SuRVoS Workbench

Super-**R**egion **Vo**lume **S**egmentation

Imanol Luengo

Index

- The project
- What is **SuRVoS**
- **SuRVoS** Overview
- What can it do

- Overview of the internals
- Current state & Limitations
- Future direction
- Goal of the project

SuRVoS Project

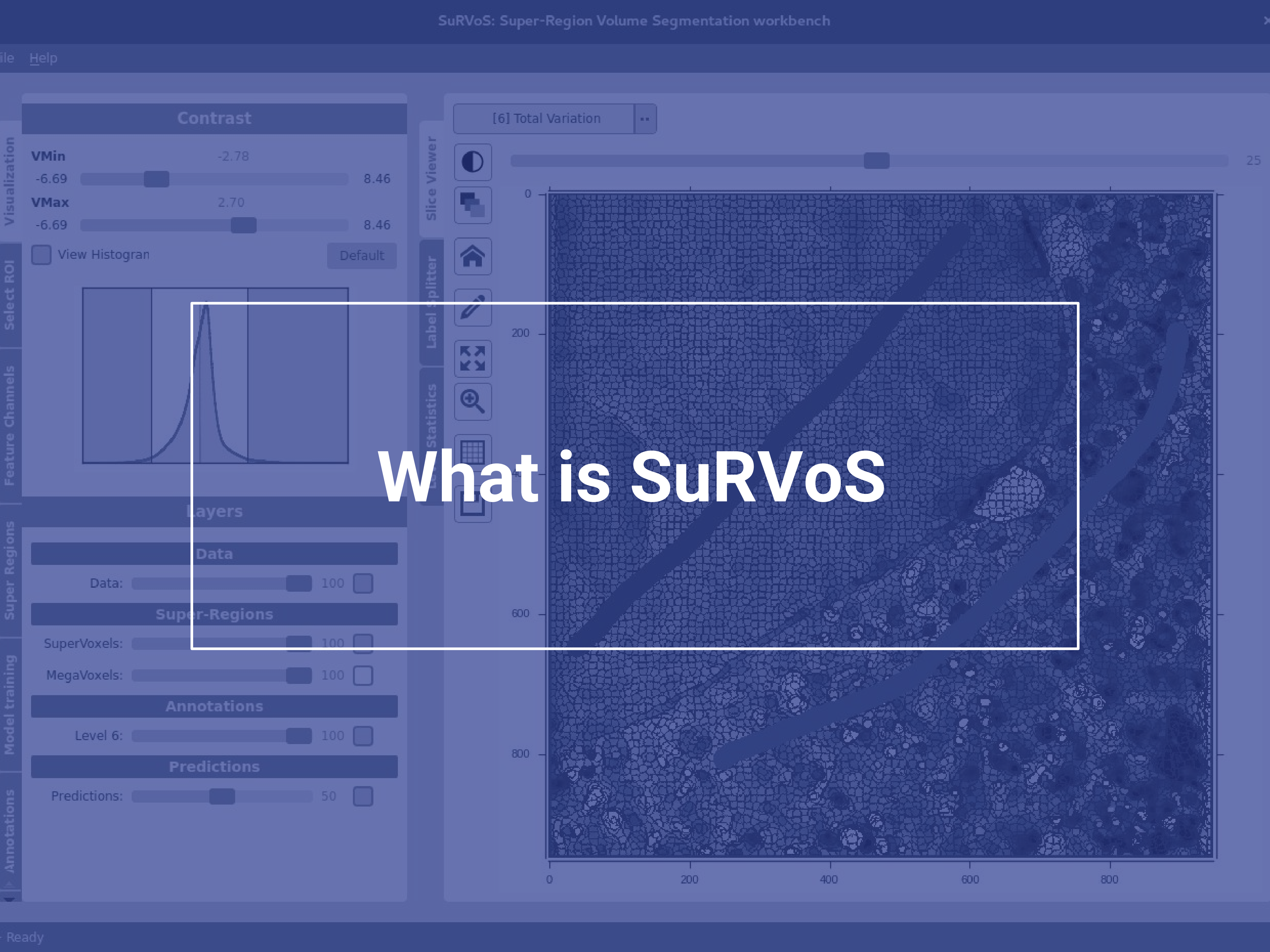
University of Nottingham:

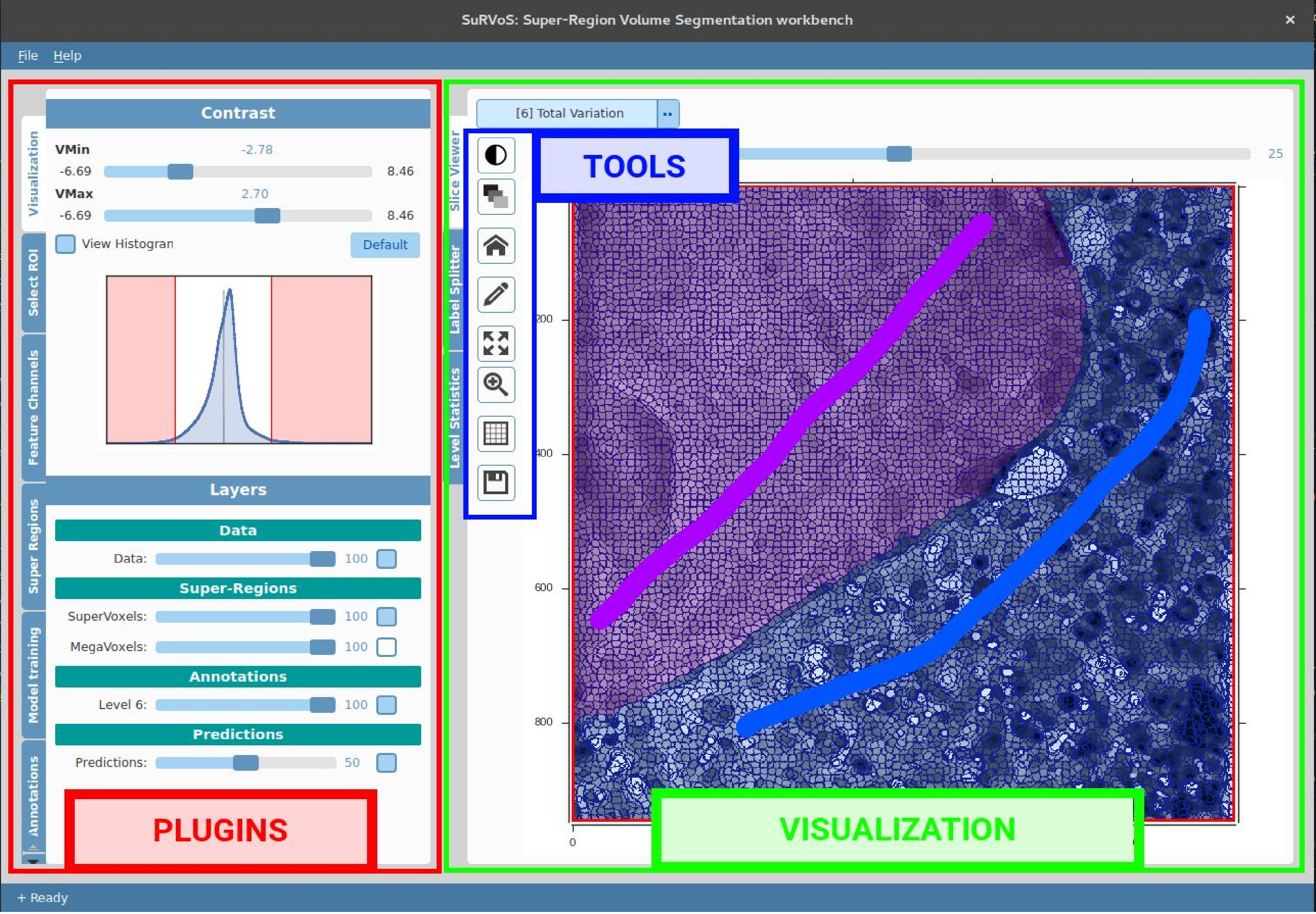
- Computer Vision Laboratory

Diamond Light Source:

- B24: Cryo Transmission X-ray Microscopy

What is SuRVoS





https://DiamondLightSource.github.io/SuRVoS

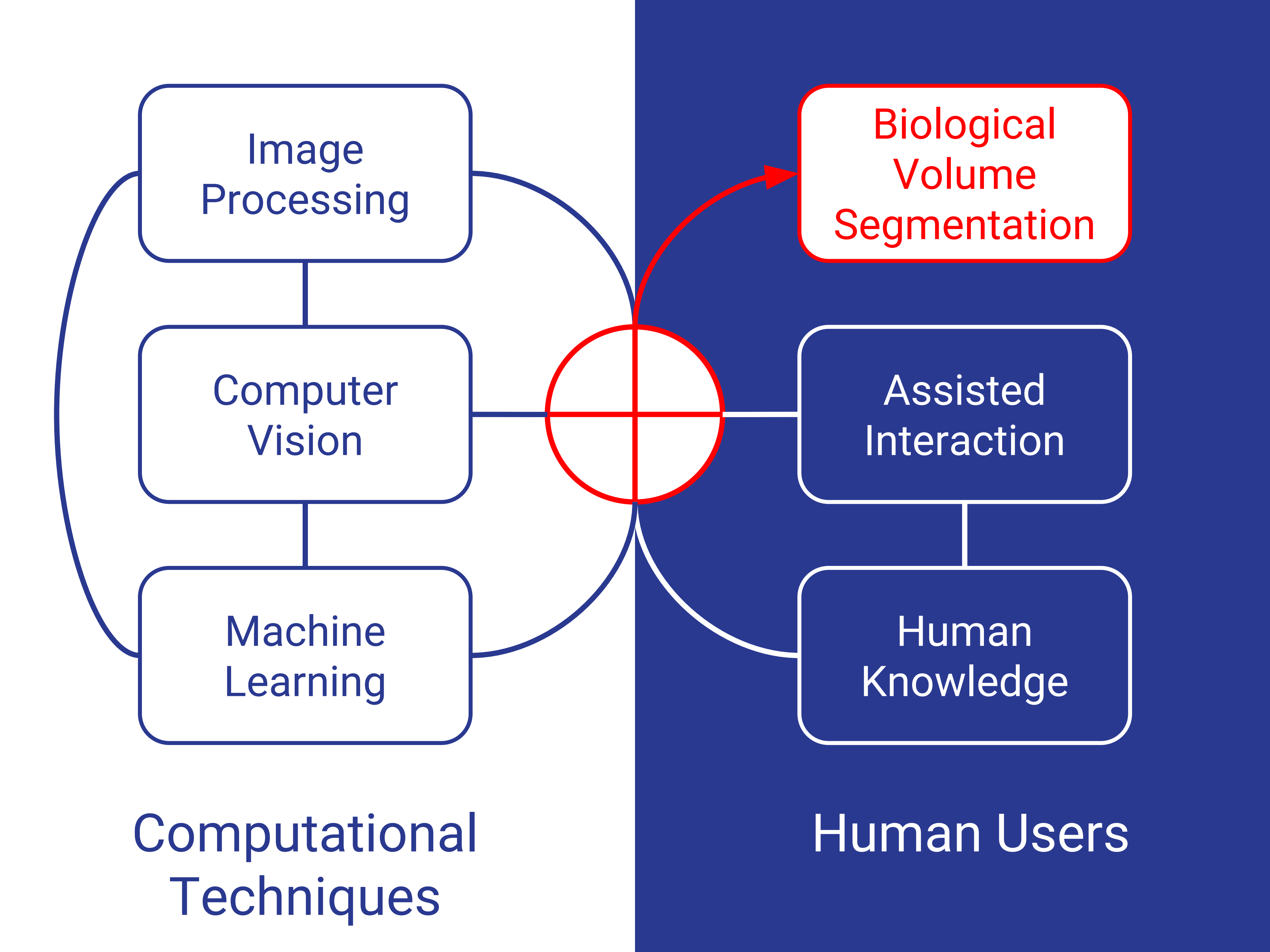
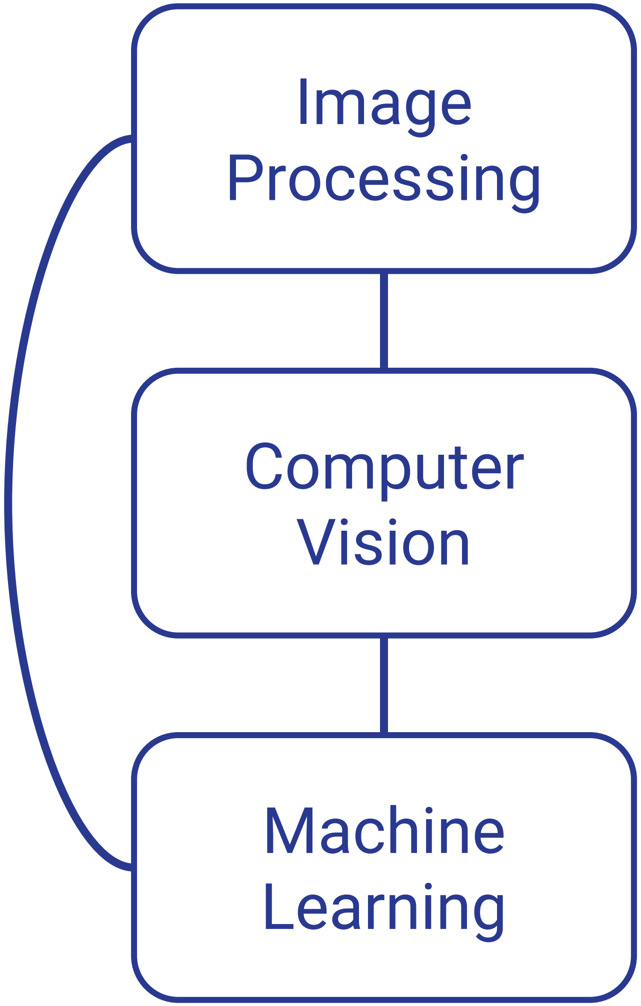


