ZEN Intellesis Trainable Segmentation

Dr. Sebastian Rhode

Product Owner, Technology Center Software & Digital Solutions, Munich

General Problem Statement

"I can get decent images from anywhere but I am looking for information"
- Potential customer during her visit to Pleasanton

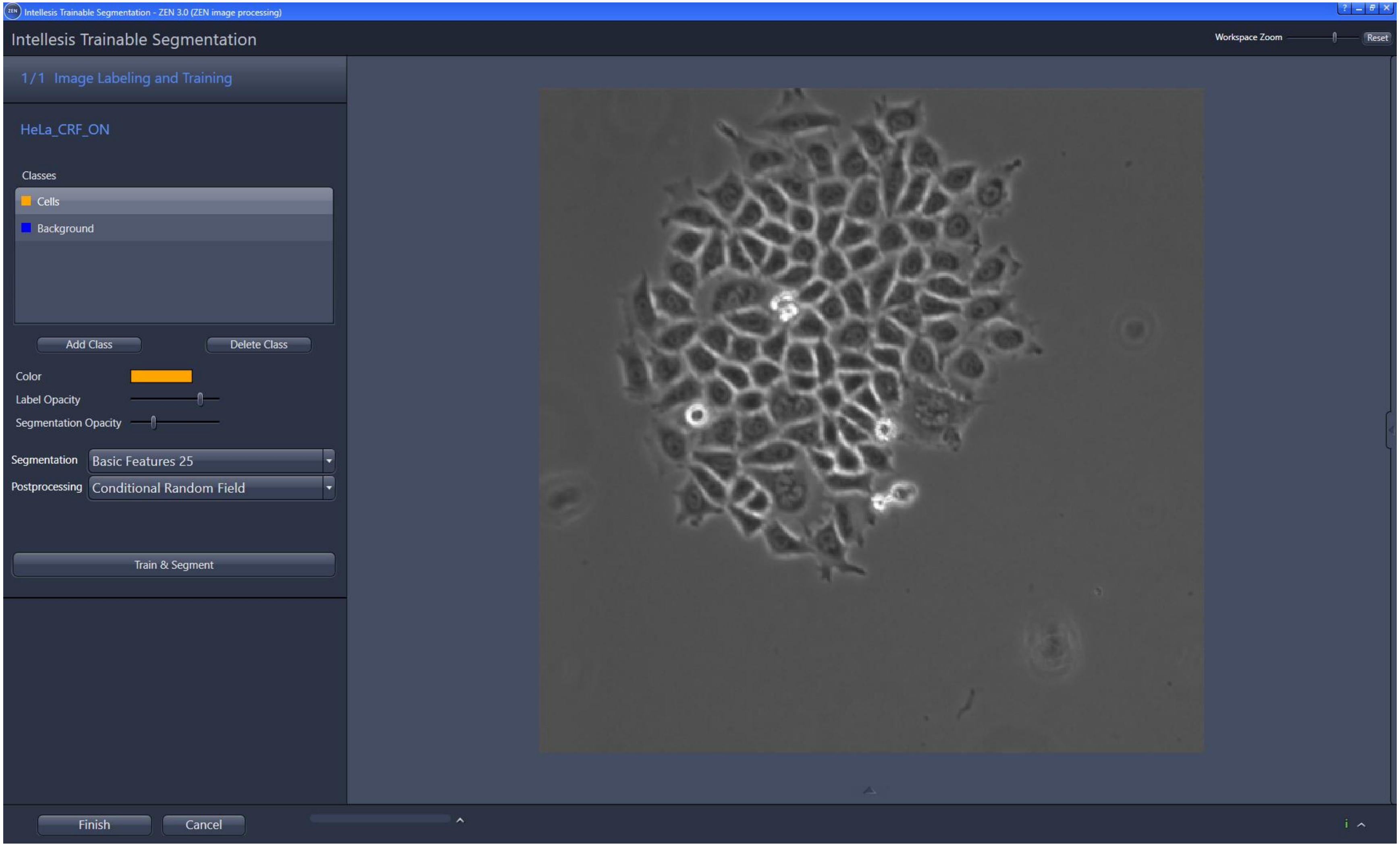
Segmentation is today's biggest problem.

A powerful and reproducible segmentation is the mandatory precursor for any downstream image analysis and measurements.

The actual purpose of an image is "to be analyzed"

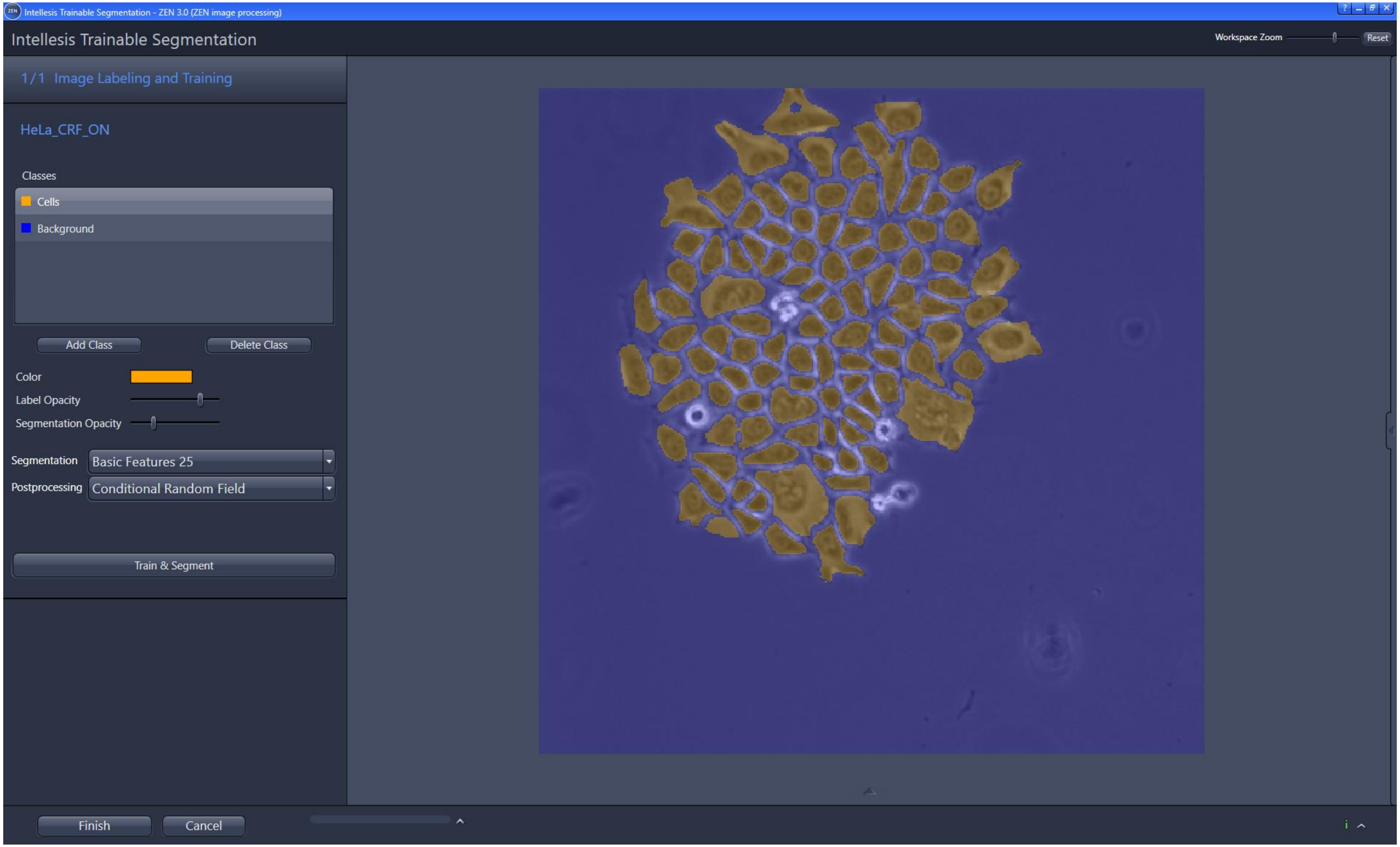
- Images can be "hard" or challenging to segment because of their nature.
- User is not an expert and needs a simple tool, even for "easy-to-segment" images from an algorithm point of view.

The core functionality of Intellesis - Segmentation



The core functionality of Intellesis - Segmentation

Machine learning



Key Points - ZEN Intellesis Trainable Segmentation

Machine learning

- **Simple User Interface for Labelling and Training**
  - The tool aims for the non-expert by providing an “easy-to-use” interface
  - Not all “parameters” in machine-learning (an expert might expect) can be adjusted. They are hidden “on purpose”

- **Integration into ZEN Measurement Framework**
  - As segmentation is only the required first step for subsequent measurements the **integration** into the actual measurement tools is key

- **Support for Multi-dimensional Datasets**
  - Intellesis, especially when considering the BioFormats option, can be used to segment any image even from non-Zeiss systems. 3D stacks, Tiles, Multi-Channel, …

Tech Notes - ZEN Intellesis Trainable Segmentation

Machine learning

- Real Multi-Channel Feature Extraction – All channels will be used to segment a pixel
- Class Segmentation – hierarchical structures with independent segmentation per class
- Engineered Feature Sets and Deep Feature Extraction (GPU) and pre-trained networks
- Pixel Classification by proven and established Random Forrest Classifier
- Option to download pre-trained DNNs for image segmentation
- Post processing by Conditional Random Fields
- Option to apply confidence thresholds
- IP-Functions for creating masks and scripting integration for advanced automation
- Client-Server Architecture (Zen Client ←→ Python-Server) with client-side tiling functionality to deal with large datasets

Fact Sheet - ZEN Intellesis Trainable Segmentation

Machine learning

- Machine-Learning Tool for Pixel Classification powered by Python
  - Dask, Scikit-Learn and Tensorflow
- Client / Server Architecture with REST-API
- Engineered Default Feature Sets (CPU)
  - 25 or 33 Features
- Neural Network (vgg19) Layers for Feature Extraction (GPU)
  - 64, 128 (red. 50) or 256 (red. 70) Features for 1st, 2nd or 3rd layer
- Classifier: Random Forrest Classifier for **Pixel Classification**
- Pre-trained DNNs for segmentation (optional download)
- Post Processing: Conditional Random Field (CRF)
- Universal automated build pipeline for ZeissPython established and integrated in official Zeiss installer

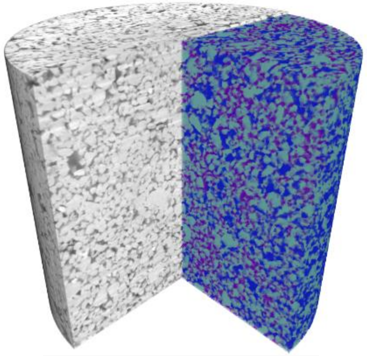
Intellesis – Things to know and understand

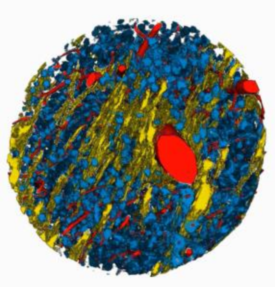
Machine learning

- Intellesis is **a SW module in ZEN Blue and ZEN Core** for image segmentation
- It task is to **segment images**, not measure parameters, but Intellesis can provide the segmentation results for downstream image analysis
- Image Segmentation is achieved by classifying **<u>every</u>** pixel of an image.
- Intellesis **extends of toolset of image segmentation algorithms** but its purpose is not to replace those.
- If an image can be easily segmented by a “classical” threshold etc. try this method first.
- By definition and the nature of the used machine-learning tools pixel classification is “computationally” very expensive and will take much, much longer than a simple threshold

Pixel Classification and Feature Extraction

Pixel based image segmentation



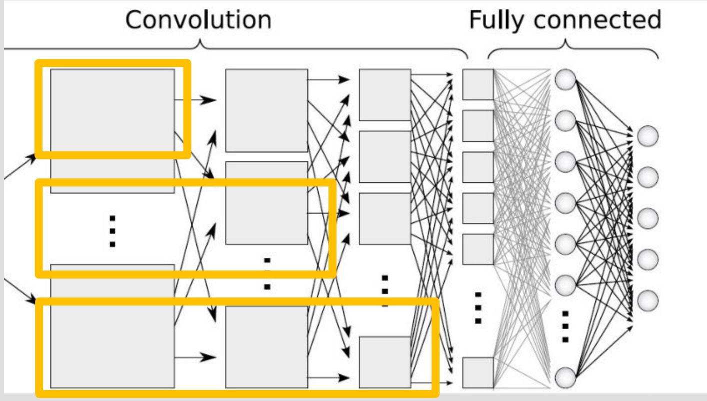


Examples:
- Segmentation of oil, brine and rock in a reservoir rock
- Segmentation of blood vessels, cells and axons in brain XRM scan

Extract Layers to use features

Deep Neural Networks

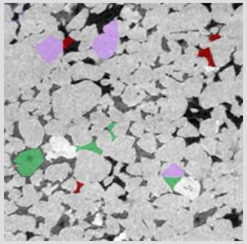
(More data beats clever algorithms)



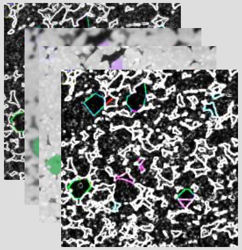
Use predefined features

Feature Engineering plus classifier

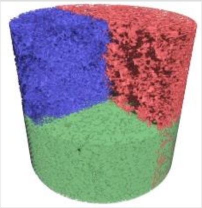
(Domain knowledge beats more data)



Manual selection of known regions (Labeled Image)



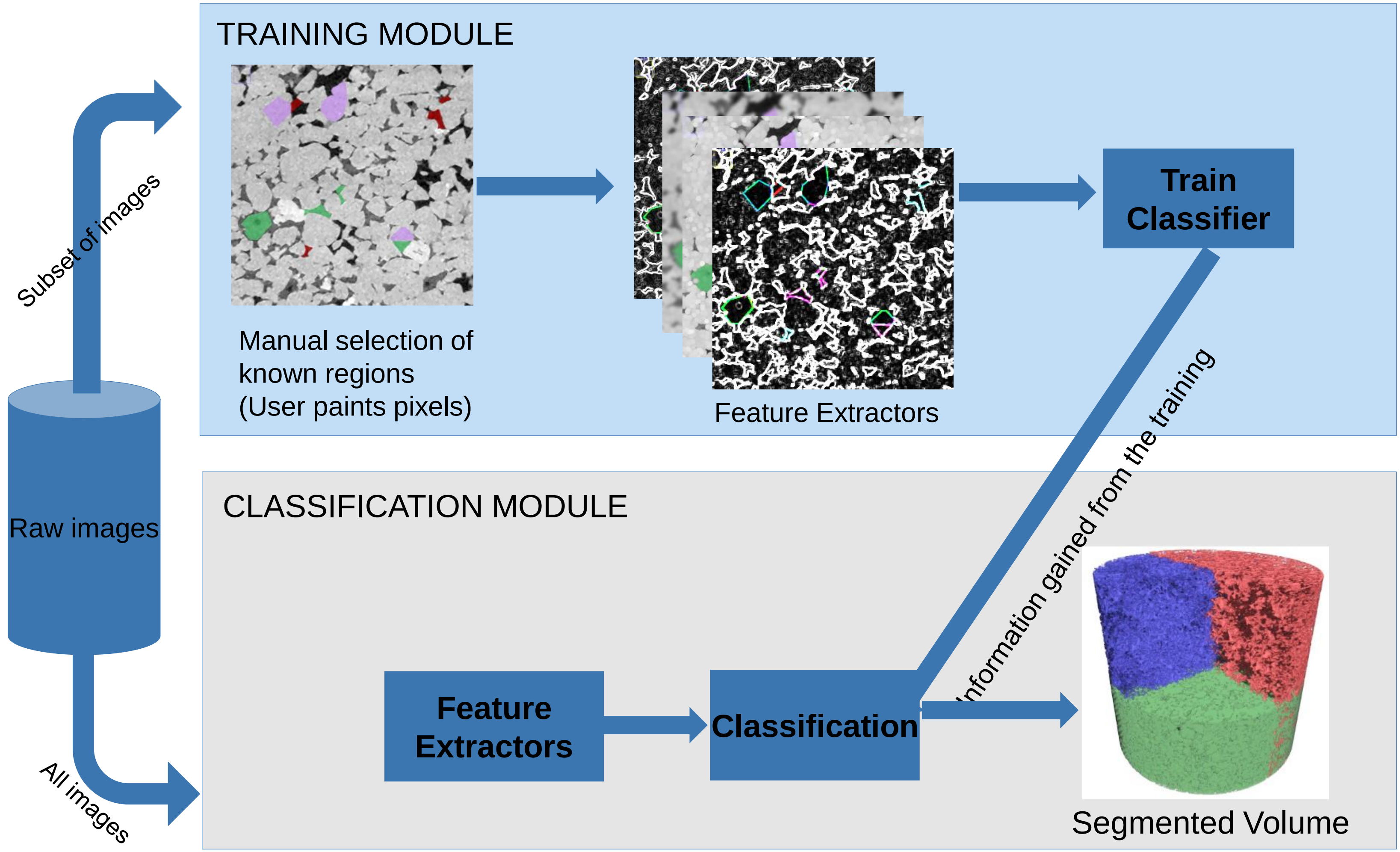
Generation of feature vectors



Segmented Volume

Segmentation workflow

Machine learning

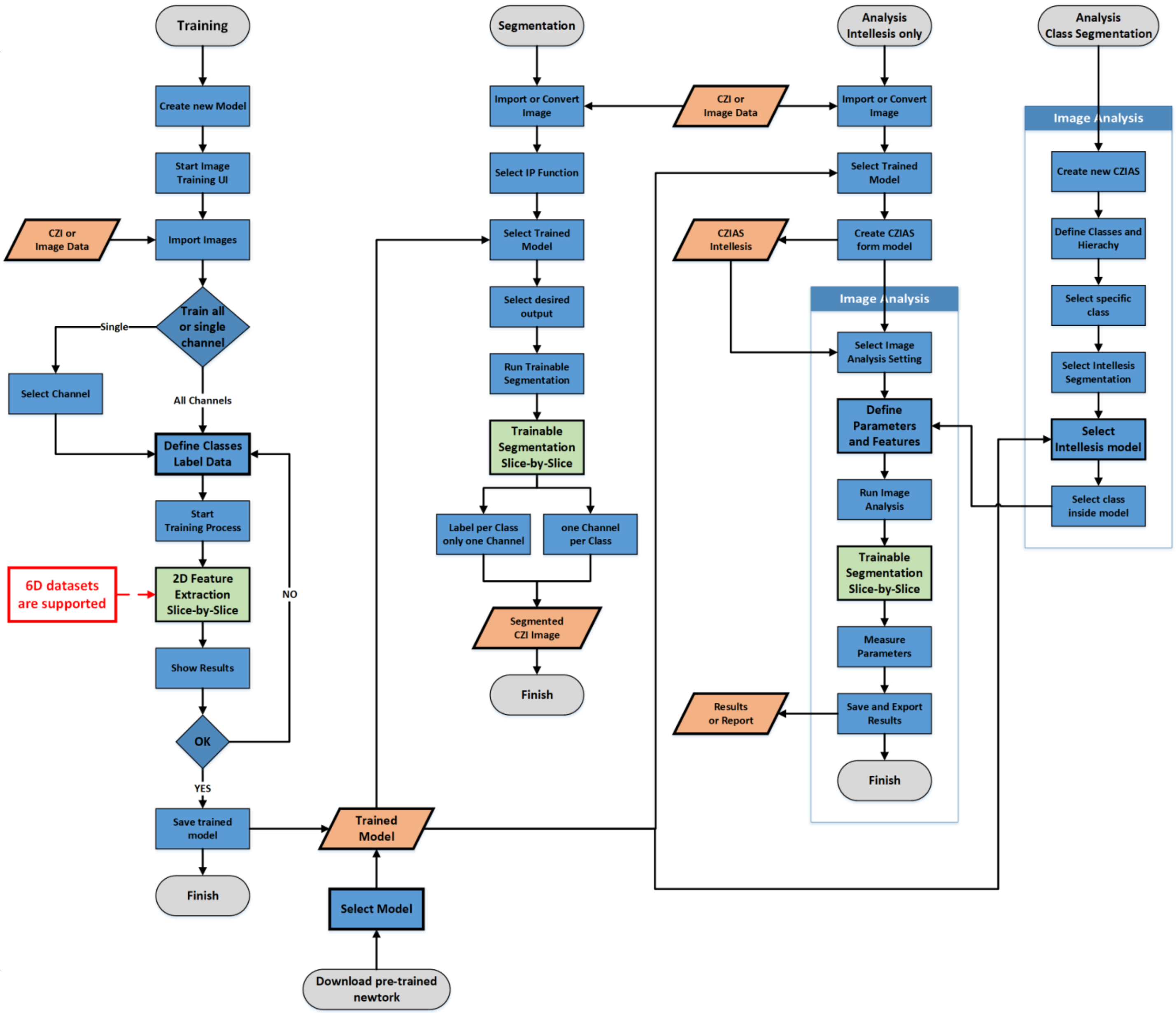


Labeling and Training – Things to know and understand

Machine learning

- A label is valid for all channels of an image and the information from all channels will be used to classify that particular pixel (X,Y)
- Label as few pixels as possible and rather do more iterations of training.
- Sometimes it is helpful to more classes in order to get better results for each class.
- Labeling large homogenous areas with many pixels will slow down the training but give not necessarily giver better results.
- It is good practice to labels “roughly” the same number of pixels for every class, e.g. try to avoid labeling much, much more pixels for one class compared to another class
- The default feature extractor are using the fewest number of parameters (start here first) while the Deep Feature extractors use the largest number. Use those only when the Default one do not work well.

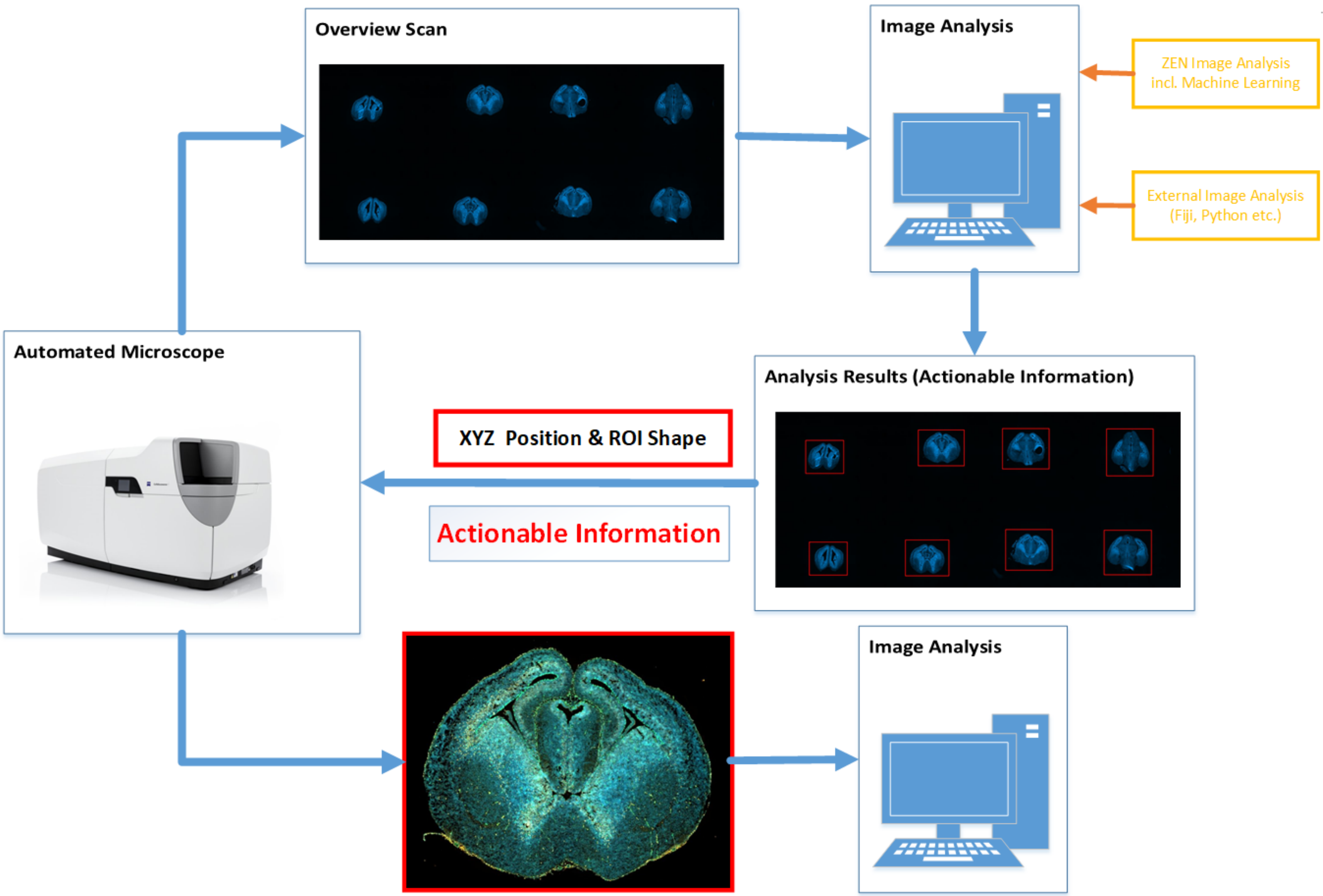
ZEN Intellesis – WorkFlows and Measurement Integration



