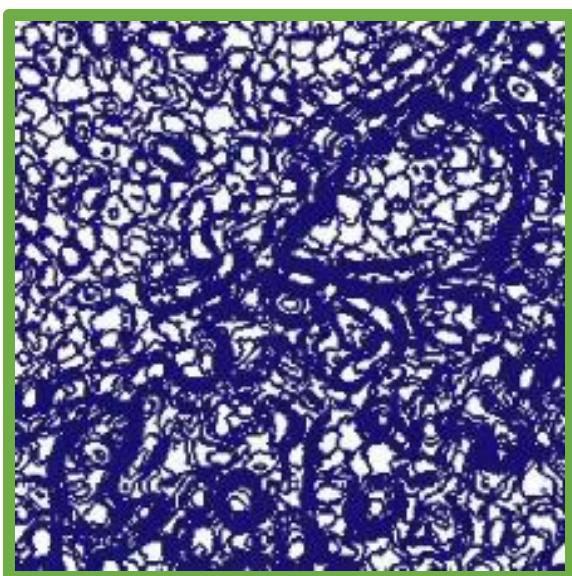
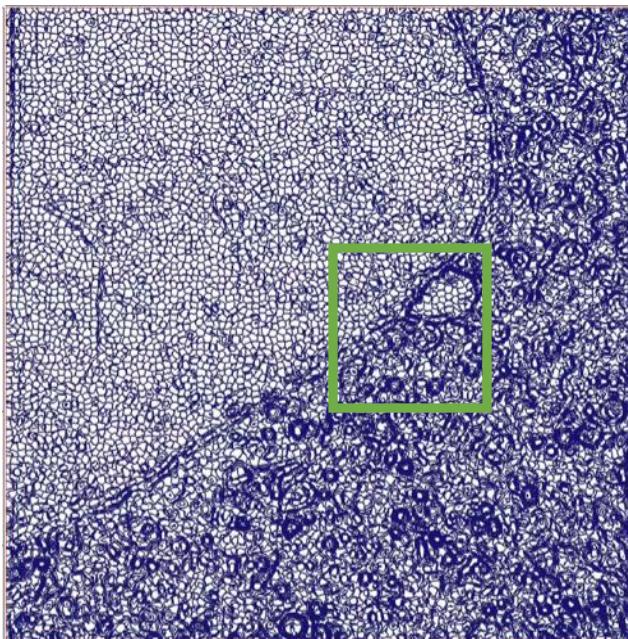
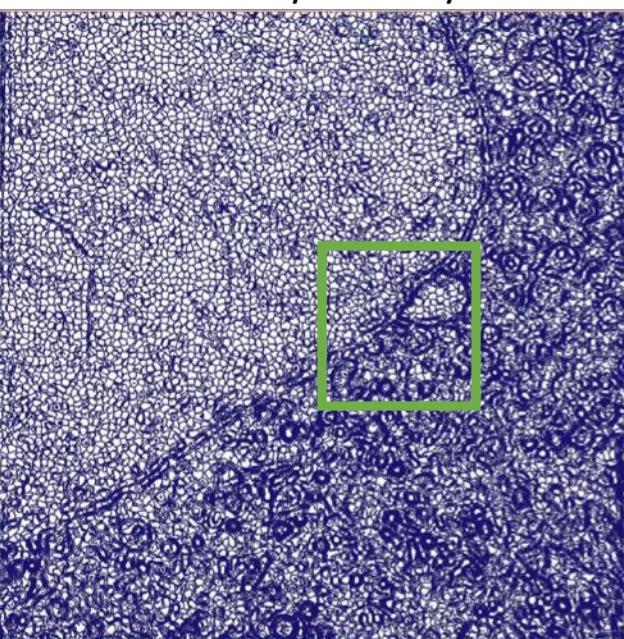
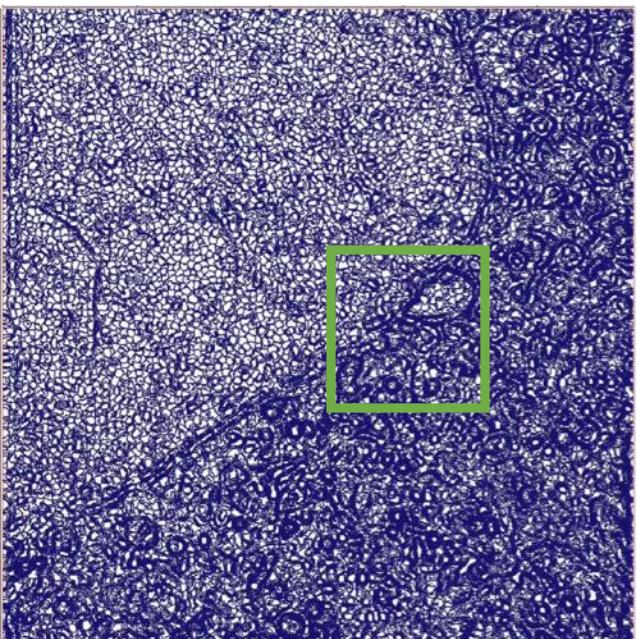
- Source: Data channel
- SP shape: Default 10
- Spacing: 1x1x1
- Compactness: 20 (range 5-100)
- Order is Z, Y, X

Default:

10x10x10 / 1x1x1 / 10

10x10x10 / 1x1x1 / 20

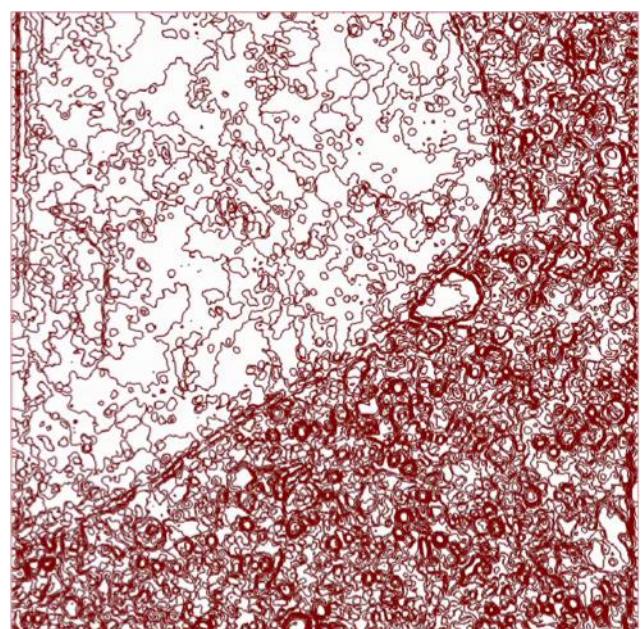
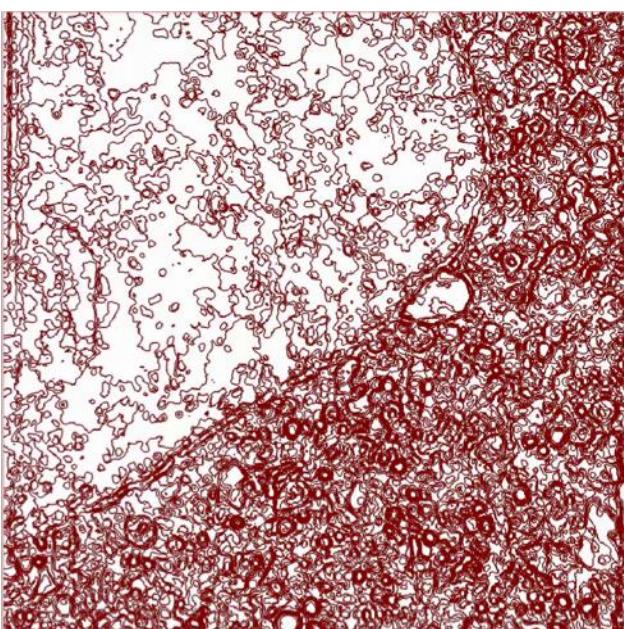
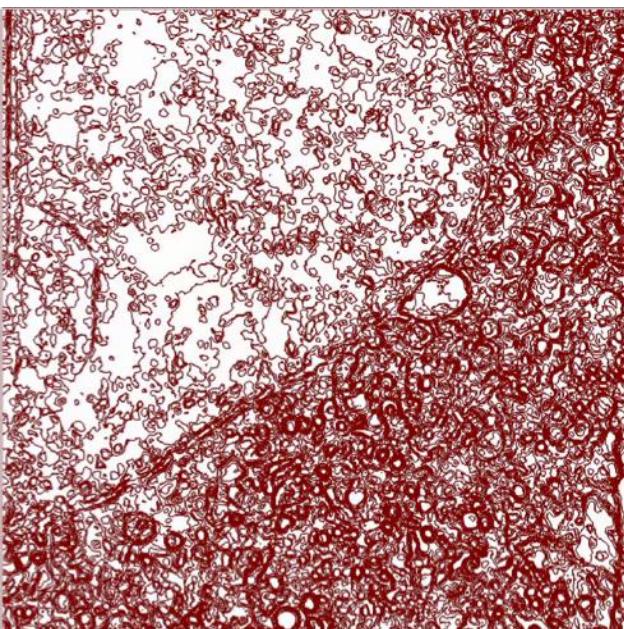
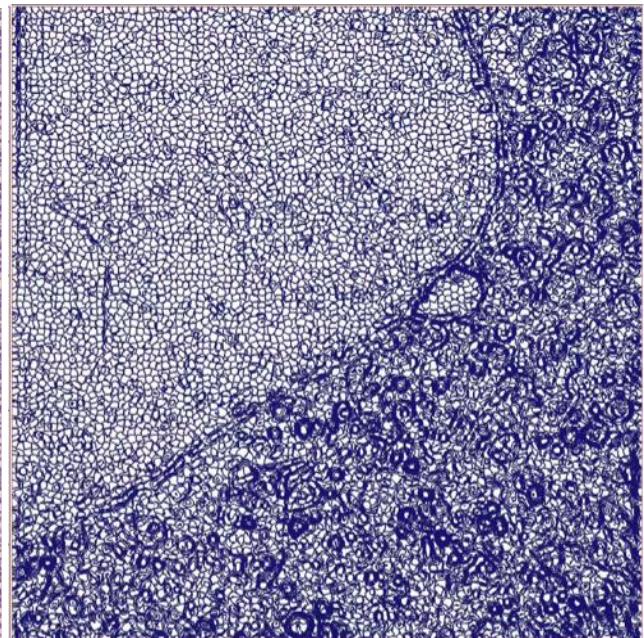
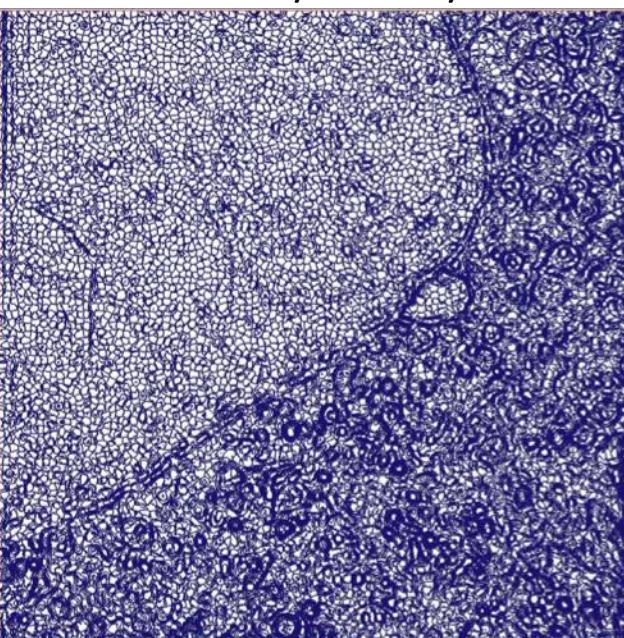
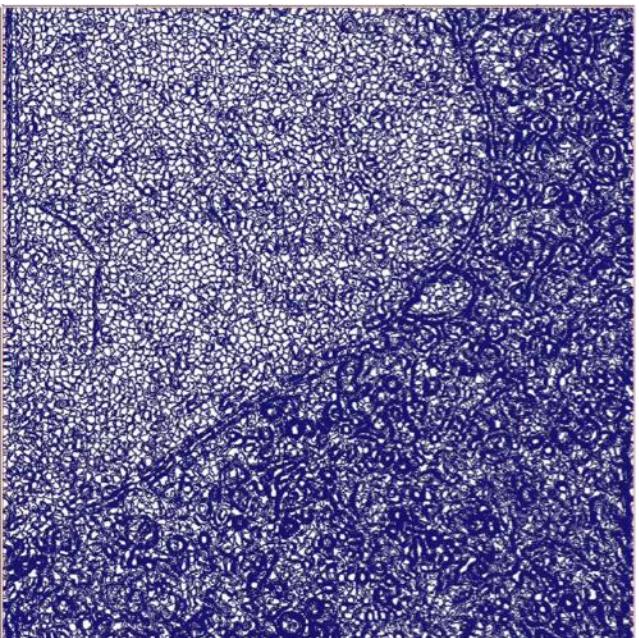
10x10x10 / 1x1x1 / 30



Default:

10x10x10 / 1x1x1 / 20

10x10x10 / 1x1x1 / 10

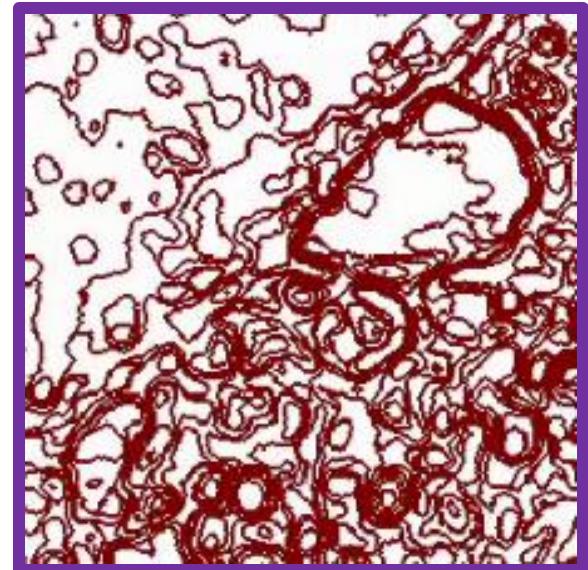
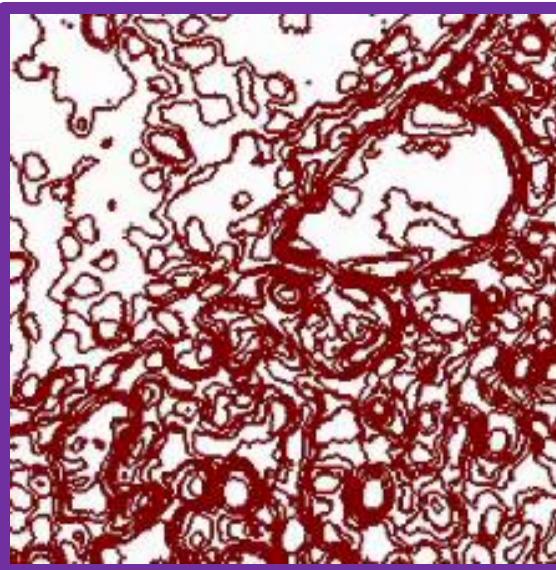
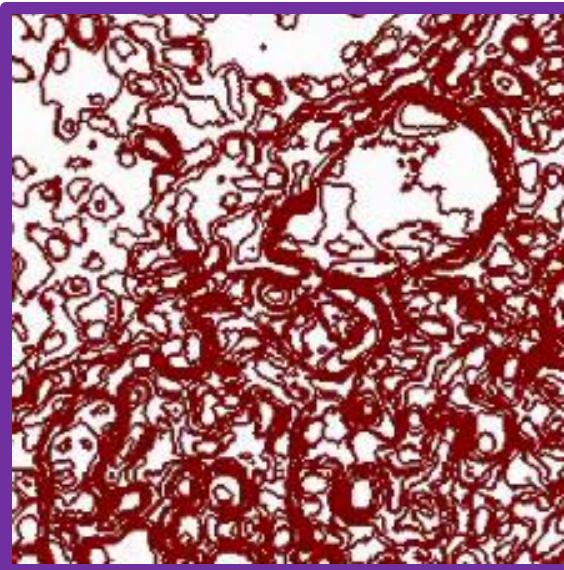
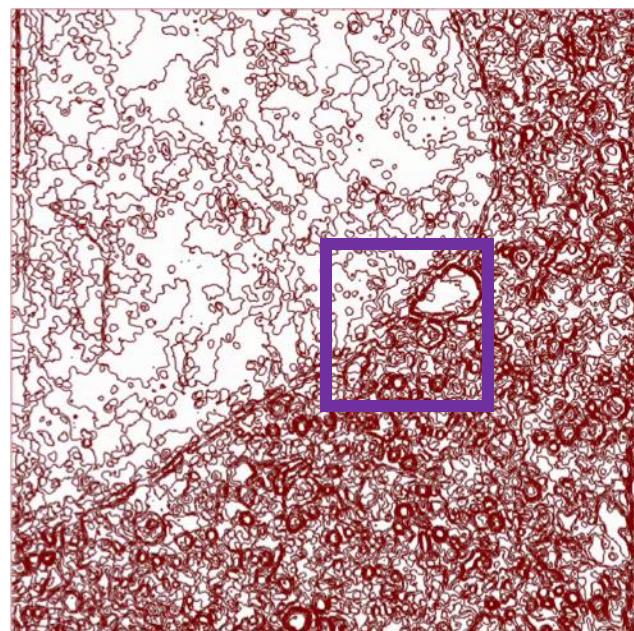
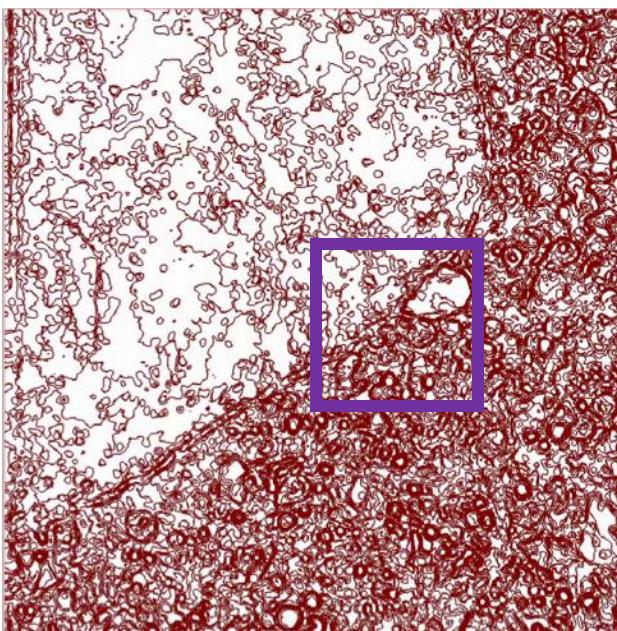
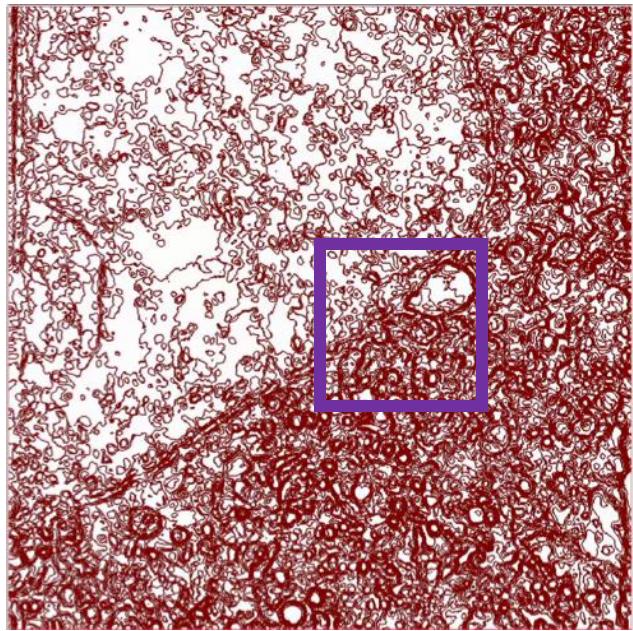


10x10x10 / 1x1x1 / 10

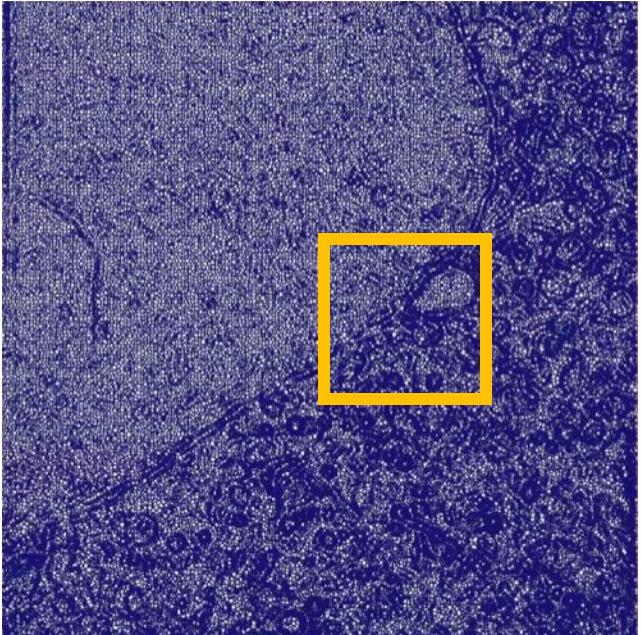
Default:

10x10x10 / 1x1x1 / 20

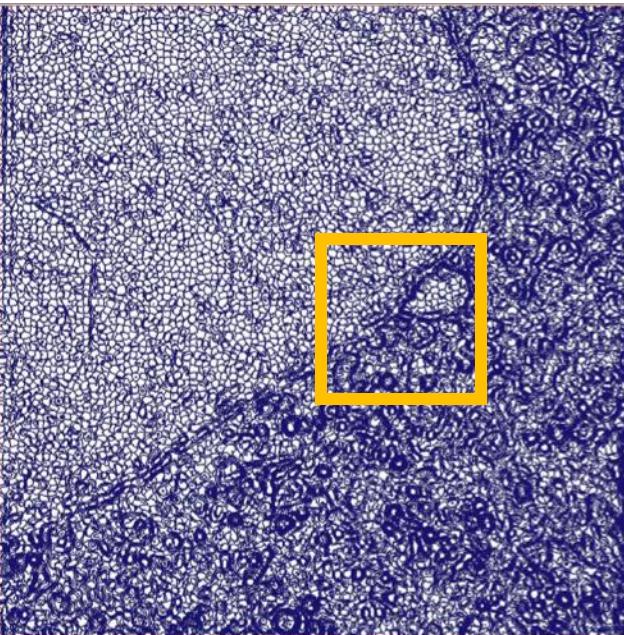
10x10x10 / 1x1x1 / 30



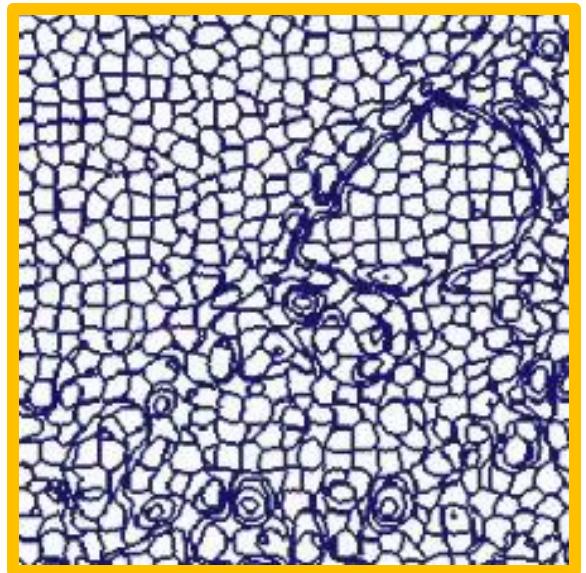
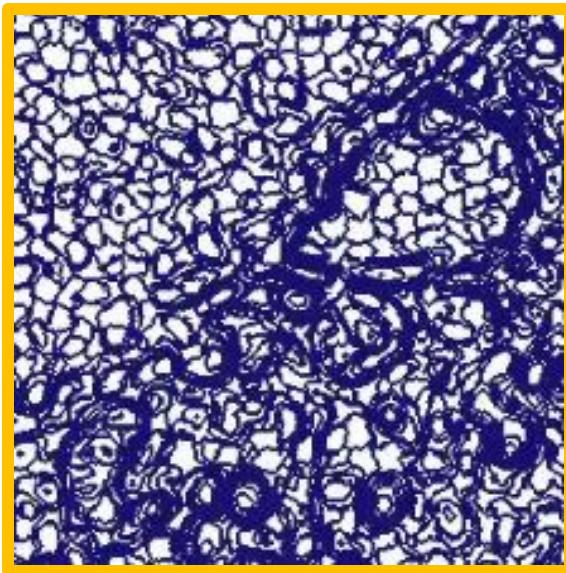
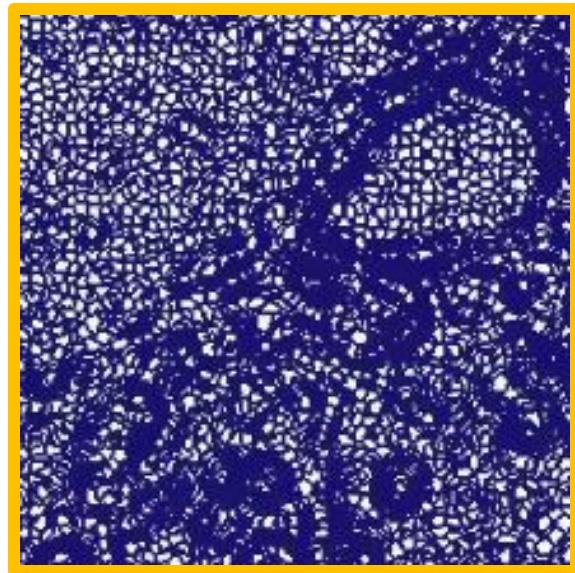
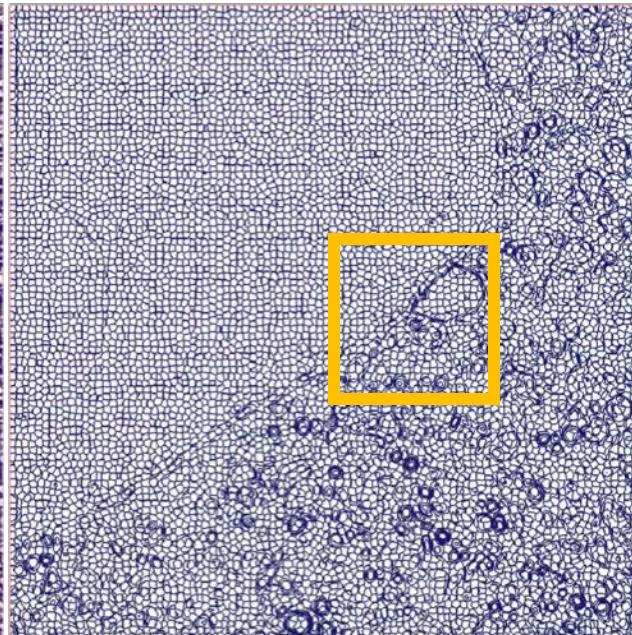
5x5x5 / 1x1x1 / 20