Image Processing
Computer Vision
Machine Learning
Biological Volume Segmentation
Assisted Interaction
Human Knowledge
Computational Techniques
Human Users



Computational Techniques

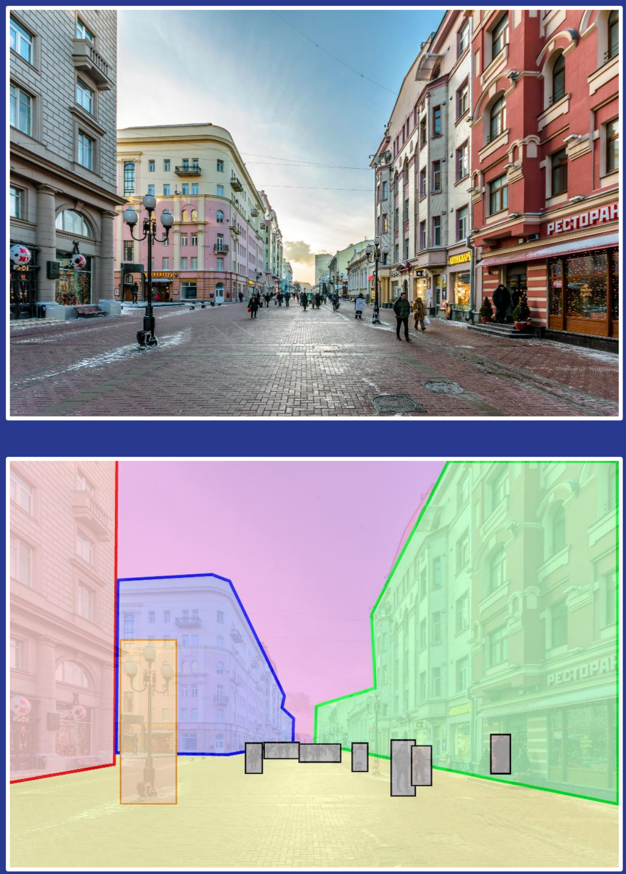
Computer Vision

Digital Image Understanding

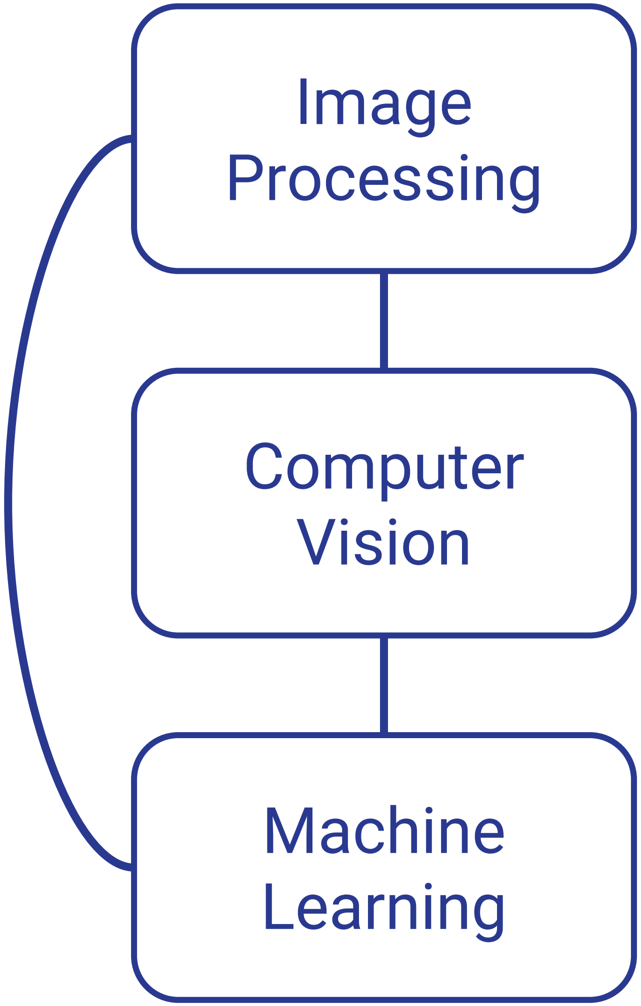
Processing

Analysis

*



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



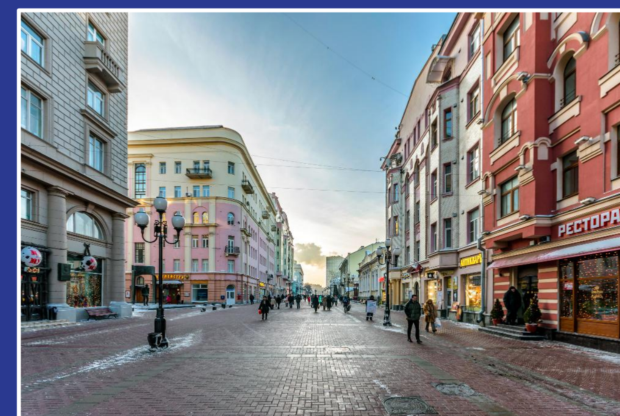
Computational Techniques

Image Processing

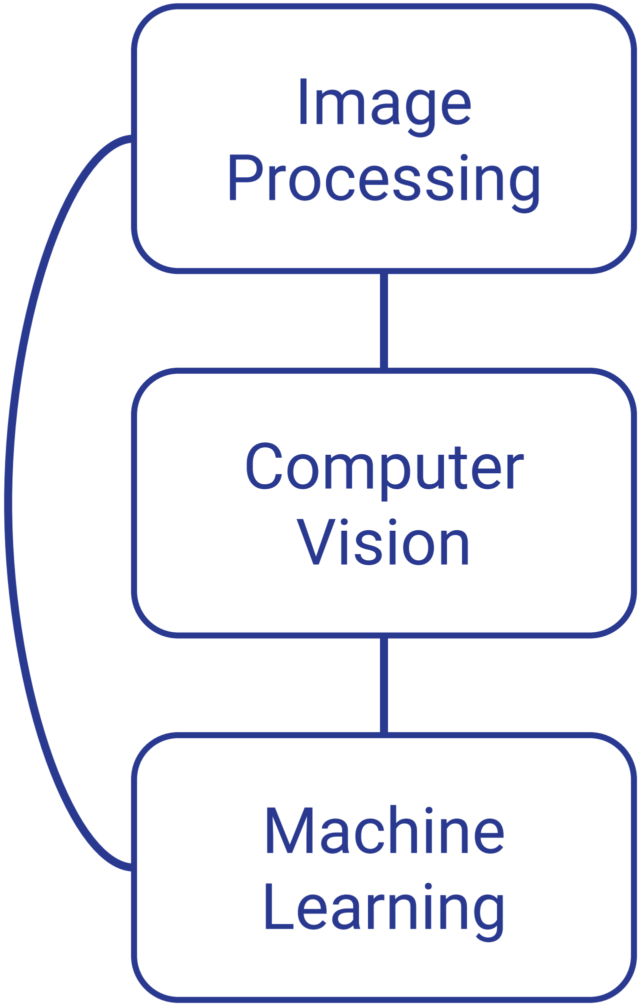
Image Manipulation and Enhancing

Noise Reduction | Feature Extraction

*



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



Computational Techniques

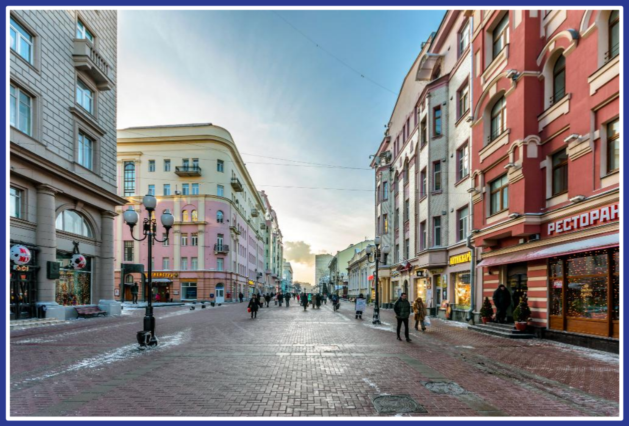
Machine Learning

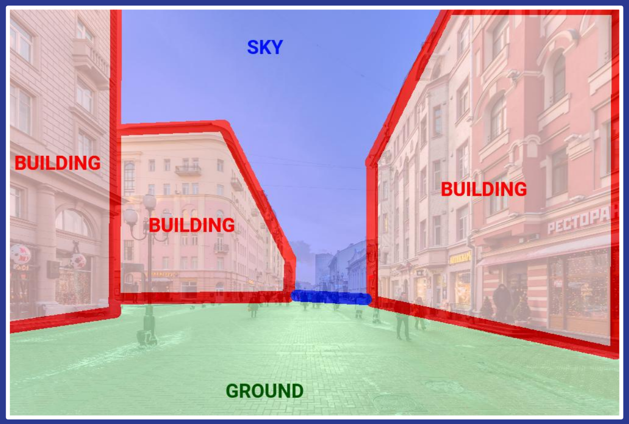
Computational Learning

Classification

Regression

*

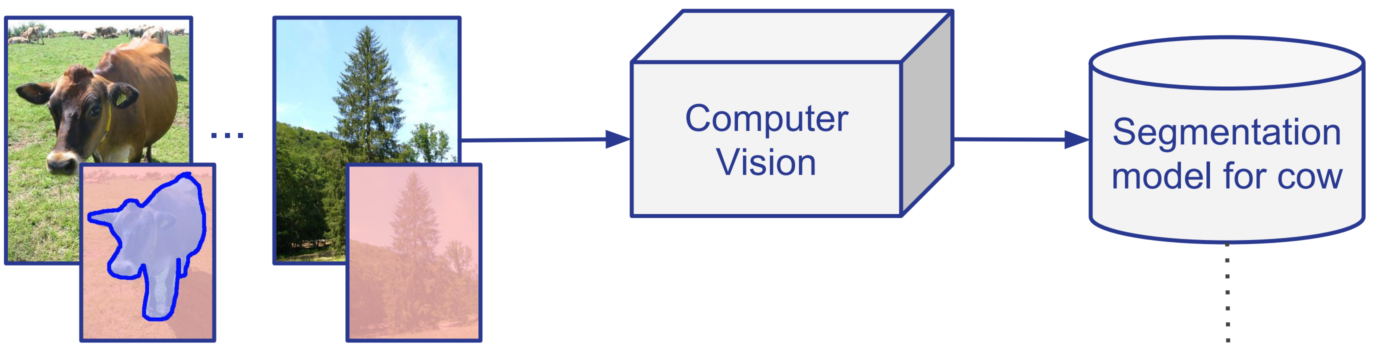




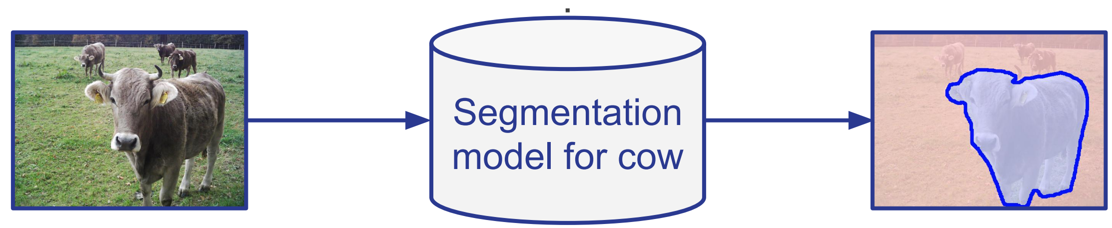
* Original Image: https://en.wikipedia.org/wiki/Arbat_Street

Automatic Computer Vision

Training: Learning to identify cows



Testing: Predicting if there is a cow in the image



Automatic Computer Vision

Pros:

- Learns from annotations.
- Fast and accurate results.
- Completely automatic.
- Generalizable for new images.