Machine learning

Actionable Information – Where can trained models be useful

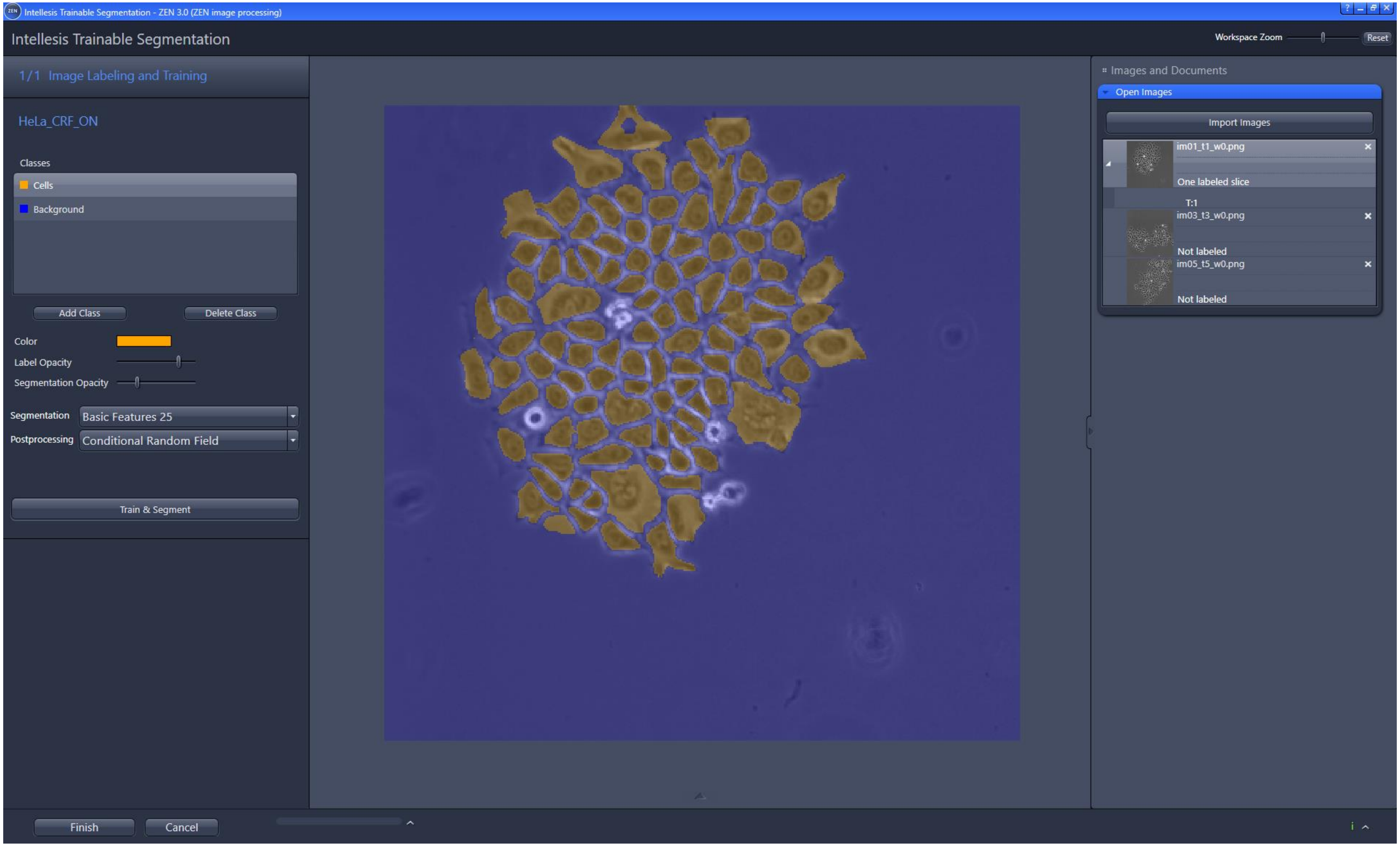
Machine learning



UX Design – Training in ZEN Blue

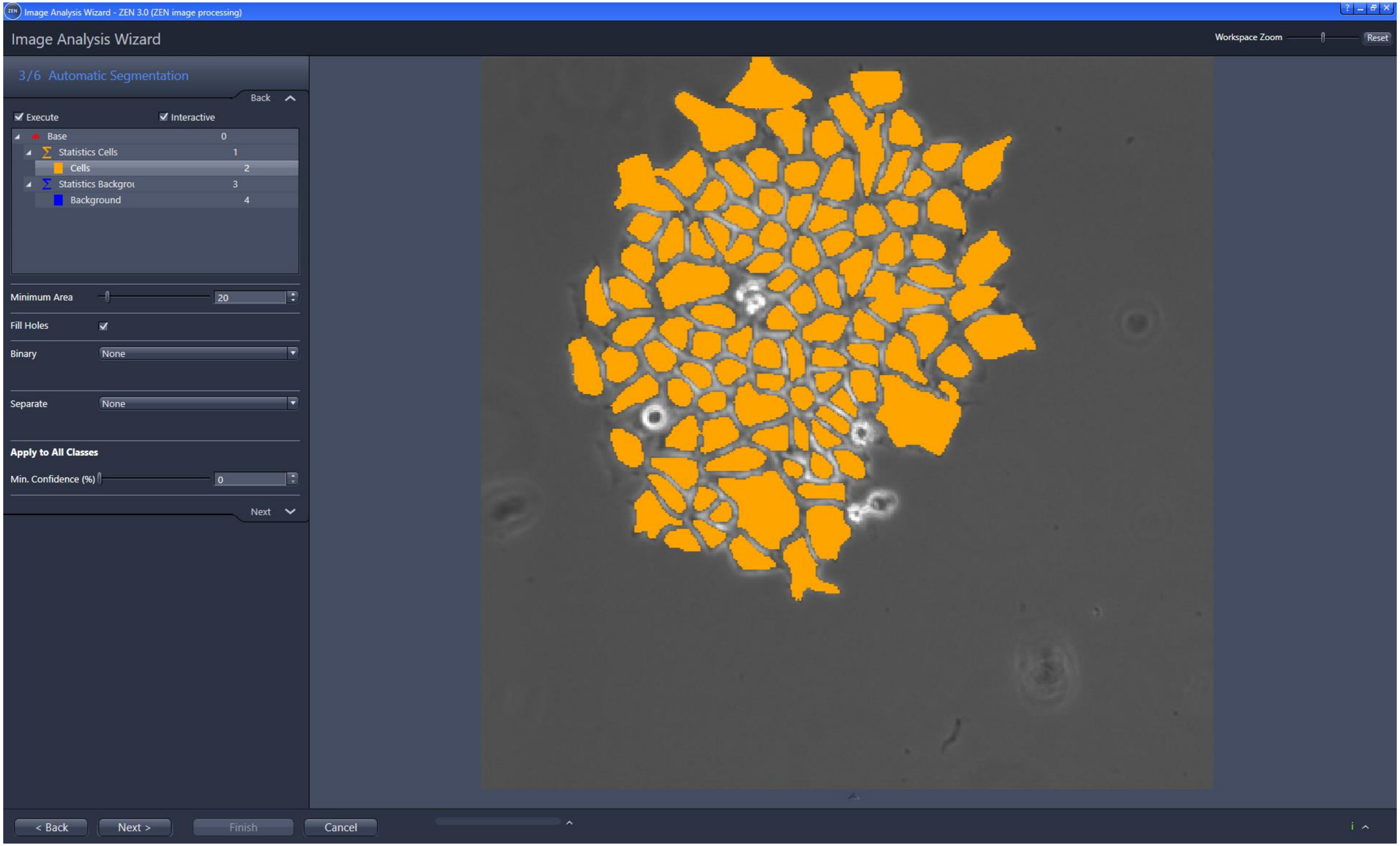
Machine learning

ZEISS



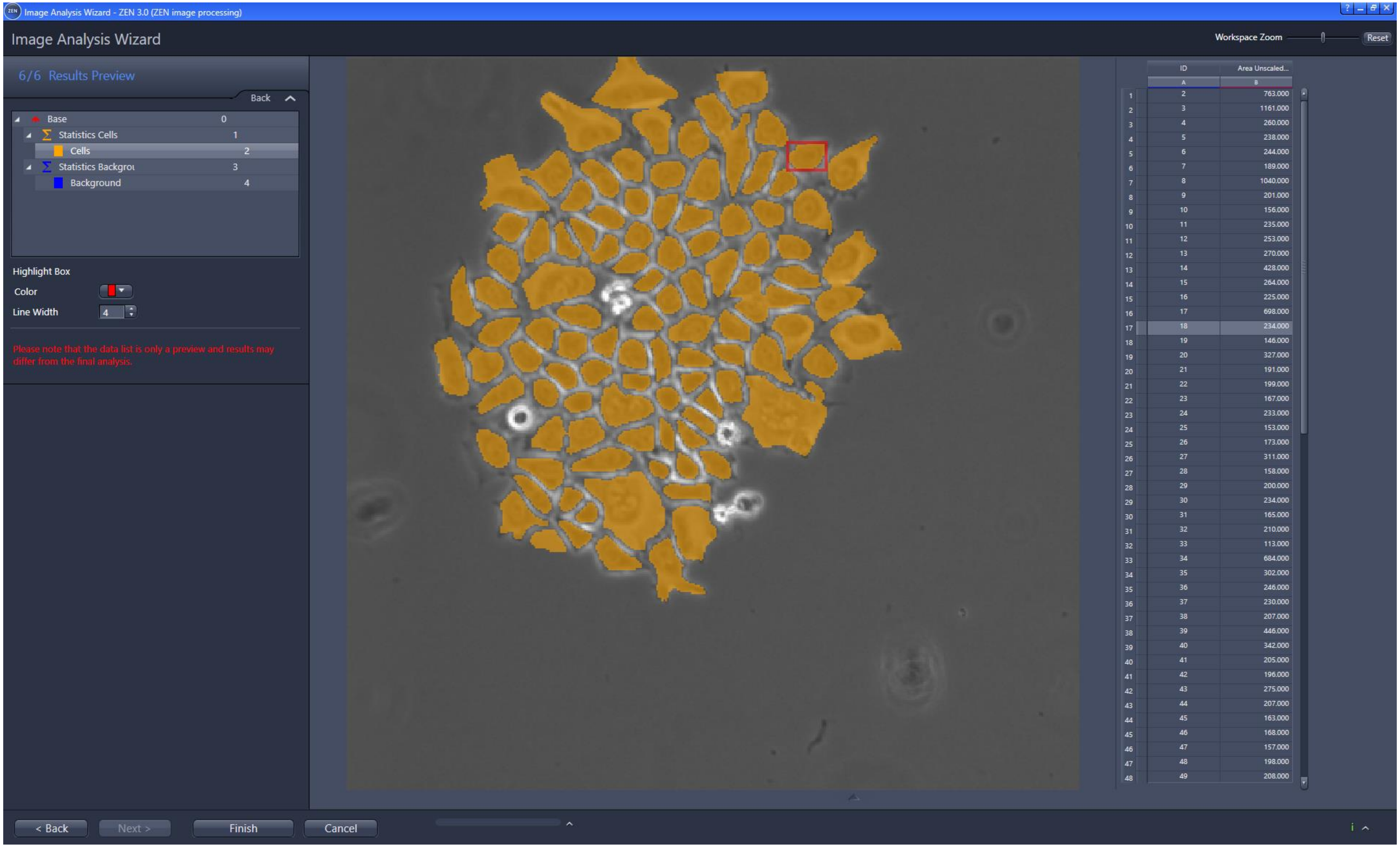
Integration into Image Analysis Pipelines in ZEN Blue

Machine learning



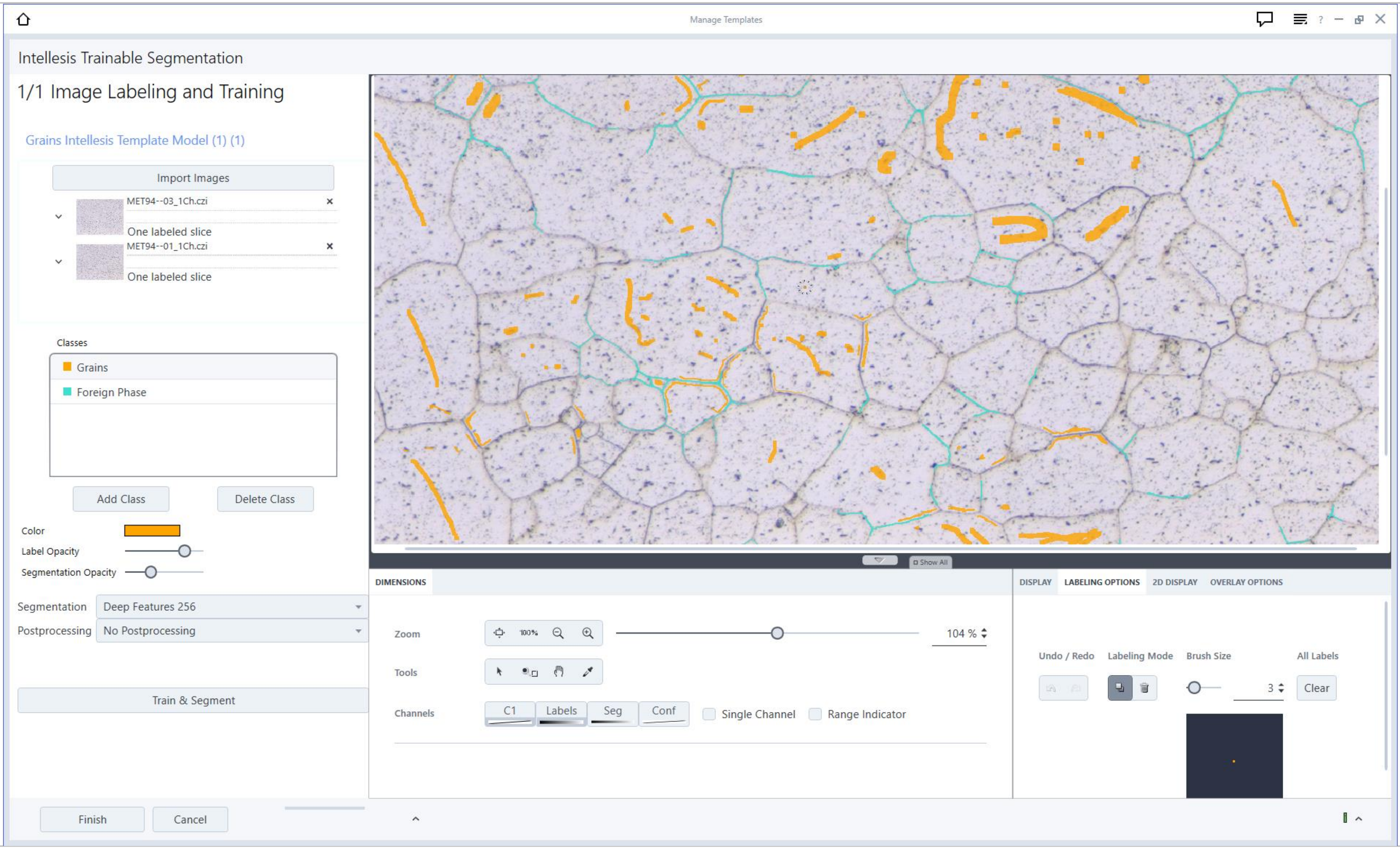
Integration into Image Analysis Pipelines in ZEN Blue

Machine learning



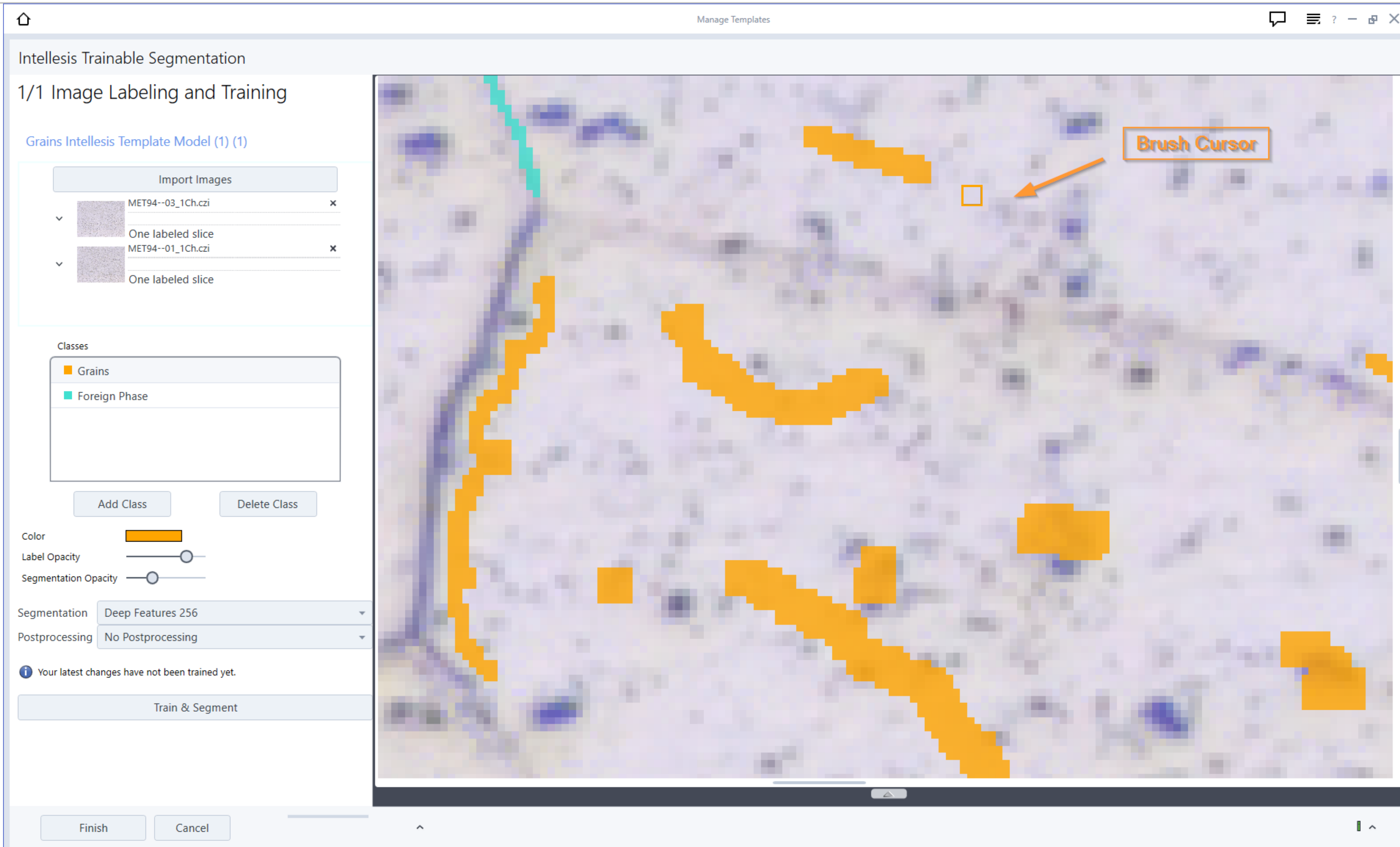
New UX Design – Intellesis Training in ZEN Core

Machine learning



Intellesis – Brush Cursor

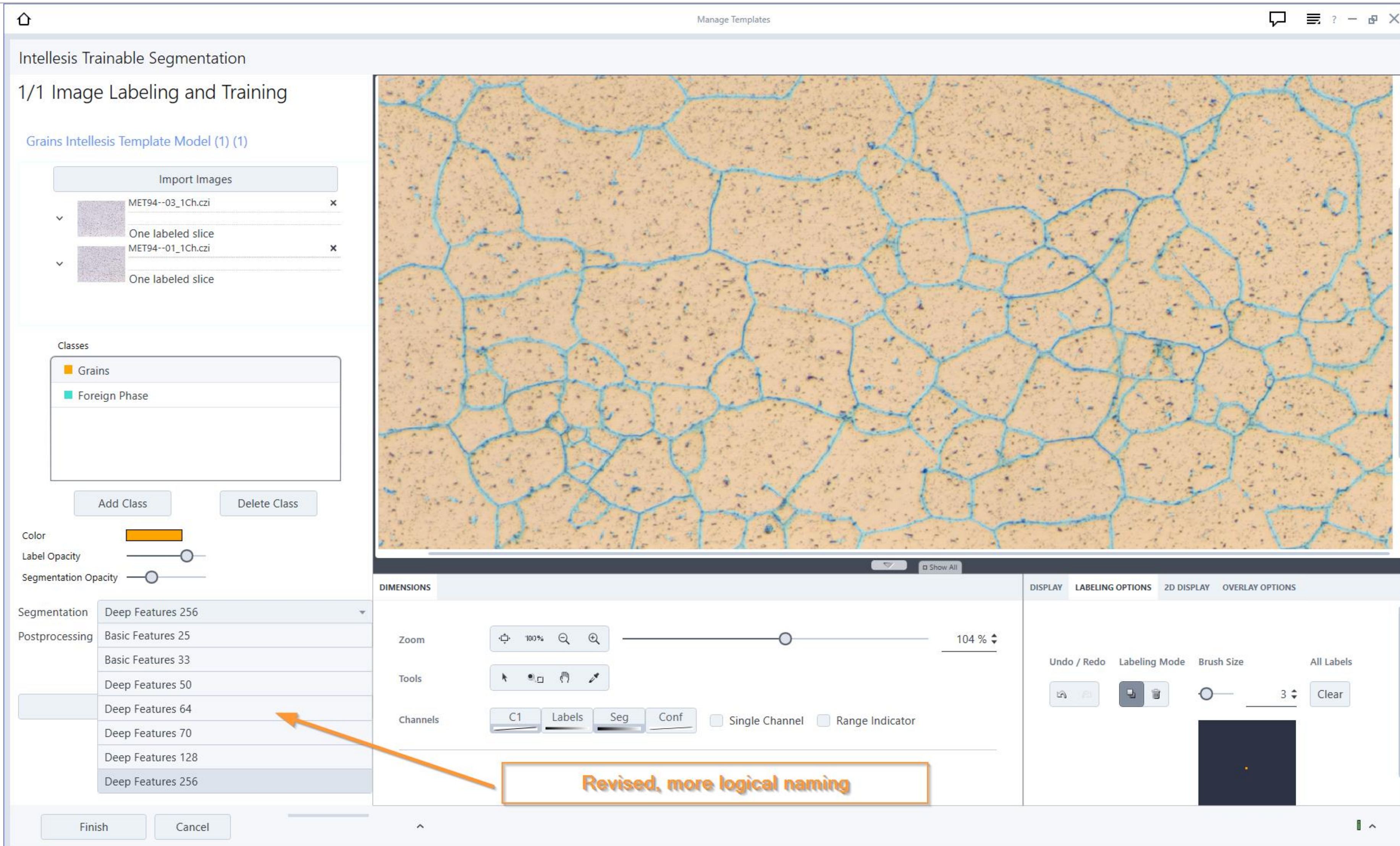
Machine learning



Intellesis – Revised Naming Scheme

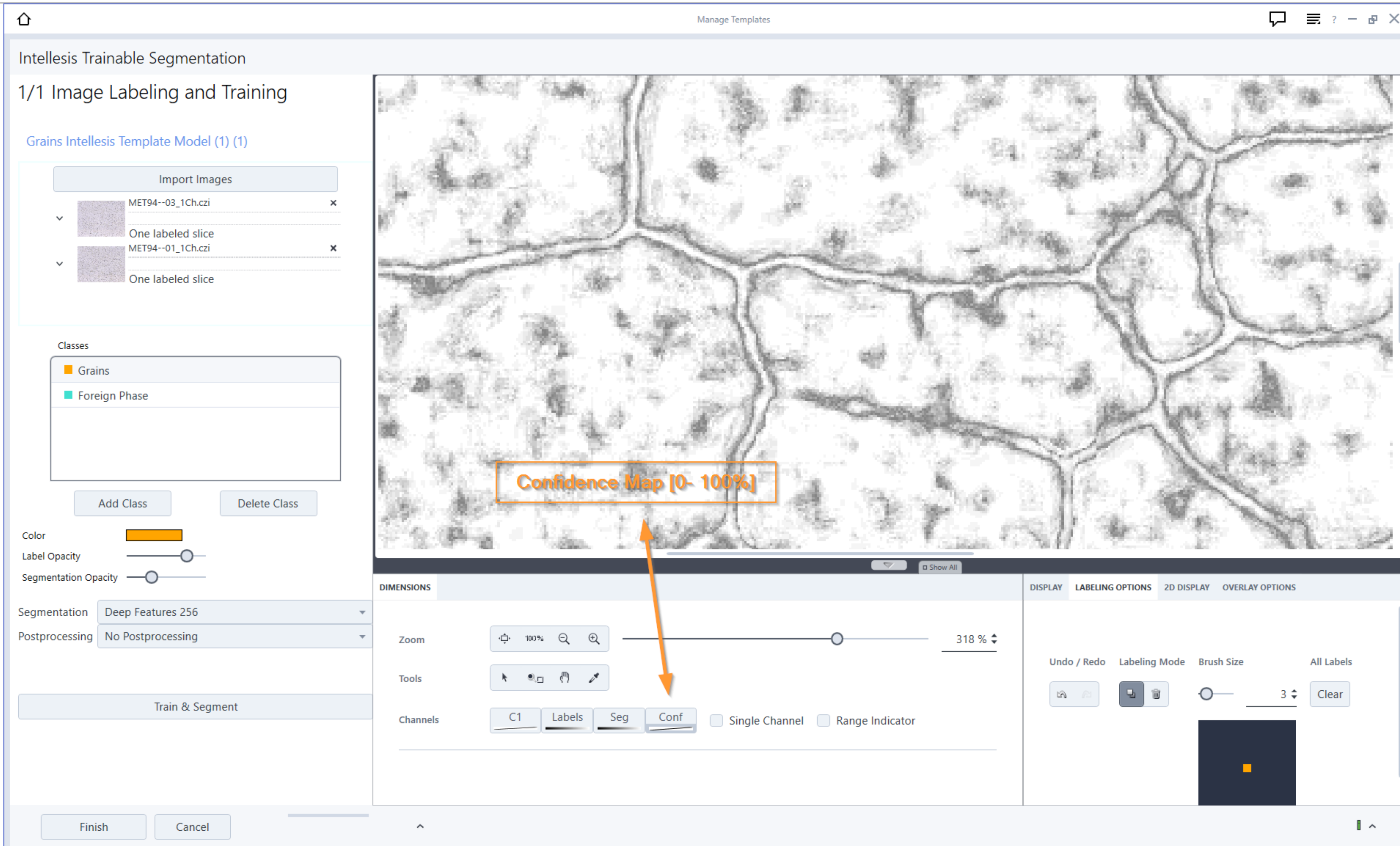
Machine learning

ZEISS



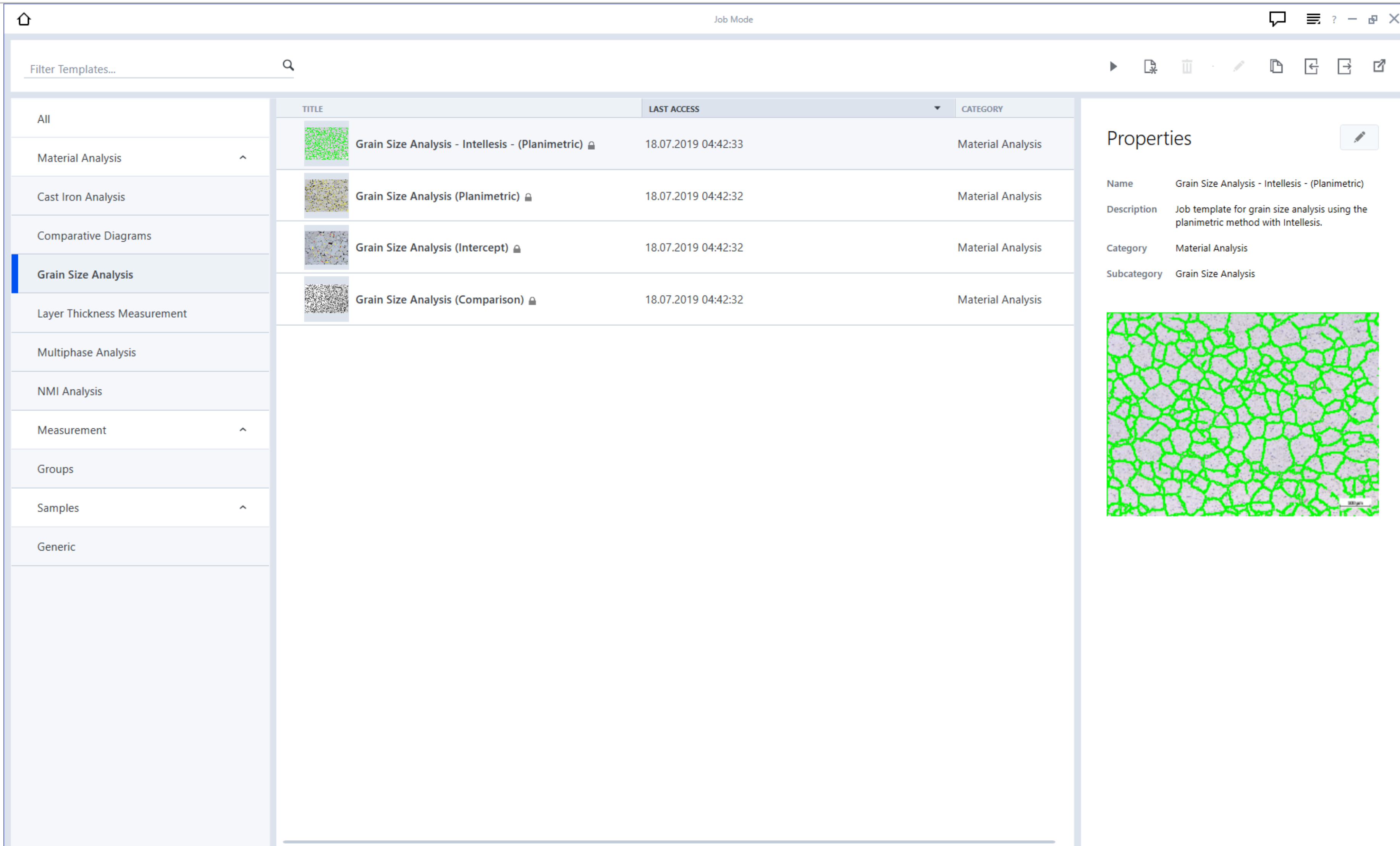
Intellesis – Confidence map in Training UI