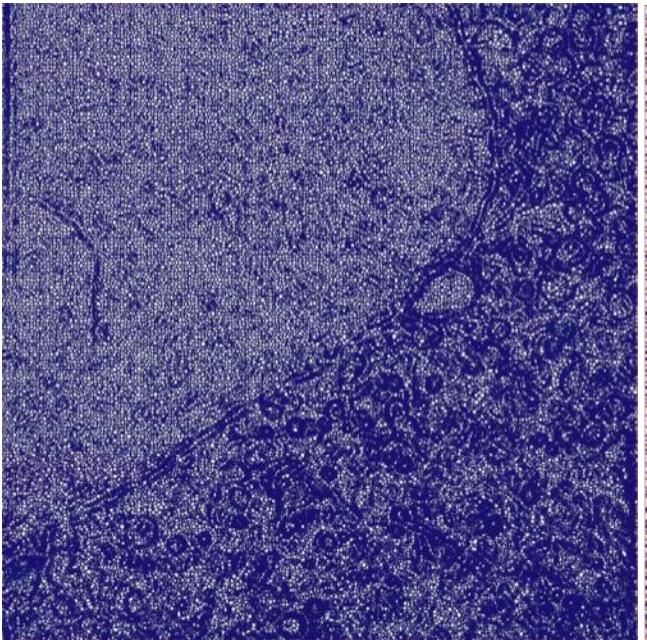
**Default:
10x10x10 / 1x1x1 / 20**



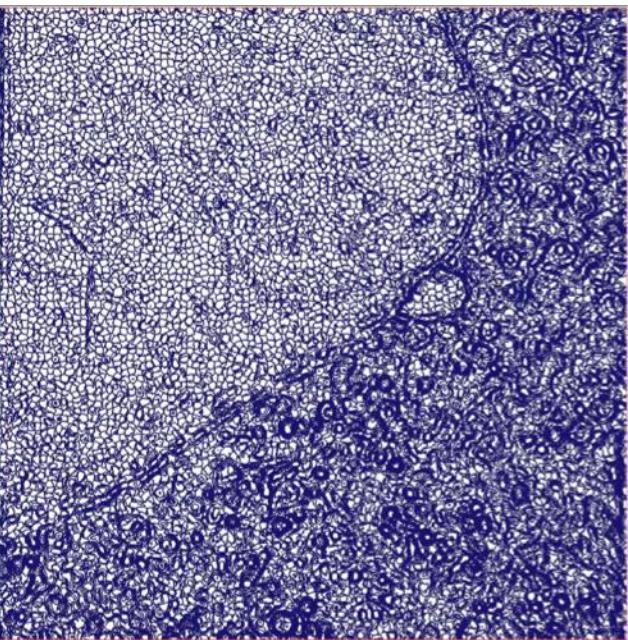
10x10x10 / 3x3x3 / 20



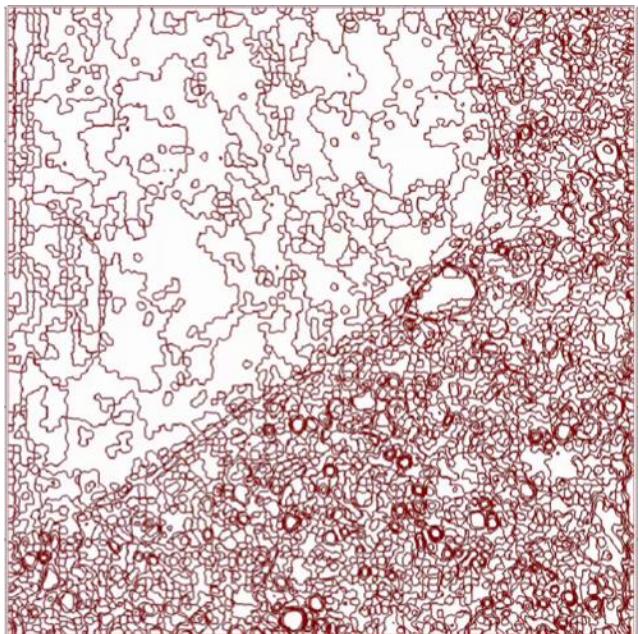
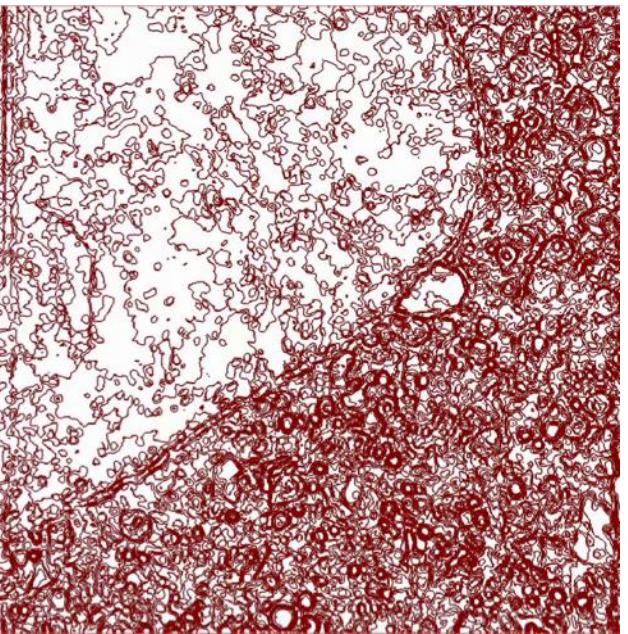
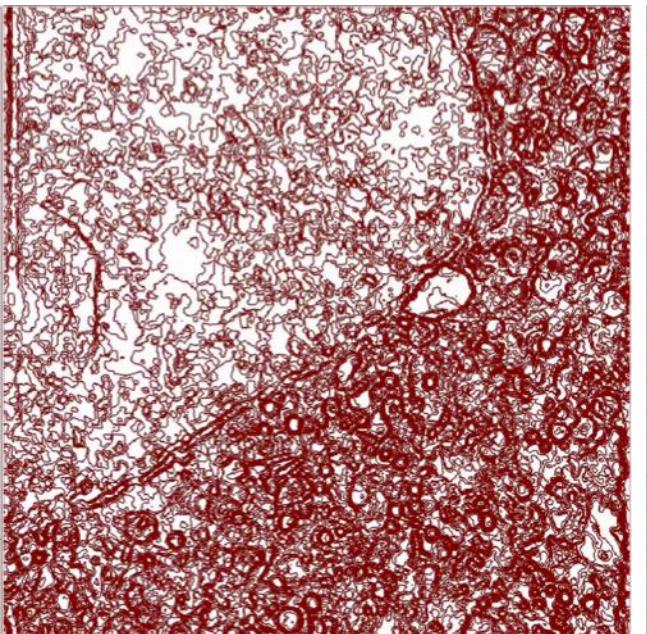
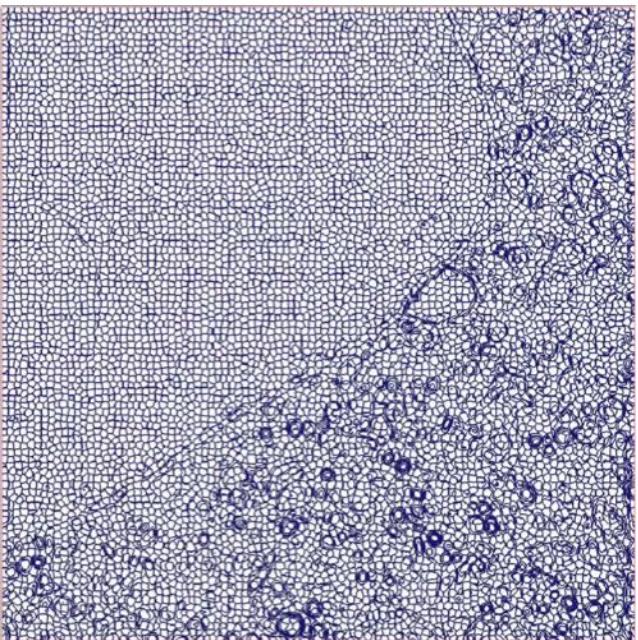
5x5x5 / 1x1x1 / 20



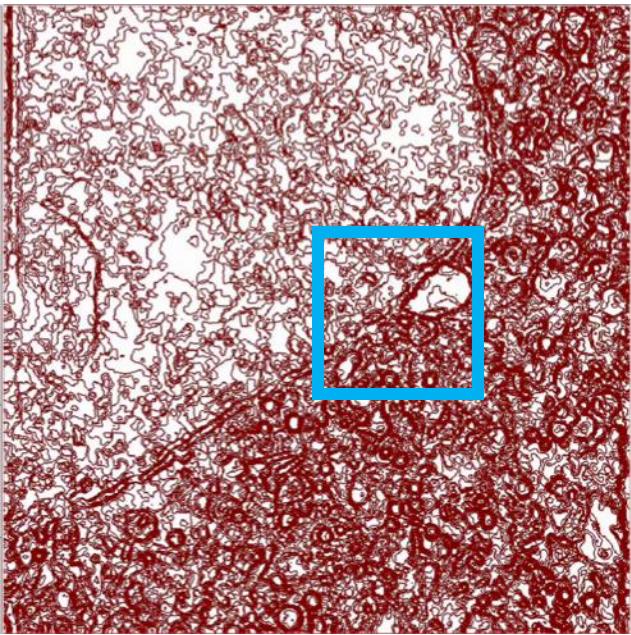
**Default:
10x10x10 / 1x1x1 / 20**



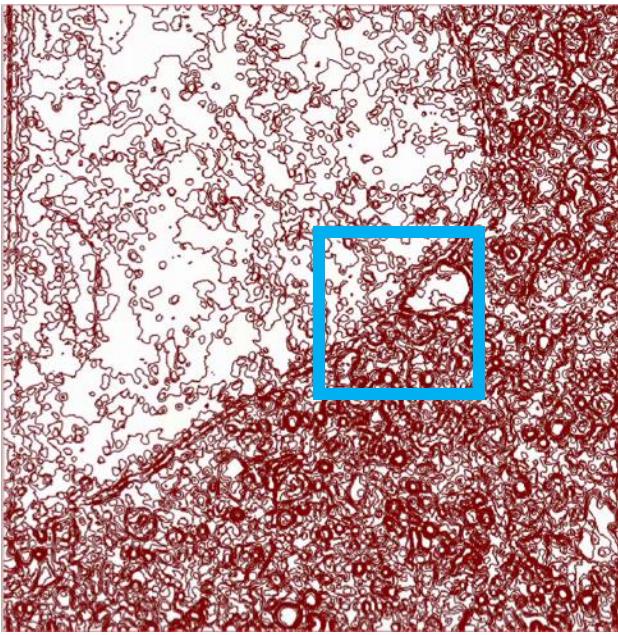
10x10x10 / 3x3x3 / 20



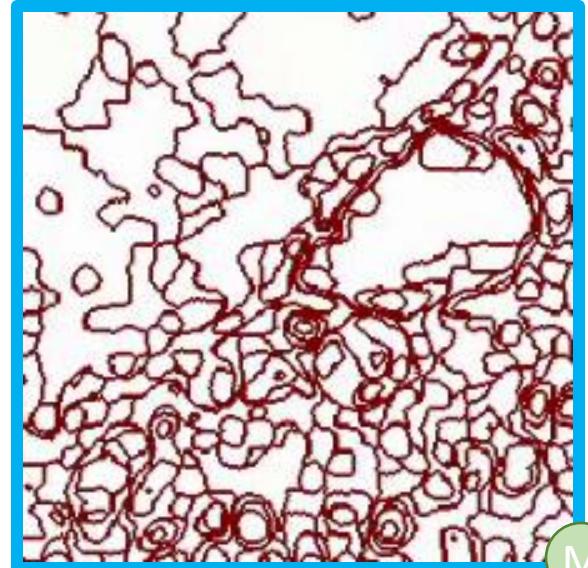
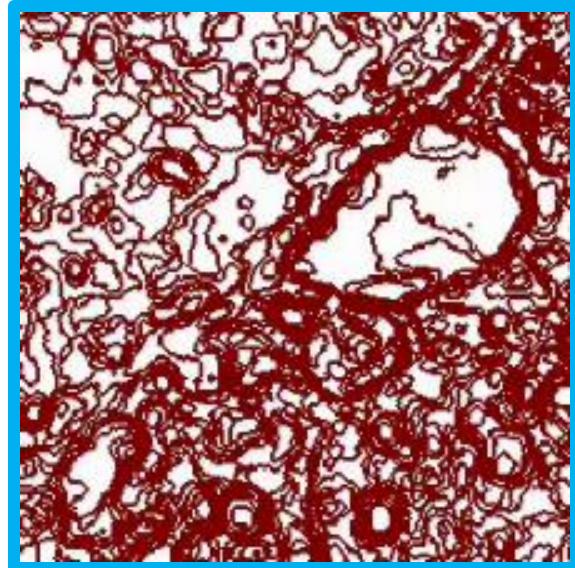
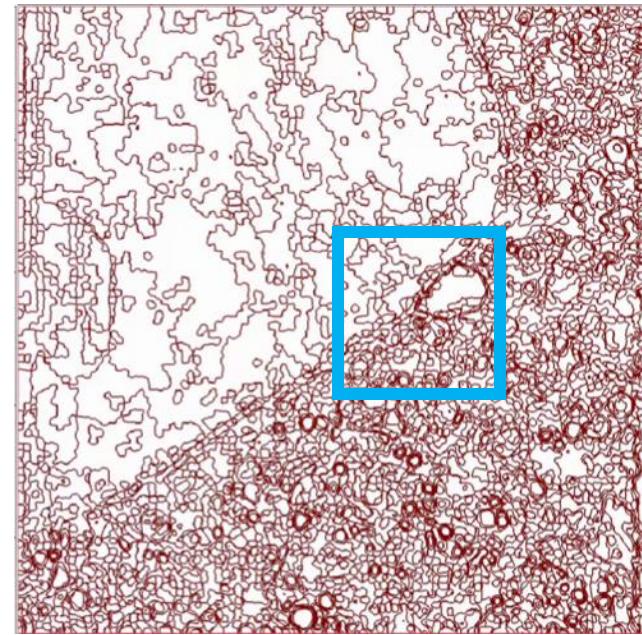
5x5x5 / 1x1x1 / 20