Cons:

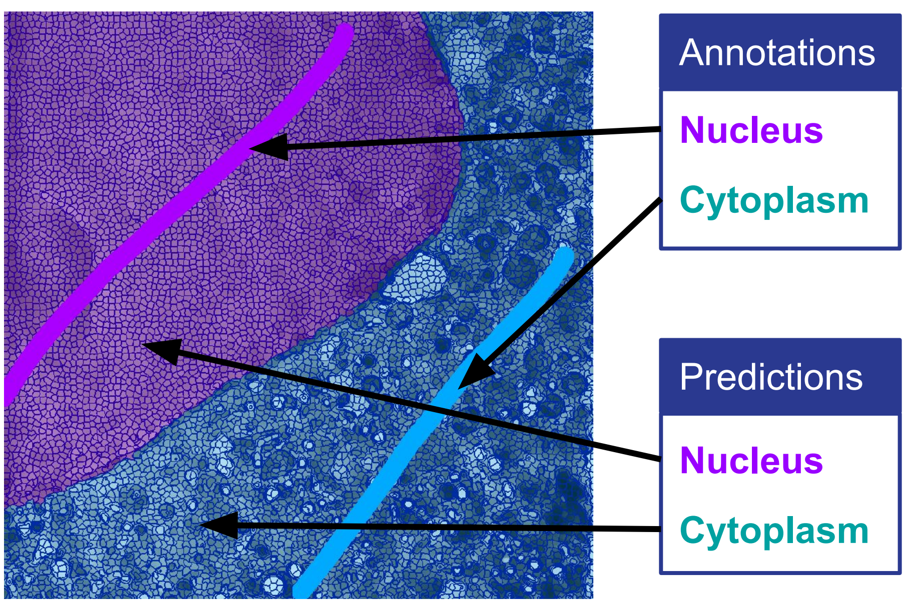
- Requires a lot of annotated data.
- Slow learning process.
- Can only learn specific tasks.
- Data to be analysed has to be similar.

SuRVoS

Biological Volume Segmentation

Problem

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



SuRVoS

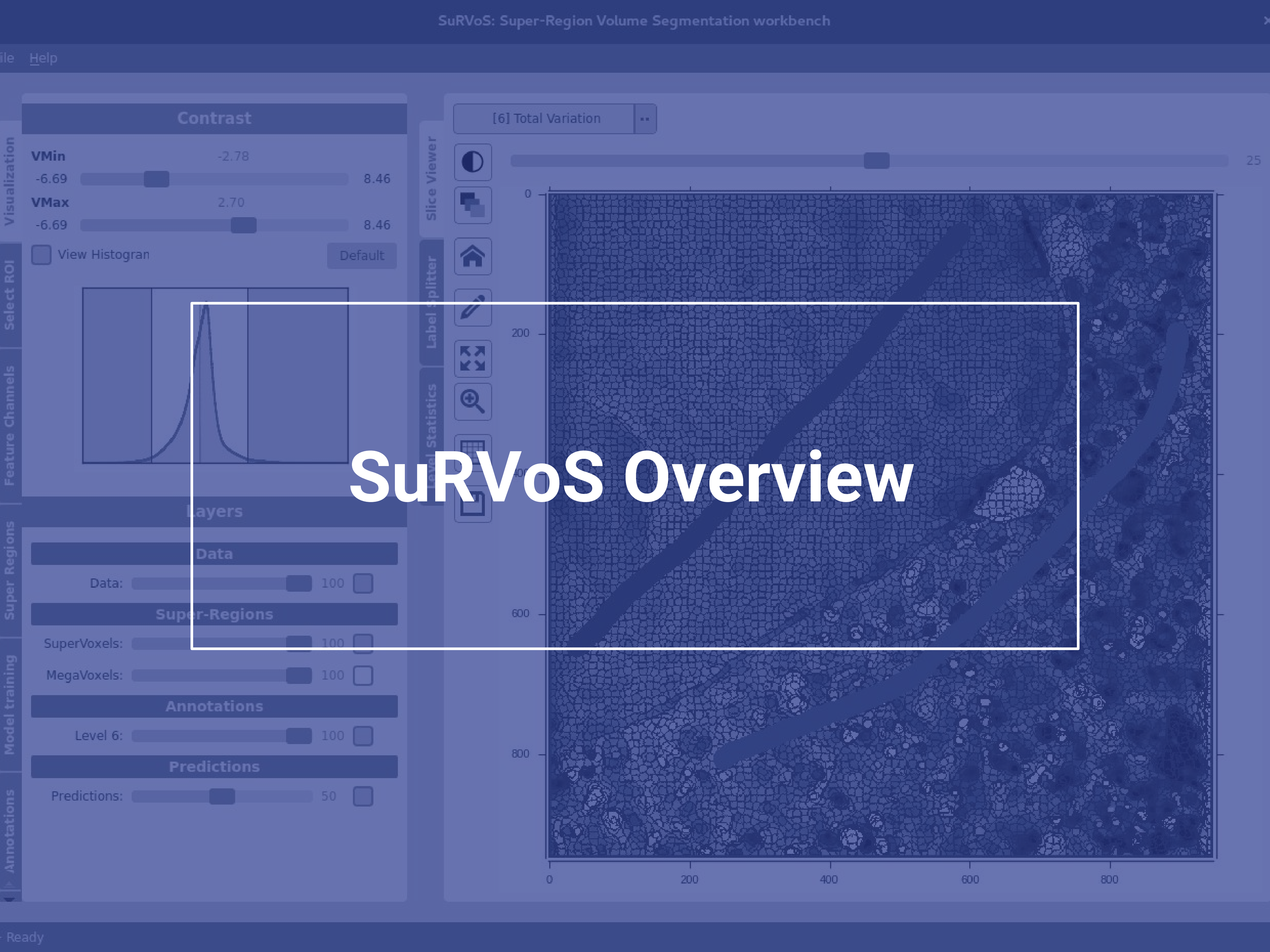
- Assist the user to annotate data.
- Learn to segment with user annotations.