Machine learning

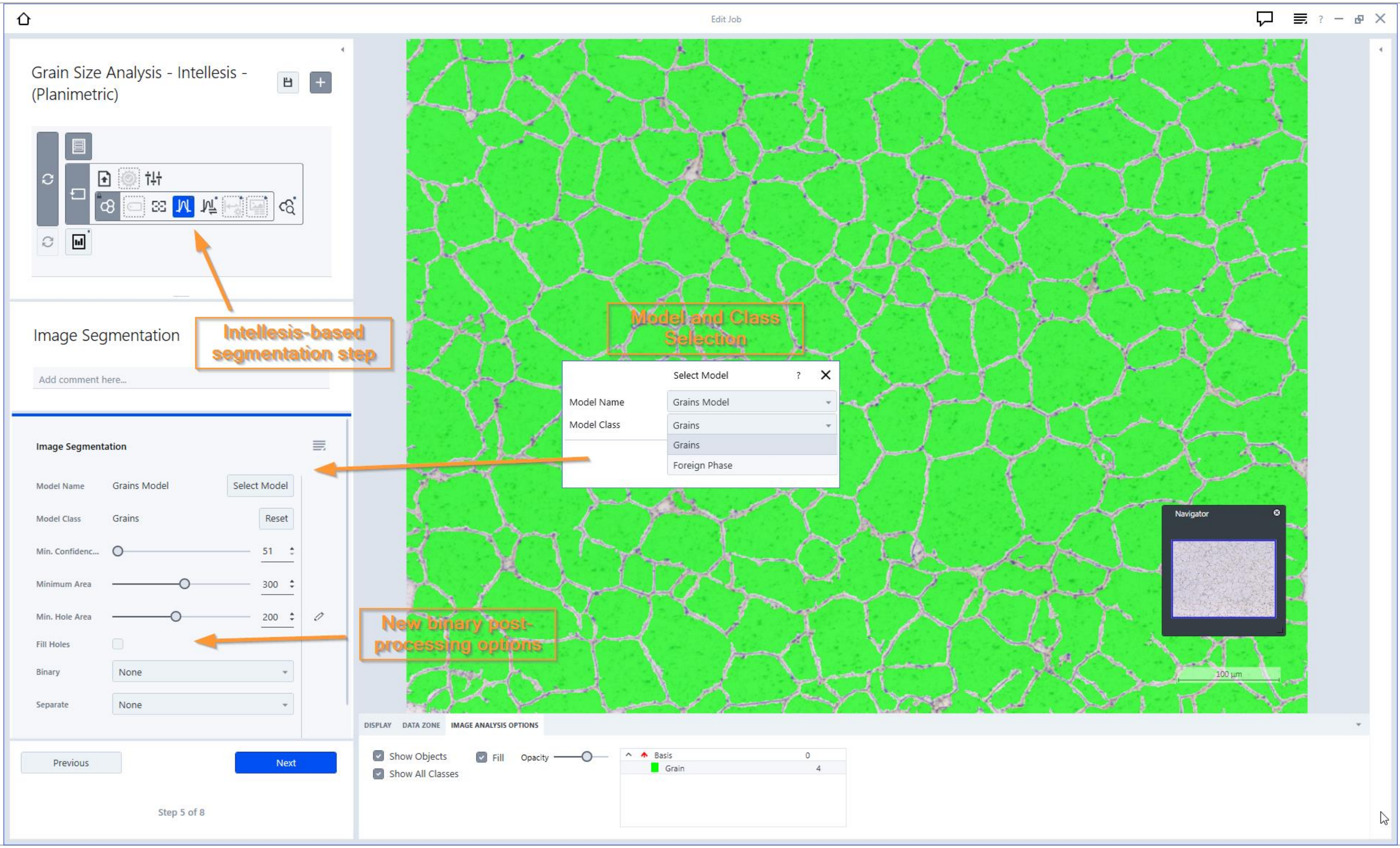


Intellesis – Grains Integration

Machine learning

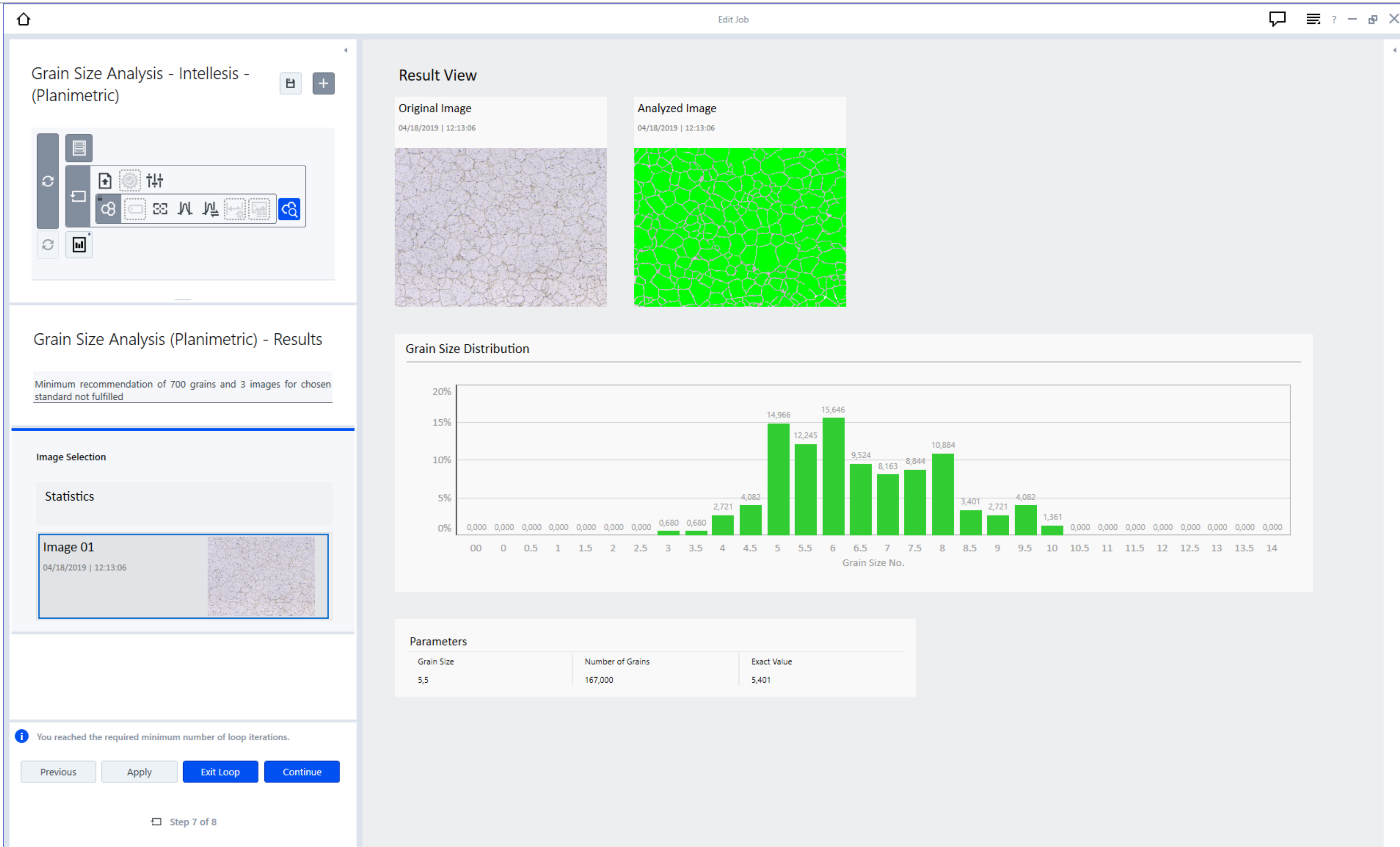


Intellesis – Grains Integration



Intellesis – Grains Integration

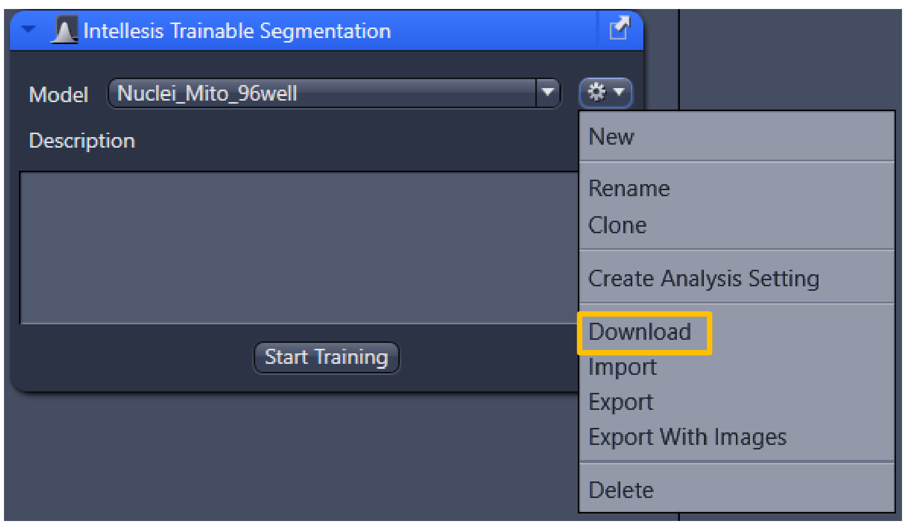
Machine learning

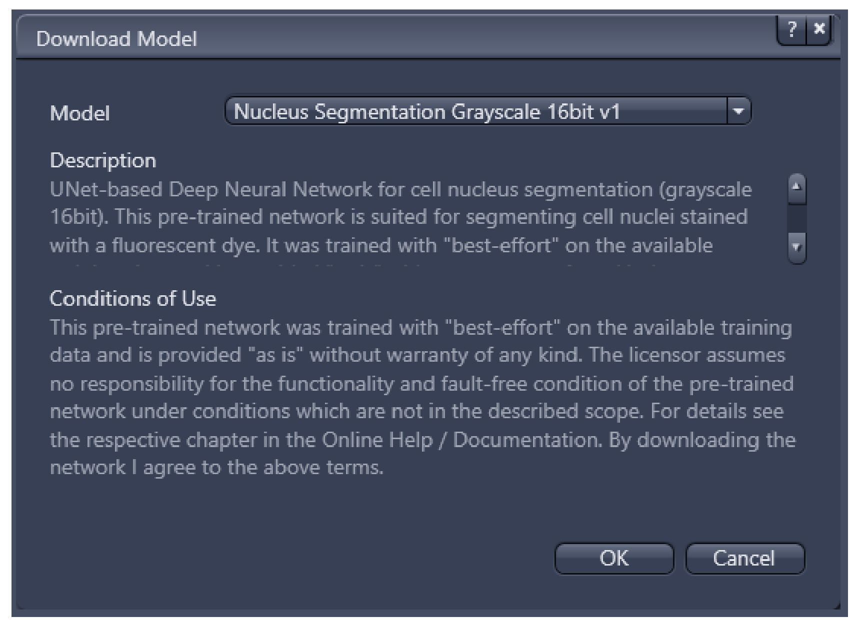


Download Deep Neural Networks for specific application

Machine learning

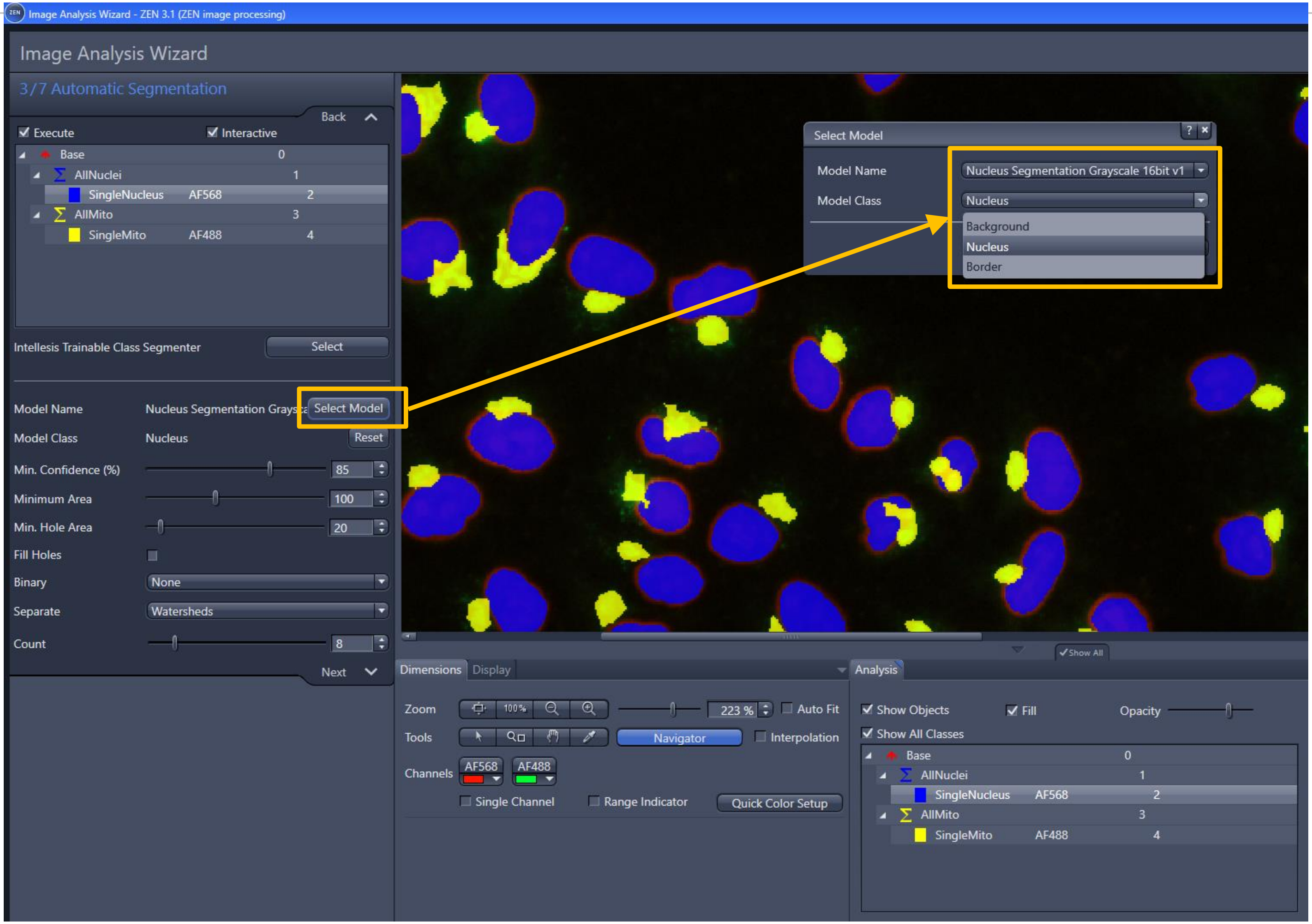
- Download dialog to select networks(s) from ZEISS
- First Application: Pre-trained Nucleus Detection
- Use directly for Segmentation, Image Analysis incl. Class Segmentation





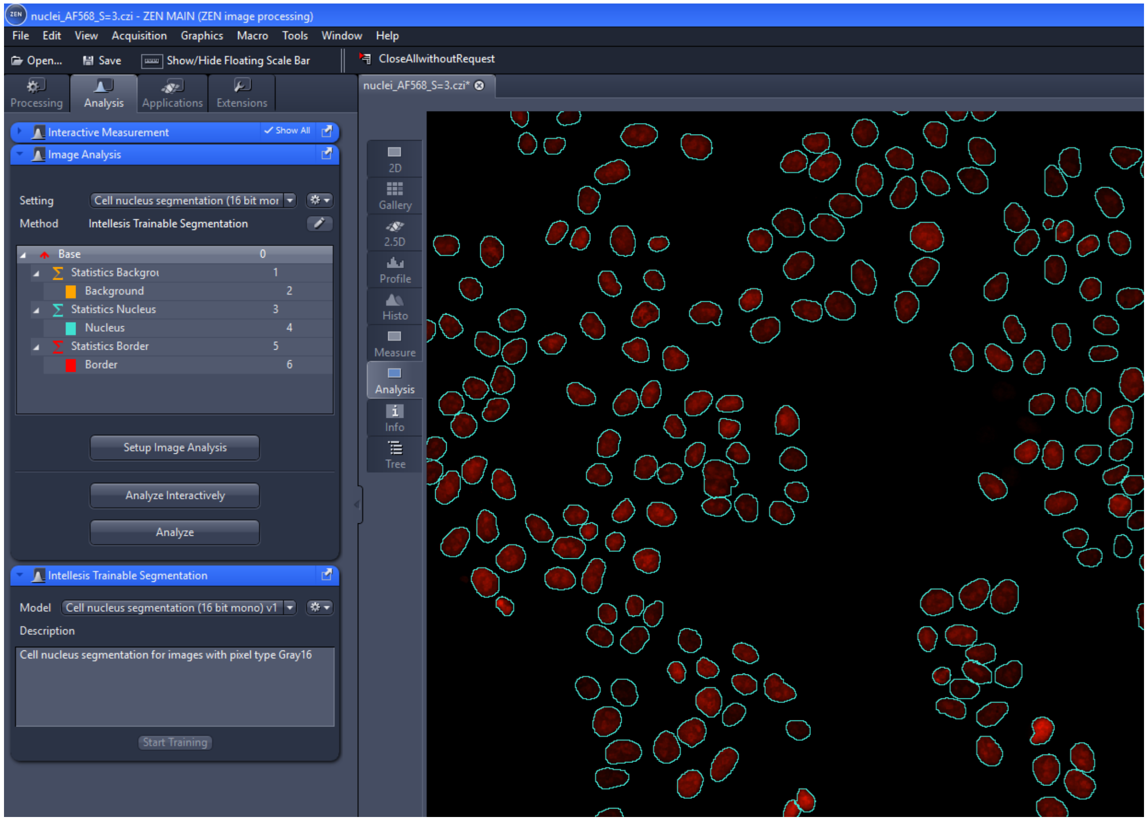
Nucleus Detector - Use inside Class Segmentation

Machine learning



Pretrained Network – Integration with Image Analysis

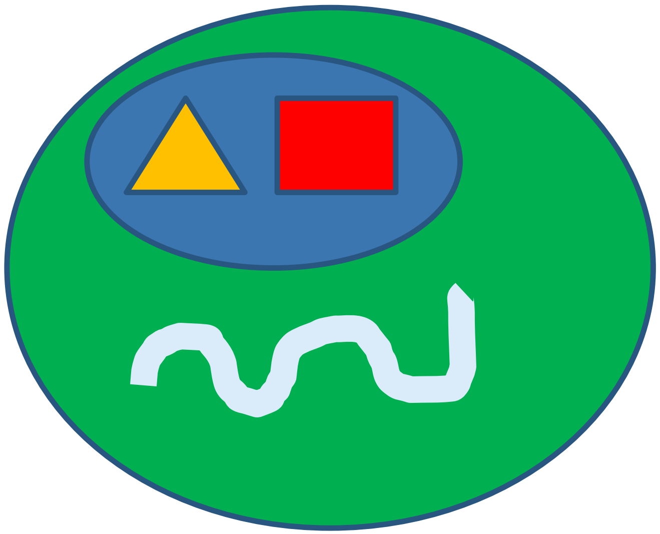
Machine learning

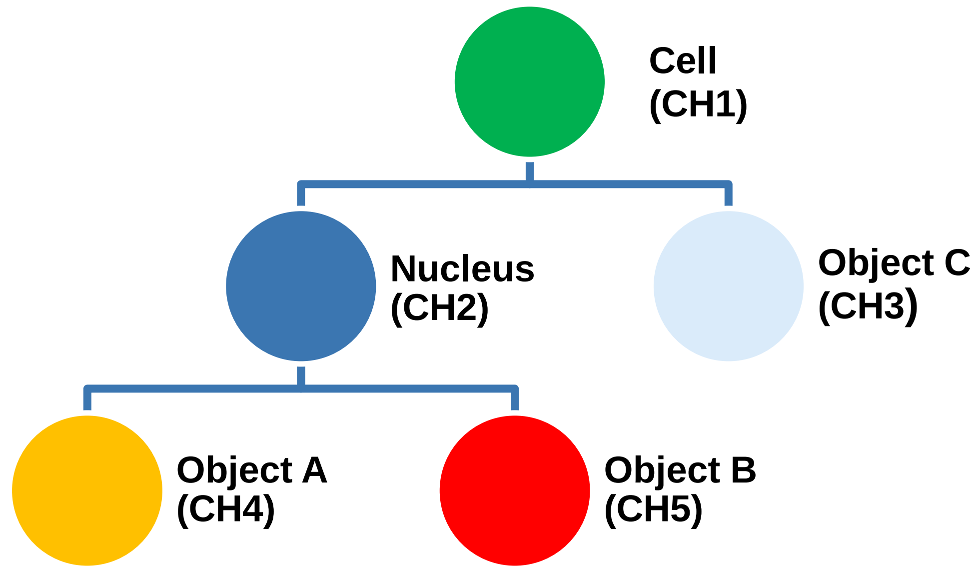


- Network can be “converted” to a fully functional Image Analysis Pipeline by one single click
- Such settings will now also have the advanced binary postprocessing options
- Confidence Threshold is also available
- There are three models available and more to come

Class Segmentation - Hierarchical structures with independent segmenter per class

Machine learning



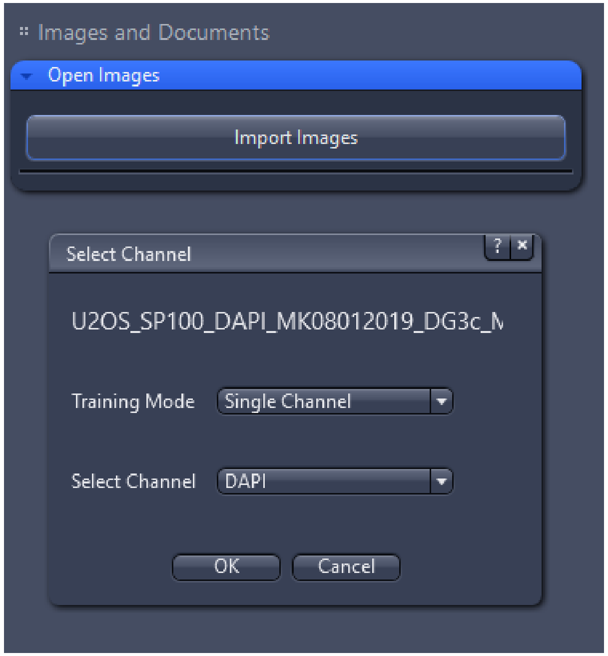


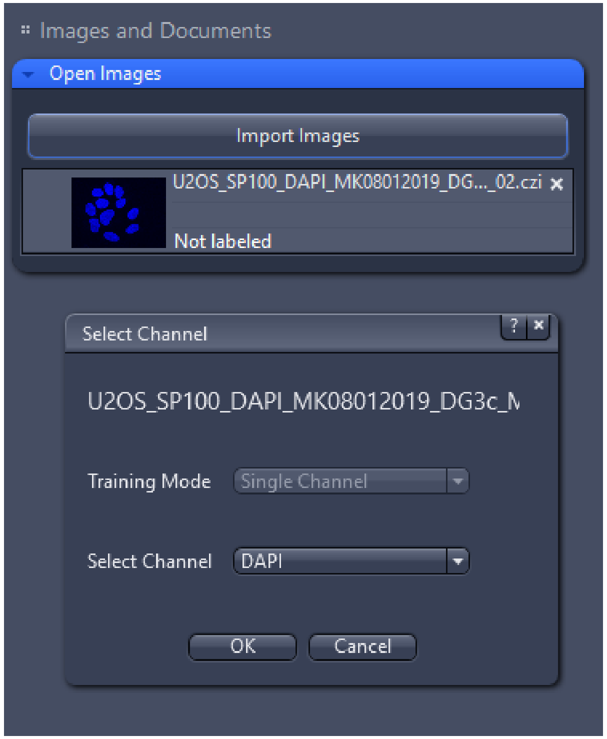
<u>Flexible & Hierarchical Segmentation Scheme (Class Segmentation)</u>

- Cell - classical threshold on CH1
  - Nucleus - classical threshold on CH2
    - Object A - trained Intellesis model on CH3
    - Object B - trained Intellesis model on CH4
  - Object C - classical threshold on CH5

Class Segmentation - Hierarchical structures with independent segmenter per class

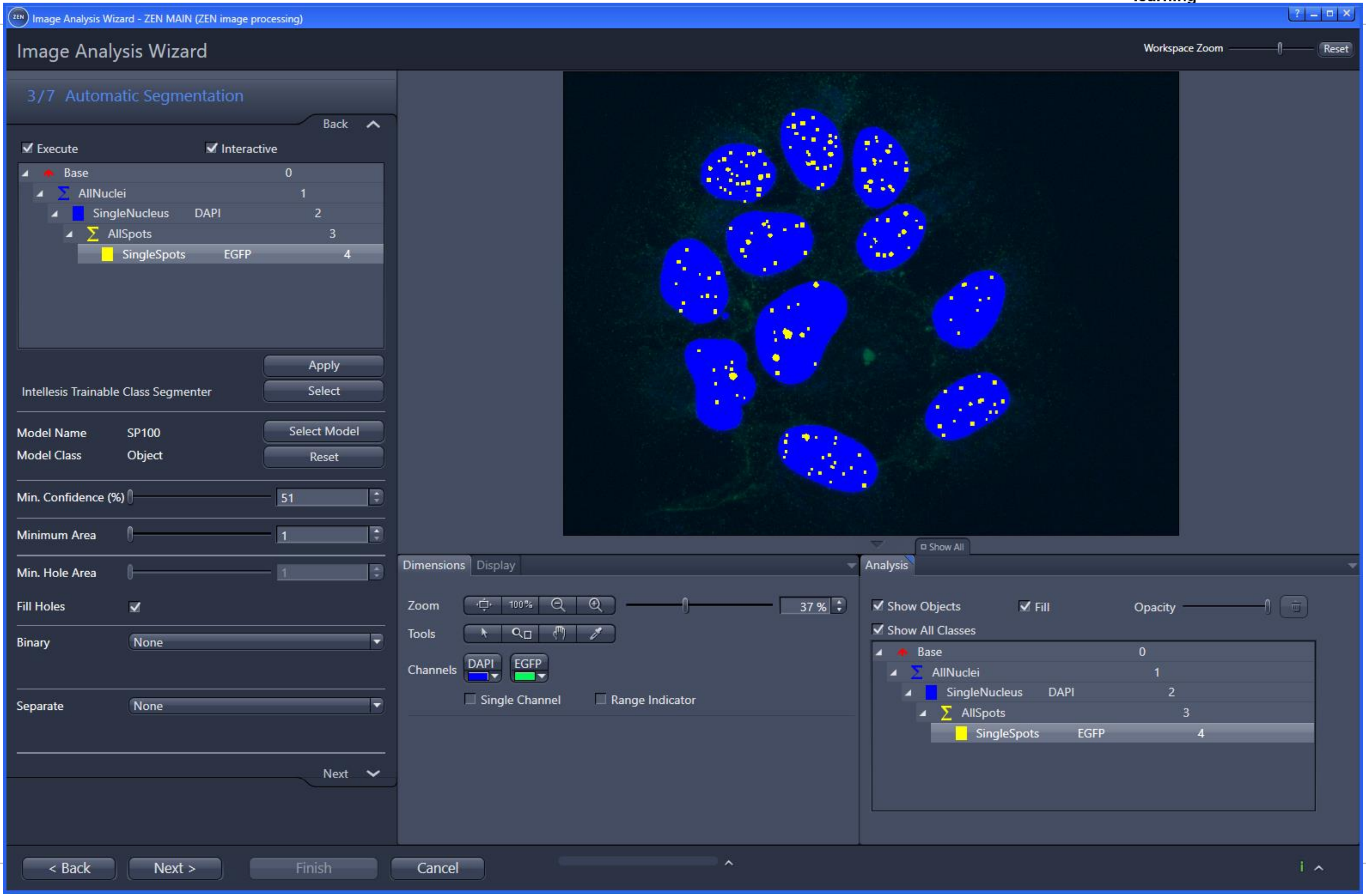
- Import dialog for channel selection incase of multichannel image
- Single Channel is the default
- Next import allows only selecting the channel