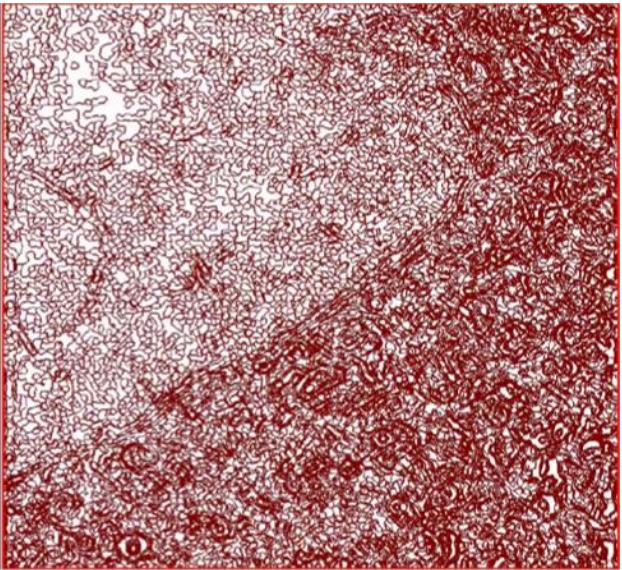
**Default:
10x10x10 / 1x1x1 / 20**



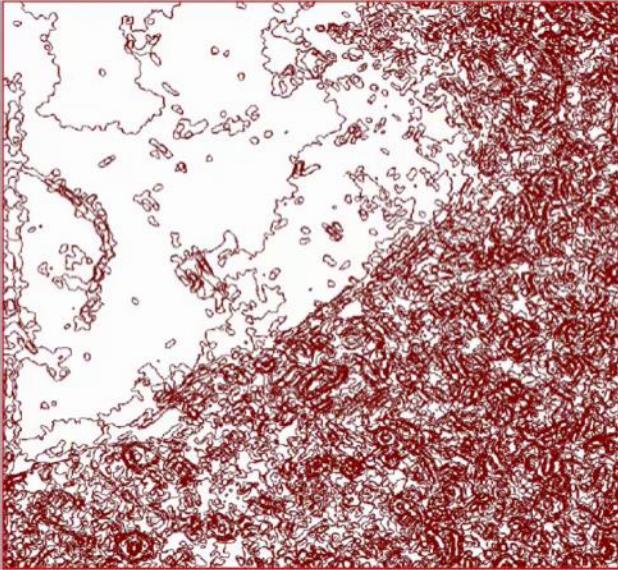
10x10x10 / 3x3x3 / 20



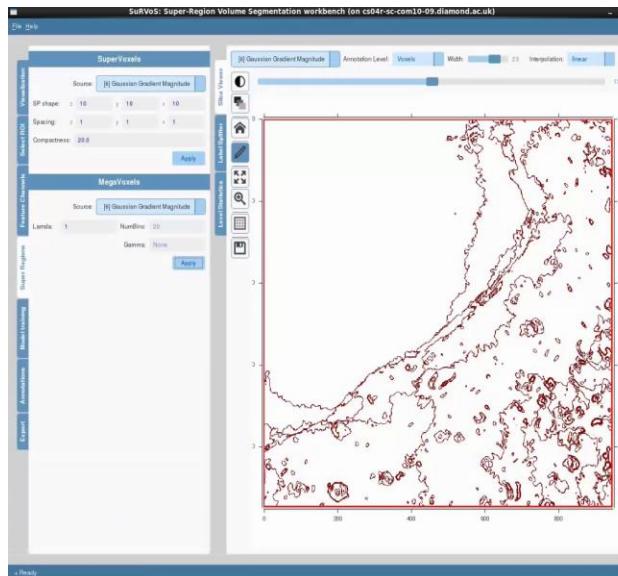
$\lambda=0.01$ /NumBins=20



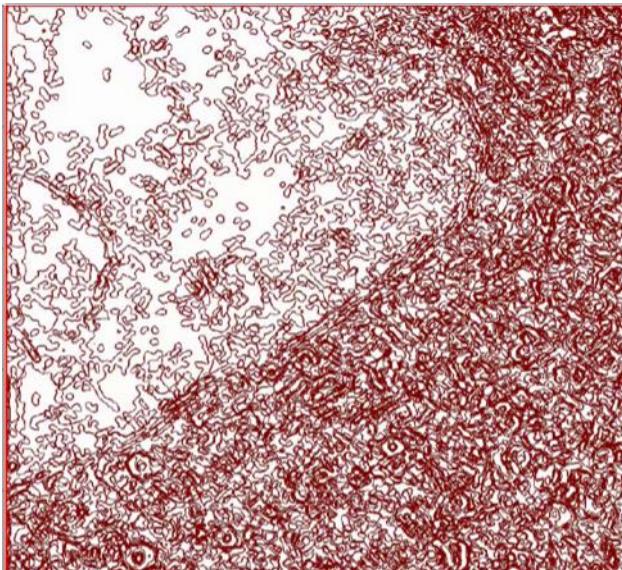
**Default:
 $\lambda=0.1$ /NumBins=20**



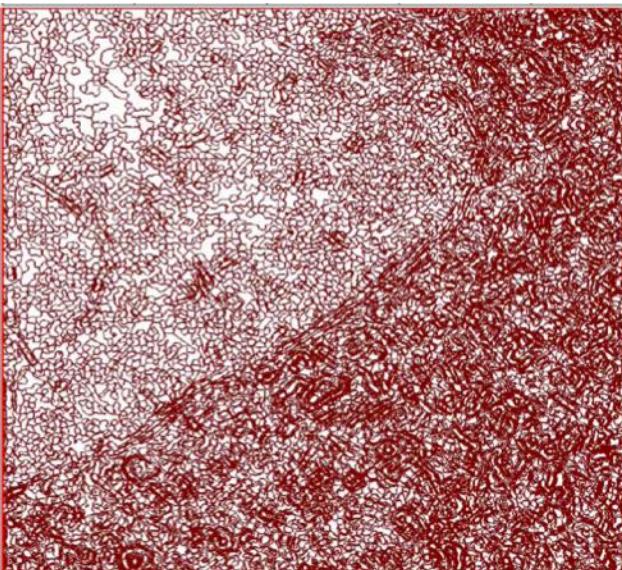
$\lambda=1$ /NumBins=20



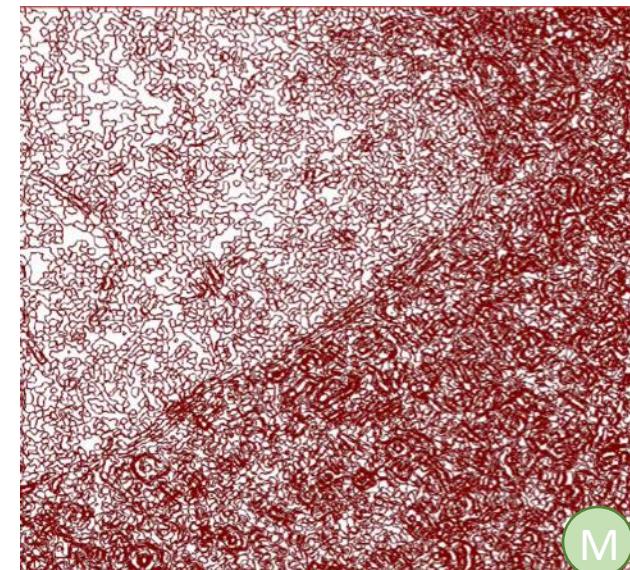
$\lambda=0.1$ /NumBins=10



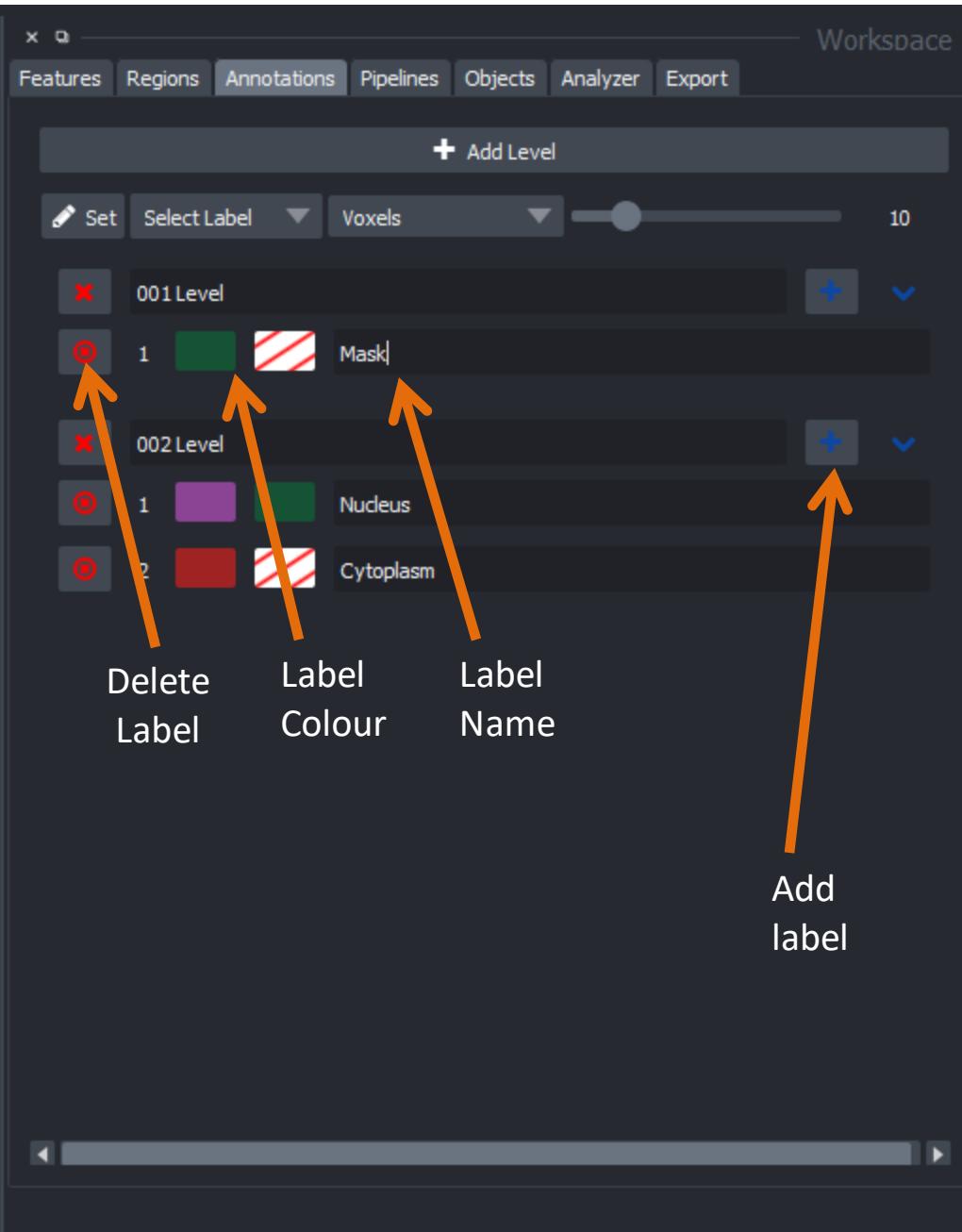
$\lambda=0.1$ /NumBins=30



Defaults, gamma=auto

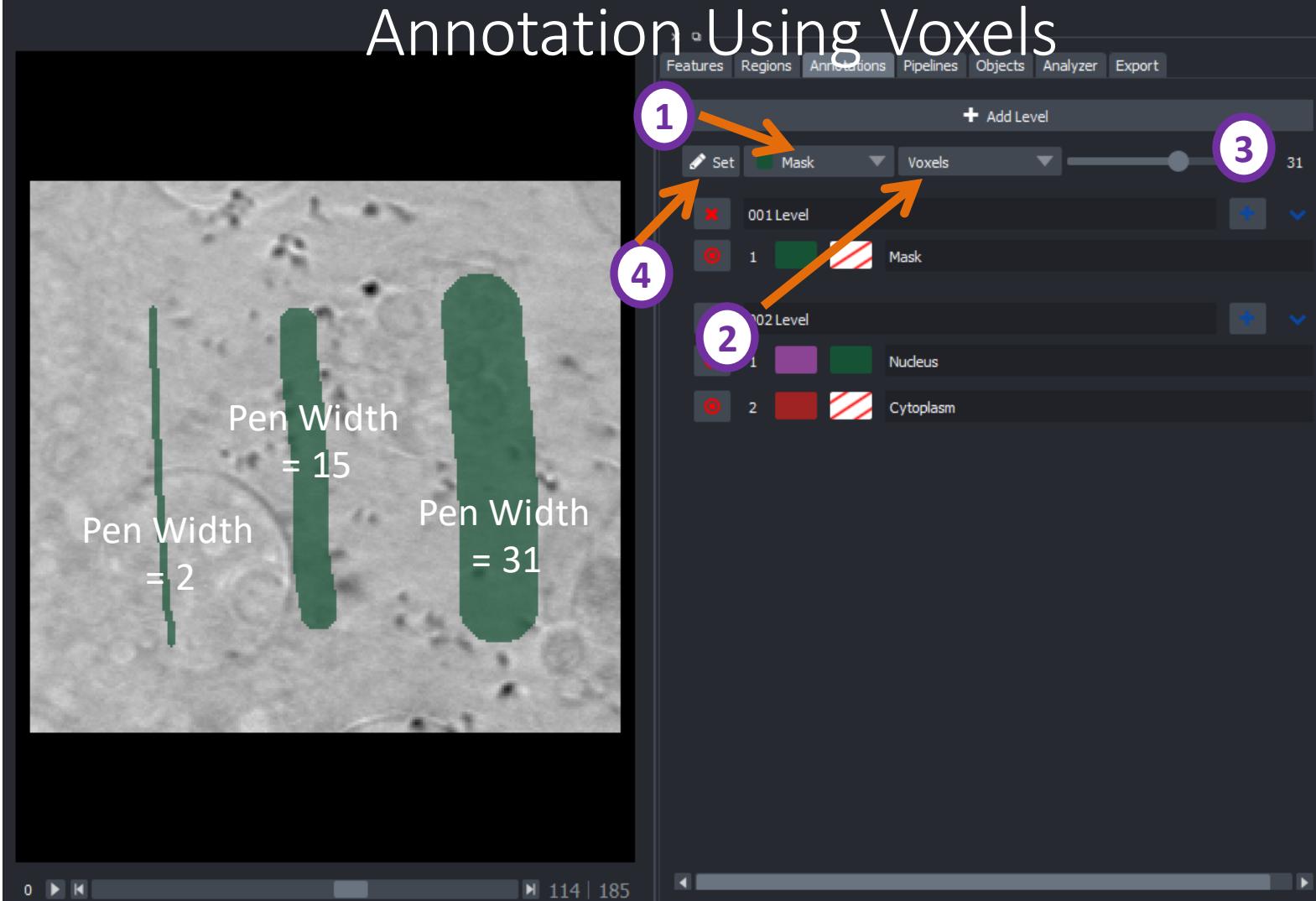


The Annotation Tab



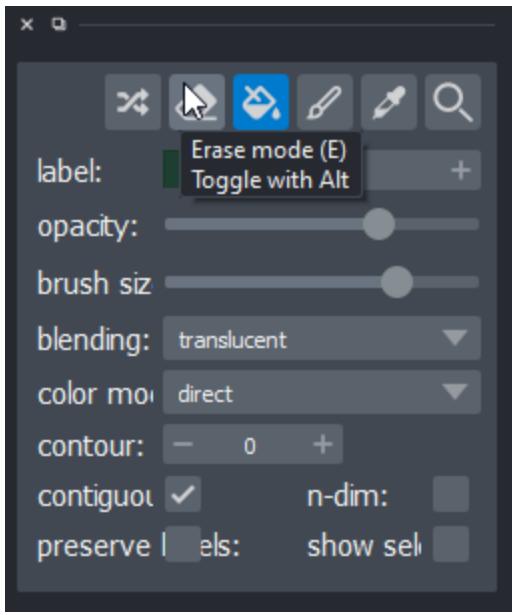
- To add a Level click on Add Level. Levels are given a default name but can be renamed.
- To add Labels to a Level, click Add Label. Multiple labels can be added to any Level.
- Labels can be given names, the colour used to represent them in annotations can be changed. Or if they are no longer needed, labels can be deleted.

Annotation Using Voxels



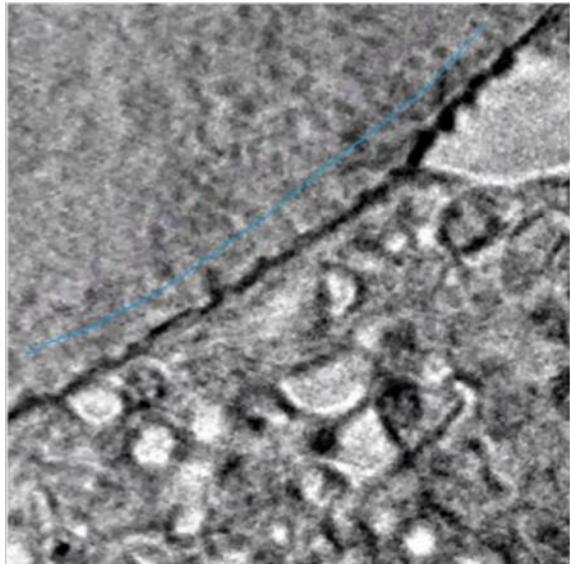