Assisted Interaction

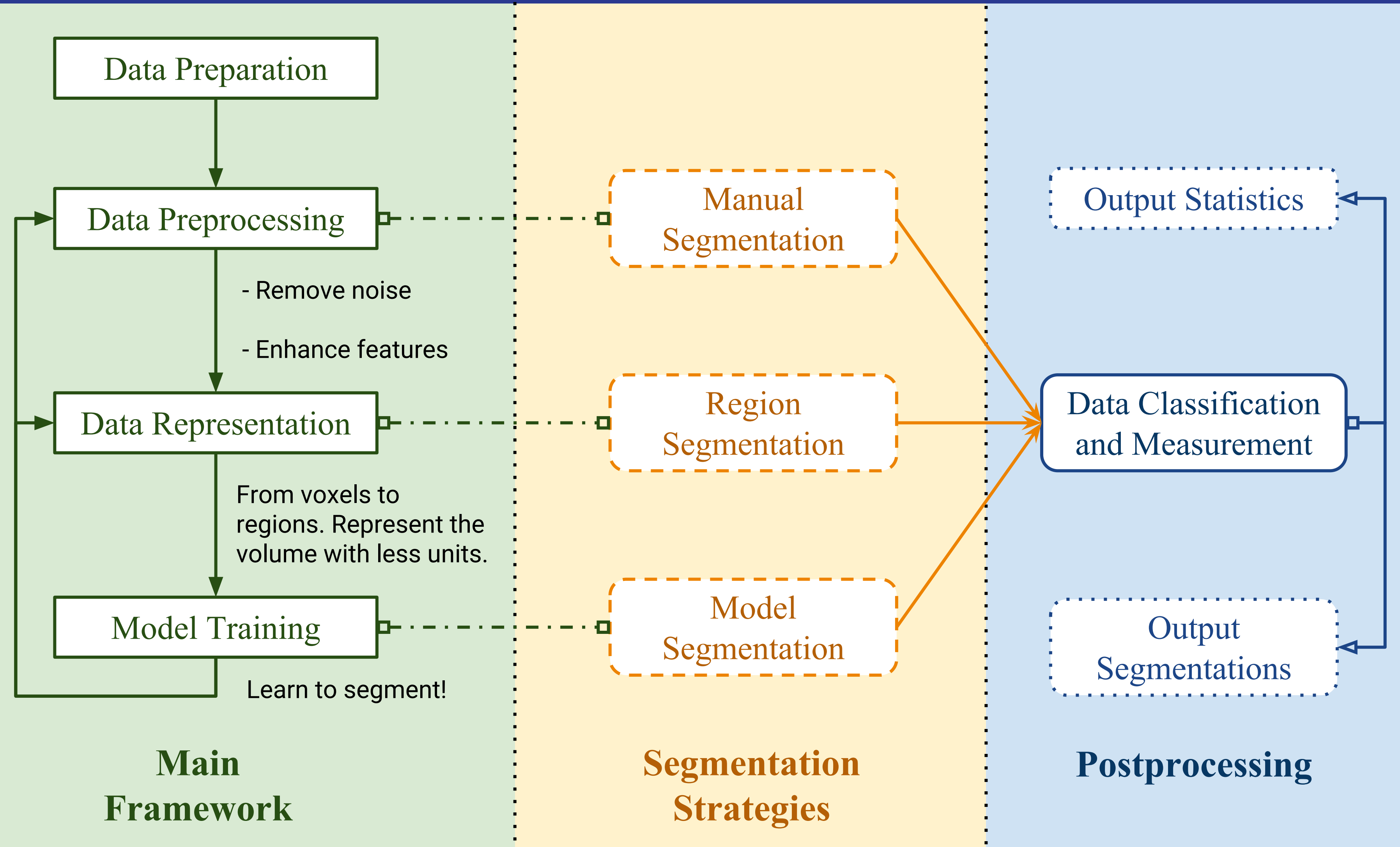
Human Knowledge

Human Users

SuRVoS Overview



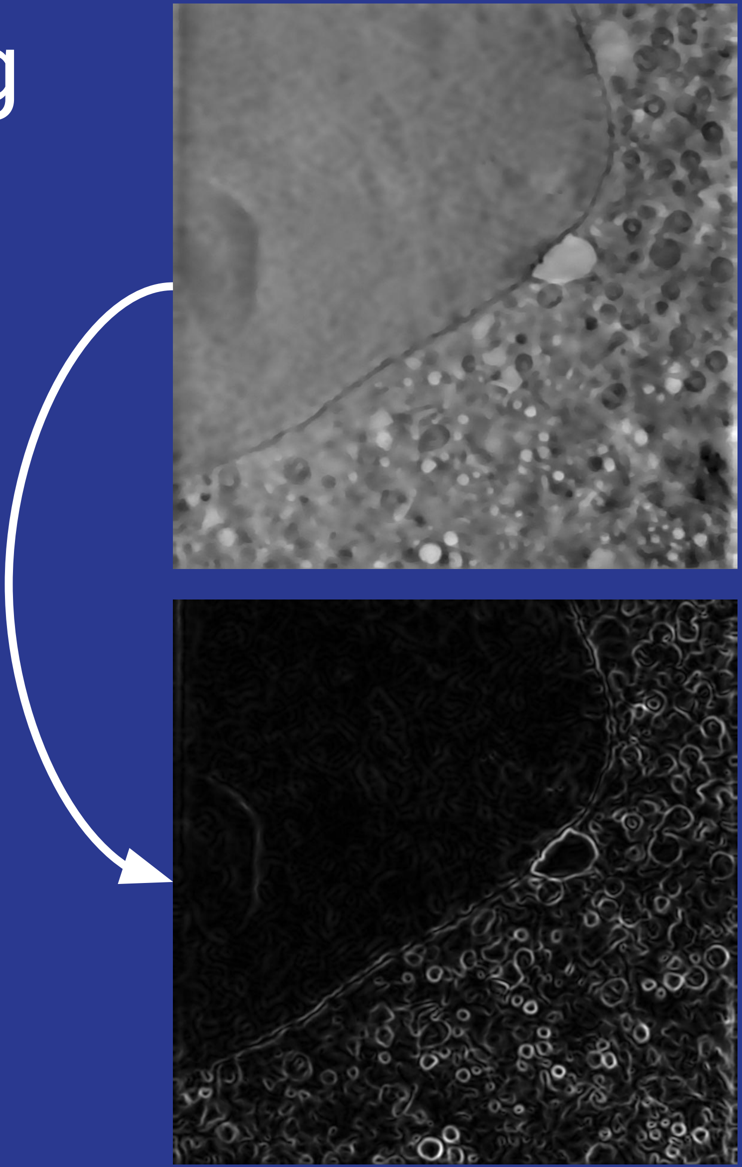
SuRVoS Overview



Data Preprocessing

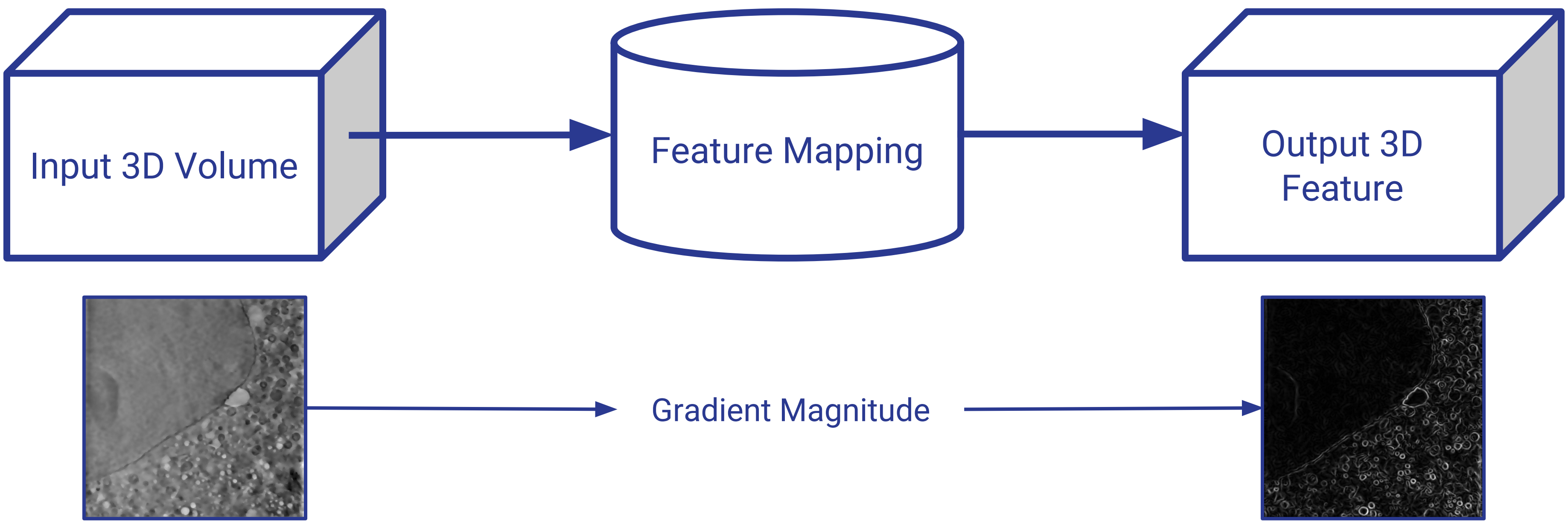
Enhancing data properties

- Raw Features
- Denoising
- Local Features
- Gaussian Features
- Blob-like Detection
- Texture and Structure
- Robust Features



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

- **Raw Features**
- Denoising
- Local Features
- Gaussian Features
- Blob-like Detection
- Texture and Structure
- Robust Features

Thresholding

$$x \rightarrow \begin{cases} x & \text{If } x > \mathbf{T} \\ 0 & \text{otherwise} \end{cases} \rightarrow y$$

Inverse Thresholding

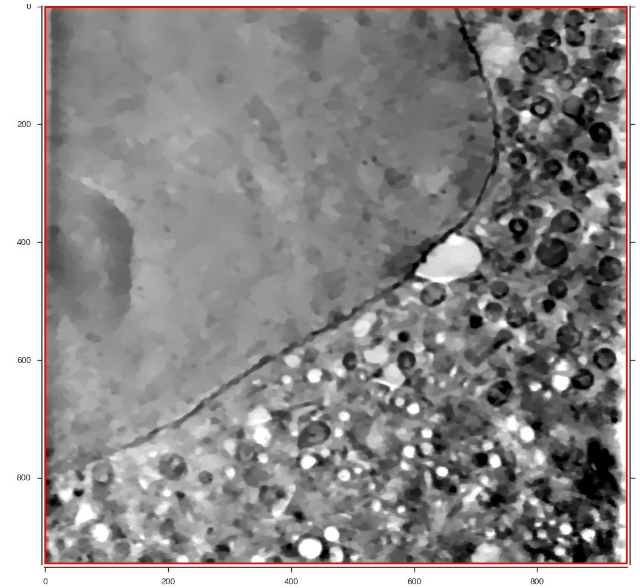
$$x \rightarrow \begin{cases} -x & \text{If } x < \mathbf{T} \\ 0 & \text{otherwise} \end{cases} \rightarrow y$$

Data Preprocessing

- Raw Features
- **Denoising**
- Local Features
- Gaussian Features
- Blob-like Detection
- Texture and Structure
- Robust Features

Original Image

Gaussian Smooth

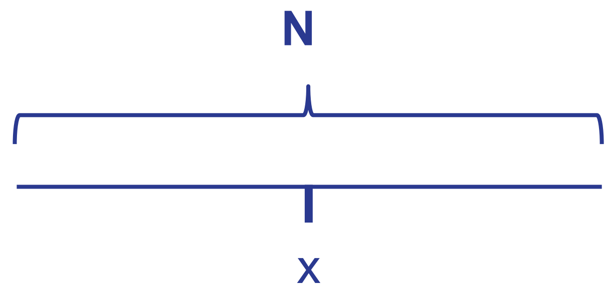


Total Variation

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

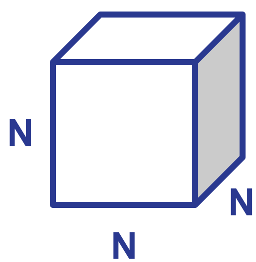
Data Preprocessing

- Raw Features
- Denoising
- **Local Features**
- Gaussian Features
- Blob-like Detection
- Texture and Structure
- Robust Features

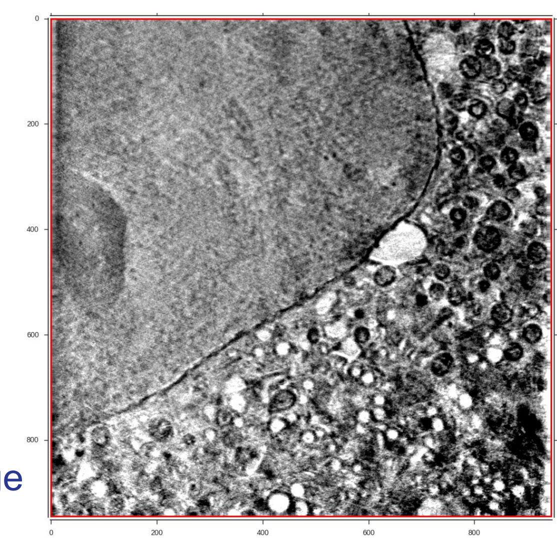


1D Uniform Neighbourhood of size N:

Every pixel in the neighbour is aggregated to **x** with equal importance

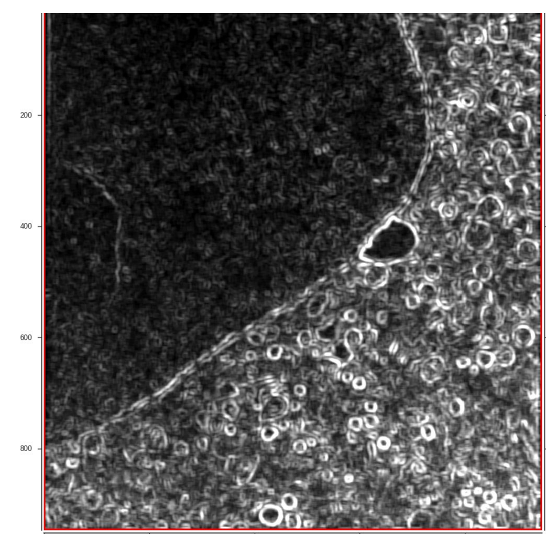


- A cube of shape N x N x N centered on each voxel **x**
- Function of all the intensities in that cube



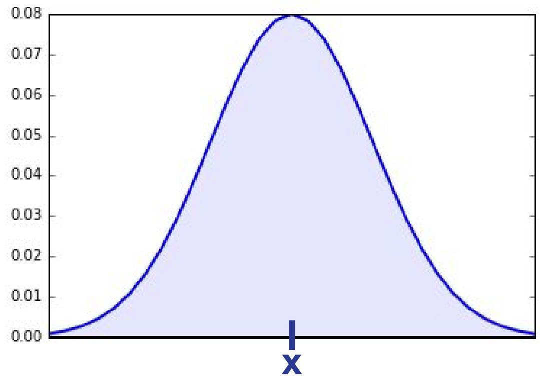
Original Image

Standard Deviation



Data Preprocessing

- Raw Features
- Denoising
- Local Features
- **Gaussian Features**
- Blob-like Detection
- Texture and Structure
- Robust Features



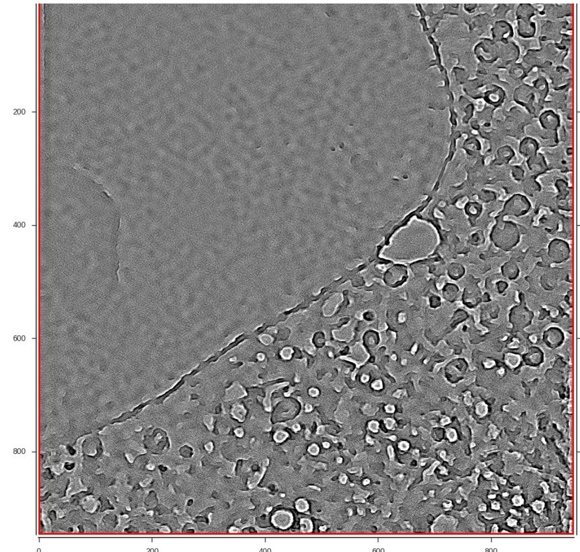
1D Gaussian:

Pixels near the center have more importance.

- A Gaussian neighbourhood of size N x N x N centered on every pixel
- Better data fidelity.

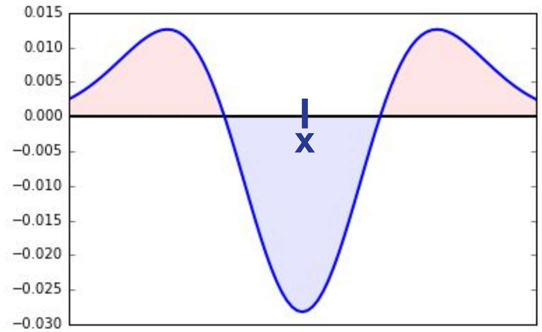
Original Image

Mean Subtraction



Data Preprocessing

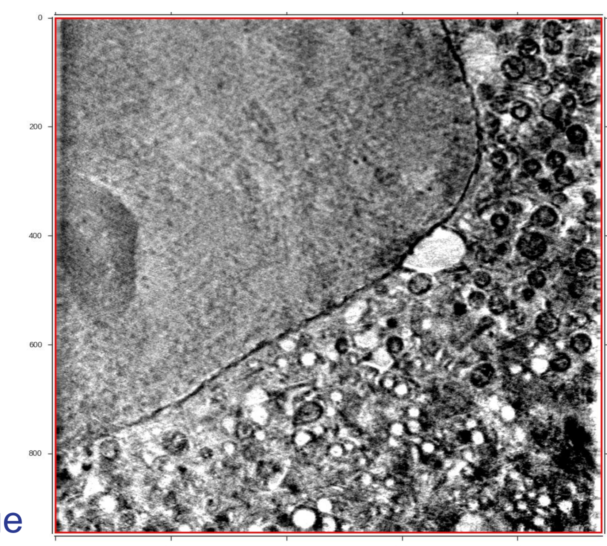
- Raw Features
- Denoising
- Local Features
- Gaussian Features
- **Blob-like Detection**
- Texture and Structure
- Robust Features



1D Laplacian of Gaussian:

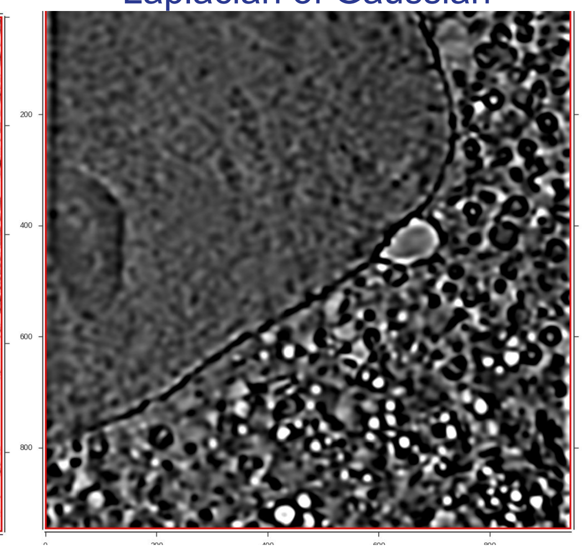
Intensity near the center is substracted to the surroundings: (*x = red - blue*).