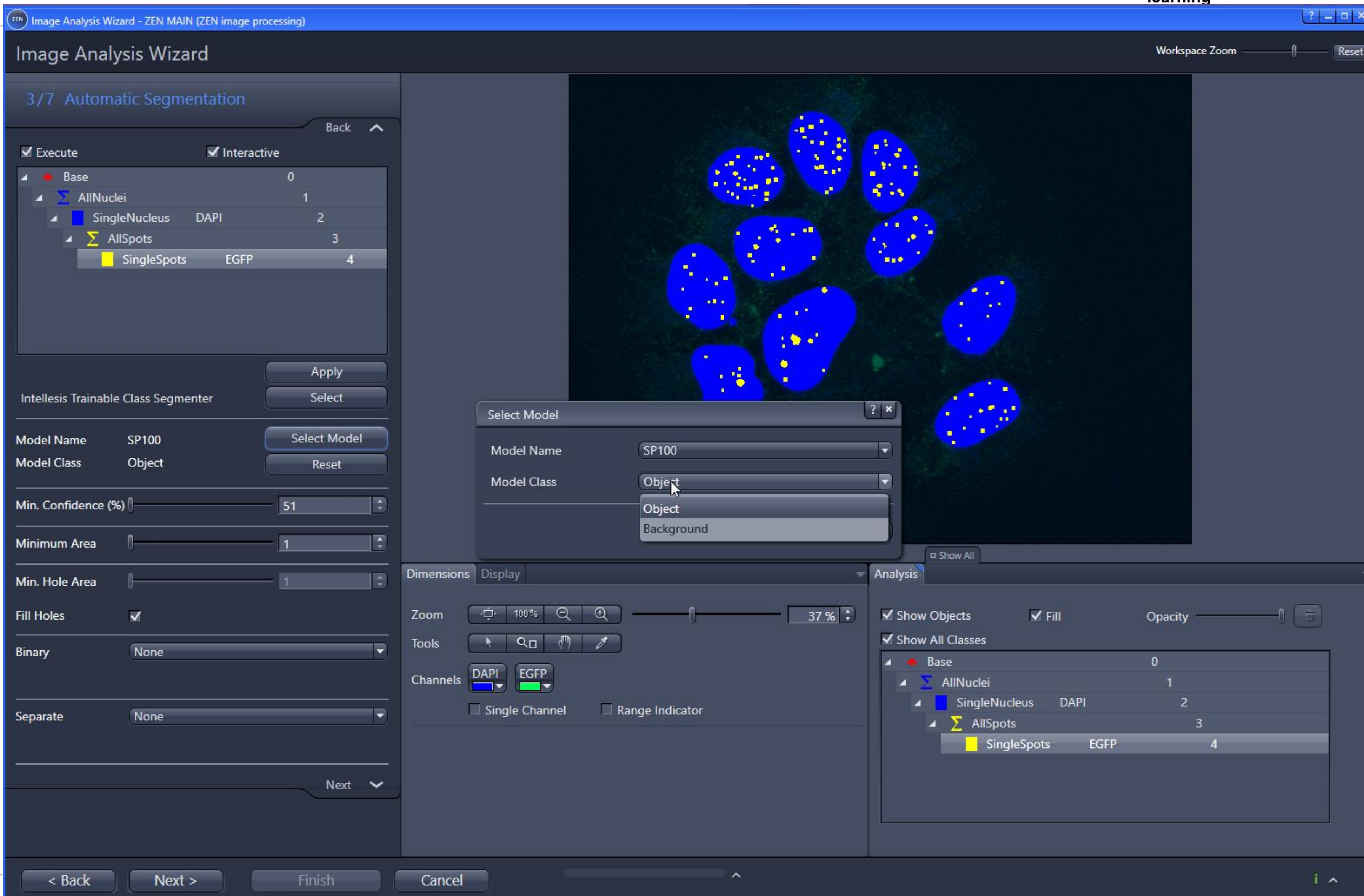
Class Segmentation - Hierarchical structures with independent segmenter per class

Machine learning



Class Segmentation - Hierarchical structures with independent segmenter per class

Machine learning



Deep neural Networks versus Threshold

- Starting with ZEN Blue 3.1 it is possible to download pretrained networks for Segmentation

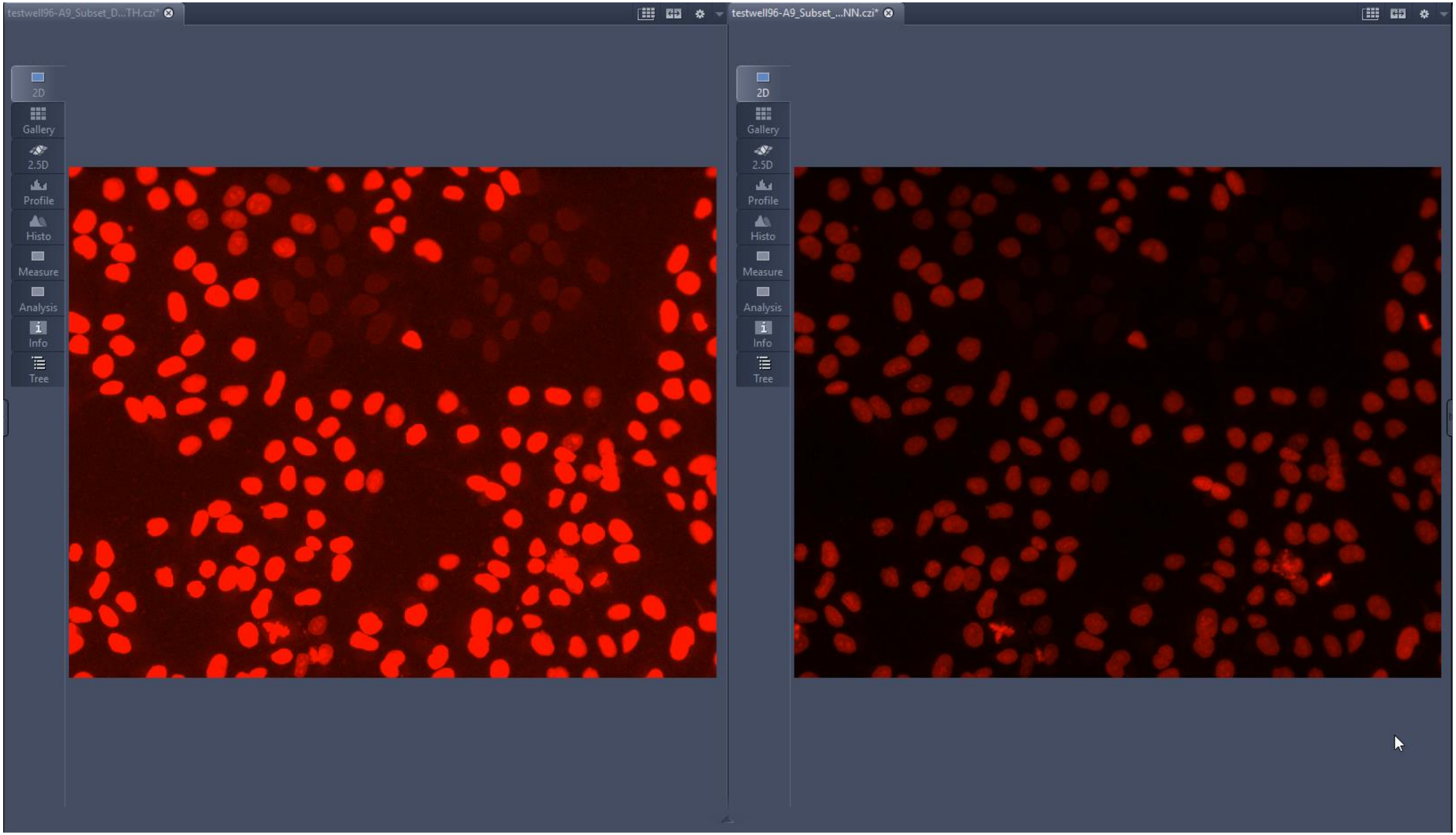
- While Thresholding combined with classical PreProcessing function is really fast, it can be challenging (or impossible), especially for the non-expert user, to get the same results

- To illustrate this the following slides shows the same image analyzed by an Image Analysis Setting using

  – preProcessing + Thresholding
  – DNN-based Segmentation

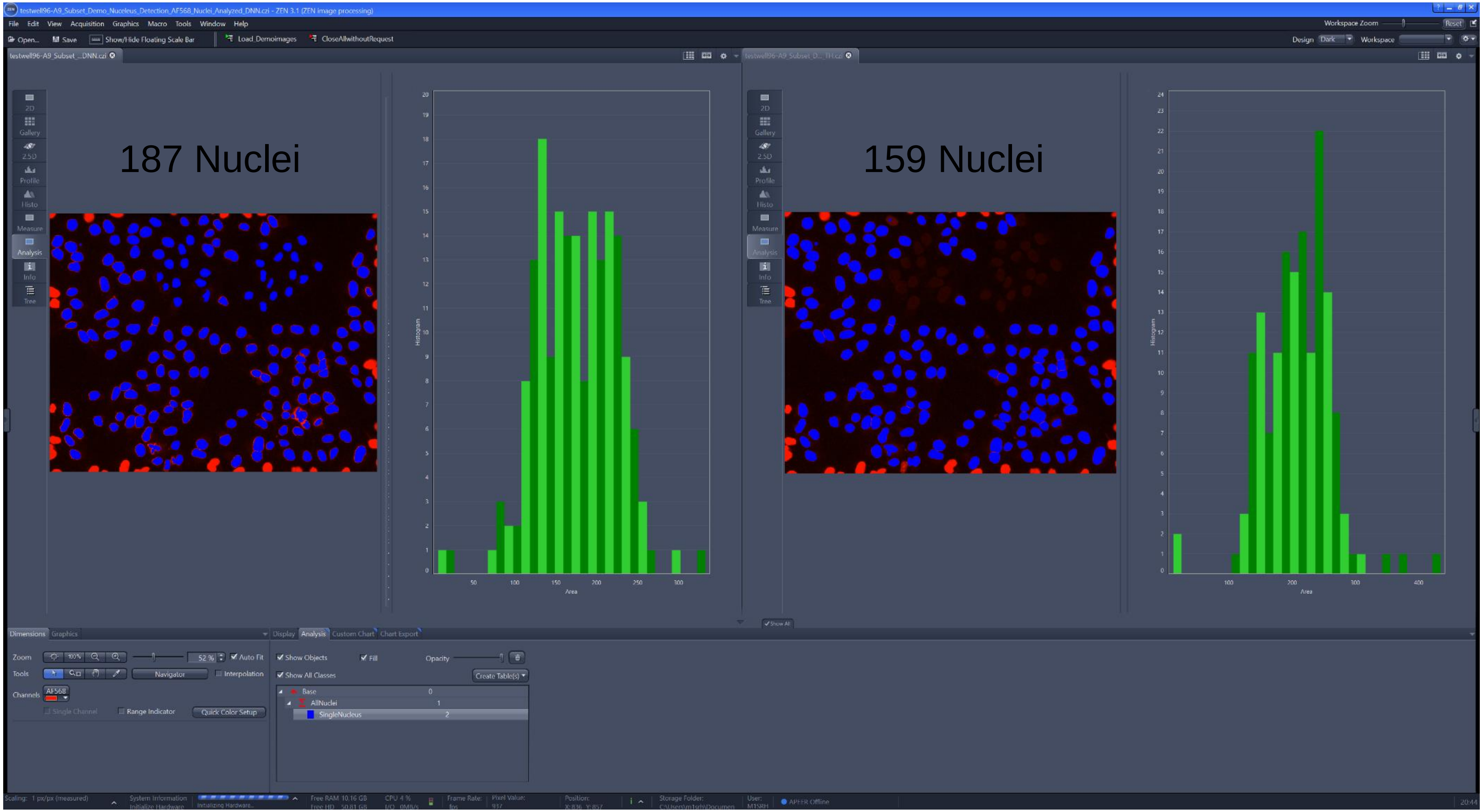
Deep neural Networks versus Threshold
Same Image Data – different Display Curve

Machine learning



Deep neural Networks versus Threshold

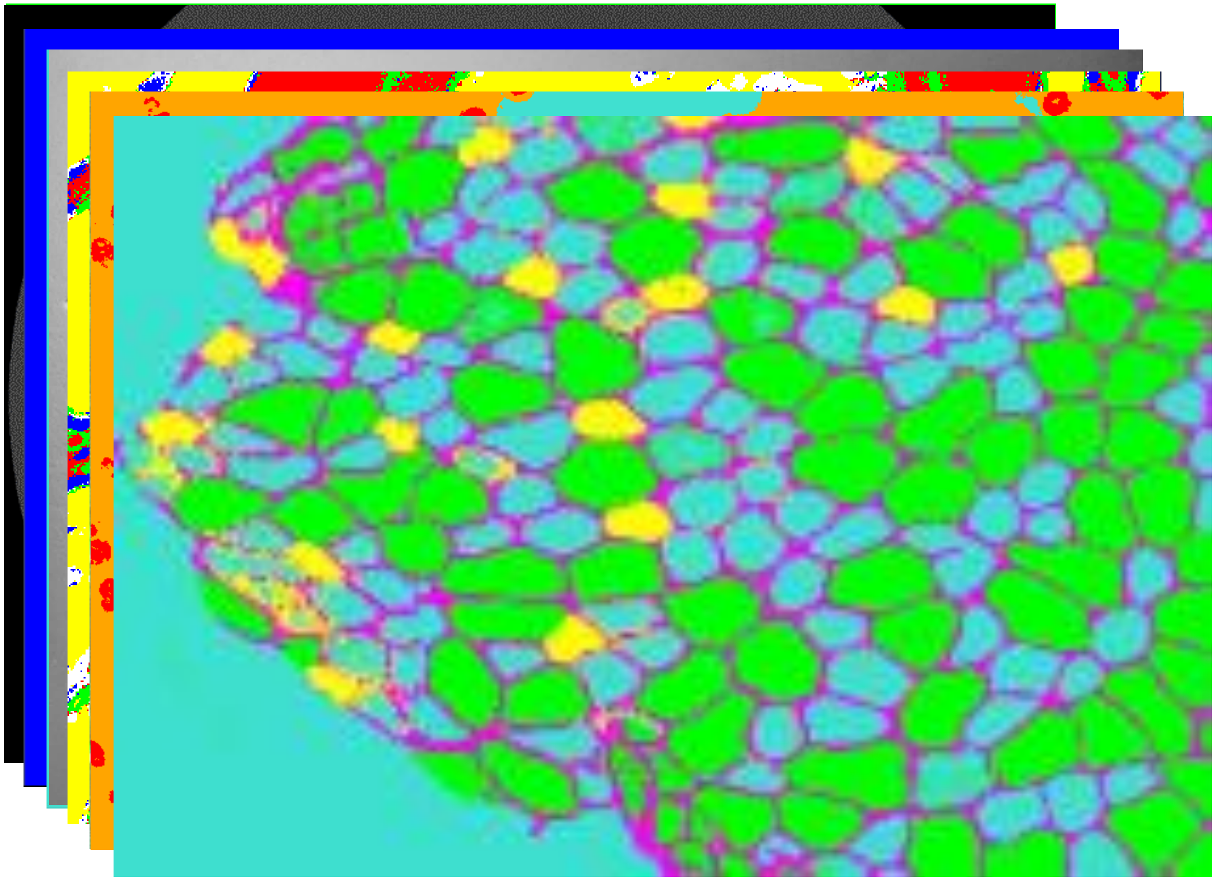
Same Image Data – different results



ZEN Intellesis

Machine Learning based image segmentation

Machine learning

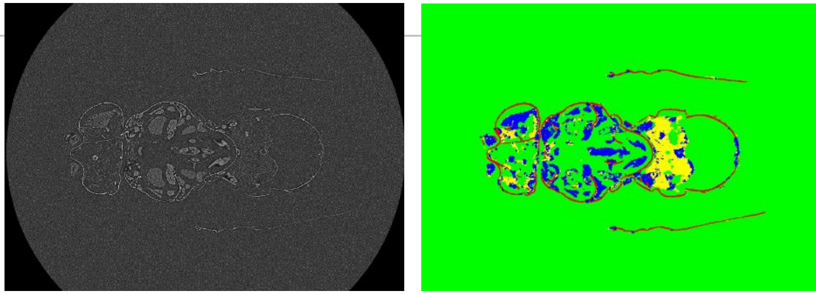


Sample courtesy [illegible] Institute, London

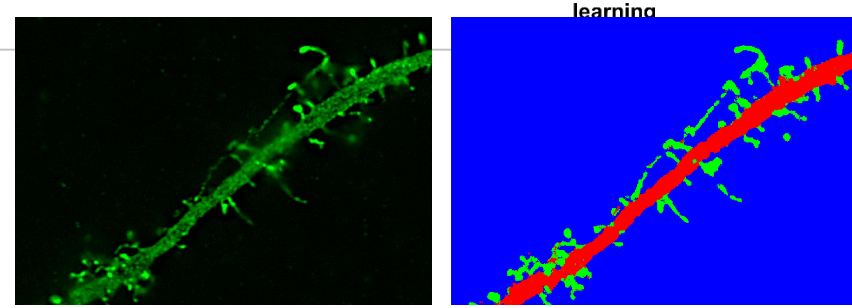
- XRM scan: Drosophila
- Elyra SIM: Neuron – spines & dendrite
- Celldiscoverer 7 (BF): Organoid and migrated cells
- EM: classification of cell compartments
- Celldiscoverer 7 (BF + FL): Scratch assay – area and mitotic cells
- Airyscan Fast: Muscle section – different myosin types

Intellesis Applications: Life Science Examples

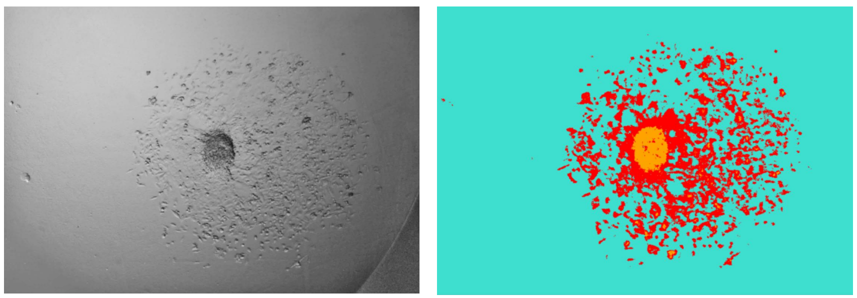
Machine learning



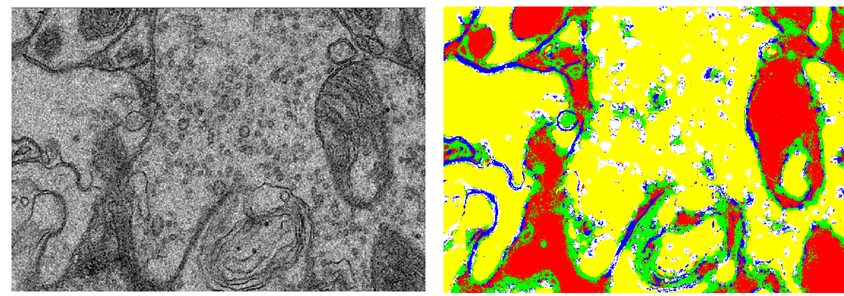
XRM scan: Drosophila



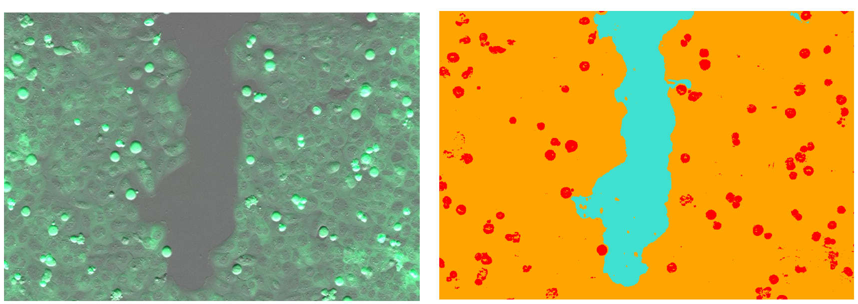
Neuron – spines & dendrite



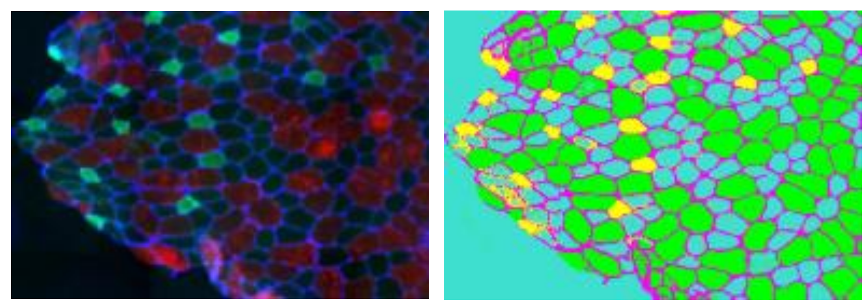
Organoid and migrated cells



Classification of cell compartments



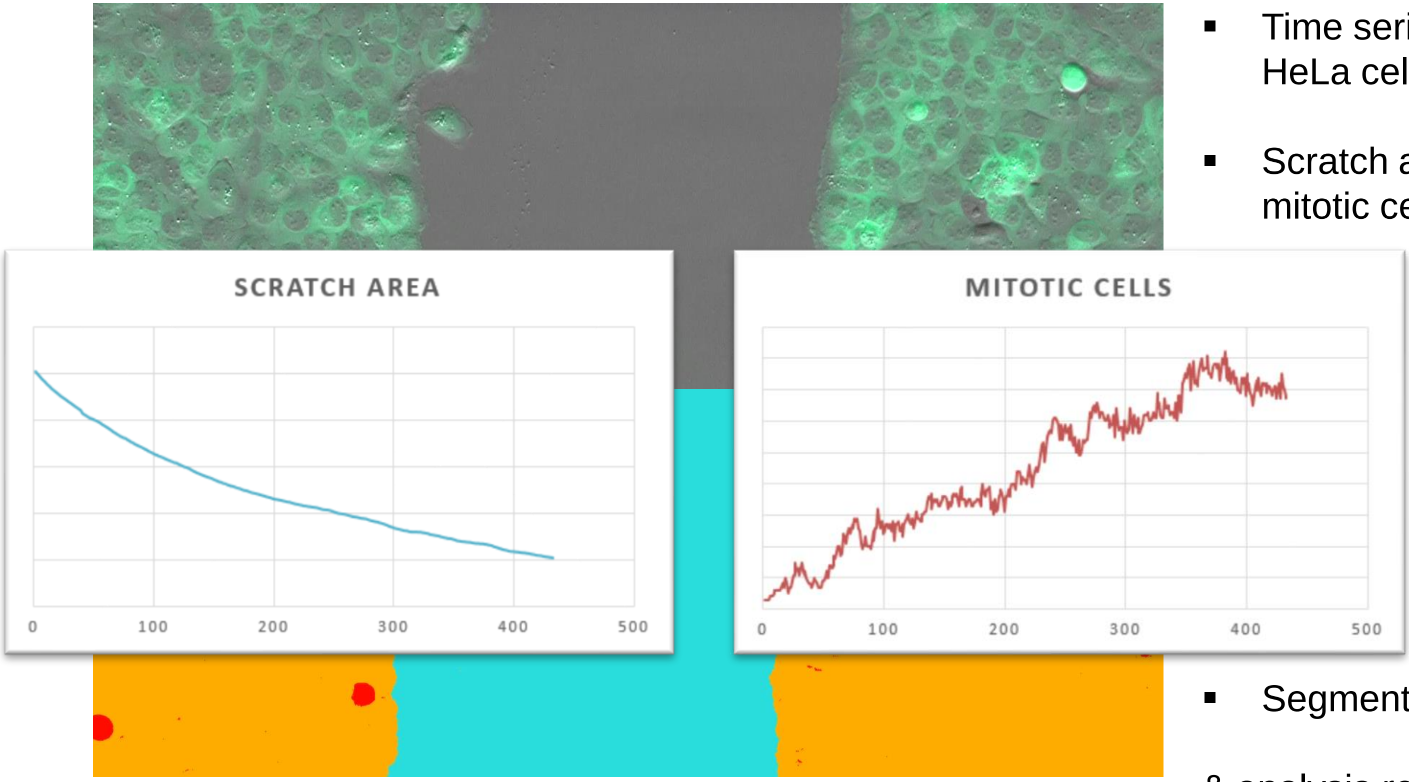
Scratch assay – area and mitotic cells



Muscle section – different myosin types

ZEN Intellesis
Scratch Assay

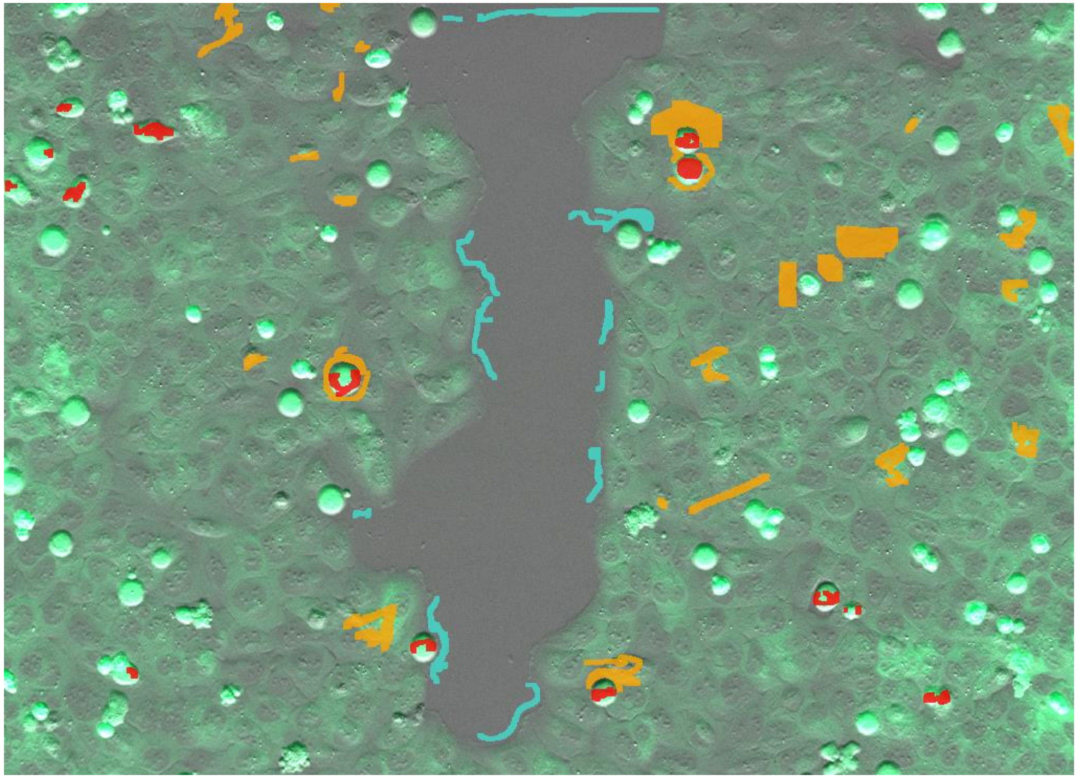
Machine learning



- Time series of GFP expressing HeLa cells
- Scratch area & number of mitotic cells shall be measured
- Segmented data