- 1) In the annotation tab 'Select label' dropdown select the label you wish to annotate with.
- 2) Select annotation level (**voxel** or **supervoxel**)
- 3) Choose an appropriate brush width for the feature that you are annotating.
- 4) Press set to confirm.
- 5) Paint on the view.

Erasing Annotations

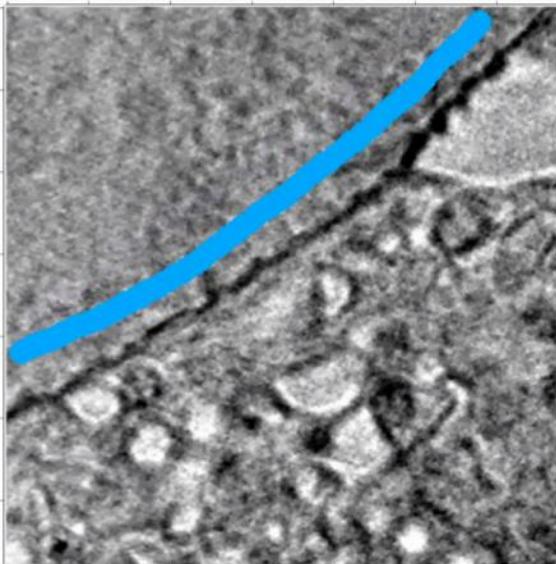


- 1) In the layer controls select the eraser tool.
- 2) Change the brush size.
- 3) To begin annotating again, select the appropriate label and parameters in the annotation panel.

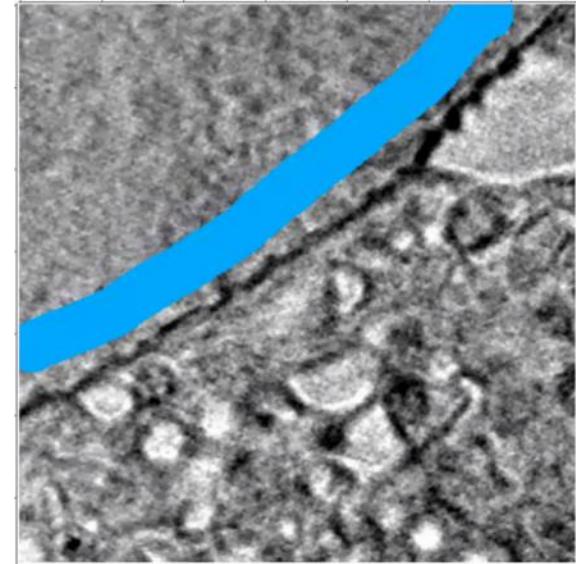
Annotation Using Voxels



Pen Width = 1



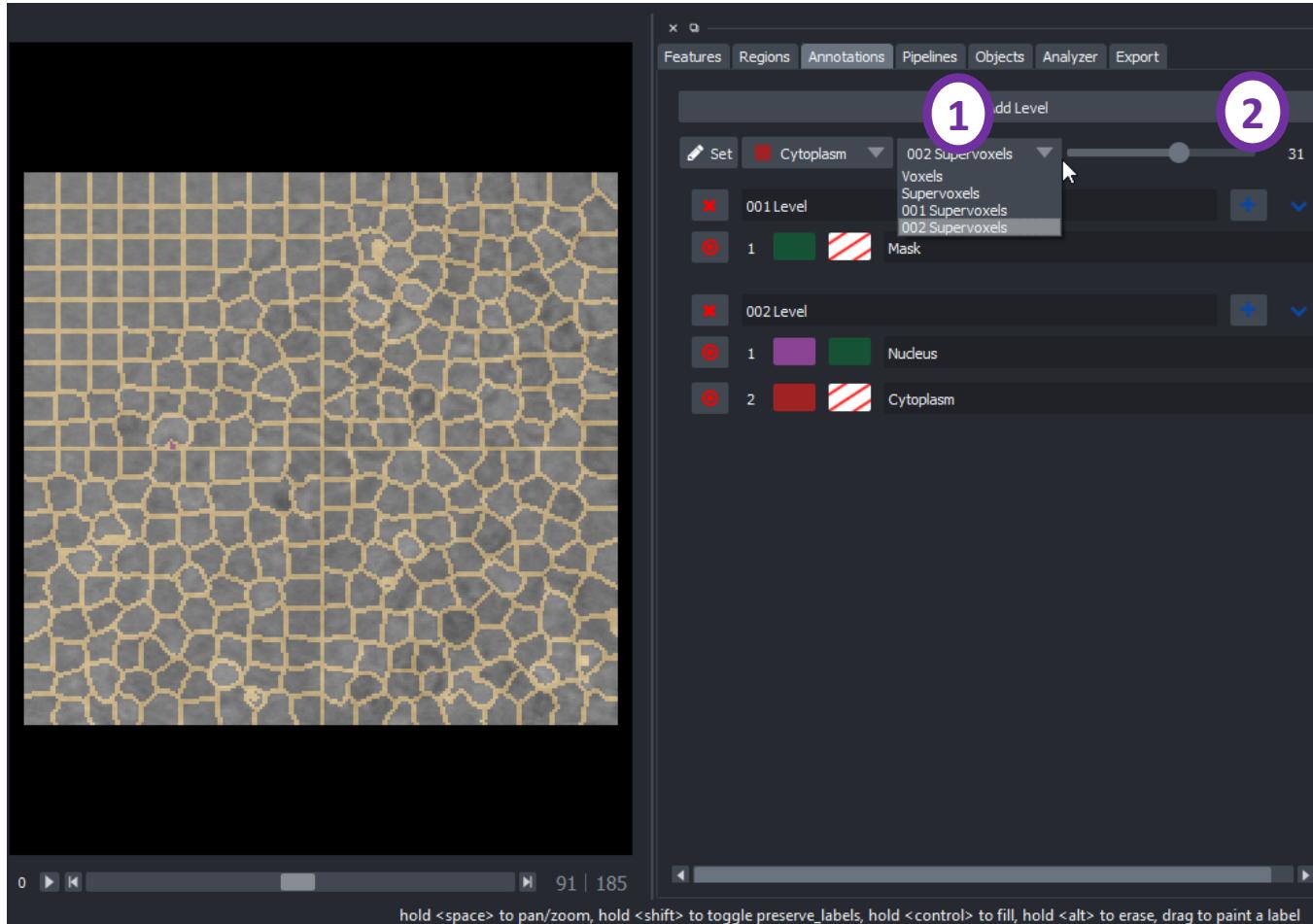
Pen Width = 15



Pen Width = 31

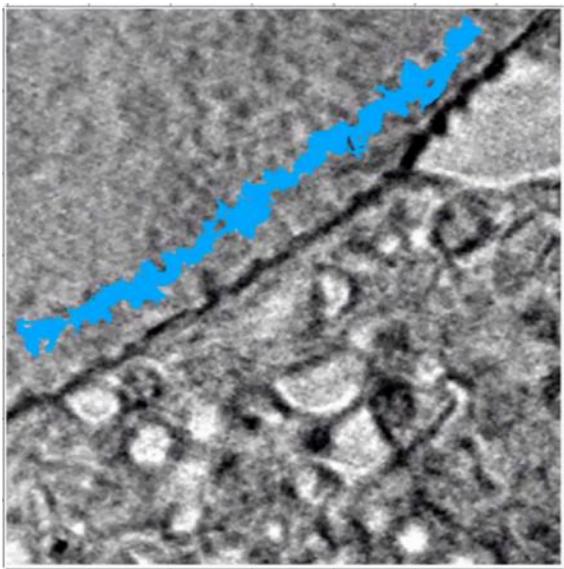
- Annotation using voxels can be done with different pen widths.
- Whilst annotating in voxels annotations do not penetrate the volume in Z

Annotation Using Supervoxels

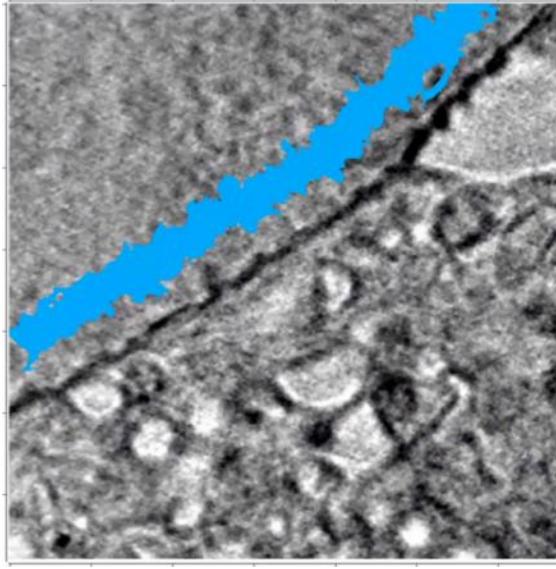


- 1) Select annotation level (voxel, **supervoxel**, or megavoxel)
- 2) Choose a width appropriate for the feature that you are annotating and draw using the left mouse button.