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



Original Image

Laplacian of Gaussian

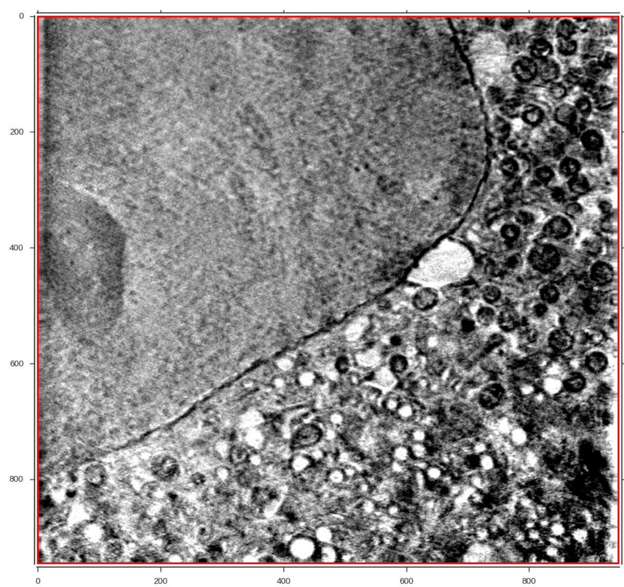


Data Preprocessing

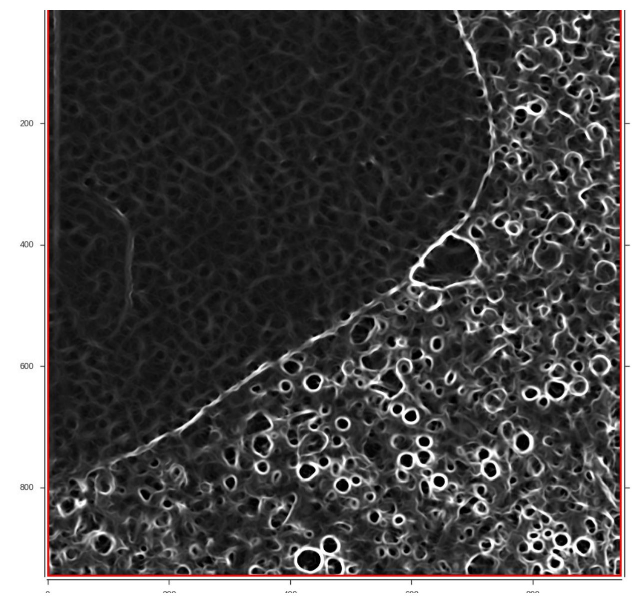
- Raw Features
- Denoising
- Local Features
- Gaussian Features
- Blob-like Detection
- **Texture and Structure**
- Robust Features

Projects the data to analyze its main axis of variance

- Hessian Eigenvalues: texture
- Structure Tensor Eigenvalues: structure



Original Image

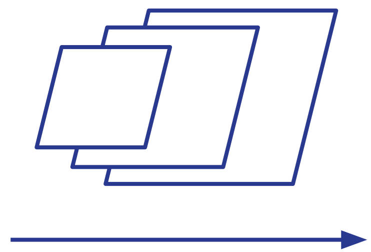


Largest Eigenvalue of the Hessian Matrix

Data Preprocessing

- Raw Features
- Denoising
- Local Features
- Gaussian Features
- Blob-like Detection
- Texture and Structure
- **Robust Features**

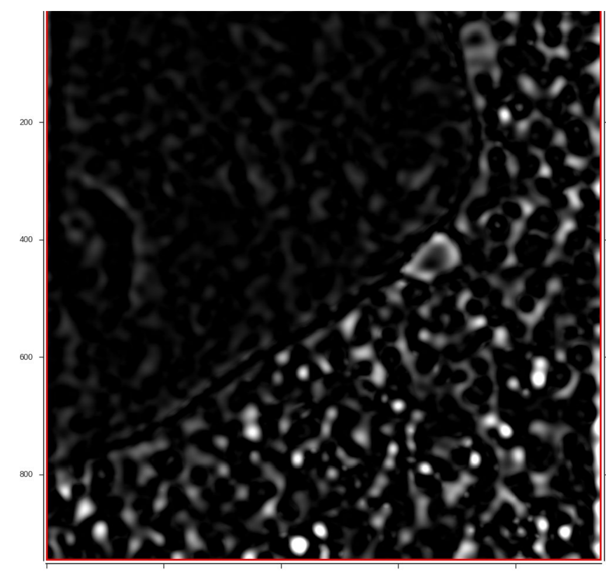
Apply any of the previous ones in a Multi-scale fashion



- Filter the scale with maximum value.

In other words, find objects of any size at once.

Original Image

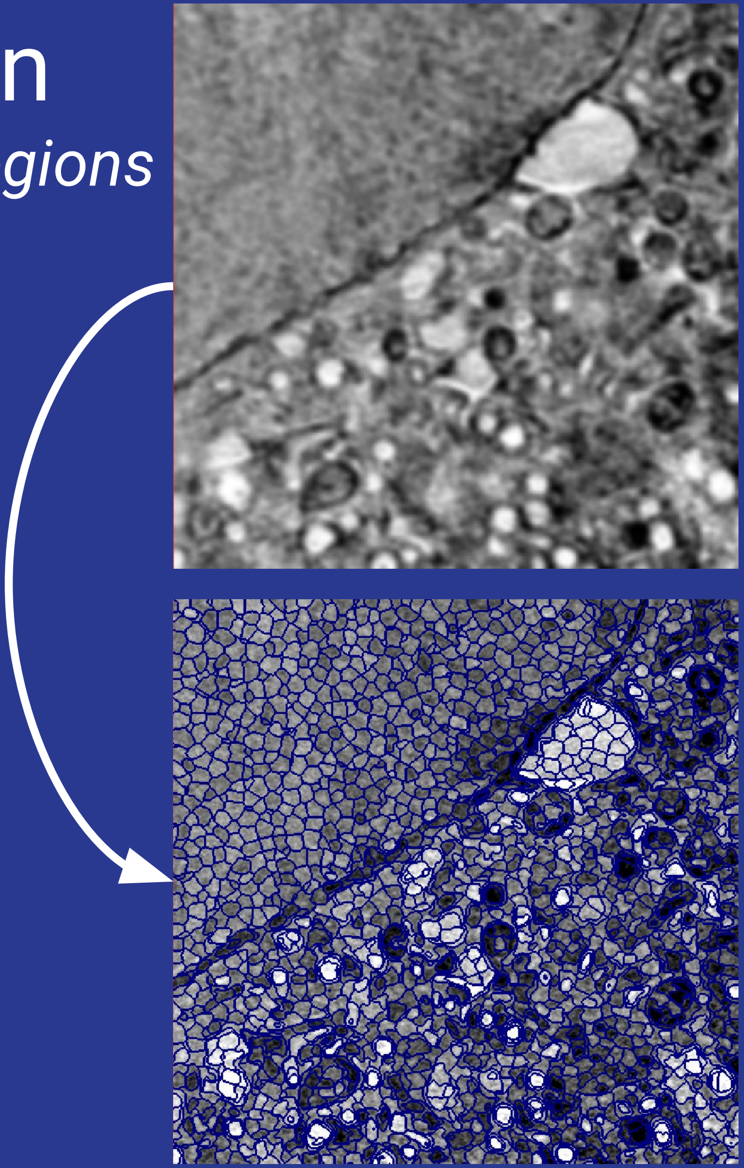


Multi-scale Laplacian of Gaussian

Data Representation

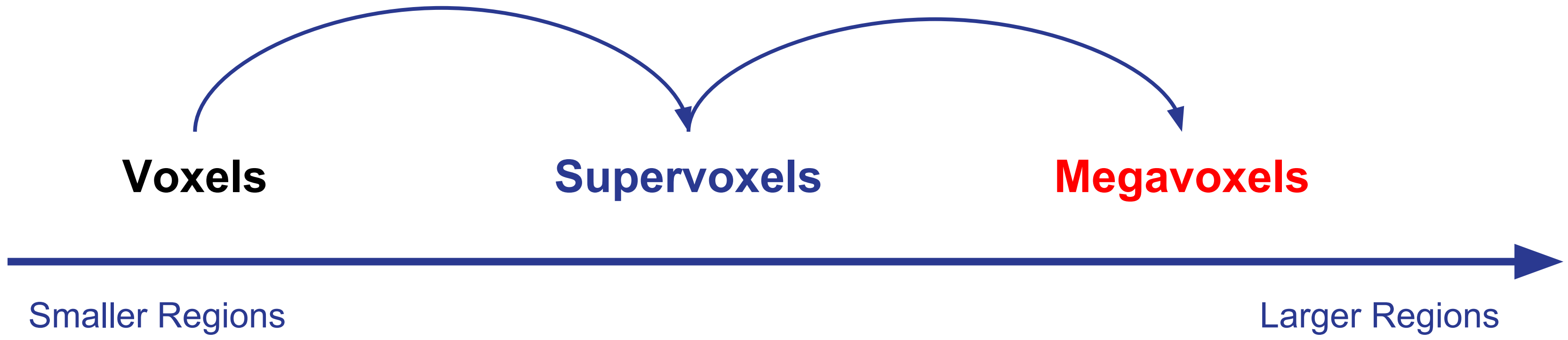
Represent data in coherent regions

- Voxels
- SuperVoxels
- MegaVoxels



Super-Regions

- Hierarchical Region representation of the volume
- **Voxels** are the smallest representative units
- **Supervoxels** are groups of similar and adjacent **voxels**
- **Megavoxels** are groups of similar and adjacent **Supervoxels**



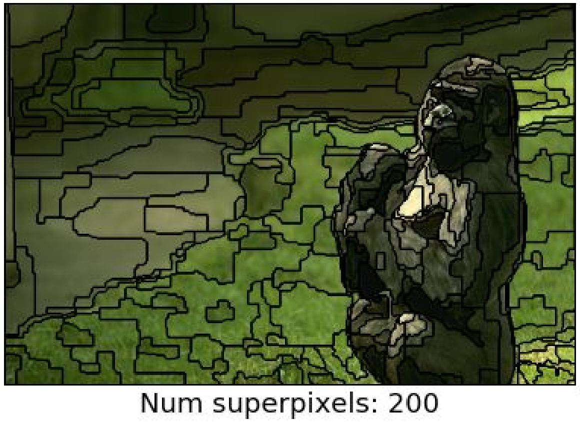
Data Representation

321x481 = 154401
Pixels

400
Superpixels

50
Megapixels

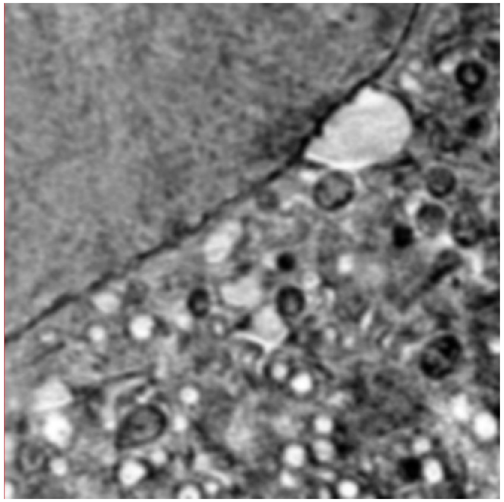
Data Representation

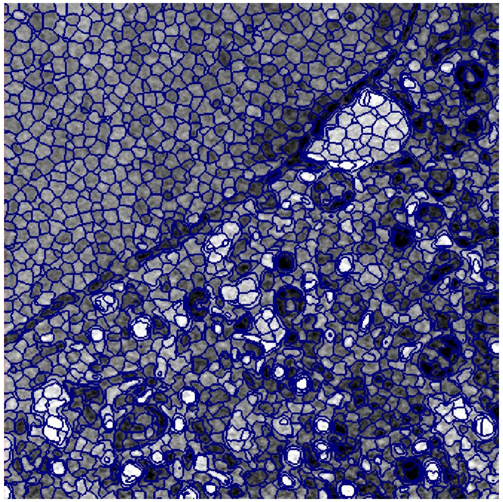


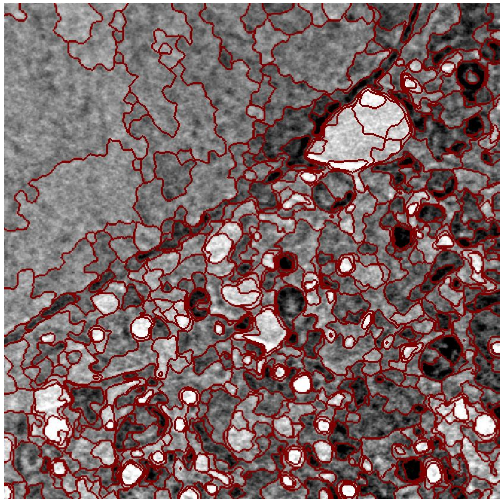
Superpixels extracted

Original Image → Assign to each pixel the mean color of all the pixels that belong to that superpixel → **Reconstructed Image**
Only 200 superpixels