& analysis results

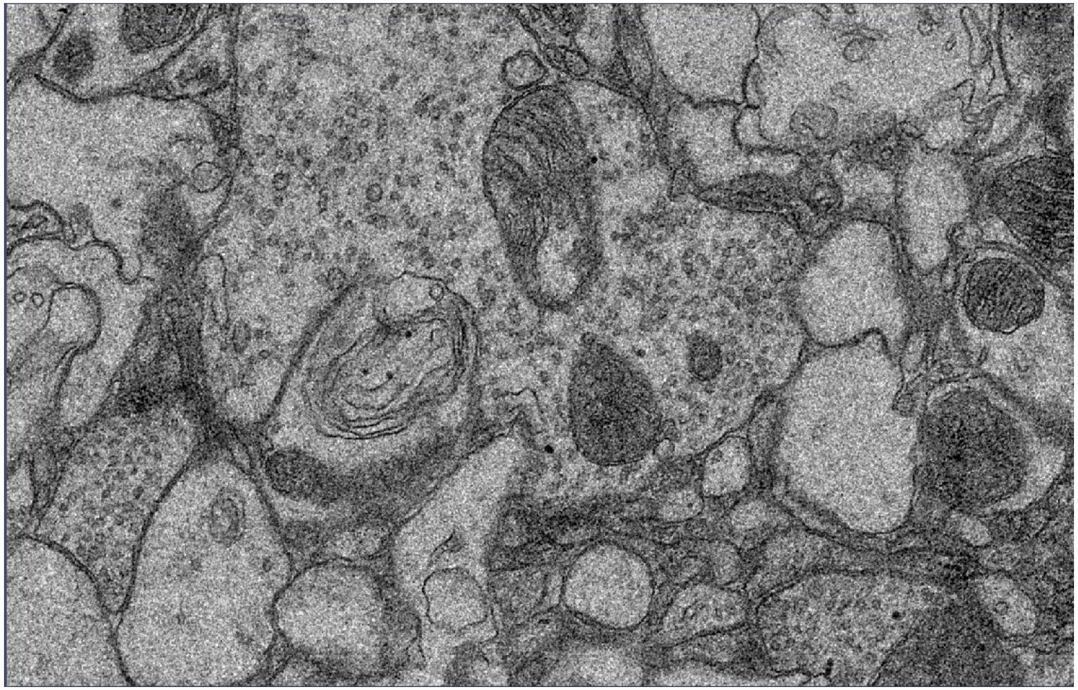
ZEN Intellesis
Scratch Assay



- Time series of GFP expressing HeLa cells
- Scratch area & number of mitotic cells shall be measured
- Training („painted pixels“)

ZEN Intellesis
EM dataset – pre-synaptic vesicles

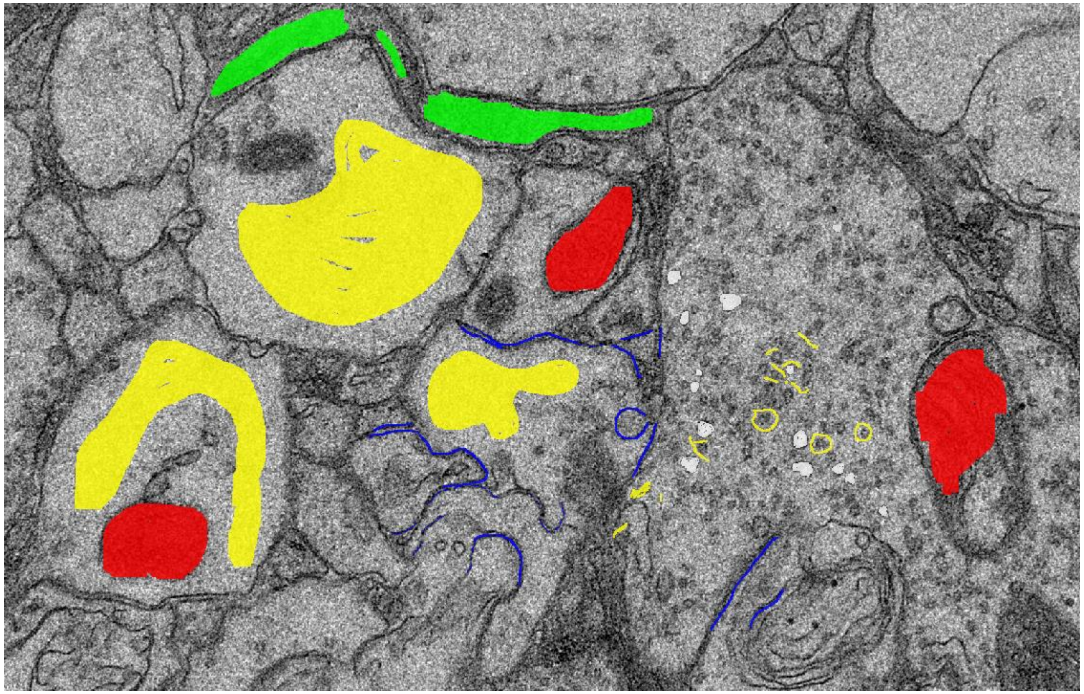
Images Courtesy of
MPI for infection biology, Berlin



- Section through the calyx region of 30d Drosophila melanogaster, aim
- quantification of synaptic vesicles and structure characterization
- Raw data

ZEN Intellesis
EM dataset – pre-synaptic vesicles

Images Courtesy of
MPI for infection biology, Berlin

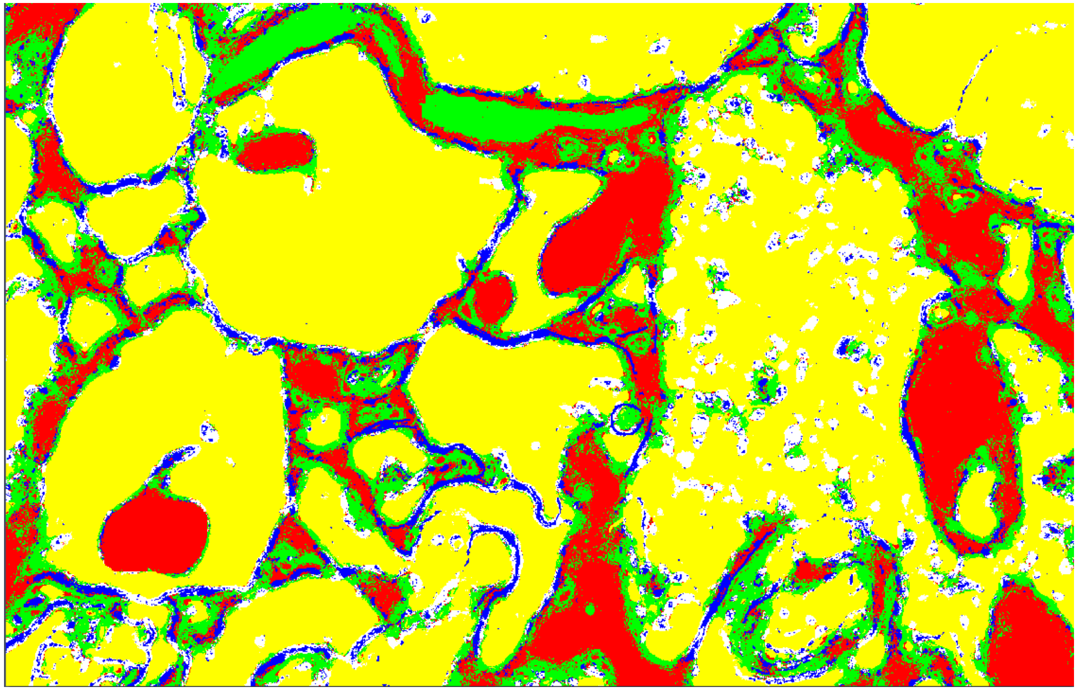


- Section through the calyx region of 30d Drosophila melanogaster, aim
- quantification of synaptic vesicles and structure characterization
- Training („painted pixels“)

ZEN Intellesis
EM dataset – pre-synaptic vesicles

Machine learning

Images Courtesy of
MPI for infection biology, Berlin

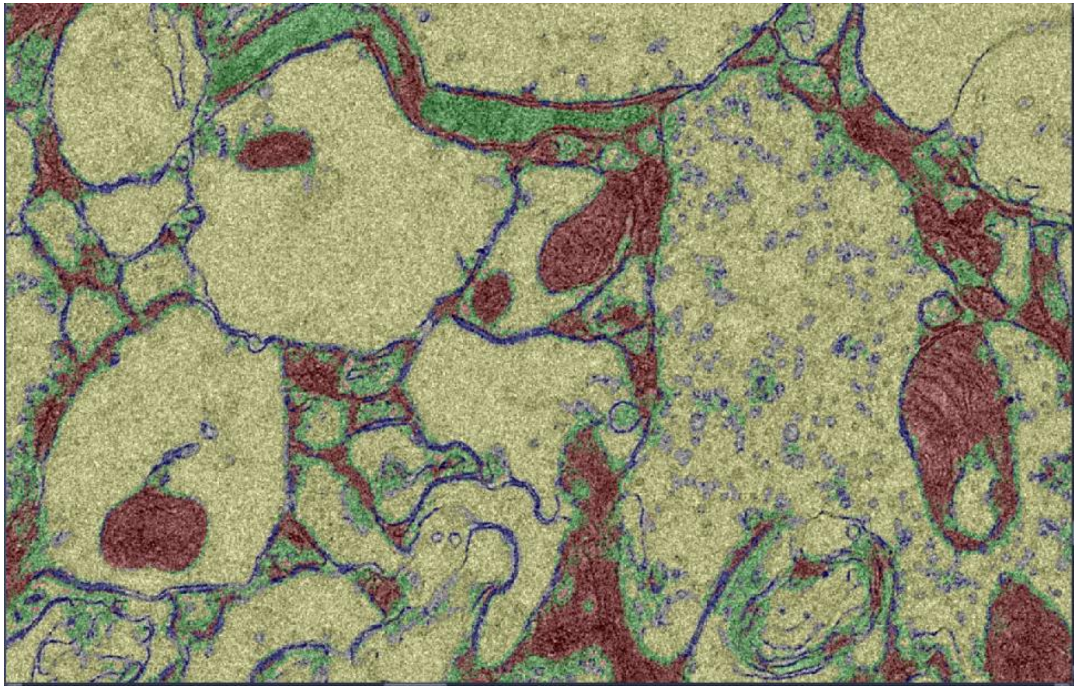


- Section through the calyx region of 30d Drosophila melanogaster, aim
- quantification of synaptic vesicles and structure characterization
- segmented data

ZEN Intellesis
EM dataset – pre-synaptic vesicles

Machine learning

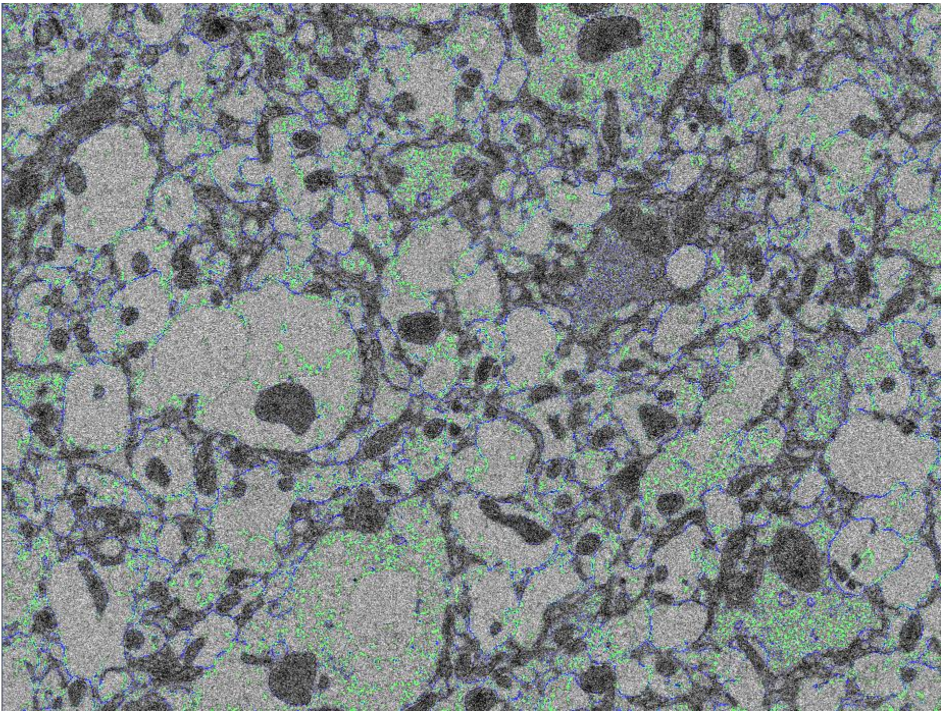
Images Courtesy of
MPI for infection biology, Berlin



- Section through the calyx region of 30d Drosophila melanogaster, aim
- quantification of synaptic vesicles and structure characterization
- segmented data + raw data overlay

ZEN Intellesis
EM dataset – pre-synaptic vesicles

Images Courtesy of
MPI for infection biology, Berlin



Full Image

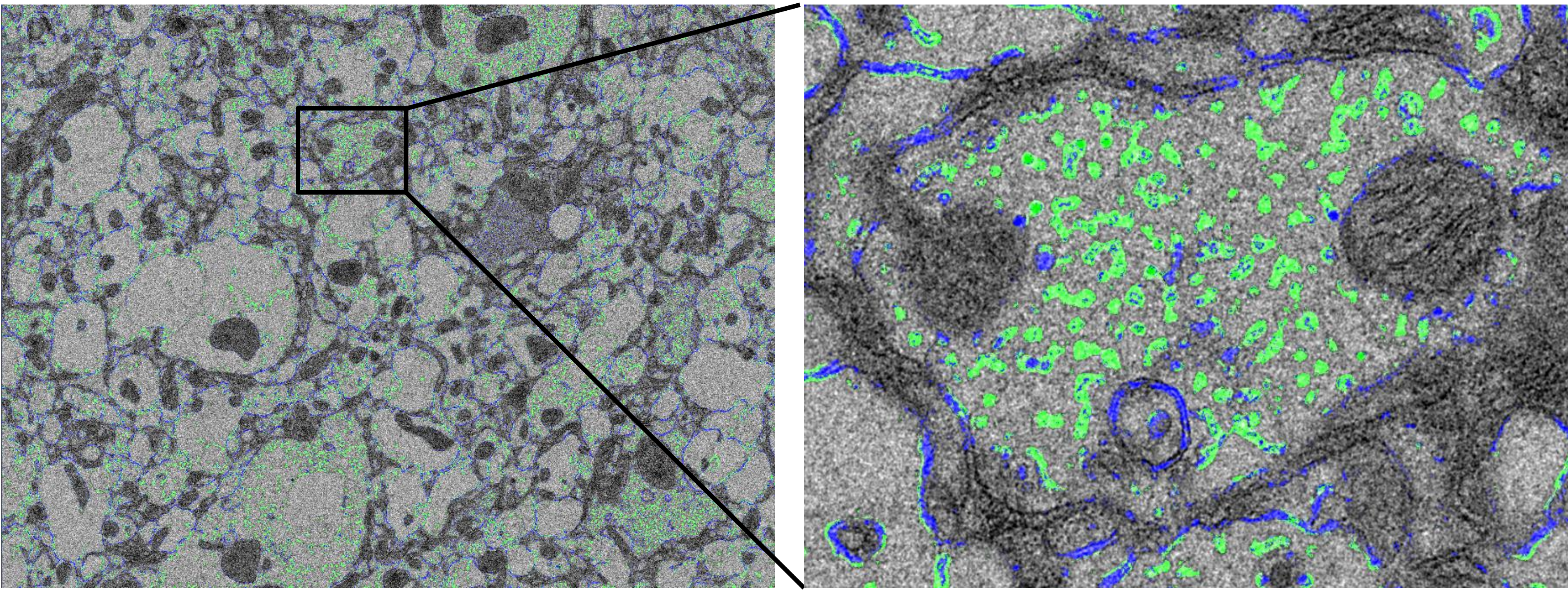
ZEN Intellesis

EM dataset – pre-synaptic vesicles

Machine learning

ZEISS

Images Courtesy of
MPI for infection biology, Berlin

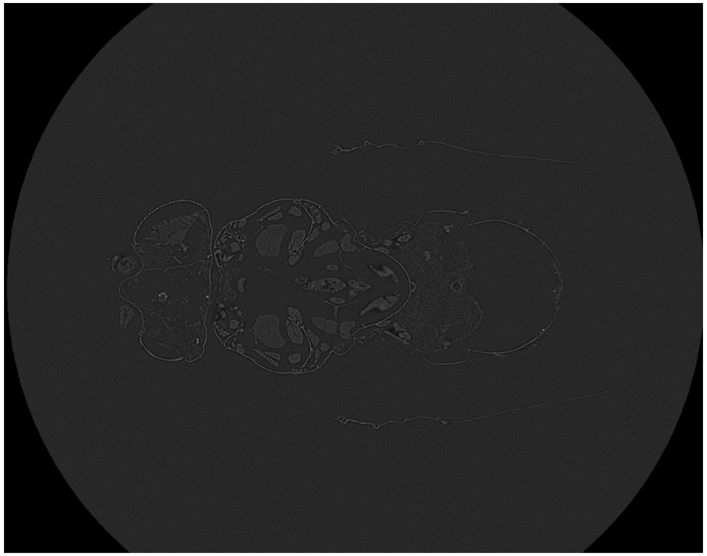


Full Image

Vesicles and Membranes

Pre-synaptic vesicles can easily be quanified based on the Intellesis segmentation.

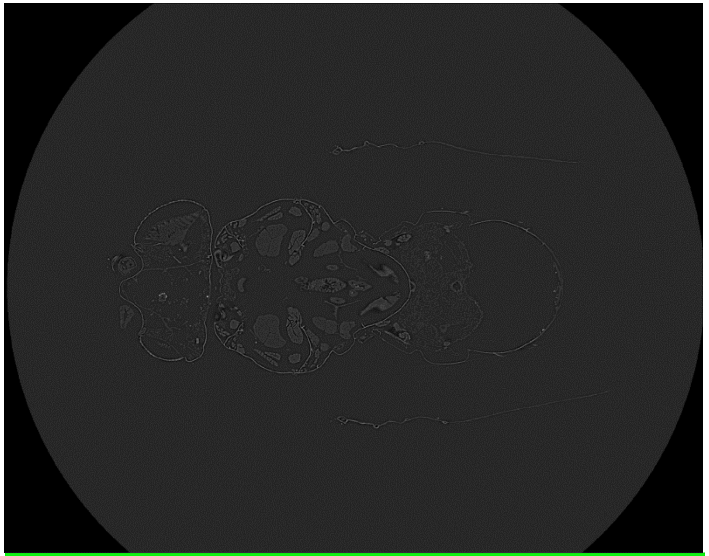
ZEN Intellesis
Example 3: XRM Drosophila



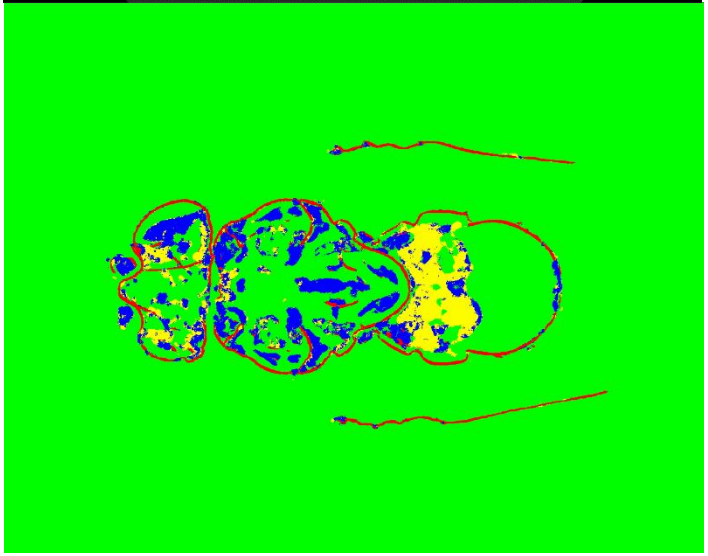
- 3D X-Ray microscope dataset

Dataset Courtesy of
Francis Crick Institute , London

ZEN Intellesis
Example 3: XRM Drosophila



- 3D X-Ray microscope dataset

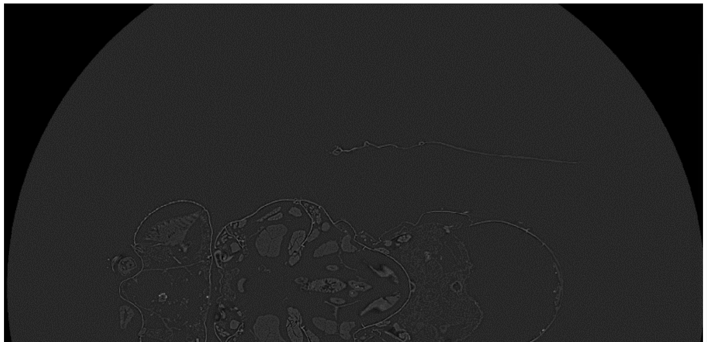


- Segmented data

Dataset Courtesy of
Francis Crick Institute , London

ZEN Intellesis

Example 3: XRM Drosophila



3D rendering

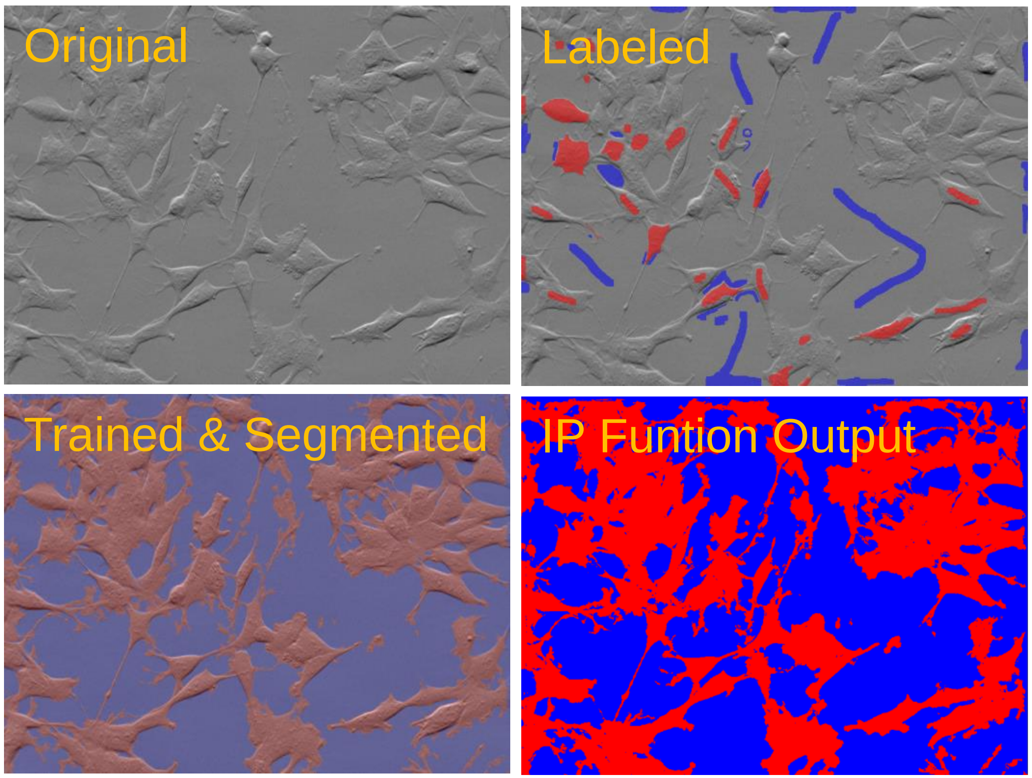
& 3D print…

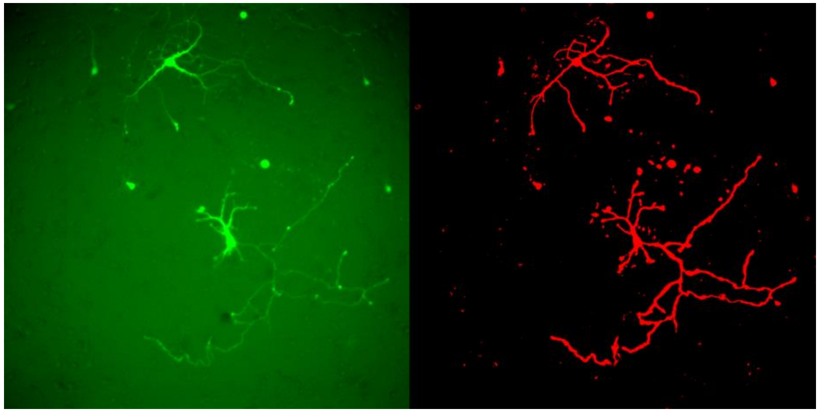
Dataset Courtesy of
Francis Crick Institute , London

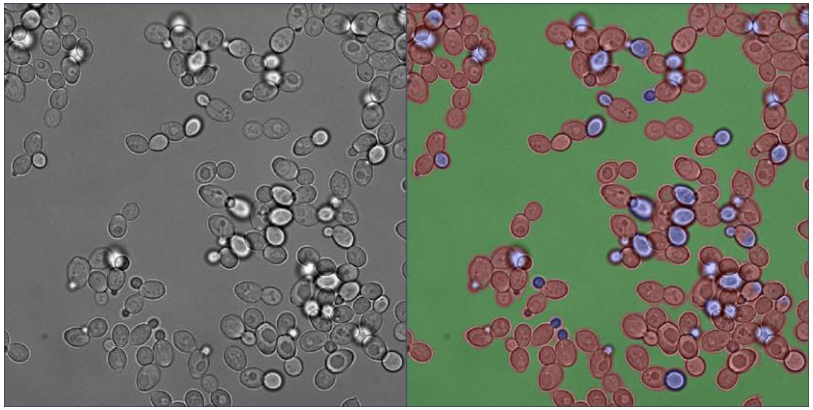
CD7 Phase Gradient Contrast, Neurons, Yeast