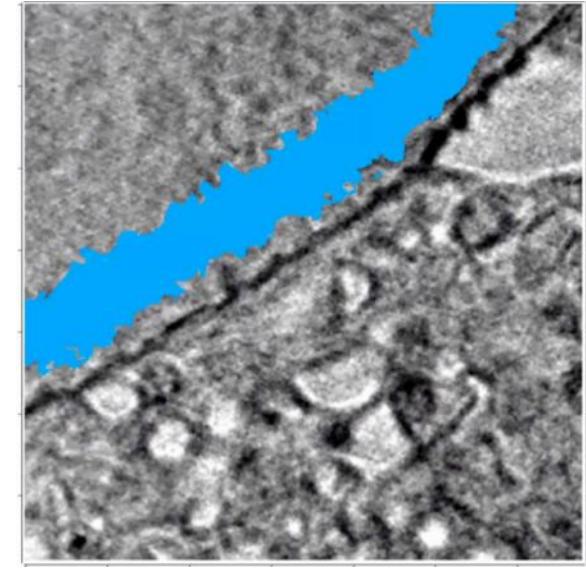
Annotation Using Supervoxels



Pen Width = 1



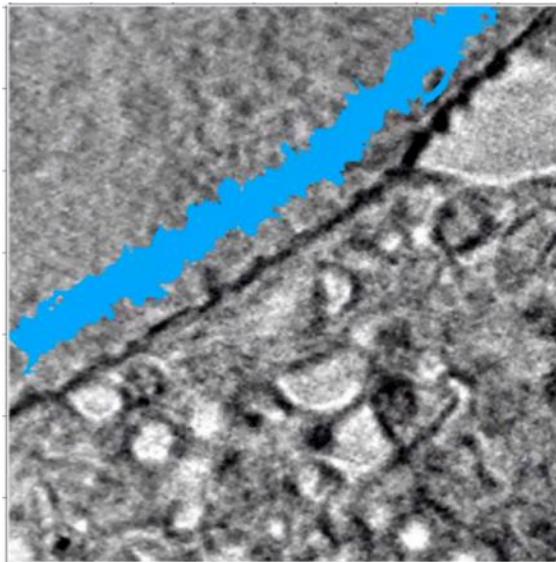
Pen Width = 15



Pen Width = 31

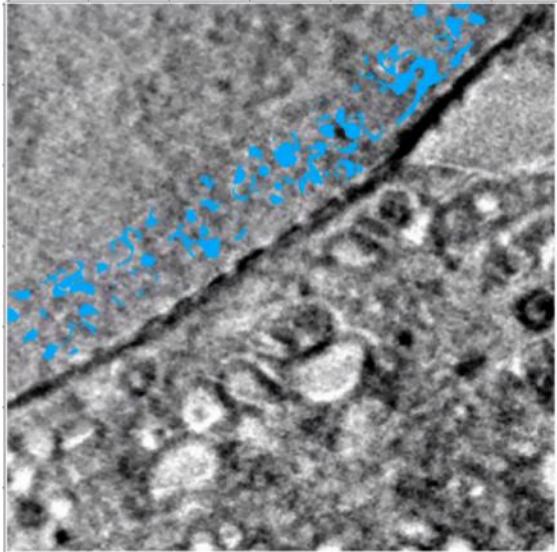
- The entire supervoxel is selected if any paint touches them.
- Whilst annotating in supervoxels annotations penetrate the volume in Z.

Annotation Using Supervoxels

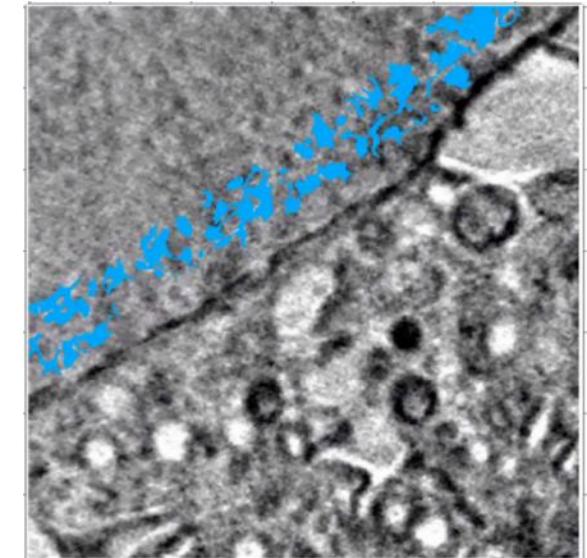


Pen Width = 15
Center Slice

10 slices below

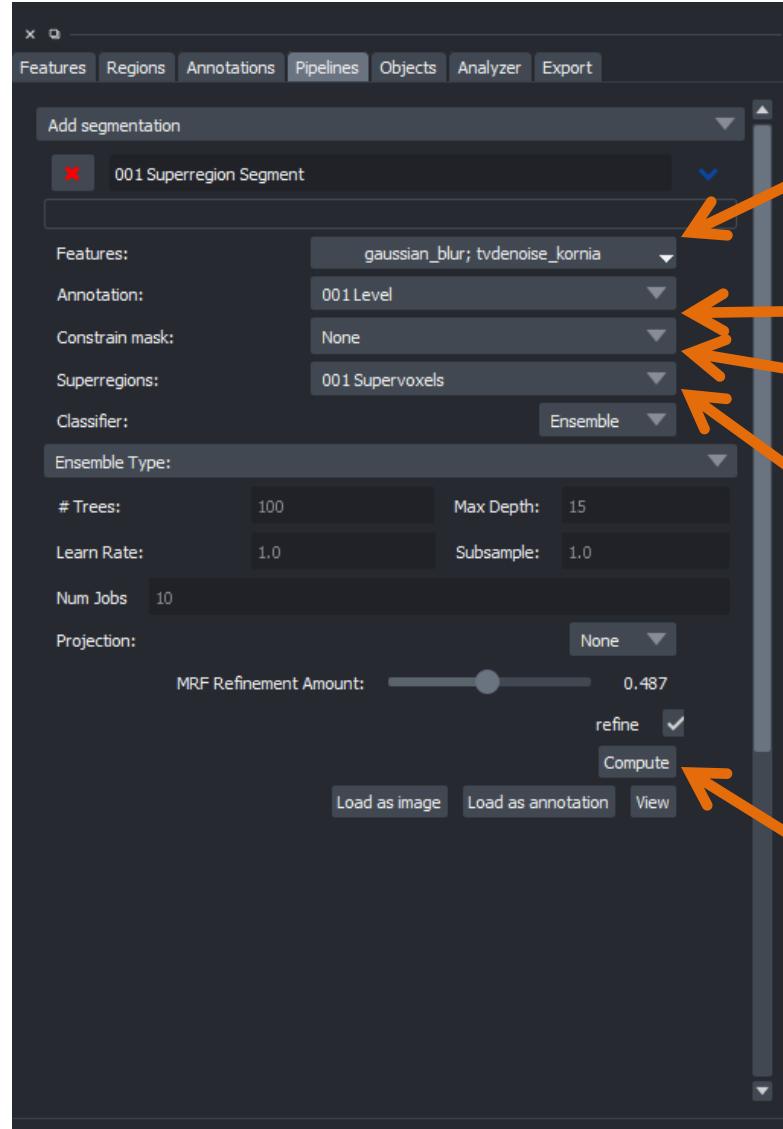


10 slices above



Annotating using
supervoxels penetrates
multiple Z-slices

Super-region Segmentation Pipeline



Select features for training and prediction

- Generally want to choose as many sources as possible
- Generally do not choose raw data
- For large areas try: Total Variation, Hessian Eigenvalues
- For small areas try: the above, plus Gaussian Feature Filters

Choose which level to train with

Optional (can constrain model training region based on annotation levels)

Choose the supervoxel region

Improves the prediction of the resultant model
(The refine checkbox turns off the refinement,
The slider value sets the amount of refinement.)

Train and predict

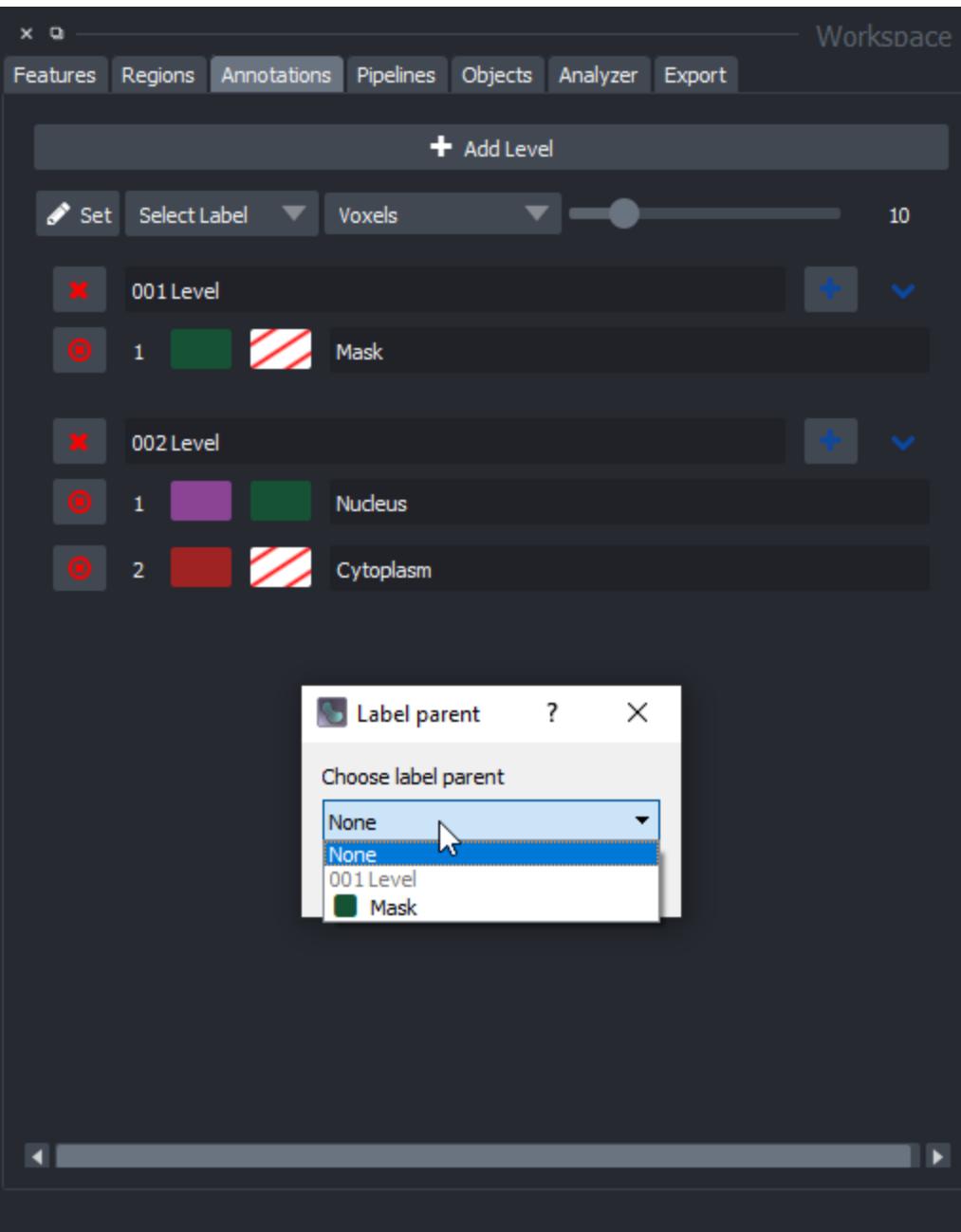
Model Training: Refining Annotations

By clicking ‘Load as Image’ Superregion Segmentation Pipelines can be converted to a feature, where the dilation, erosion and median filters can be applied before export.

Refinement methods:

- **Dilation** - This adds pixels to the inner and outer boundary of the annotation.
- **Erosion** - This strips away layers of pixels from the inner and outer boundary of the annotation. Small objects can be deleted easily with erosion.
- **Median** – This smooths the annotation

Parent-child Relationship Between Levels



- Add a new level
- Add labels to this level
- A parent of the label can be specified by clicking on the parent-child relationship box (next to the color box, initially with diagonal red lines for no parent) for each label.
- In the example shown (left) we are wanting to segment organelles which are located in the cytoplasm, so we choose the mask level (001 Level).

Note: Objects next to each other must be different labels to be considered different objects!