Data Representation

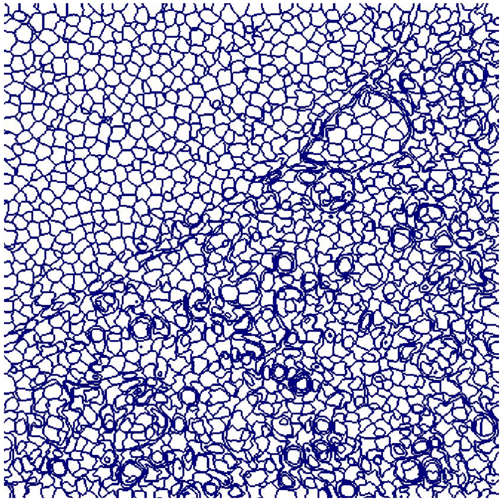


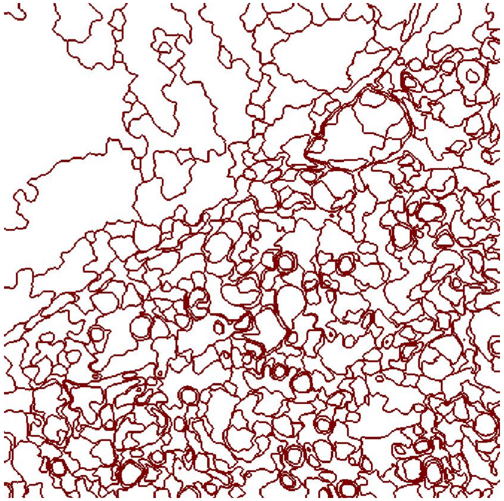




Supervoxels

Megavoxels





Data Representation

- In 3D, **Supervoxels** are groups of similar and adjacent voxels
- **Megavoxels** are groups of similar and adjacent **Supervoxels**
- **Supervoxels** and **Megavoxels** adhere to volume boundaries.
- Both are **completely unsupervised** (don't require human interaction)

Voxels	**Supervoxels**	**Megavoxels**
1000x1000x500 = 500M	10x10x10 => 500K	20-50K

Smaller Regions / More regions → Larger Regions / Less Regions

- Fast to compute and reduce further processing several orders of magnitude.
- By annotating **Supervoxels**, objects can be easily segmented without having to manually delineate boundaries.

Model Training

Learn from annotations. Propagate through volume.

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

Model Training

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

Data units that are going to be:

Used for learning

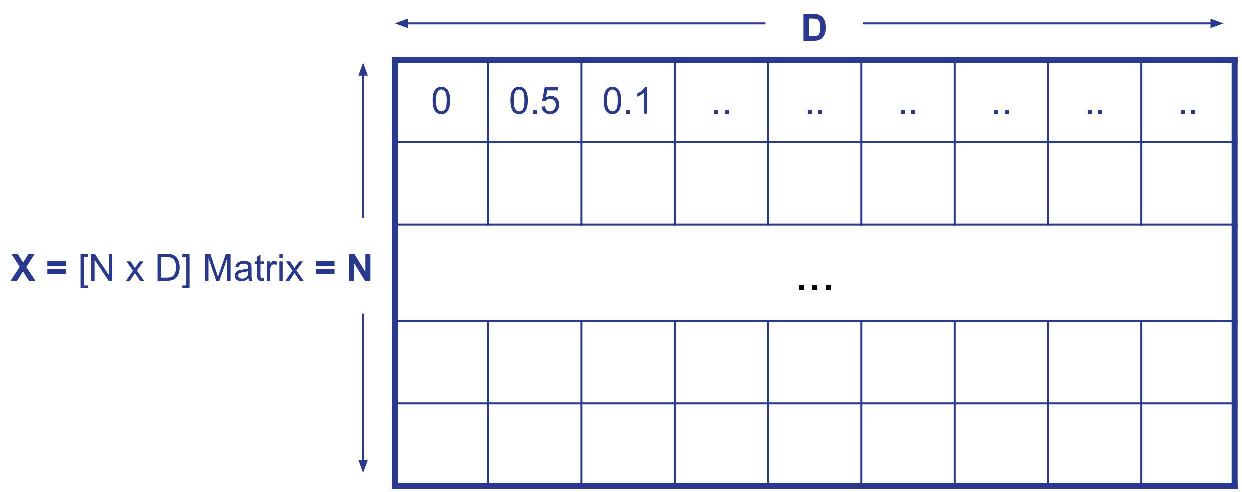
Predicted

- Voxels:
  - Learn to model voxels to predict voxels
- Supervoxels
  - Learn to model Supervoxels to predict Supervoxels
  - Several orders of magnitude faster.

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

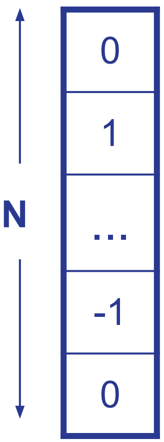
Model Training

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

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

X = [N x D] Data Matrix

Y = [N x 1] Matrix = **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	1.0
1.0	0.0
1.0	1.0

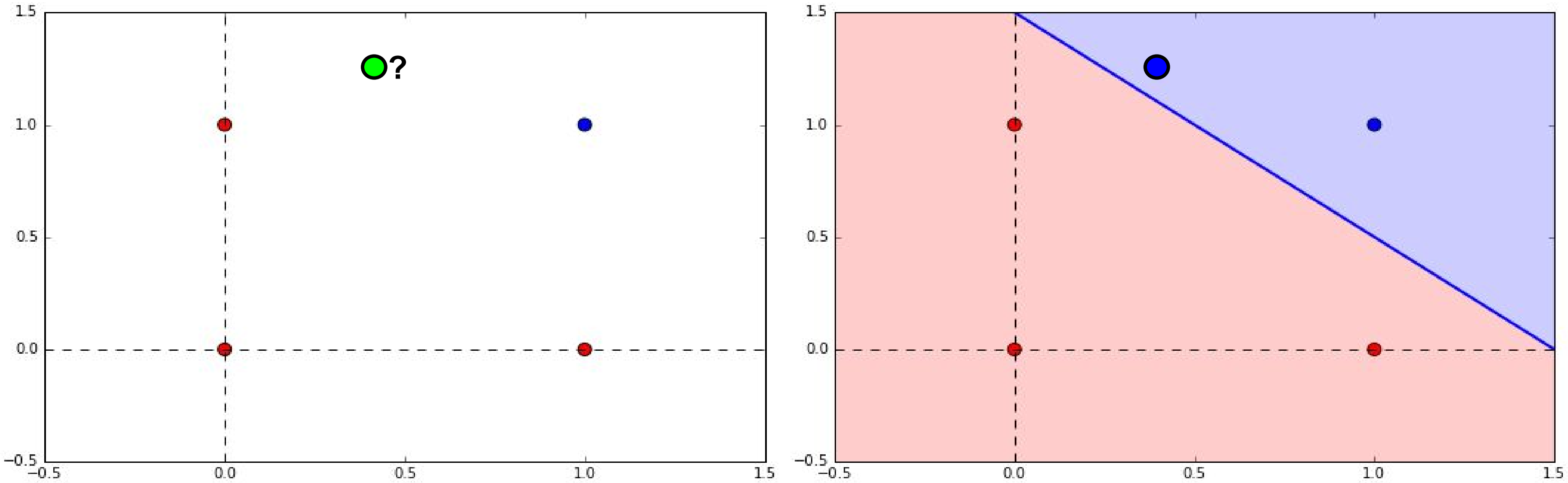
Y =

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

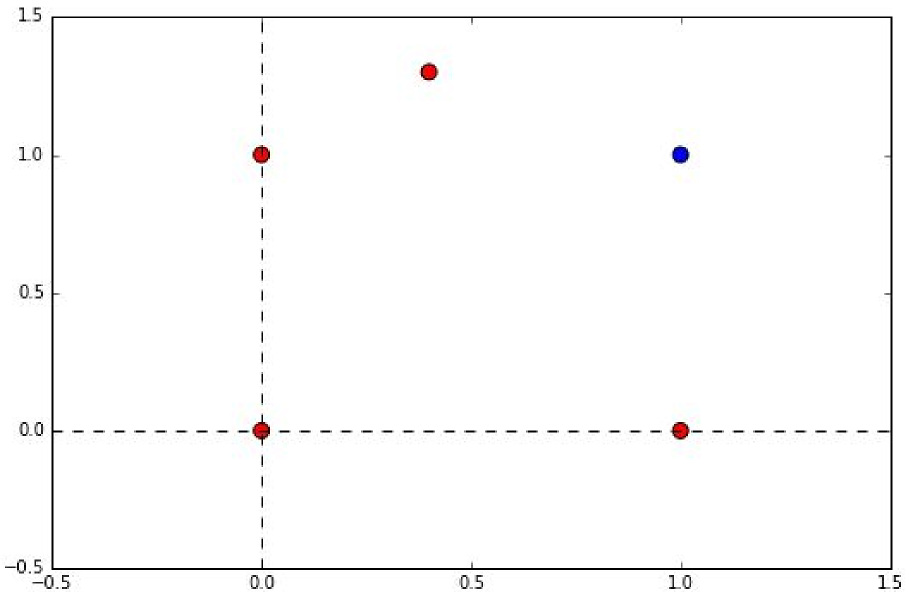
PROBLEM:

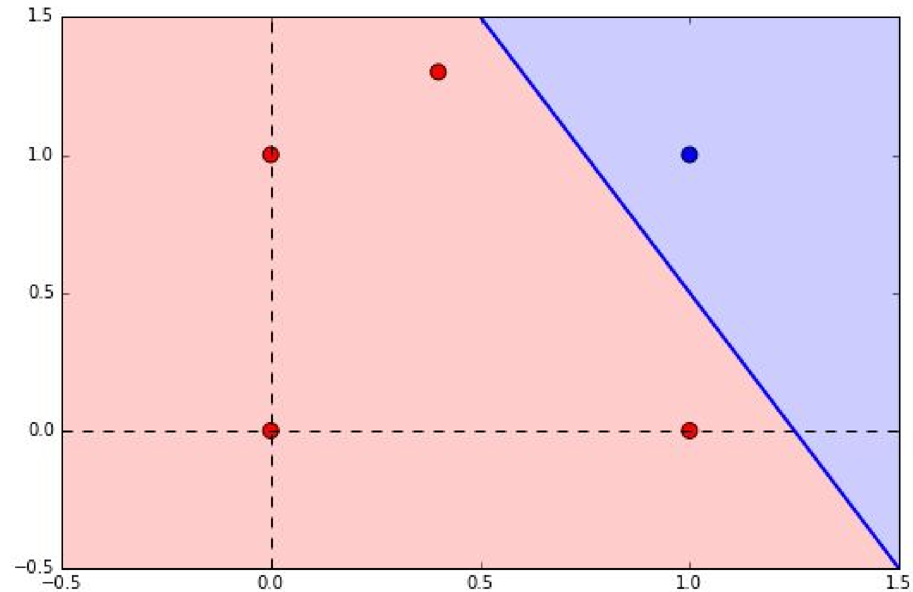
- On interactive Segmentation we only have limited data available.