Machine learning

ZEISS

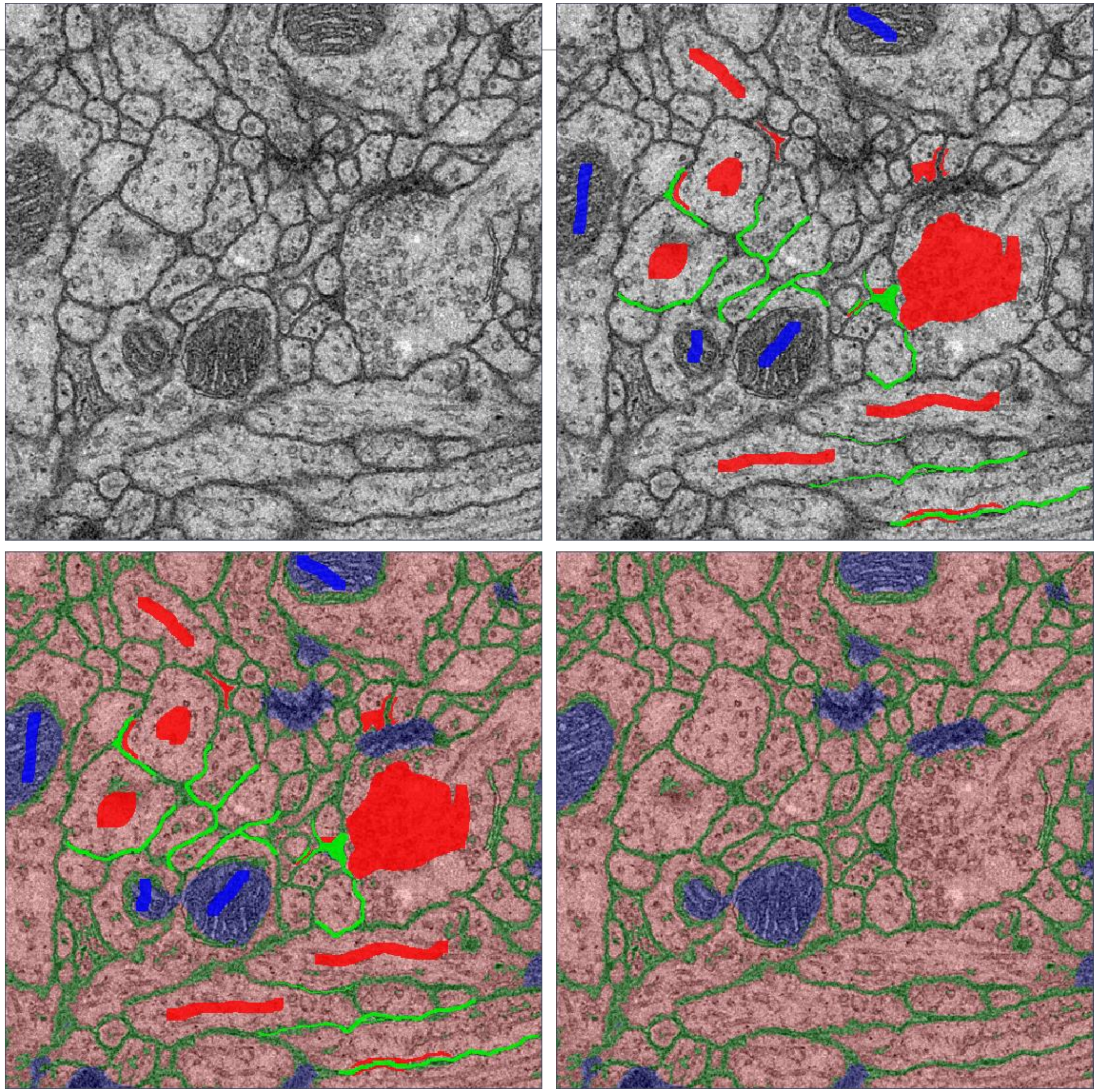




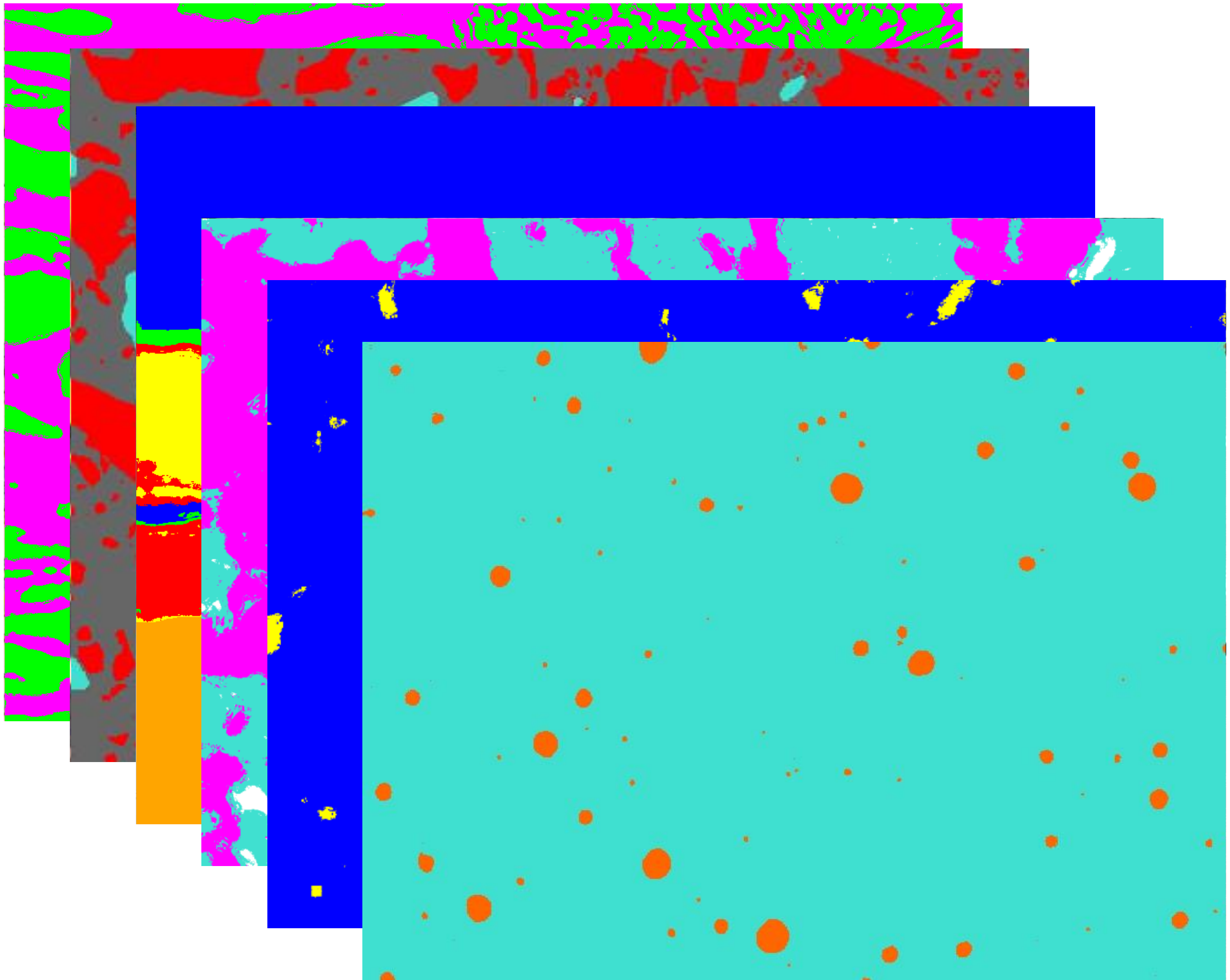


EM-Neurons

Machine learning



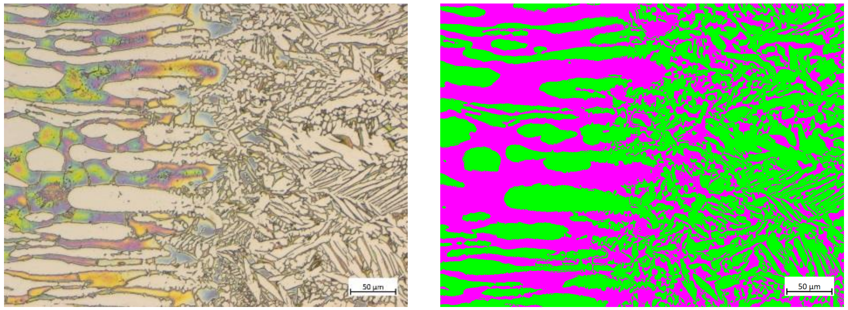
Applications:
Light and electron microscopy



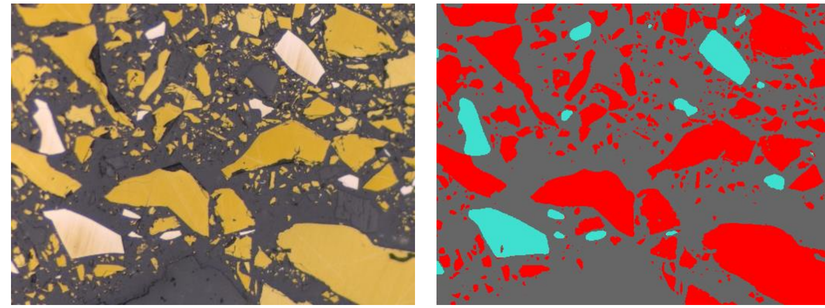
- Multiphase steels
- Ore and mineral samples
- Corrosion scale mapping
- Surface corrosion mapping
- Impurities or inclusions
- Porosity in ceramics

Intellesis Applications: Light and Electron Microscopy

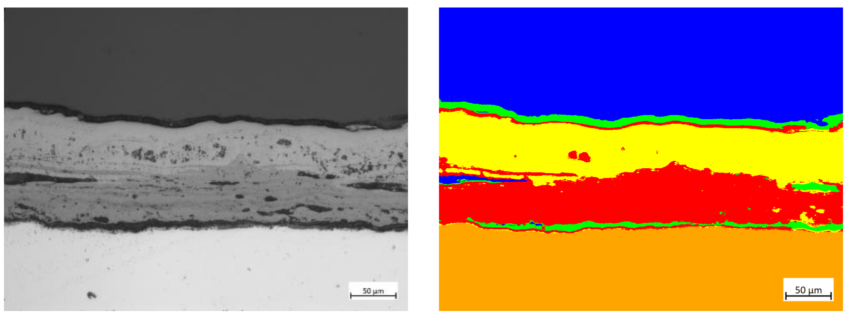
Machine learning



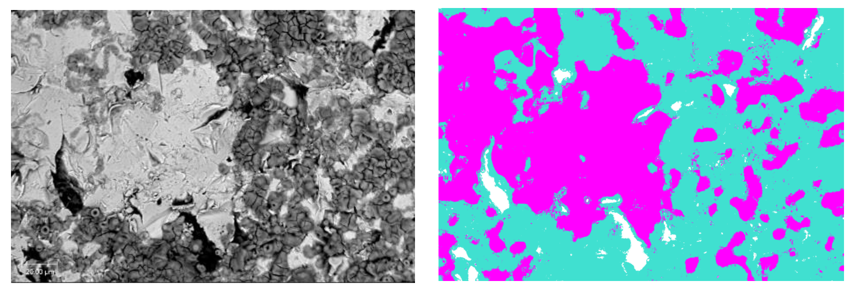
Multiphase steels



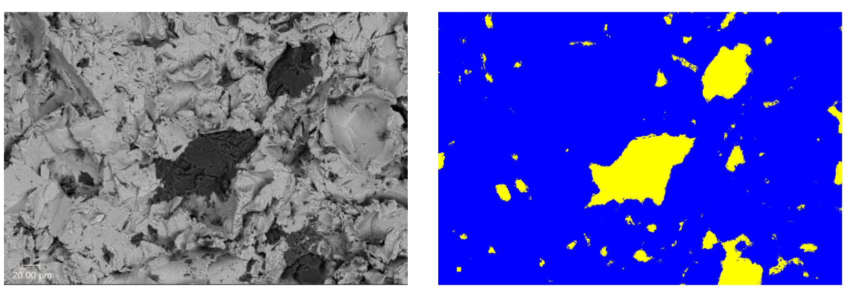
Ore and mineral samples



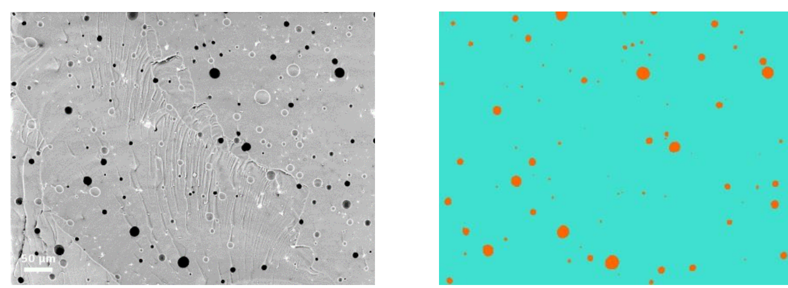
Corrosion scale mapping



Surface corrosion mapping

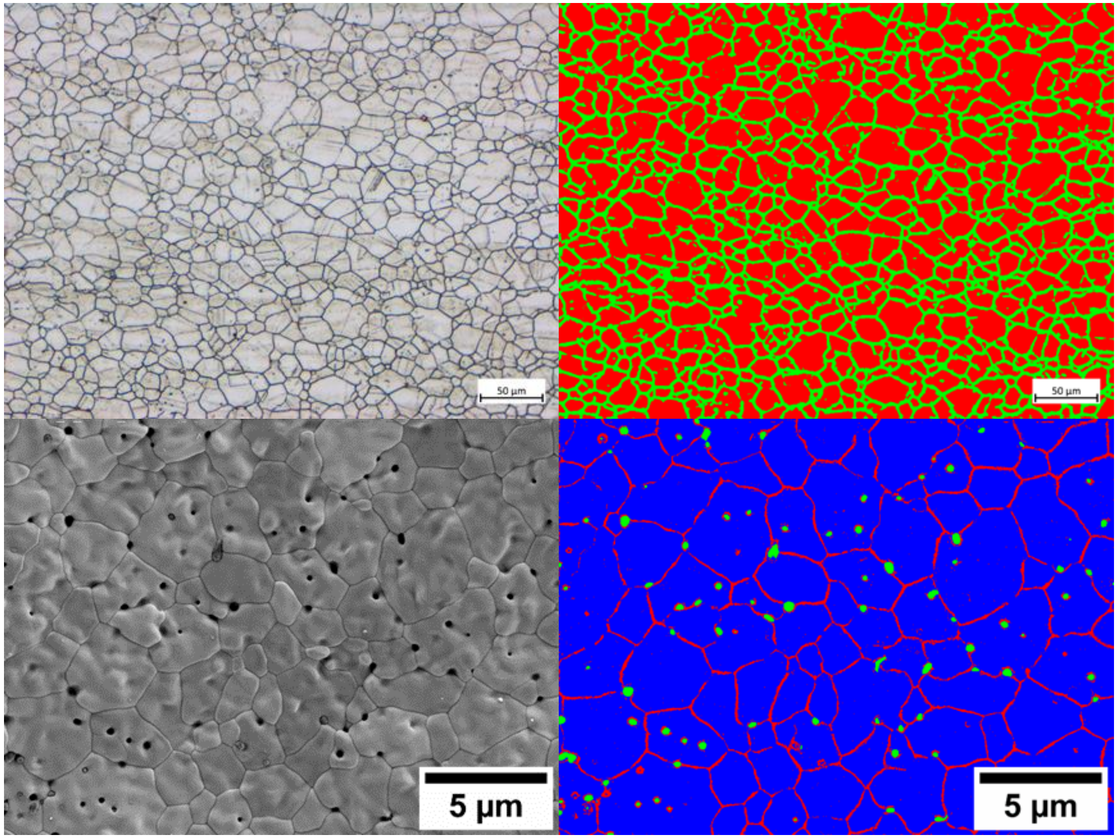


Impurities or inclusions



Porosity in ceramics

Applications:
Light and electron microscopy

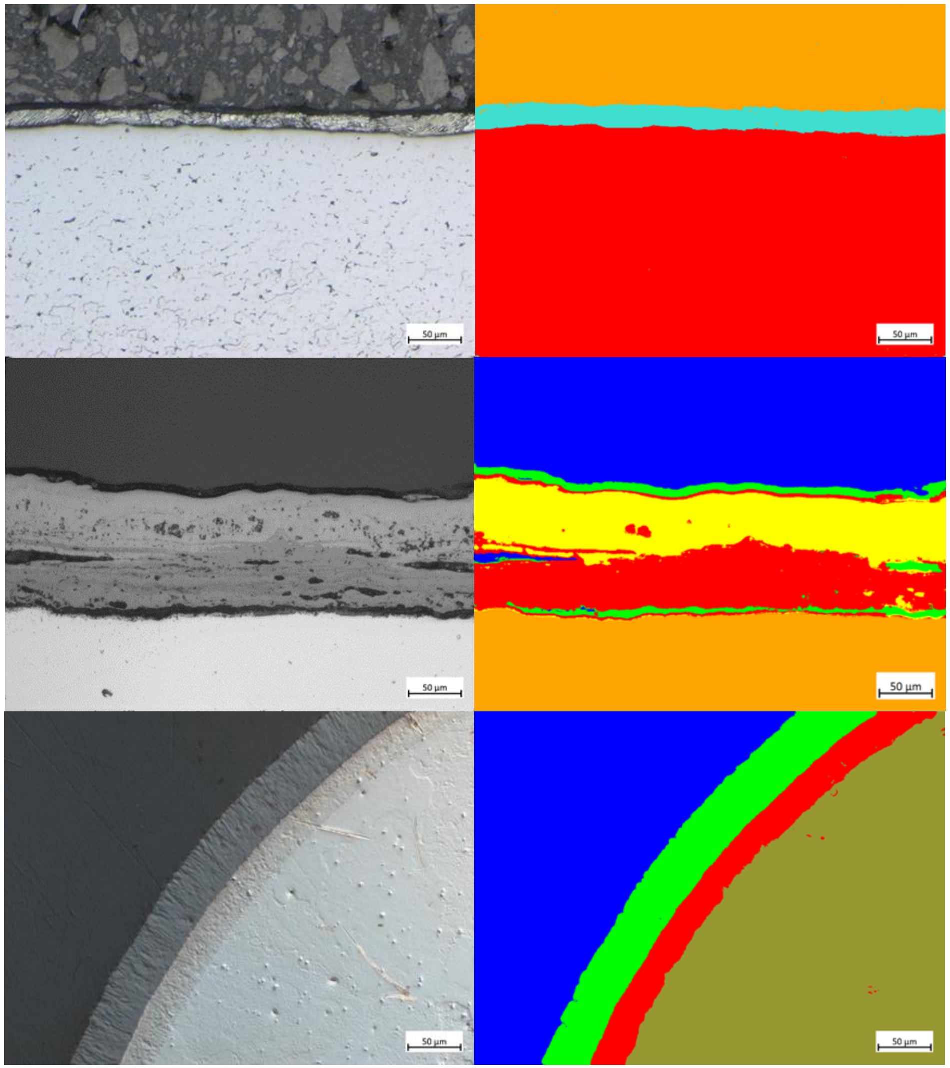


ZEISS ZEN Intellesis segmentation of metallographic samples.

- (top left) Nickel Alloy 600 after metallographic preparation and electro etching. Brightfield imaging on a ZEISS Axio Imager Z2.m.
- (top right) ZEISS ZEN Intellesis segmentation of this image, showing grains in red and grain boundaries in green.
- (bottom left) Zirconia in the as-received condition, secondary electron imaging at 1kV at 30Pa in a ZEISS Sigma 300 VP.
- (bottom right) ZEISS ZEN Intellesis segmentation of this image, showing grains in blue, grain boundaries in red, pores in green.

Applications:
Light and electron microscopy

Machine learning

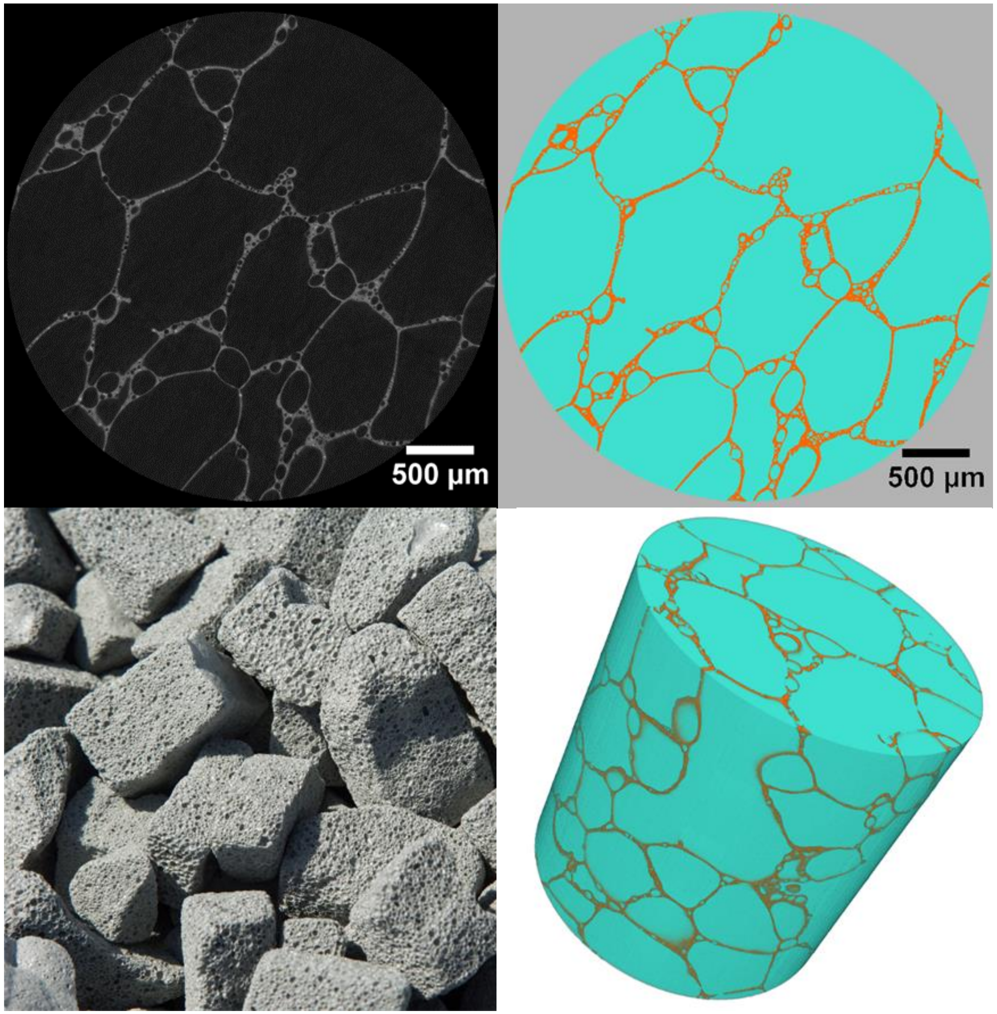


ZEISS ZEN Intellesis segmentation of coating cross-sections.

- Each colour on the segmented image represents a different coating layer.
- Left: Galvanised steel (bright field)
- Middle: High temperature corrosion scale on 9% chromium steel (bright field)
- Right: Thermal spray coating, taken using C-DIC contrast

Applications: Light and electron microscopy

Machine learning



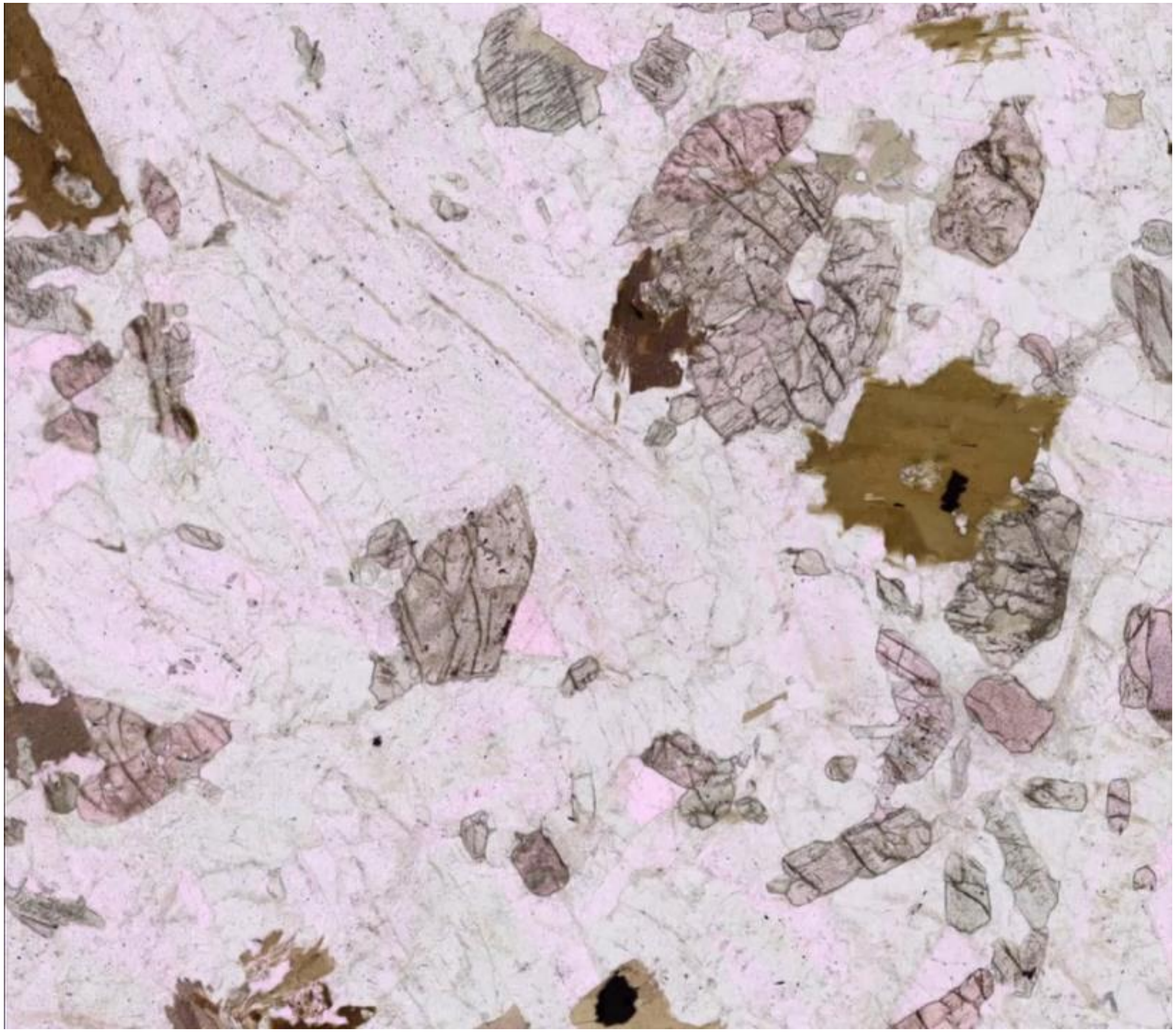
ZEISS ZEN Intellesis segmentation of a foam glass

- Virtual 2D slide of X-ray micrograph data set.
- Segmented microstructure showing pores in blue, glass walls in red.
- (c) Image of a typical foam glass structure.
- (d) 3D model of the foam glass, using segmentation results.