Morphology Filters

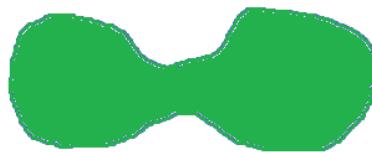
Morphology:

- **Dilation** - This adds pixels to the inner and outer boundary of the annotation.
- **Erosion** - This strips away layers of pixels from the inner and outer boundary of the annotation. Small objects can be deleted easily with erosion.
- **Opening** - Erosion followed by dilation. (todo)
- **Closing** - Dilation followed by erosion. (todo)
- **Fill holes** - Fills holes when a label surrounds them entirely. (todo)

Segmenting Organelles: Refinement

Opening

Starting annotation



Erosion



Dilation



Closing

Dilation



Erosion



SuRVoS Workspace

HDF5: on-disk storage (.h5/.hdf5 extension)

- Read data to memory on-the-fly
- Only load required data

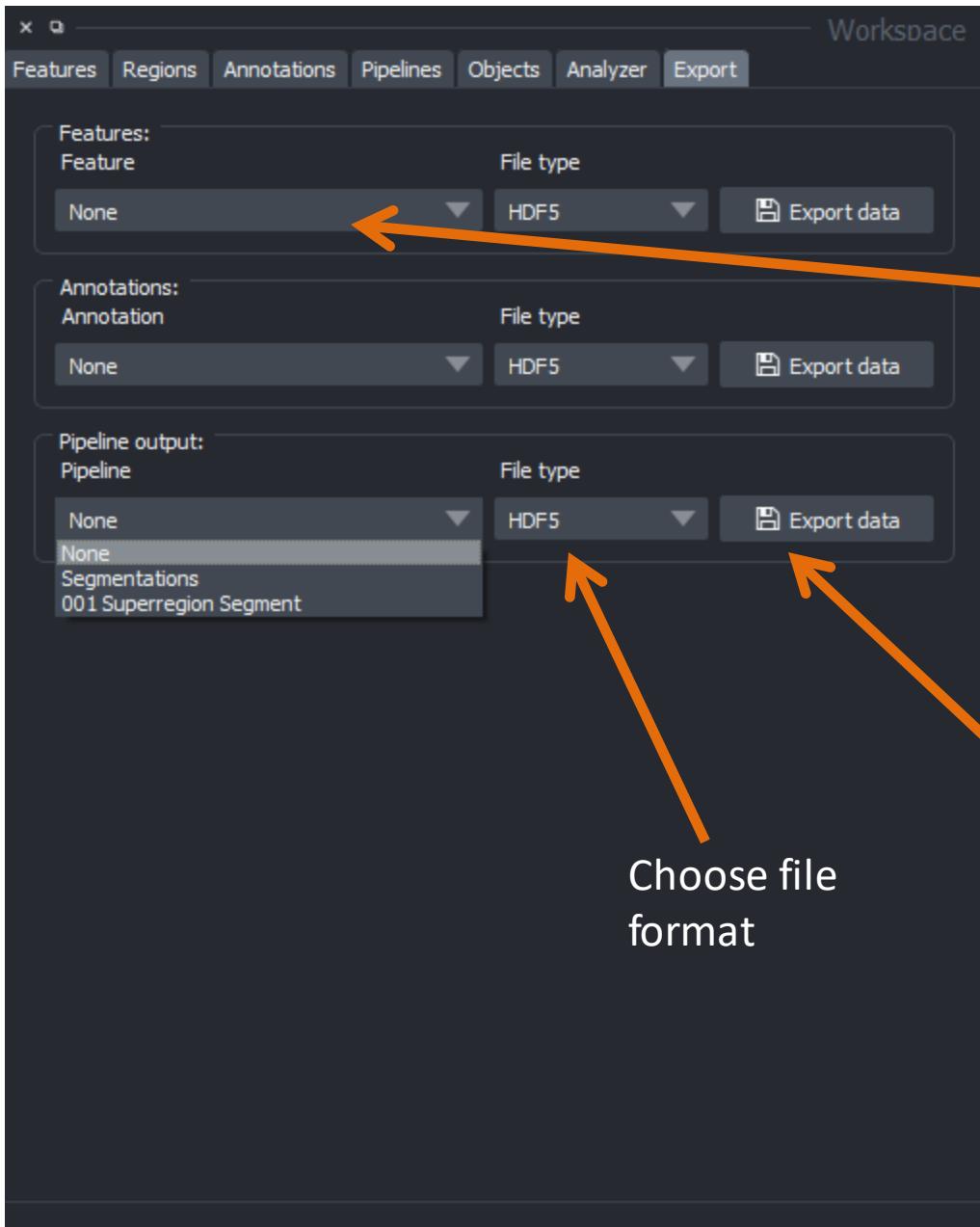
Pros:

- Work with very large data (larger than RAM)
- Work on Region of Interests efficiently
- Safe. Robust.

Cons:

- Performance loss on loading data to memory and saving to disk.

Exporting Data

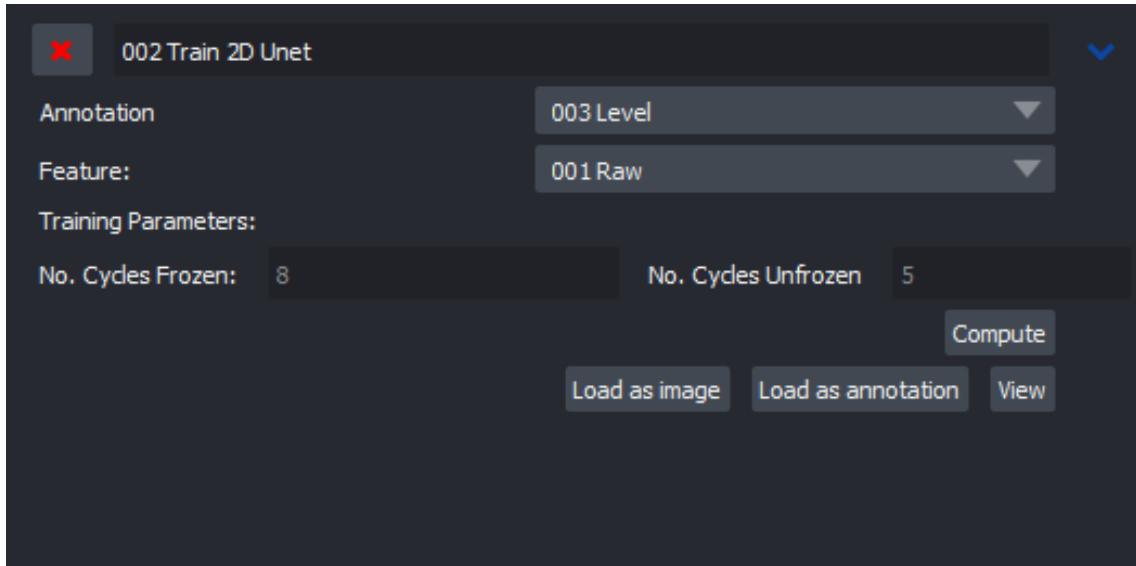


Choose data to export

Choose file format

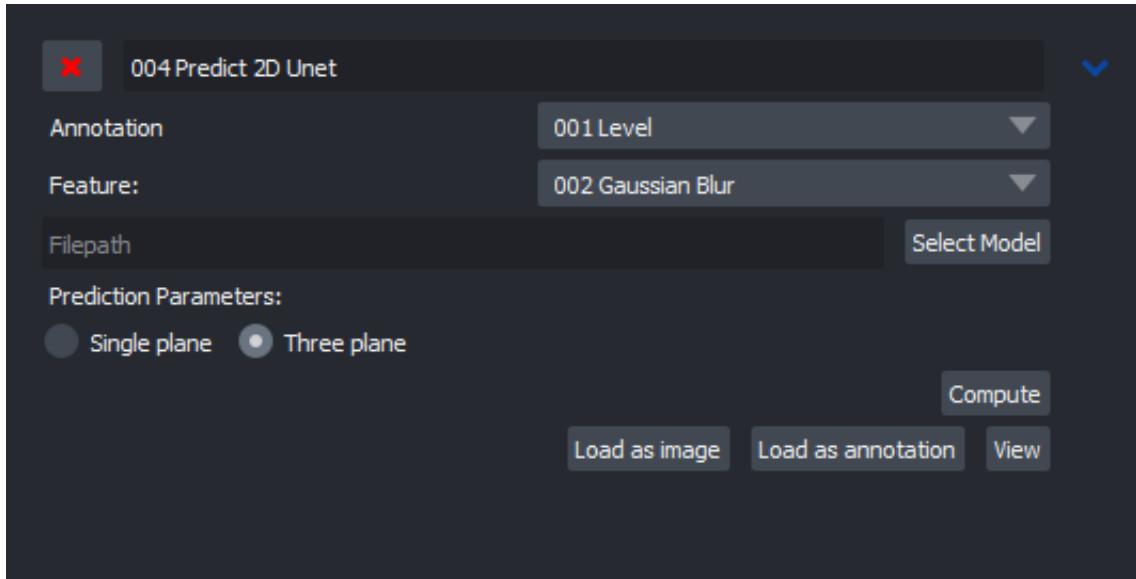
When ready, click export and a file dialog pops up to choose the location to export to.

U-net: Training



1. Choose annotation level to use for training U-net
2. Choose input feature to use as the image data to train on.
3. Set the Number of cycles, both frozen and unfrozen. The defaults here are often effective.
4. Compute. This prints a lot of activity to the terminal.
5. View the results or load them as an image or as annotation (e.g. for editing the output).

U-net: Prediction



1. Predict on a new image volume by selecting the annotation level and feature as in the training step.
2. Select Model, which opens up a file dialog where you have to locate the saved trained U-net model from a training plugin.
3. Select Single-Plane or Three-Plane, where Three Plane predicts in 3 directions and combines the output.
4. Compute.
5. View or Load in as image or as annotation.



Acknowledgements

Nottingham University

Andrew P. French

Tony Pridmore

Diamond Light Source

Imanol Luengo

Michele Darrow

Mark Basham

Avery Pennington

Olly King

Win Min Tun

Liz Duke

Matt Spink

Alan Ashton

Kyle Dent

Also thanks to

Nvidia Corporation for the donation of a Tesla K40.

Questions?

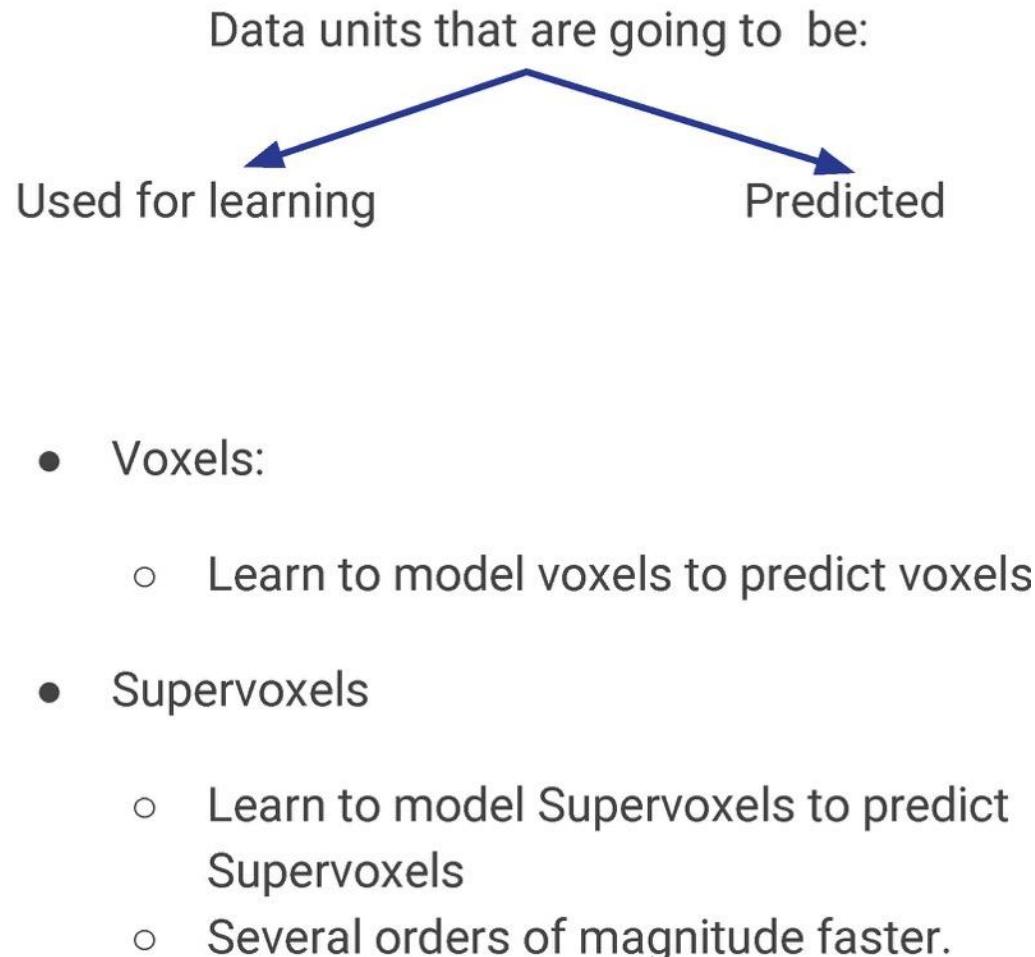
Model Training

Learn from annotations. Propagate through volume.

- Data Points
- Descriptors
- Annotations
- Classifier
- Refinement
- Confidence

Model Training

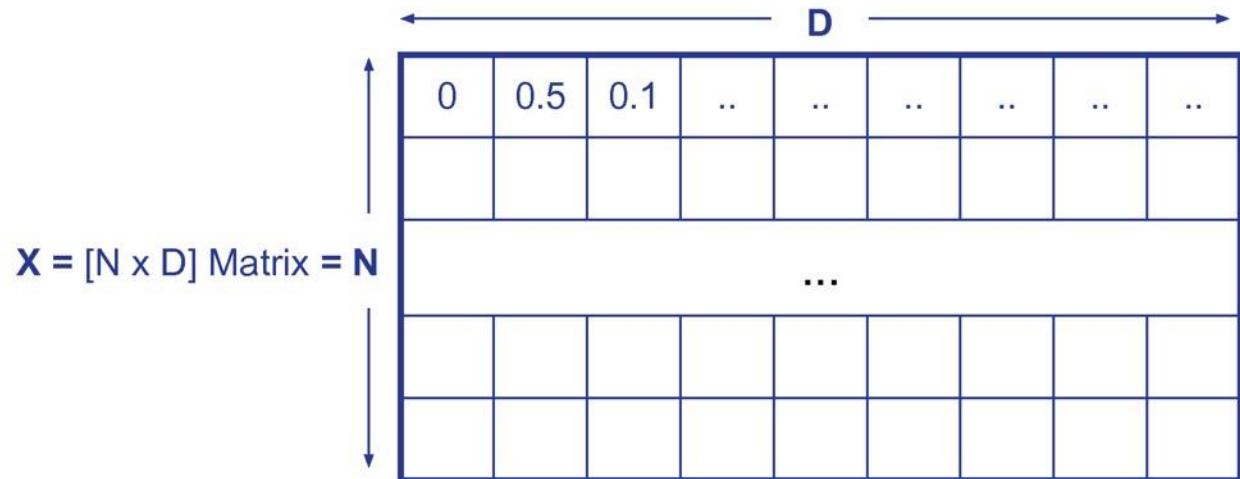
- Data points
- Descriptors
- Annotations
- Classifier
- Refinement
- Confidence



Model Training

- Data points
- **Descriptors**
- Annotations
- Classifier
- Refinement
- Confidence

Data points are represented with descriptors. E.g.
Voxel descriptors are created by concatenating
features extracted from **Data Preprocessing**.