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

Annotate + retrain

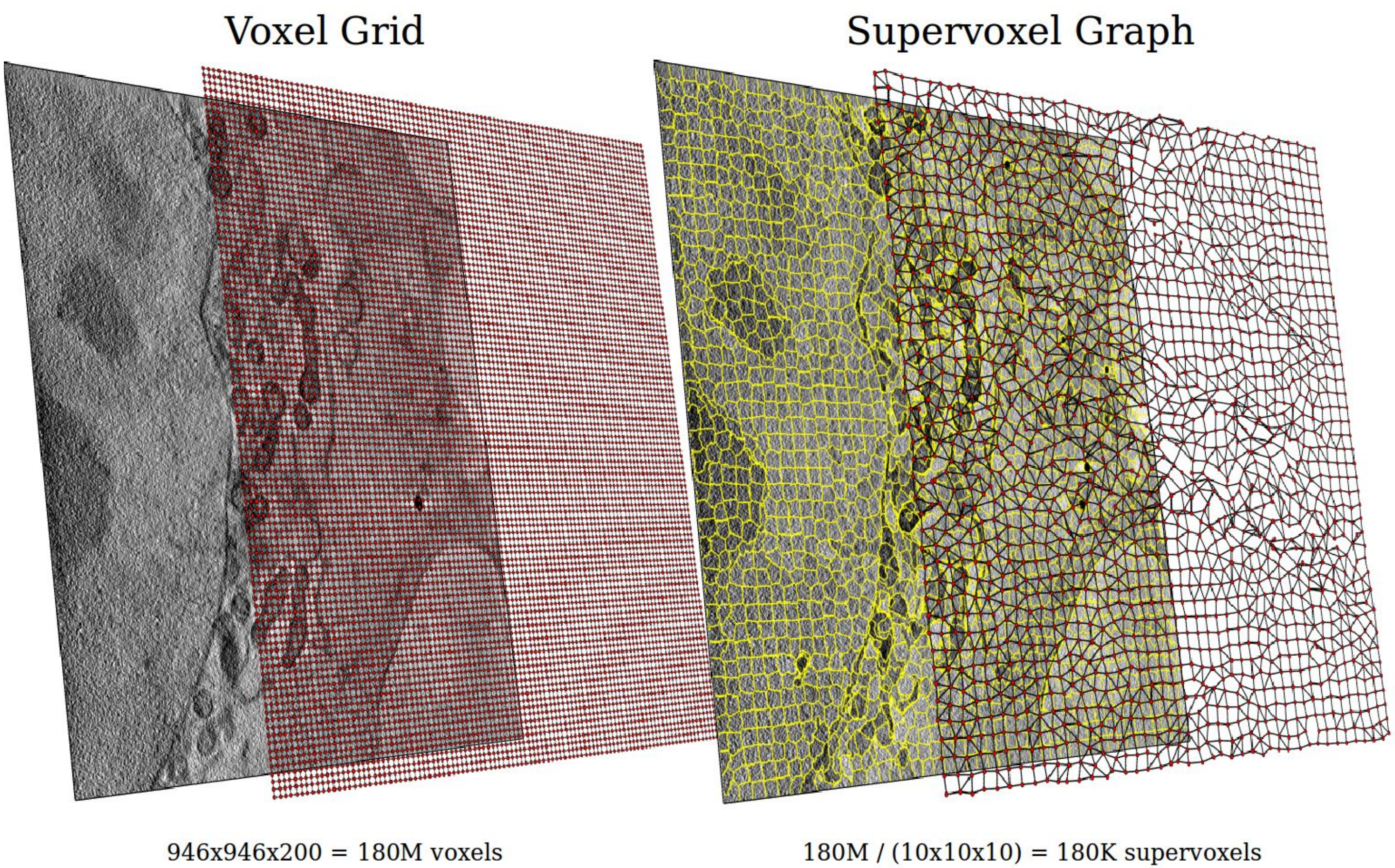
Refine Classification

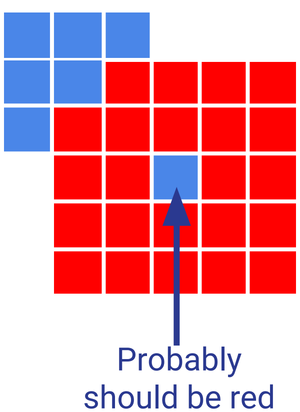




Model Training

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



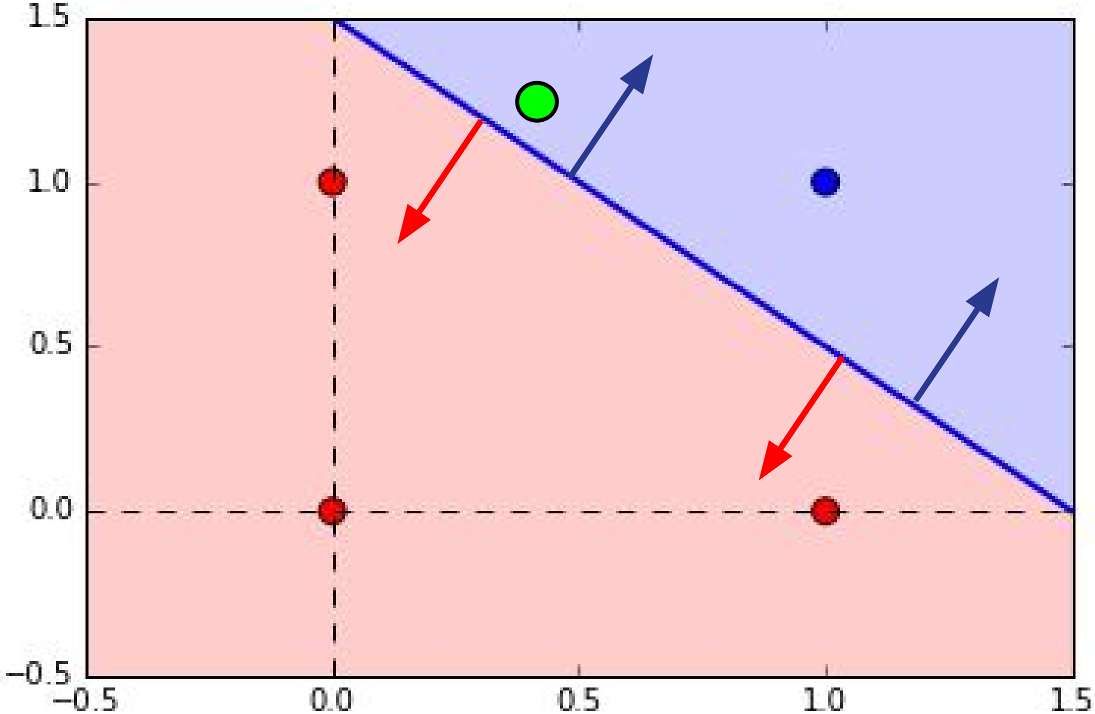


- Add Spatial Consistency to the Predictions
- Encourage nearby **voxels/supervoxels** to have belong to the same class.

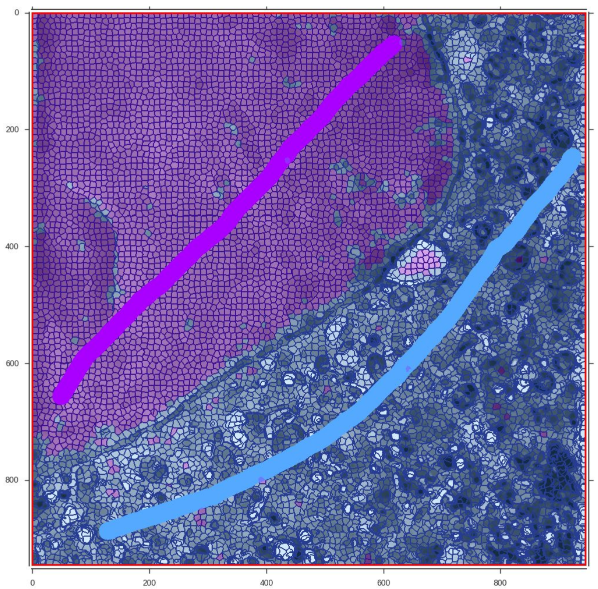
Model Training

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

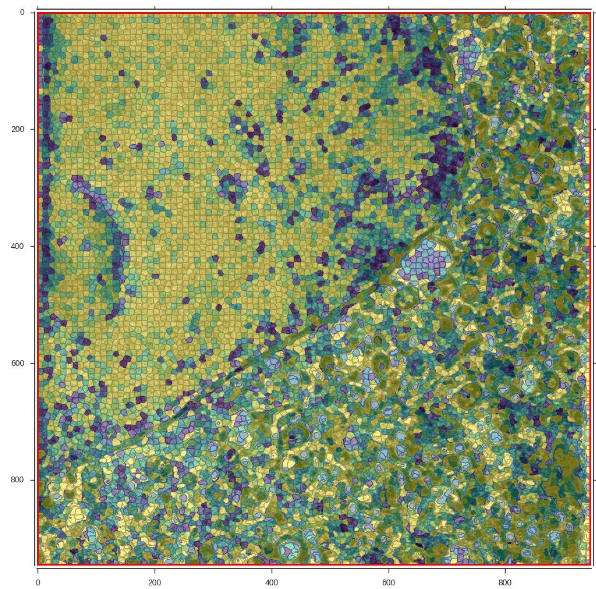
How confidence is the **Classifier** with the prediction it has made.



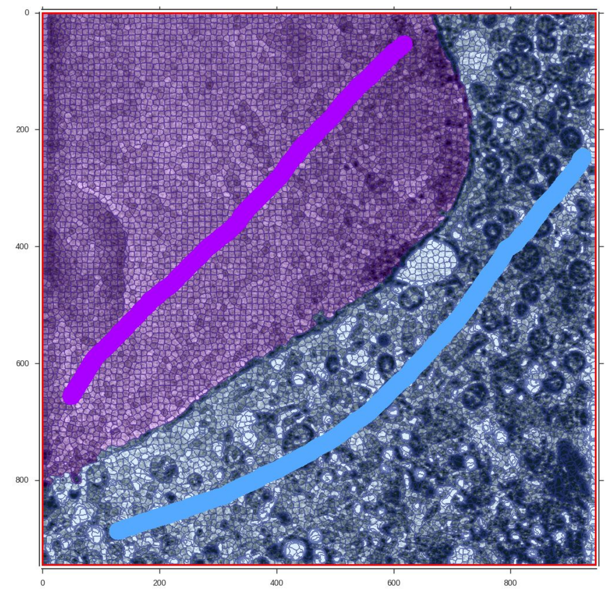
Model Training



Classification

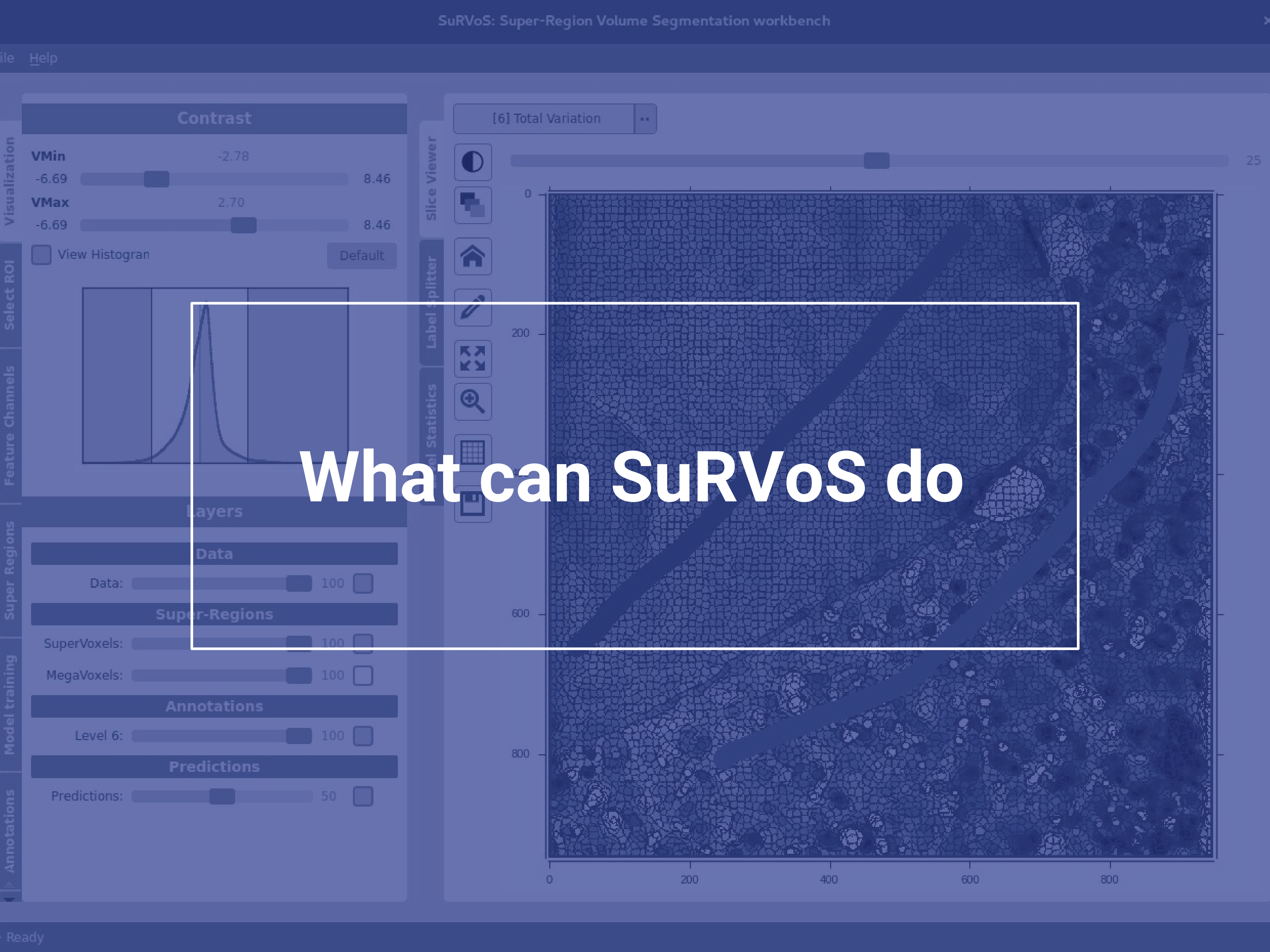


Confidence



Refinement

For large areas, a **single annotation in the central slide**, followed by training, classification and refinement is usually sufficient to accurately segment up to 100 slides of the volume.