Sample courtesy of Martin Bonderup Østergaard, Dr. Rasmus R. Petersen and Prof. Yuanzheng Yue from Aalborg University, and Dr. Jakob König from Jozef Stefan Institute

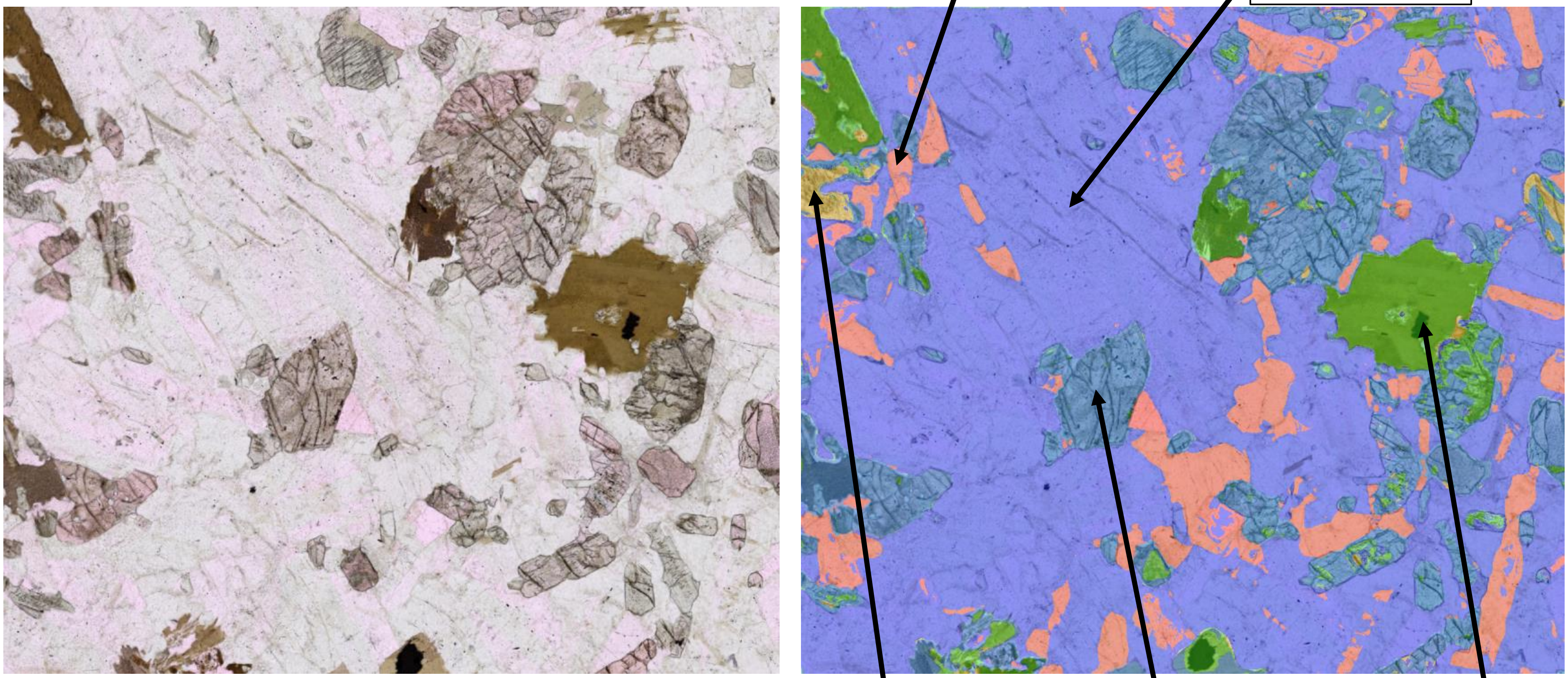
Applications: Multipolarised light microscopy

Machine learning

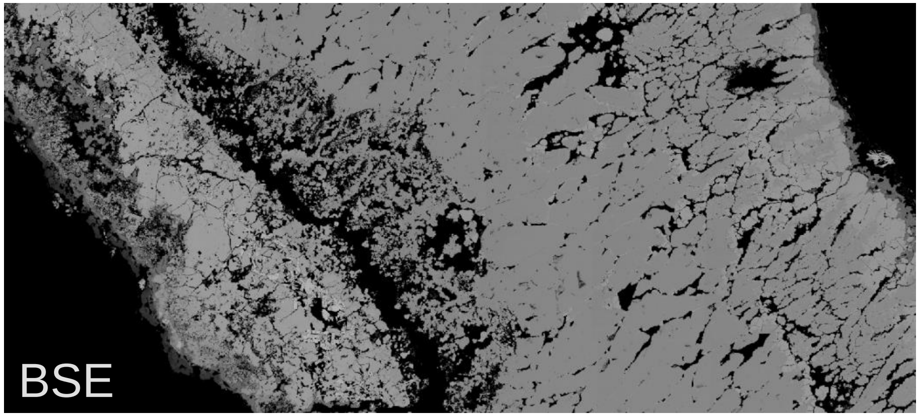


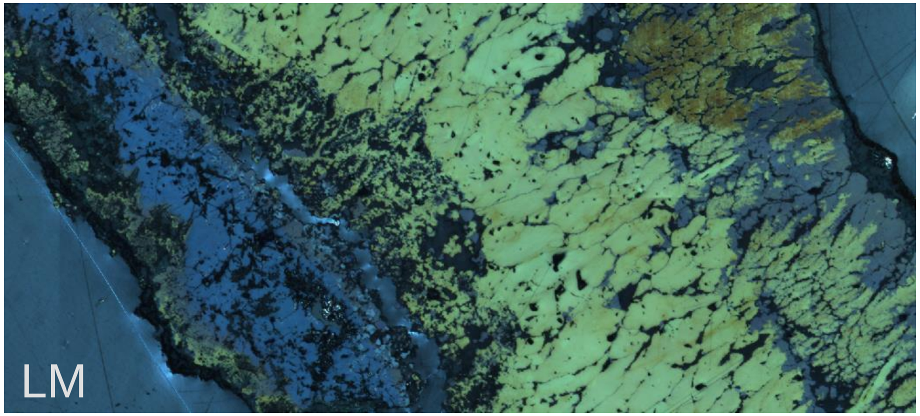
Each individual image can be considered a single channel

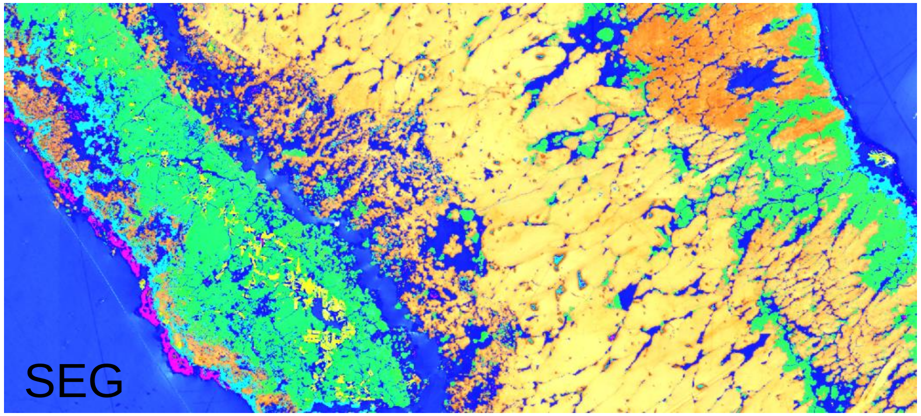
Applications: Multipolarised light microscopy
Segment structures not visible in one channel alone



Correlative Data & ZEN Intellesis



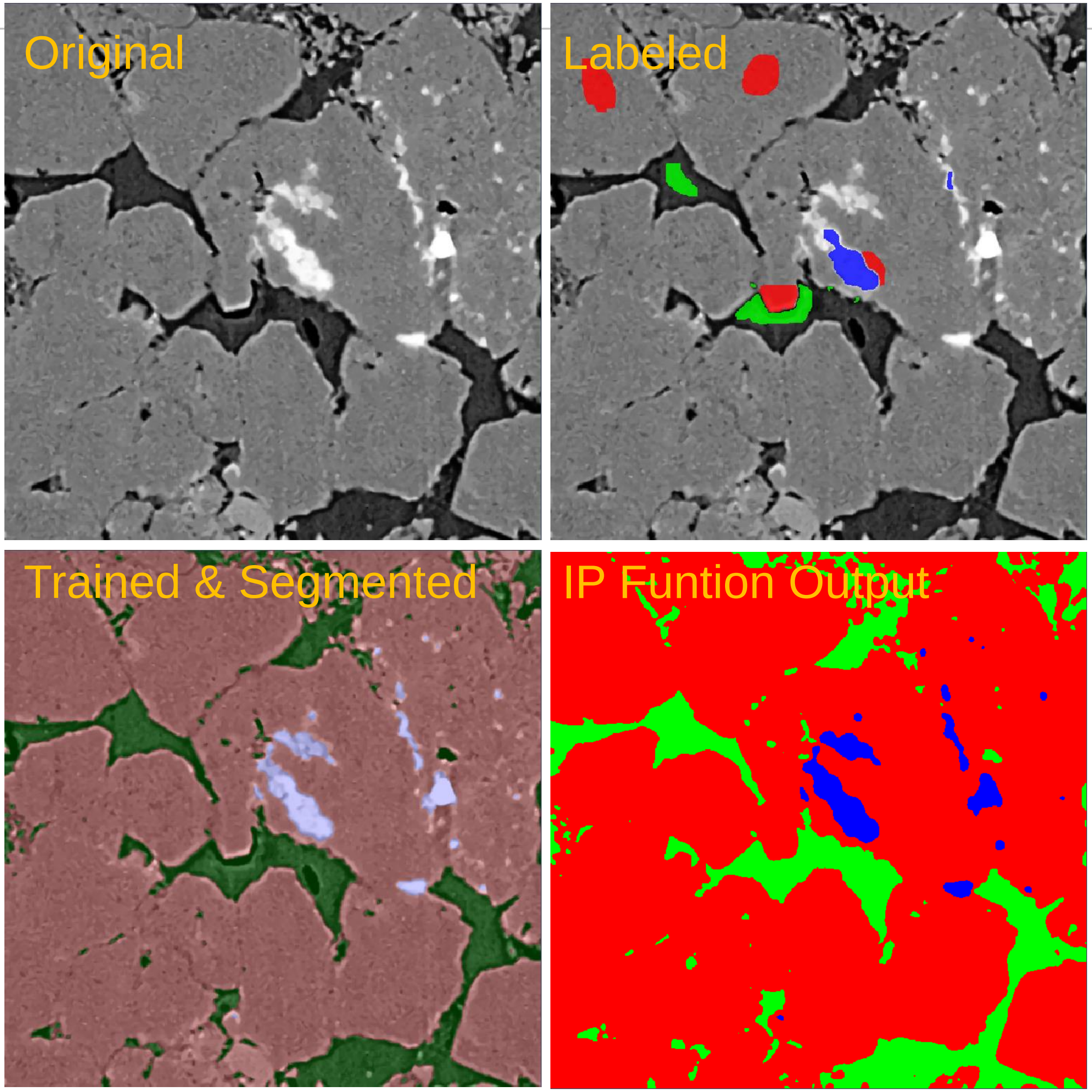




- Acquire BSE & LM data
- Correlate
- Use Correlative Data in Intellesis
- Train & Segment

XRM Sandstone – Segmentation of 3D dataset

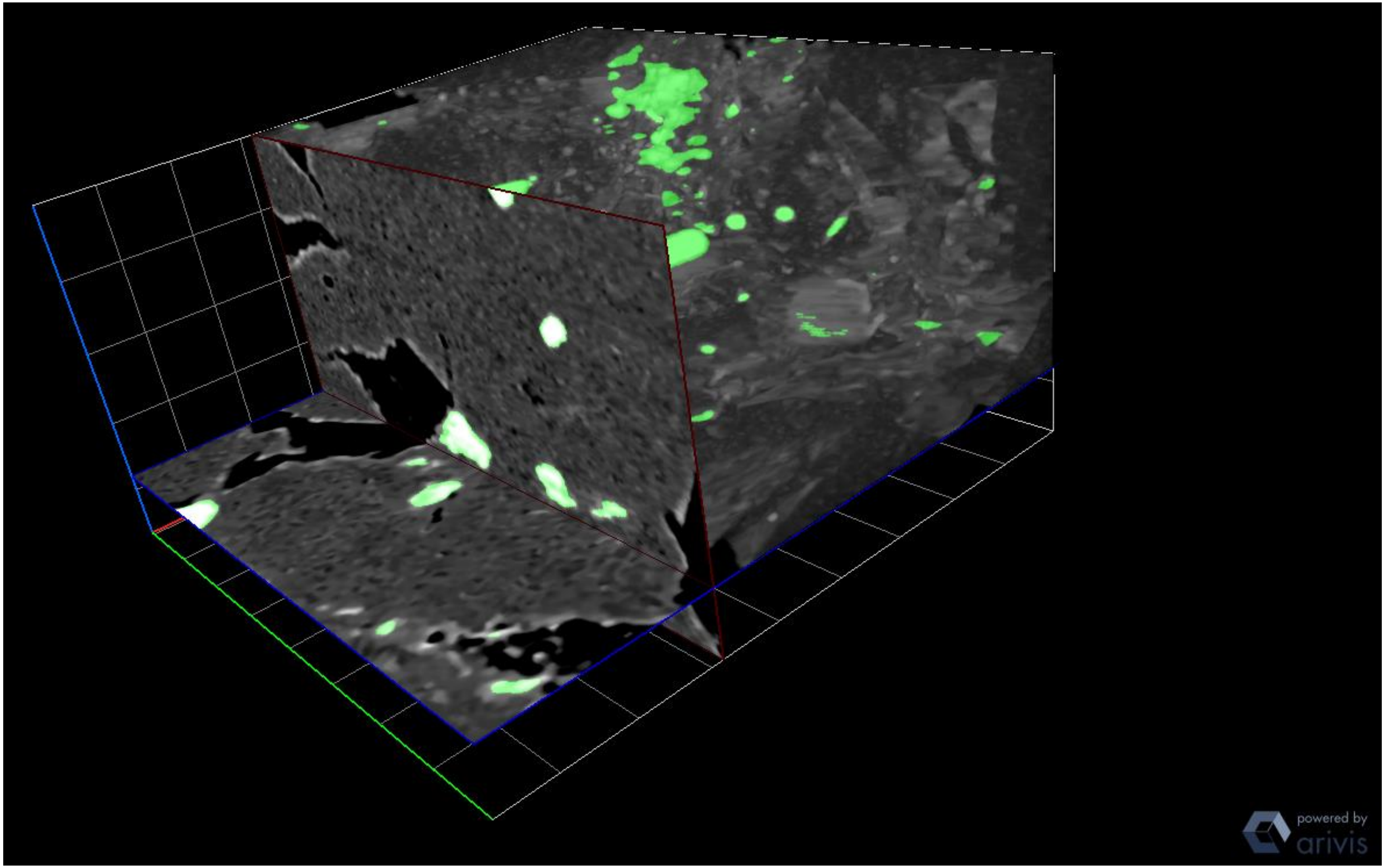
Machine learning



3D Segmentation Example: XRM – Sandstone inside 3DxL

Machine learning

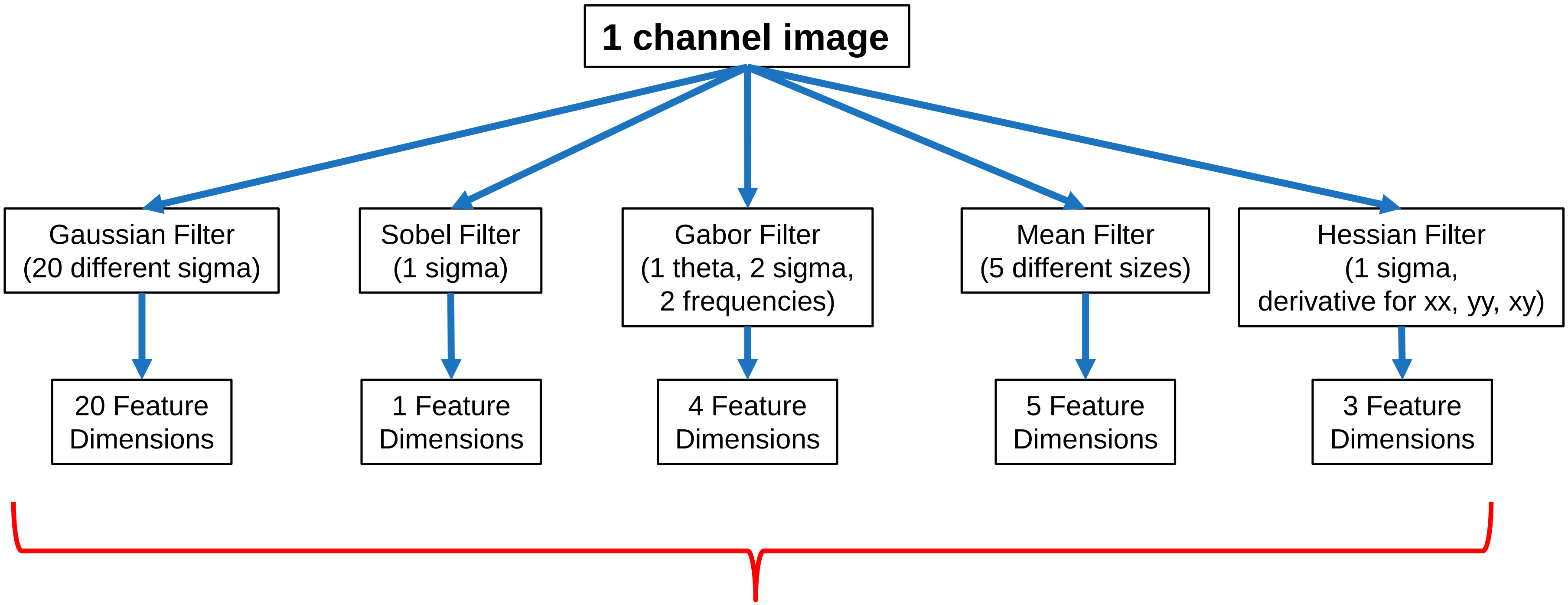
ZEISS



Intellesis – Basis Features 33

Machine learning

- For calculating the features various filters with various filter sizes and parameters are applied to the region around this pixel (2D Kernels)
- Results are concatenated yielding the final feature vector describing the pixel.



33 pieces of information per image pixel
= feature vector with 33 dimensions

Feature Extractor Information

Machine learning

ZEISS

<u>**Basic Features 33** (in ZEN 2.6 still called Default)</u>

The user labels a number of pixels in the image as part of a Class. Classes could be anything that you are interested in – structures, holes, individual cells/grains etc. Multiple different Classes can be present in one image.

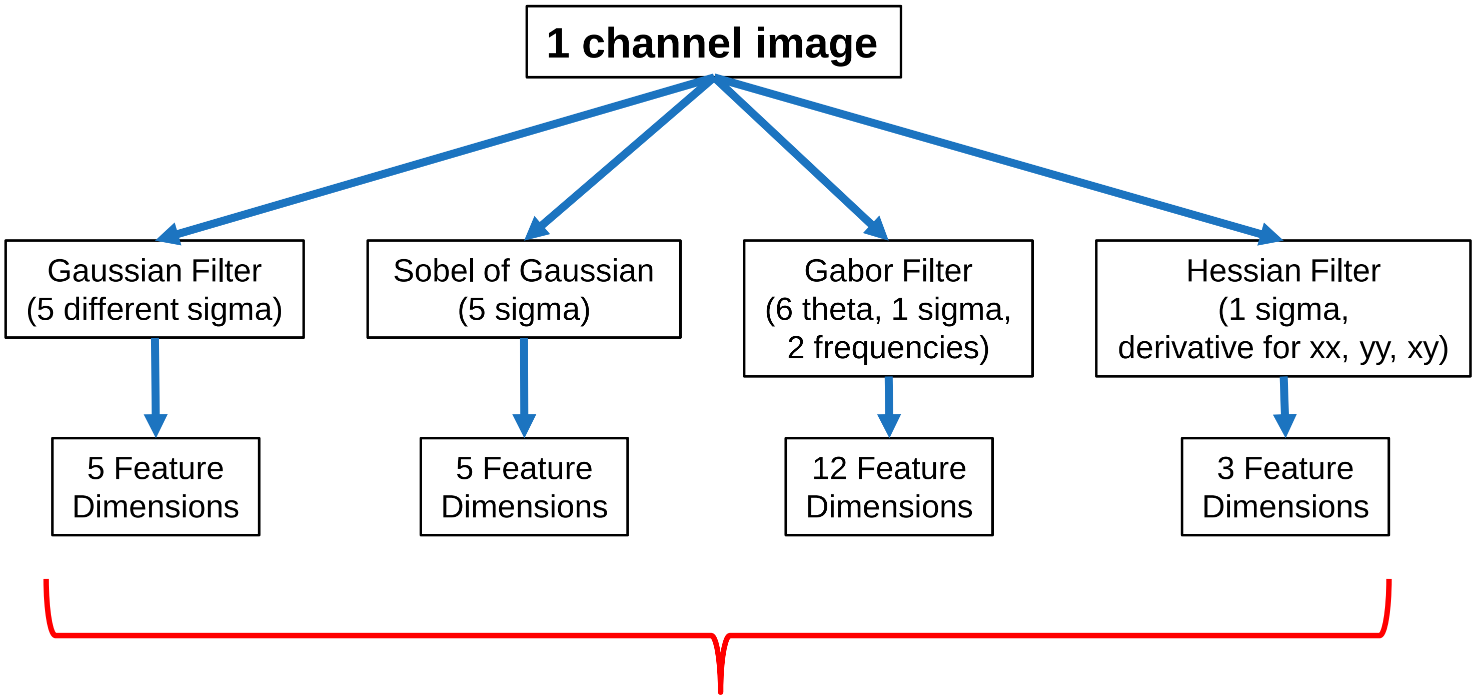
33 image analysis filters are applied to every pixel in a Class plus the region around that pixel. These filters take into account many things – including but not limited to grayscale value (or colour) of the pixel, grayscale value (or colour) of the neighbourhood at distances ranging from 1 pixel away to 20 pixels away, local edge detection and local texture in multiple directions.

Then, using a random forest classifier approach (a machine learning method for classification, averaging and prediction using multiple decision trees) every single other pixel in the image is evaluated against the feature vectors for that Class. Intellesis can therefore judge which pixels fit into that Class, and how confident it can be about this decision. The main output is an image with all pixels divided up into Classes, along with a confidence map, showing how confident Intellesis is about its decision on each pixel.

The Default method is simple but powerful and is recommended in the first instance. It is likely to work on most images.

Intellesis – Basis Features 25

- For calculating the features various filters with various filter sizes and parameters are applied to the region around this pixel (2D Kernels)
- Results are concatenated yielding the final feature vector describing the pixel.



25 pieces of information per image pixel
= feature vector with 25 dimensions

Feature Extractor Information

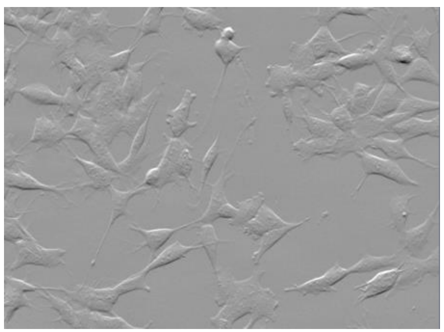
Basic Features 25 (in ZEN 2.6 still called Default 2)

Basic Features 25 uses the same process as for the Basic Features 33, except it uses fewer and different filters (25) and is therefore slightly faster.

This method is recommended for images which have clear and easy contrasts between Classes, and where faster processing is needed (very large numbers of images).

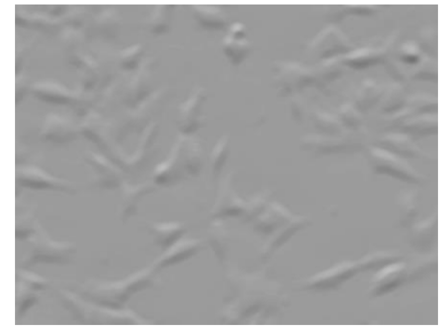
Feature Extractor Information

Machine learning

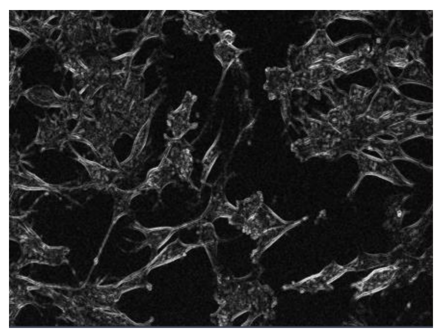


Original brightfield image

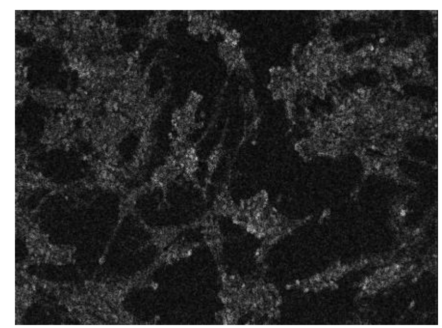
Gaussian Filter (Sigma 10)



Mean Filter (10 Pixels)



Sobel Filter (Clip)



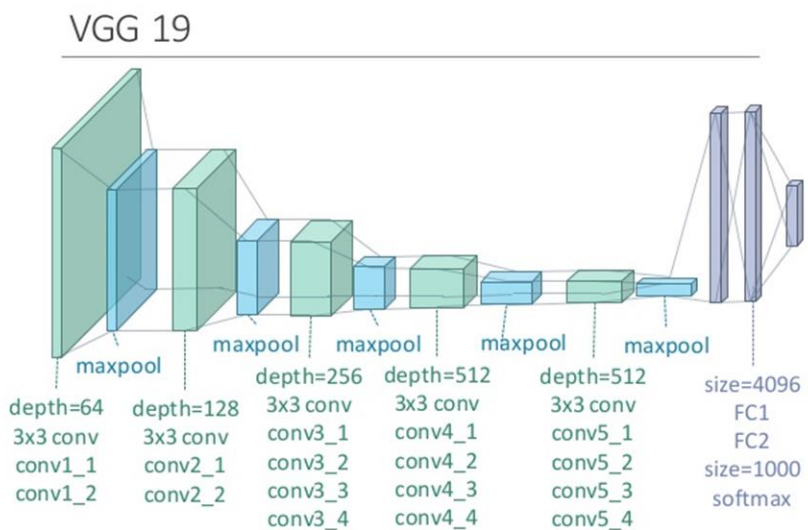
Hessian Filter (smallest)

…

Intellesis – Deep Feature Extraction from CNN Layers

Machine learning

- entire image as input for pre-trained VGG19 network
- take the output from an intermediate layer of that network as feature vector, e.g. output from layer 3 was processed by preceding layers 1 and 2
- Deep Features 50 (DF Layer 2 Reduced) : Feature dimension = 50, reduced by random transformation
- Deep Features 64 (DF Layer 1) : Feature dimension = 64
- Deep Features 70 (DF Layer 3 Reduced) : Feature dimension = 70, reduced by random transformation
- Deep Features 128 (DF Layer 2 All) : Feature dimension = 128
- Deep Features 256 (DF Layer 3 All) : Feature dimension = 256



Deep Feature - Description

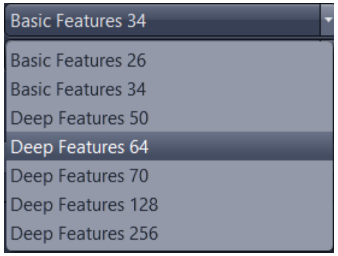
Machine learning

ZEISS

Deep Features 50 - 256

Deep Features 50 - 256 use the entire image as an input for the pre-trained network (VGG19). Each layer of this network becomes the input for the next layer as a feature vector, doubling the number of pieces of information per pixel. e.g. the output from Deep Features 3 will have been processed by the layers in Deep Features 1 and 2. To simplify this logic the Deep Features are just ordered according to the numbers of features used.

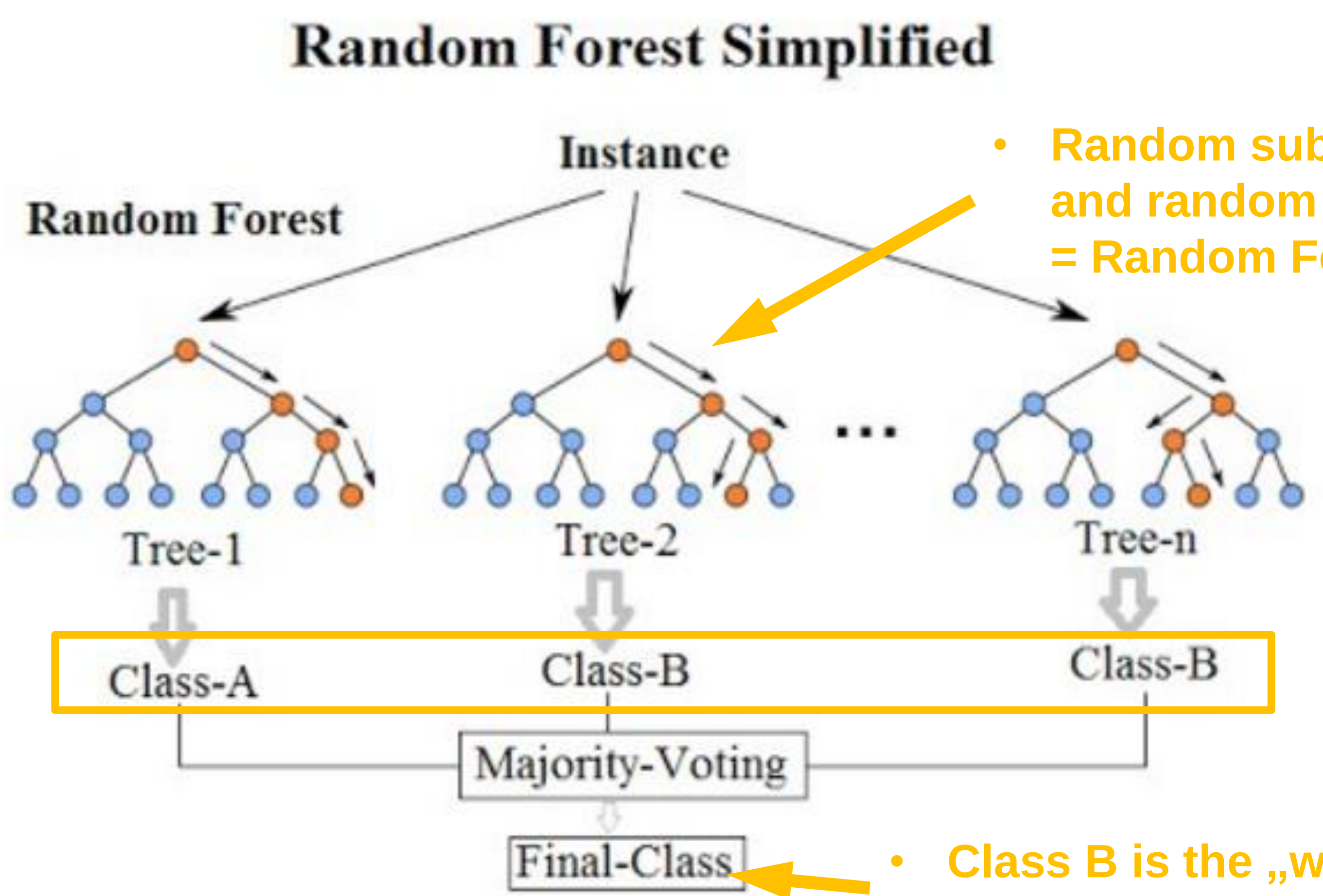
These methods take considerably more processing time (due to the complexity of the analysis) but can extract features and segment images that would not normally be possible. If the processing system is using a Nvidia GPU it will speed-up the process due to usage of the TensorFlow-GPU module.



These methods are recommended where Default methods alone cannot segment the image – e.g. subtle changes in texture - but they will take longer to run due to the added feature dimensions. If the processing time is too long, then use the Reduced setting. This reduces the number of feature dimensions by random transformation, speeding up processing.