N: Number of data points; **D:** Number of features

Descriptor Types:

- Voxels
- Supervoxels

Model Training

- Data points
- Descriptors
- **Annotations**
- Classifier
- Refinement
- Confidence

In order to learn to classify between different labels, for some data points annotations are needed.

$\mathbf{X} = [N \times D]$ Data Matrix

$\mathbf{Y} = [N \times 1]$ Matrix = \mathbf{N}



N: Number of data points

D: Number of features selected

X: Descriptor Matrix

Y: class type for each of the data points in **X**

- **>0** = class for the data point (e.g. **0=nucleus**)
- **-1** = unknown class. What we want to predict.

Model Training

- Data points
- Descriptors
- Annotations
- **Classifier**
- Refinement
- Confidence

Simplified example with $N = 4$ data points and $D = 2$ features.

$X =$

	x_1	x_2	
	0.0	0.0	0
	0.0	1.0	0
	1.0	0.0	0
	1.0	1.0	1

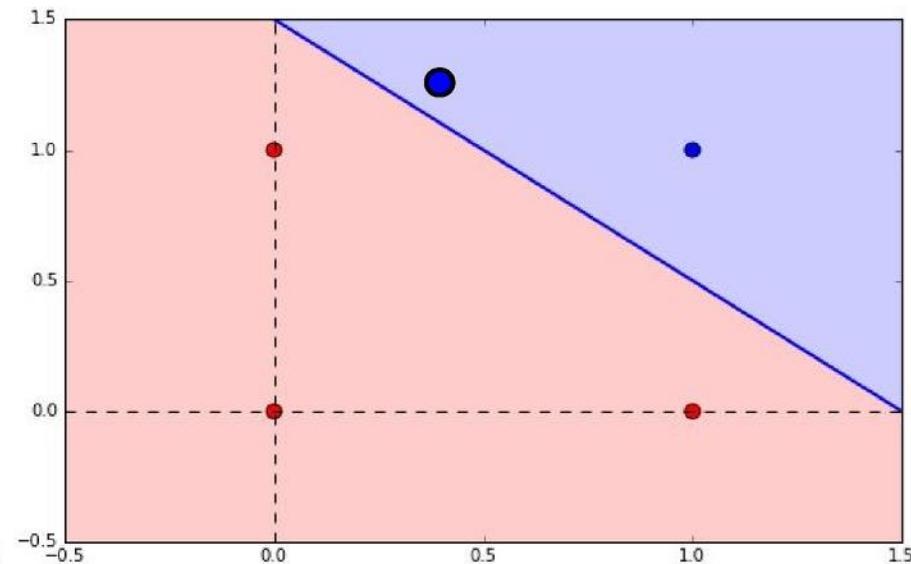
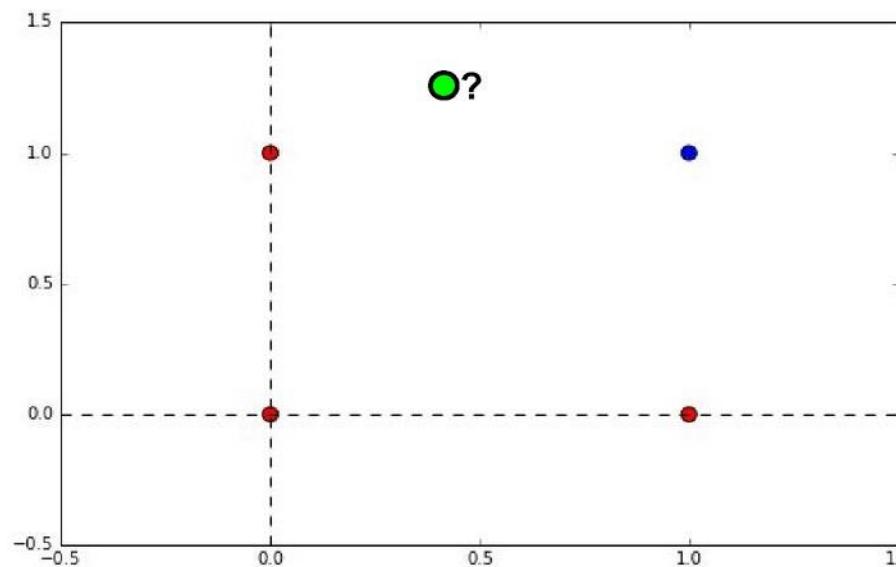
$Y =$

0
0
0
1

Model: Line

$$y = w_0 + x_1 * w_1 + x_2 * w_2$$

Guess optimal w_0, w_1, w_2



Model Training

- Data points
- Descriptors
- Annotations
- **Classifier**
- Refinement
- Confidence

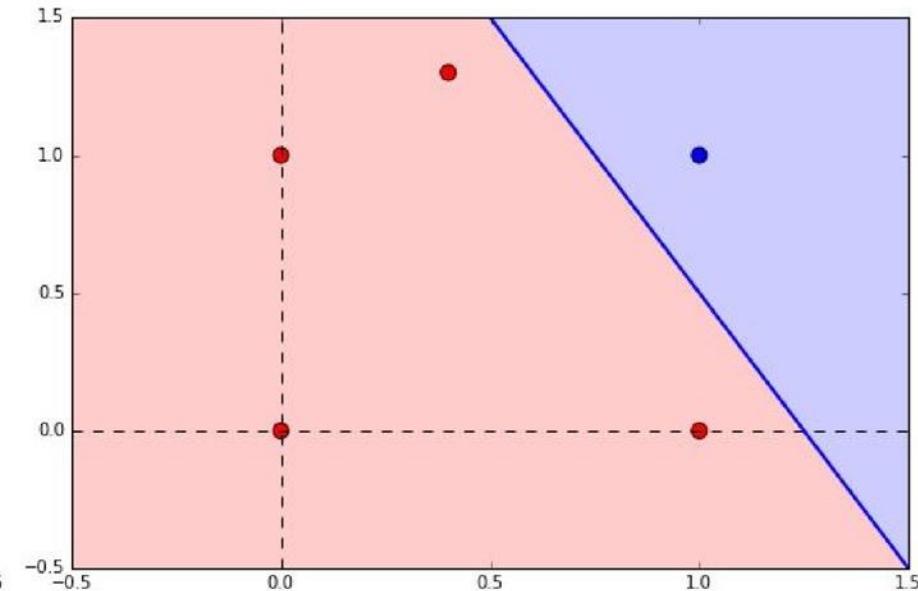
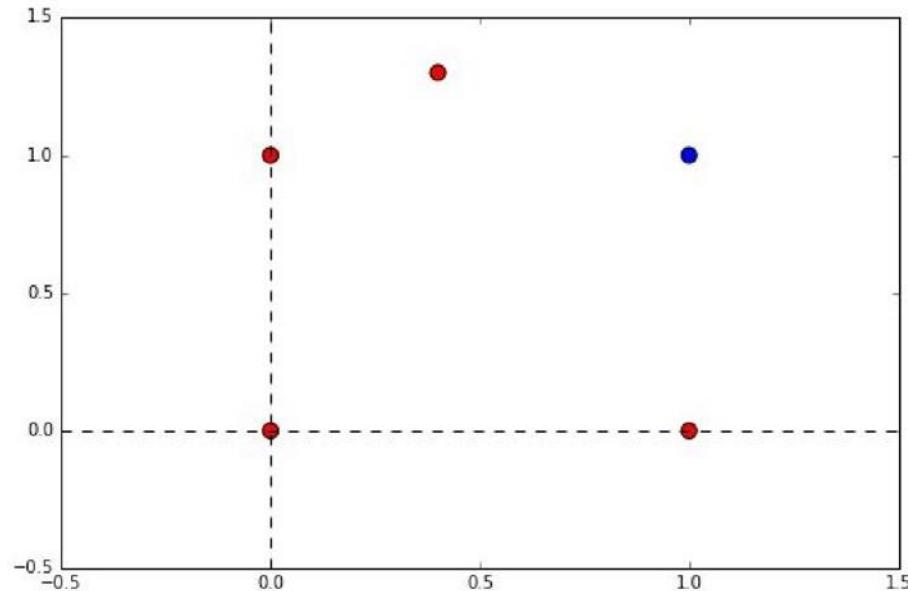
PROBLEM:

- On interactive Segmentation we only have limited data available.

What if after inspection we realise it actually should be **red**?

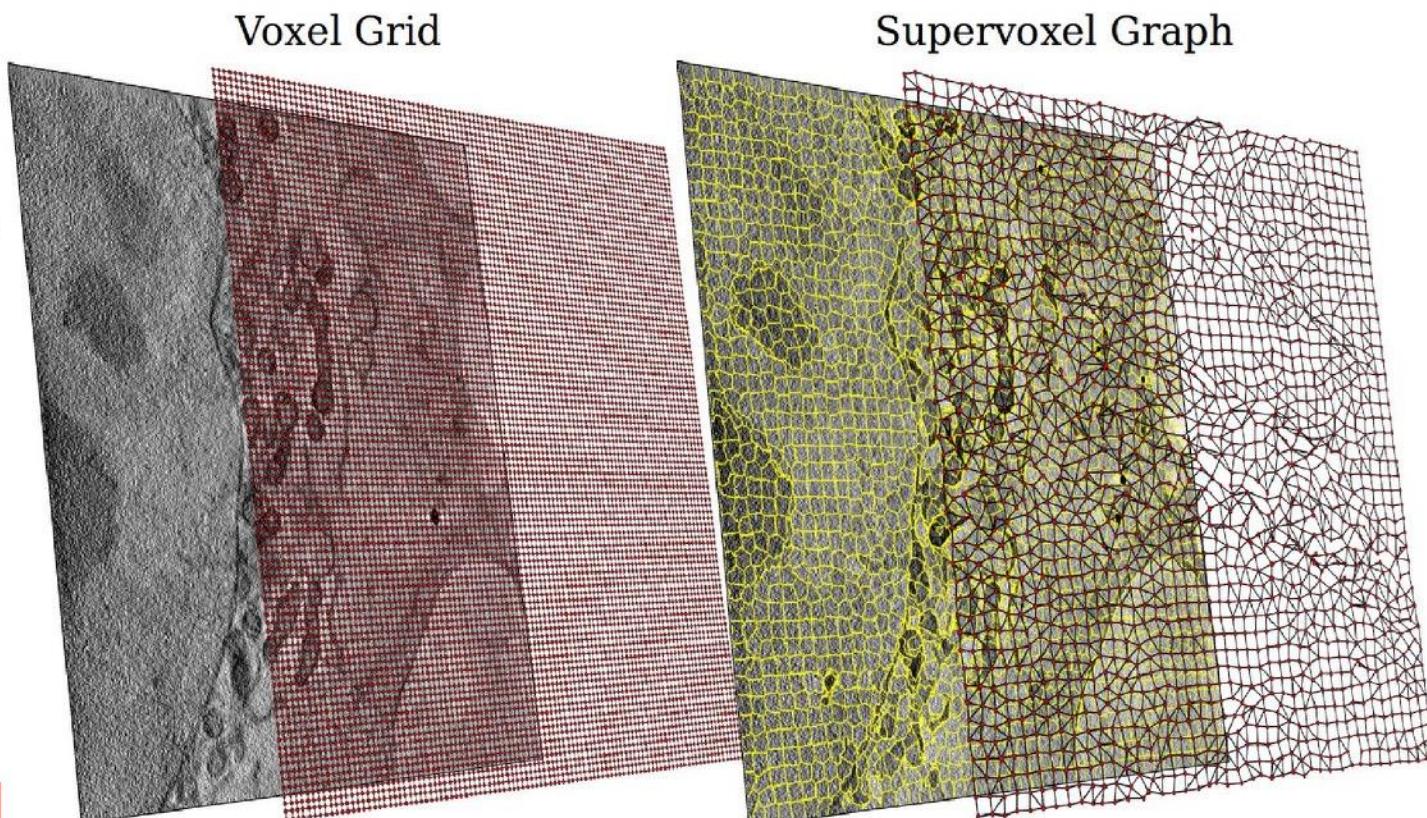
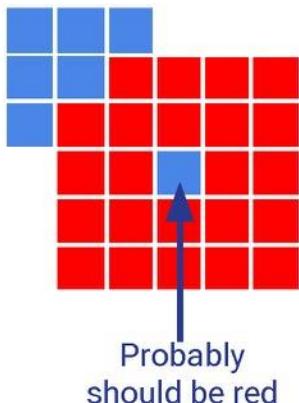
Annotate + retrain

Refine Classification



Model Training

- Data points
- Descriptors
- Annotations
- Classifier
- **Refinement**
- Confidence



- Add Spatial Consistency to the Predictions
- Encourage nearby **voxels/supervoxels** to have belong to the same class.

Model Training

- Data points
- Descriptors
- Annotations
- Classifier
- Refinement
- **Confidence**

How confidence is the **Classifier** with the prediction it has made.