What can SuRVoS do
SuRVoS: Super-Region Volume Segmentation workbench
Help
Contrast
VMin
VMax
View Histogram
Default
Layers
Data
Super-Regions
SuperVoxels:
MegaVoxels:
Annotations
Level 6:
Predictions
Predictions:
[6] Total Variation
Slice Viewer
Ready

SuRVoS Features

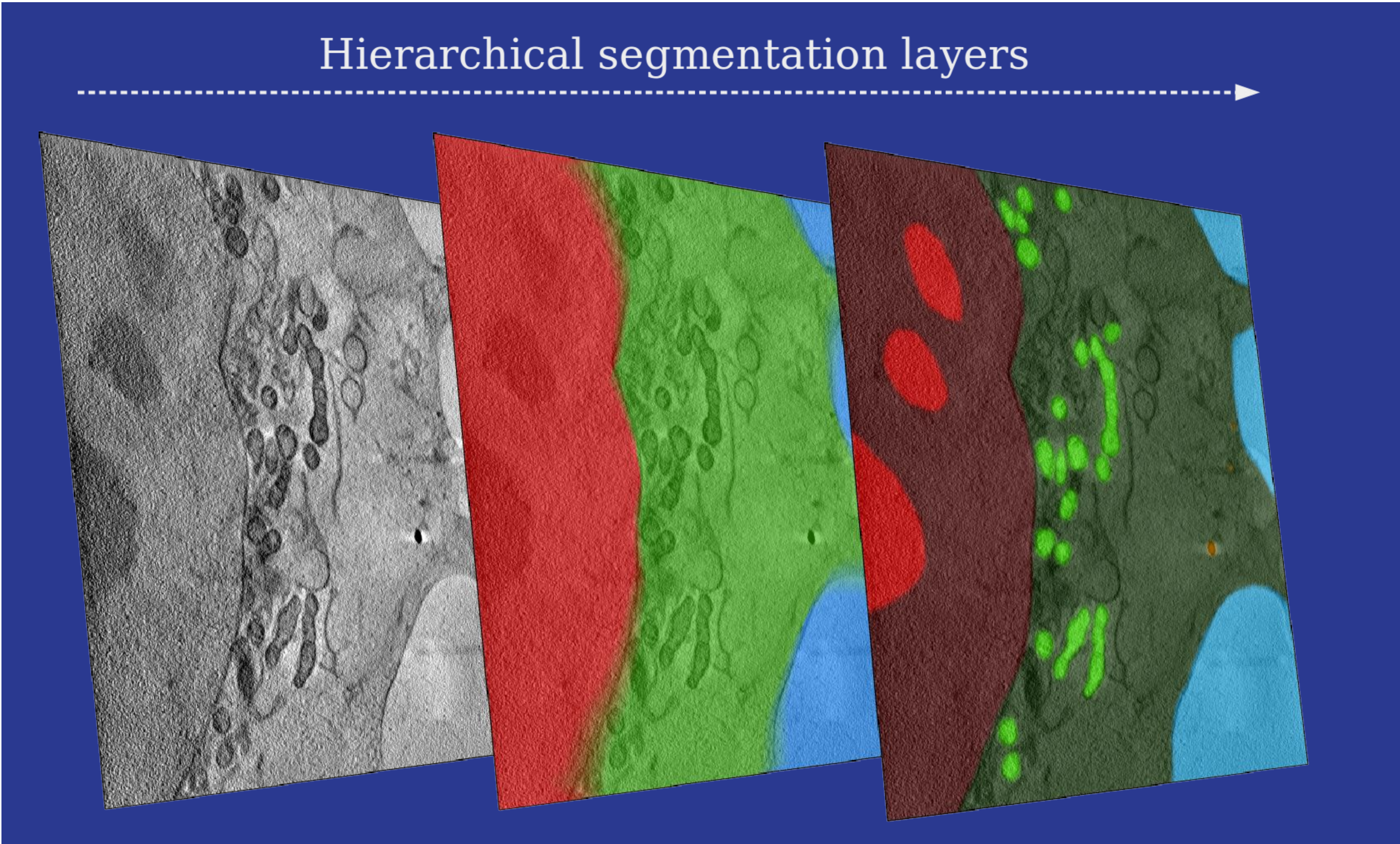
Features:

- Extract Super-Regions → Faster Annotations and Segmentation
- Compute Features → Data enhancement
- Learn Models → Automatically segment areas
- Identify individual objects → Extract measures and statistics between different object classes

Result:

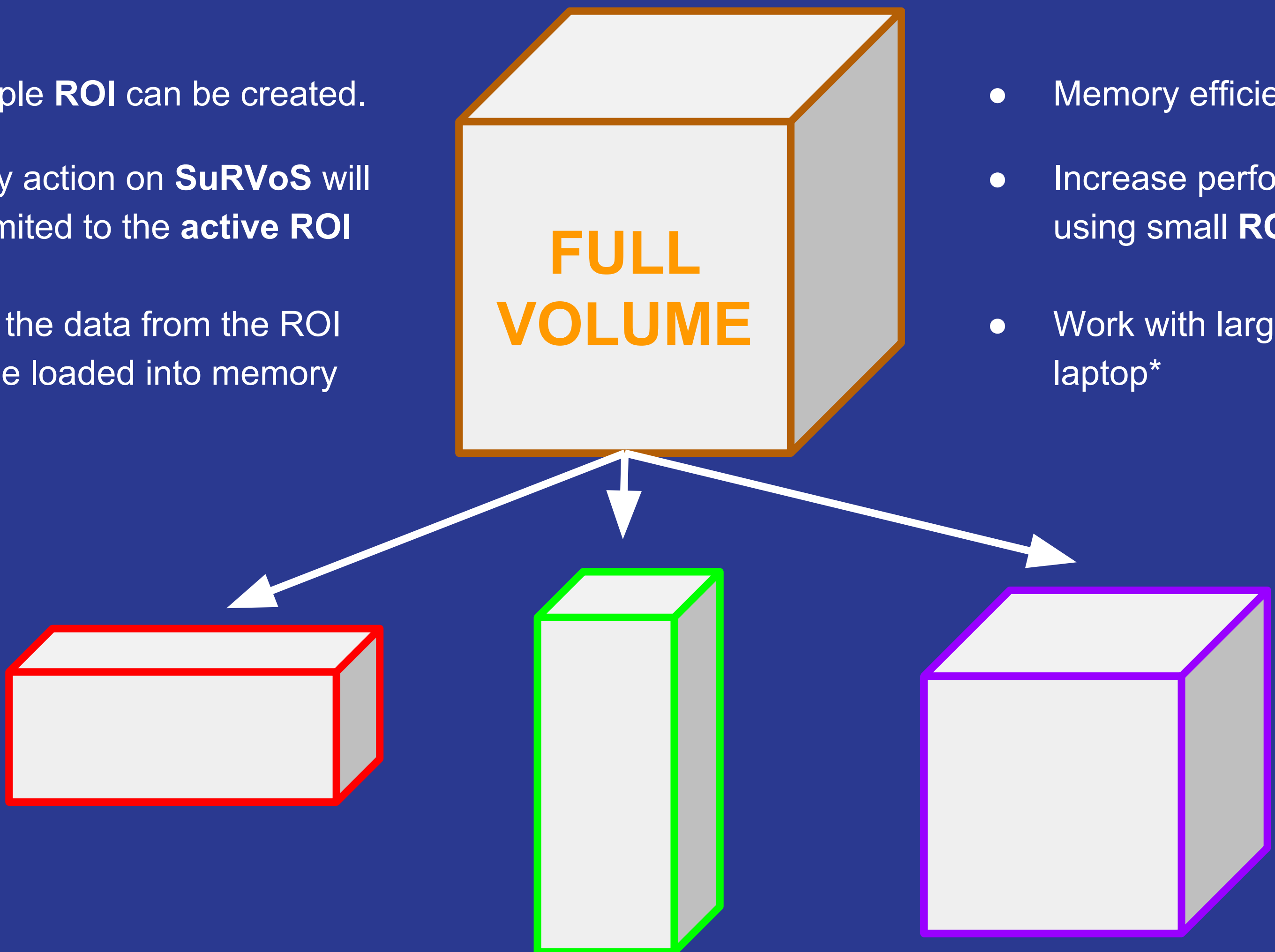
- Faster Annotations and Segmentation
- Data enhancement
- Automatically segment areas
- Extract measures and statistics between different object classes

Hierarchical Segmentation



Regions of Interest (ROI)

- Multiple **ROI** can be created.
- Every action on **SuRVoS** will be limited to the **active ROI**
- Only the data from the ROI will be loaded into memory



- Memory efficient.
- Increase performance when using small **ROI**s
- Work with large datasets in a laptop*

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.

Proyect_name/
- *data.h5*
- **annotations/**
  - *annotationsY.h5*
- **channels/**
  - *Y_channel_type.h5*
- **supervoxels/**
  - *supervoxel related .h5*
- **megavoxels/**
  - *megavoxel related .h5*

Conclusions

SuRVoS: Current State and Future Direction

Now:

- Assist user annotations with regions
- Segment large regions with models
- Identify individual objects

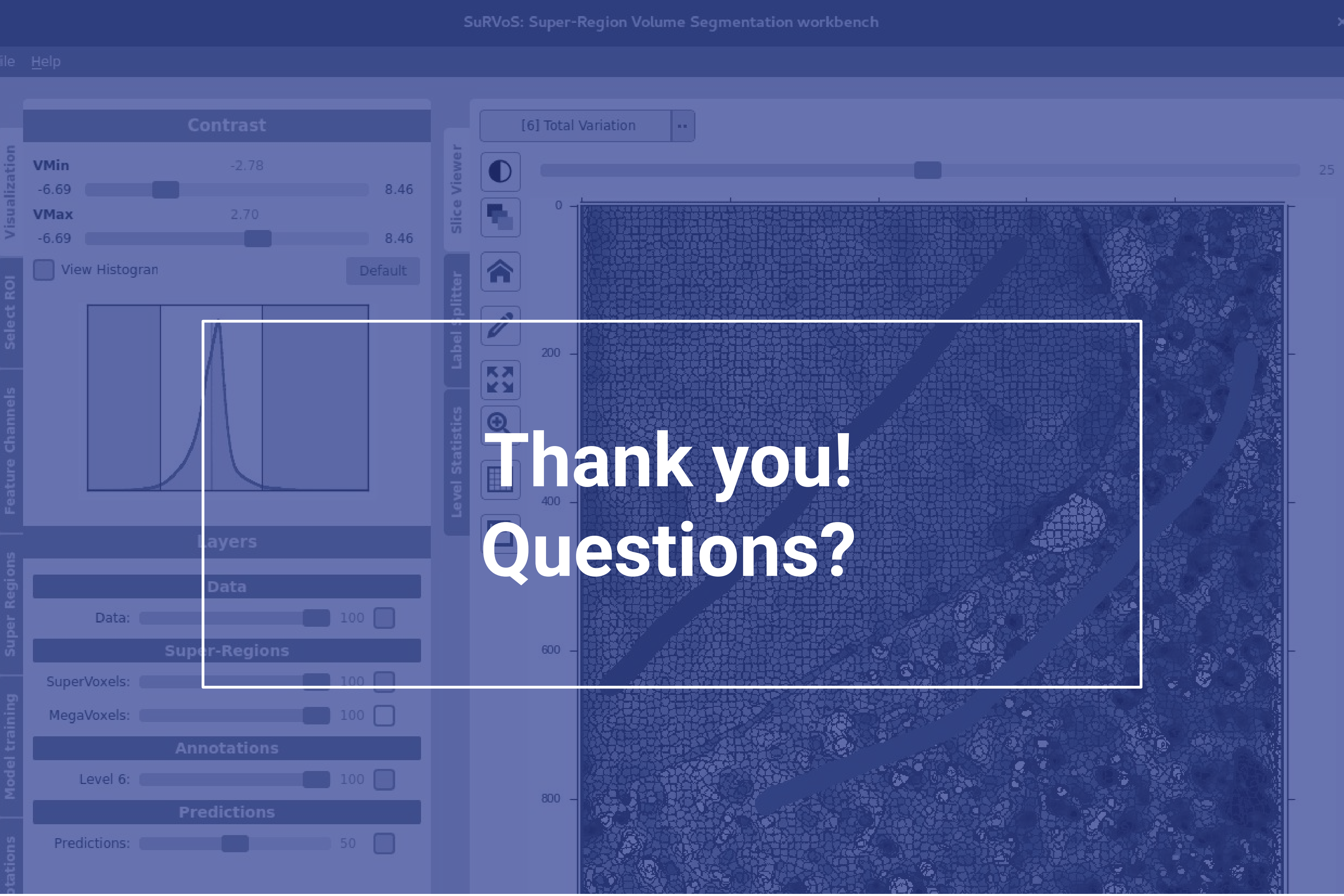
Maybe:

- Segment small organelles automatically (with enough annotations)

Future:

1. Better super-regions
   a. Multiple super-regions
2. Learn from ROI, apply to other ROI
3. Better Training Models
4. Better guidance to user using Patches
   a. Patch based Active Segmentation
5. Learn from one volume, apply to others.
6. Combine different Imaging Modalities
7. Data base of segmentations for fully automatic segmentation

Work in progress!



Thank you!
Questions?

https://DiamondLightSource.github.io/SuRVoS