Random Forest Classifier Approach

Machine learning



- Random subset of features and random decision trees = Random Forrest

- Class B is the „winner“
- 2 out of 3 vots = 66% confidence

https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d

Random Forest Classifier approach

Machine learning

- Each decision tree in the forest considers a random subset of features when forming “questions” and only has access to a random set of the training data

- This increases diversity in the forest leading to more robust overall predictions and the name ‘**Random Forest**.’

- Prediction = average of all the individual decision tree estimates

- Regression task → predicting a continuous value of …

- Classification task → discrete class labels such as A or B

  - Random Forest will take a majority vote for the predicted class

  - ***“There can be only one class!”***

https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d

scikit-learn - Random Forest

Postprocessing – Conditional Random Fields (CRF)

CRF tries to create smoother and shaper borders between objects by re-classifying pixels based on confidence levels in their neighborhood.

- Multi-class image segmentation and labeling use conditional random fields defined over pixels or image regions.
- fully connected CRFs models are defined on the complete set of pixels in an image
- CRF used a highly efficient approximate inference algorithm for fully connected CRF models in which the pairwise edge potentials are defined by a linear combination of Gaussian kernels
- the dense connectivity at the pixel level substantially improves segmentation and labeling accuracy

Conditional random fields (CRFs) are a class of statistical modeling method often applied in pattern recognition and machine learning and used for structured prediction. CRFs fall into the sequence modeling family. Whereas a discrete classifier predicts a label for a single sample without considering "neighboring" samples, a CRF can take context into account; e.g., the linear chain CRF (which is popular in natural language processing) predicts sequences of labels for sequences of input samples.

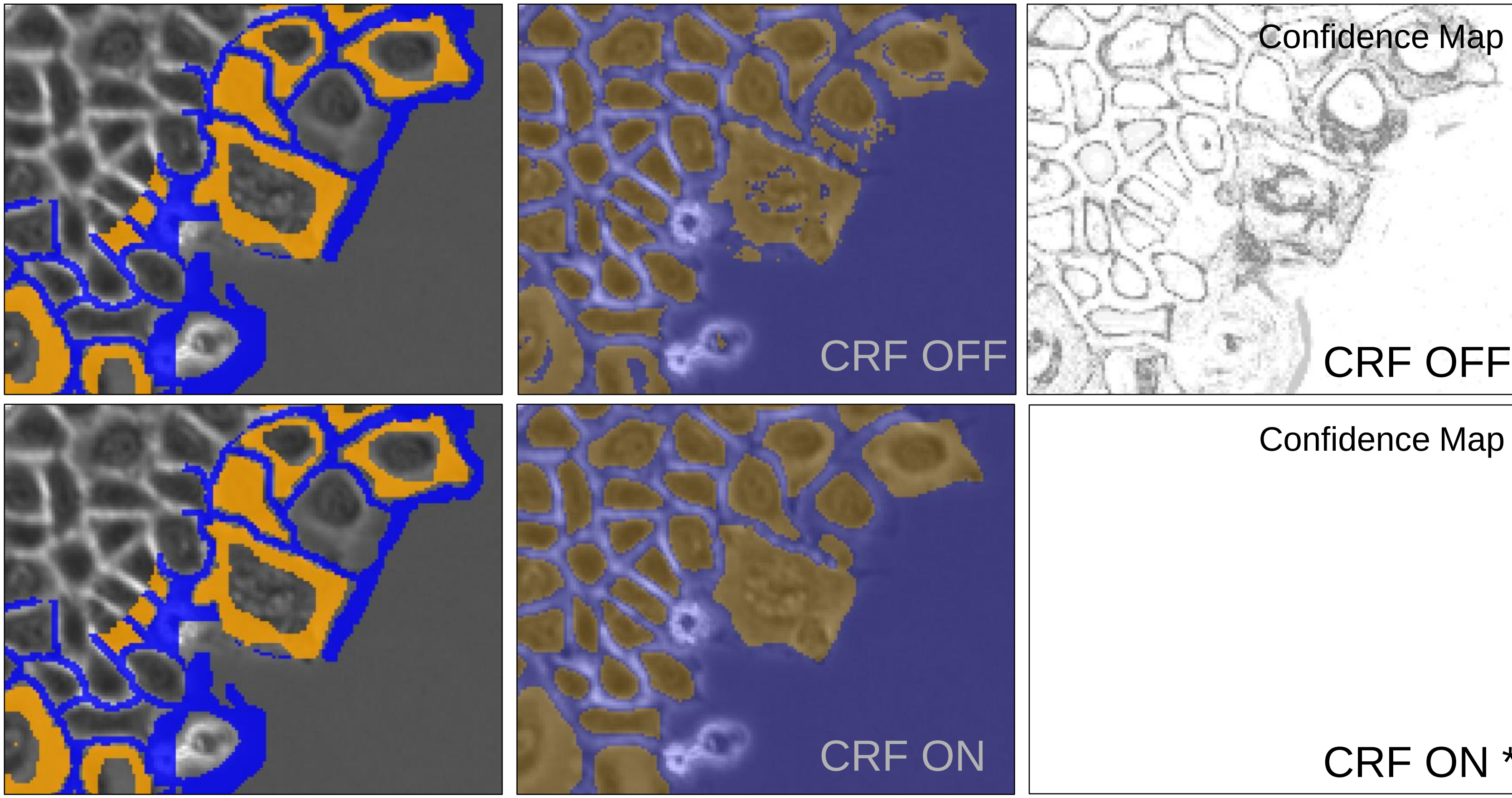
Further readings and additional information:

https://en.wikipedia.org/wiki/Conditional_random_field

Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials

Postprocessing – Conditional Random Fields (CRF)

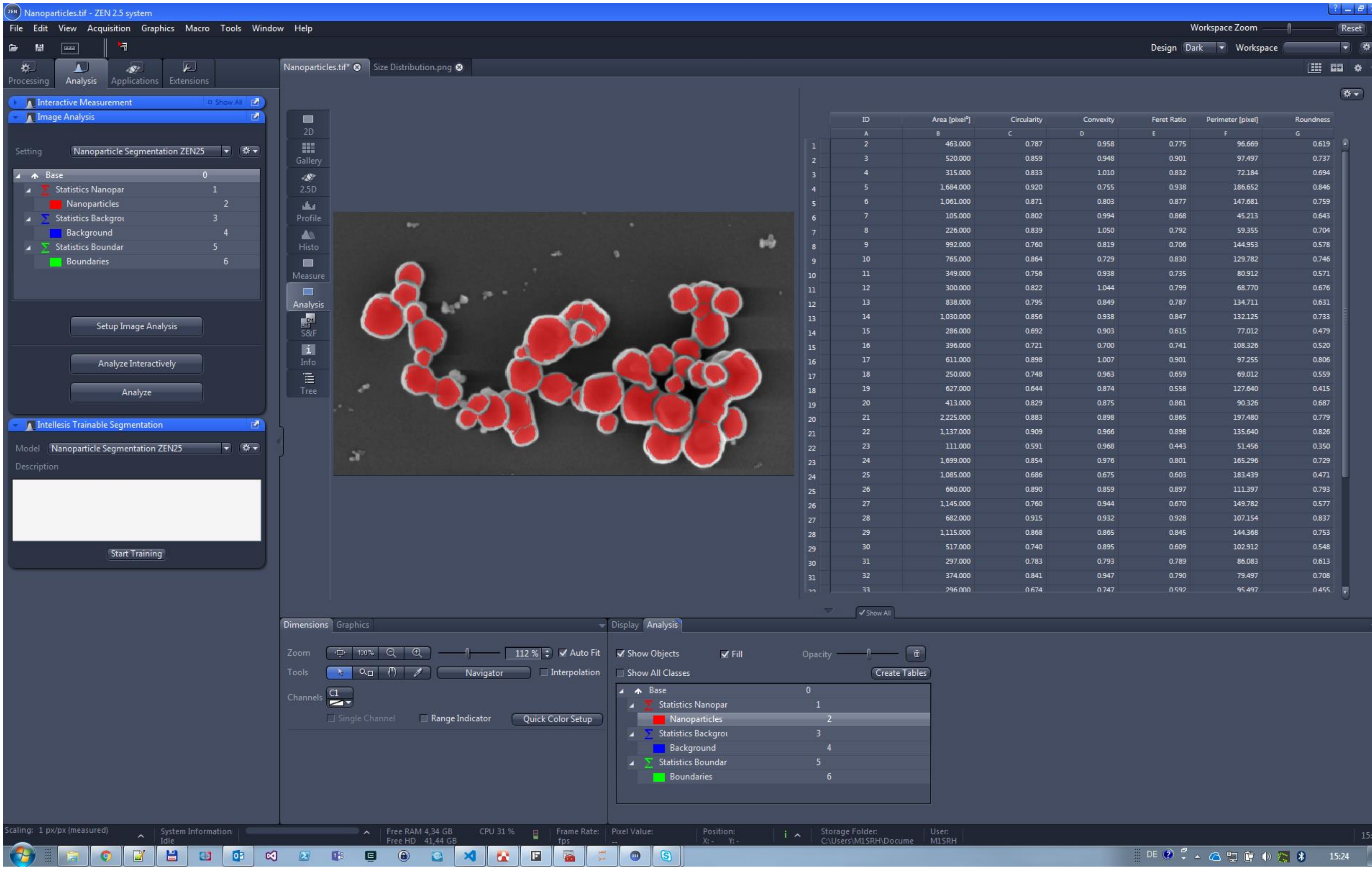
Machine learning



* - since the map after CRF does not reflect the probabilities from the Random Forrest anymore a matrix of one will be returned (from ZEN 3.0 or better) to prevent misuse of the probability threshold sliders

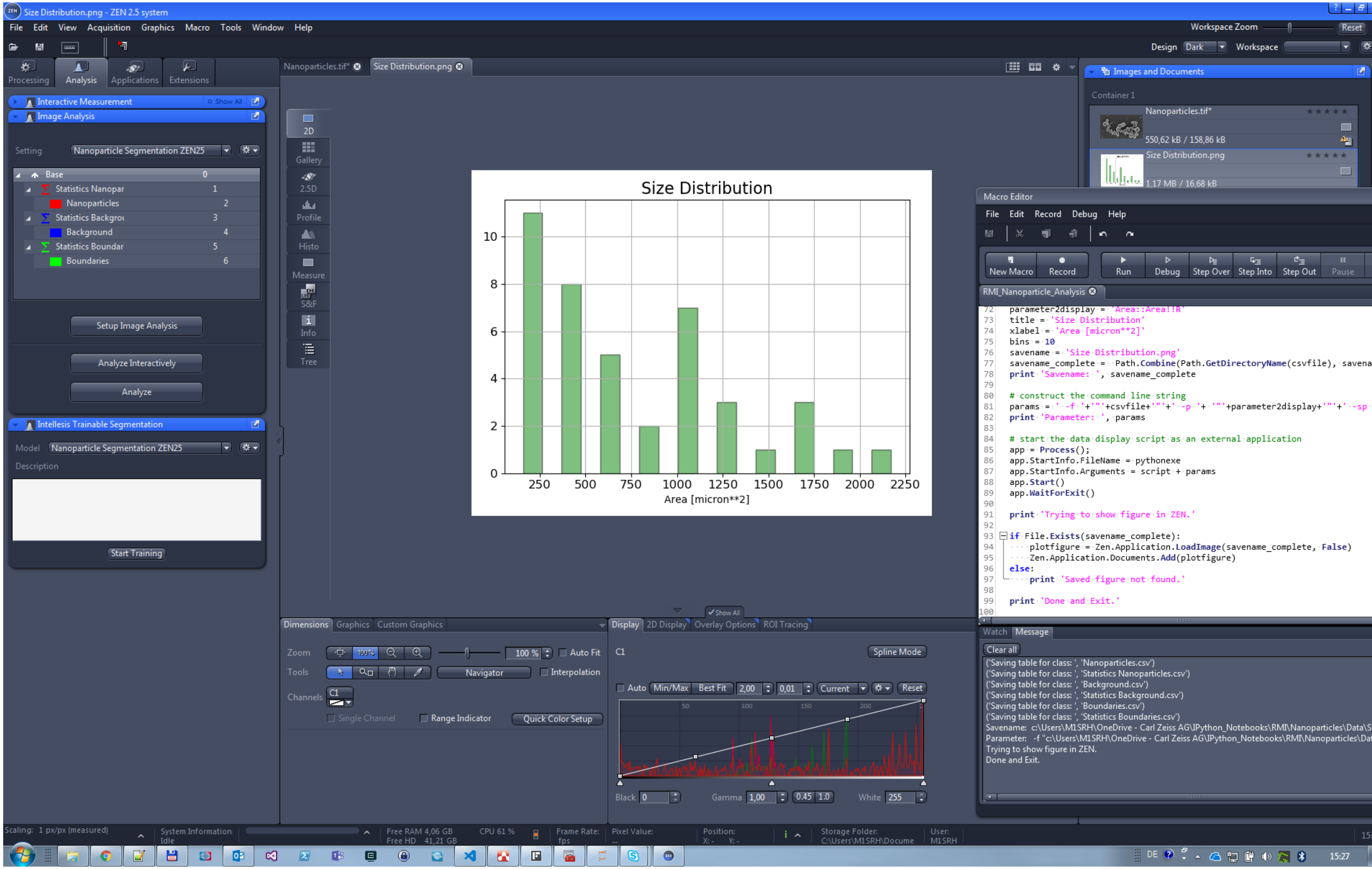
Complete Automation of Workflow

Machine learning



Complete Automation of Workflow

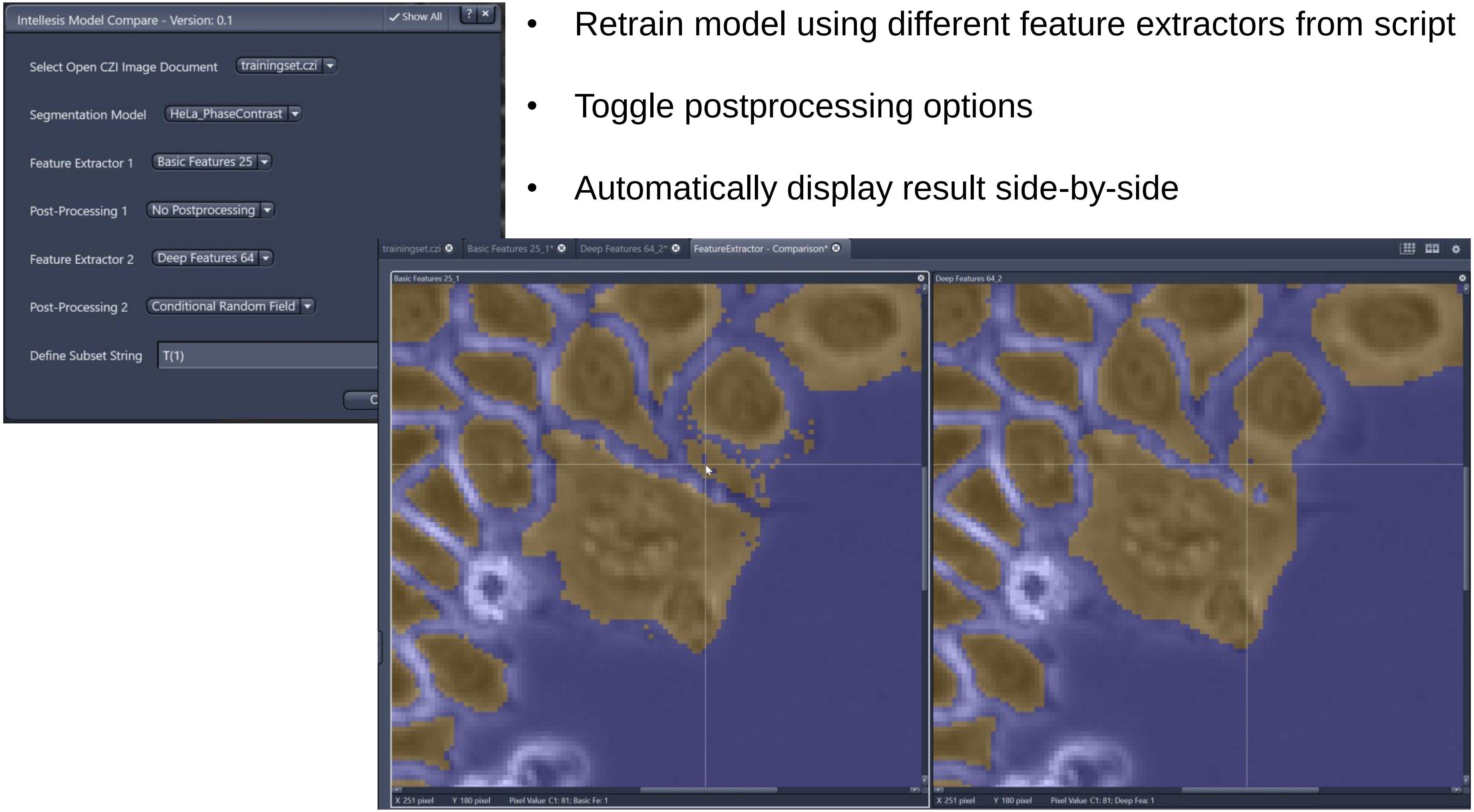
Machine learning



Compare Feature Extractors

Machine learning

- Retrain model using different feature extractors from script
- Toggle postprocessing options
- Automatically display result side-by-side

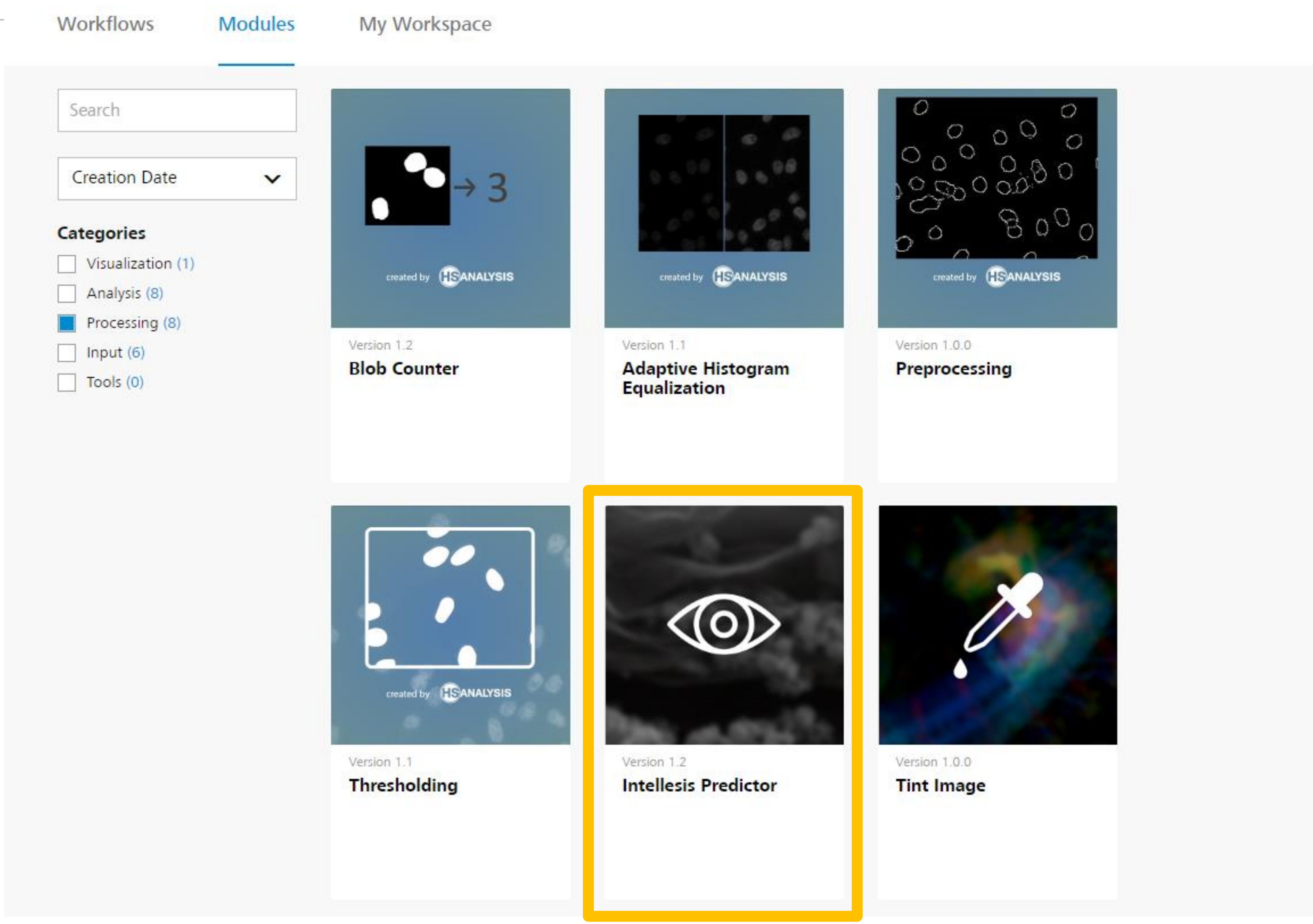


Machine learning

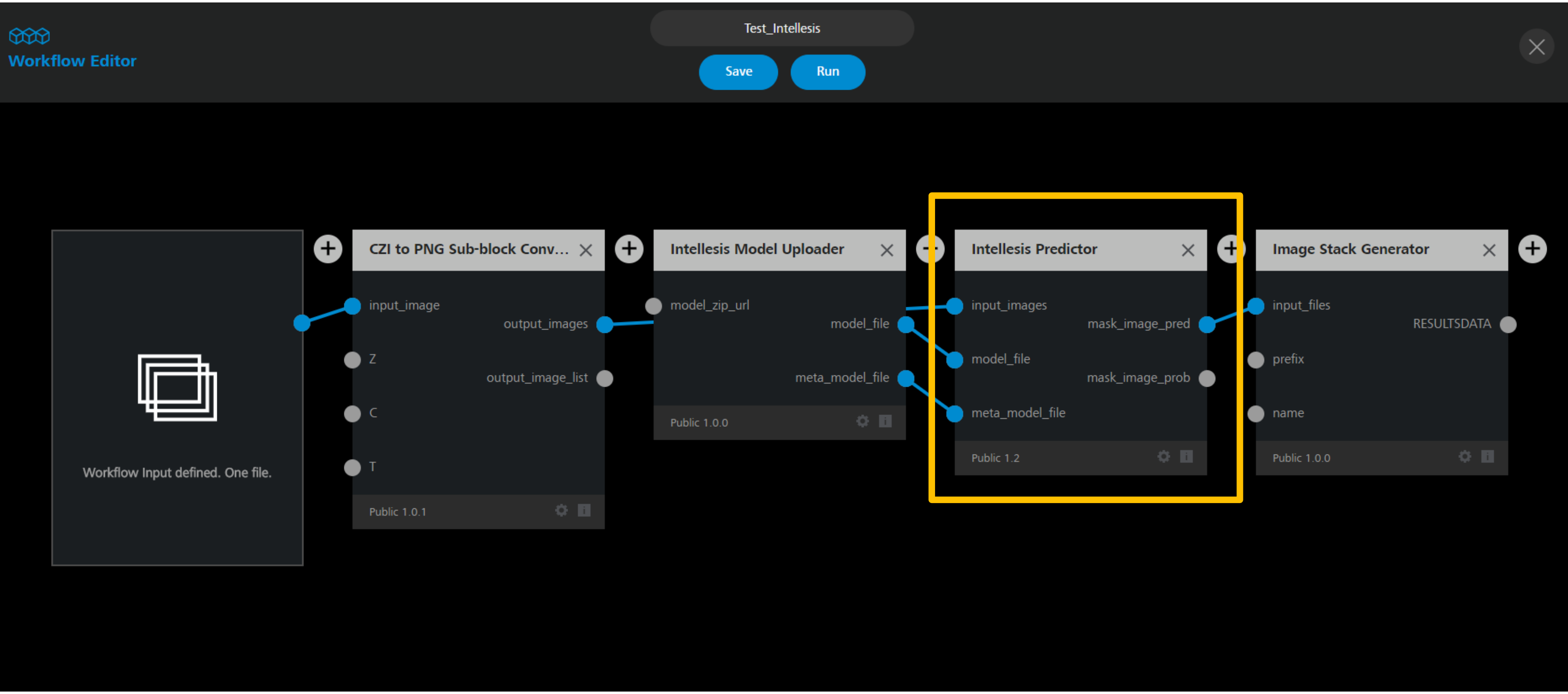
https://www.zeiss.com/microscopy/int/website/landingpages/zen-intellesis.html

Get the trial licence and try it out!

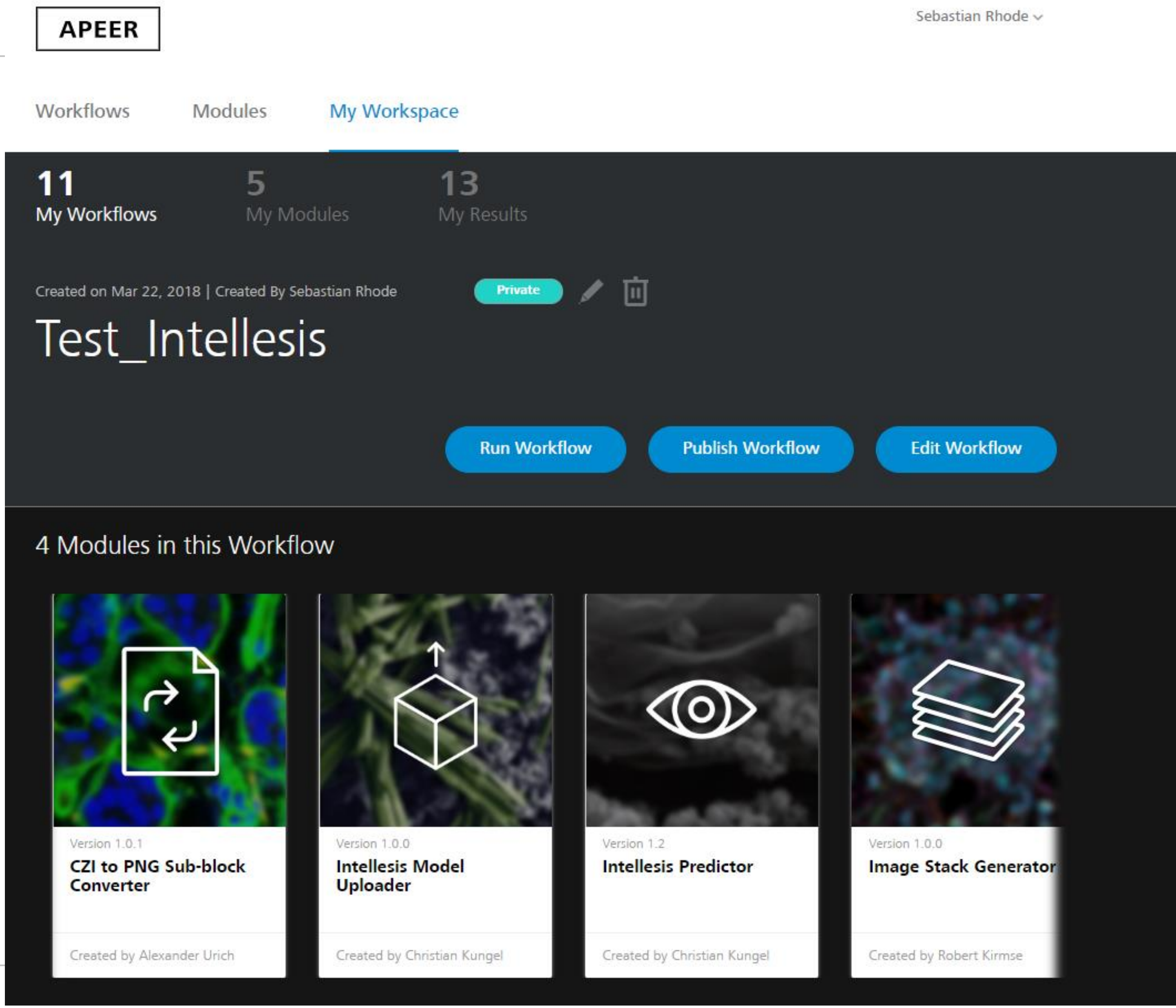
Integration in APEER platform



Intellesis - Integration in APEER platform



Intellesis - Integration in APEER platform



We make it visible.