

ZEN Intellesis Trainable Segmentation



Dr. Sebastian Rhode

Product Owner, Technology Center Software & Digital Solutions, Munich

“I can get decent images from anywhere but I am looking for information”

- Potential customer during her visit to Pleasanton

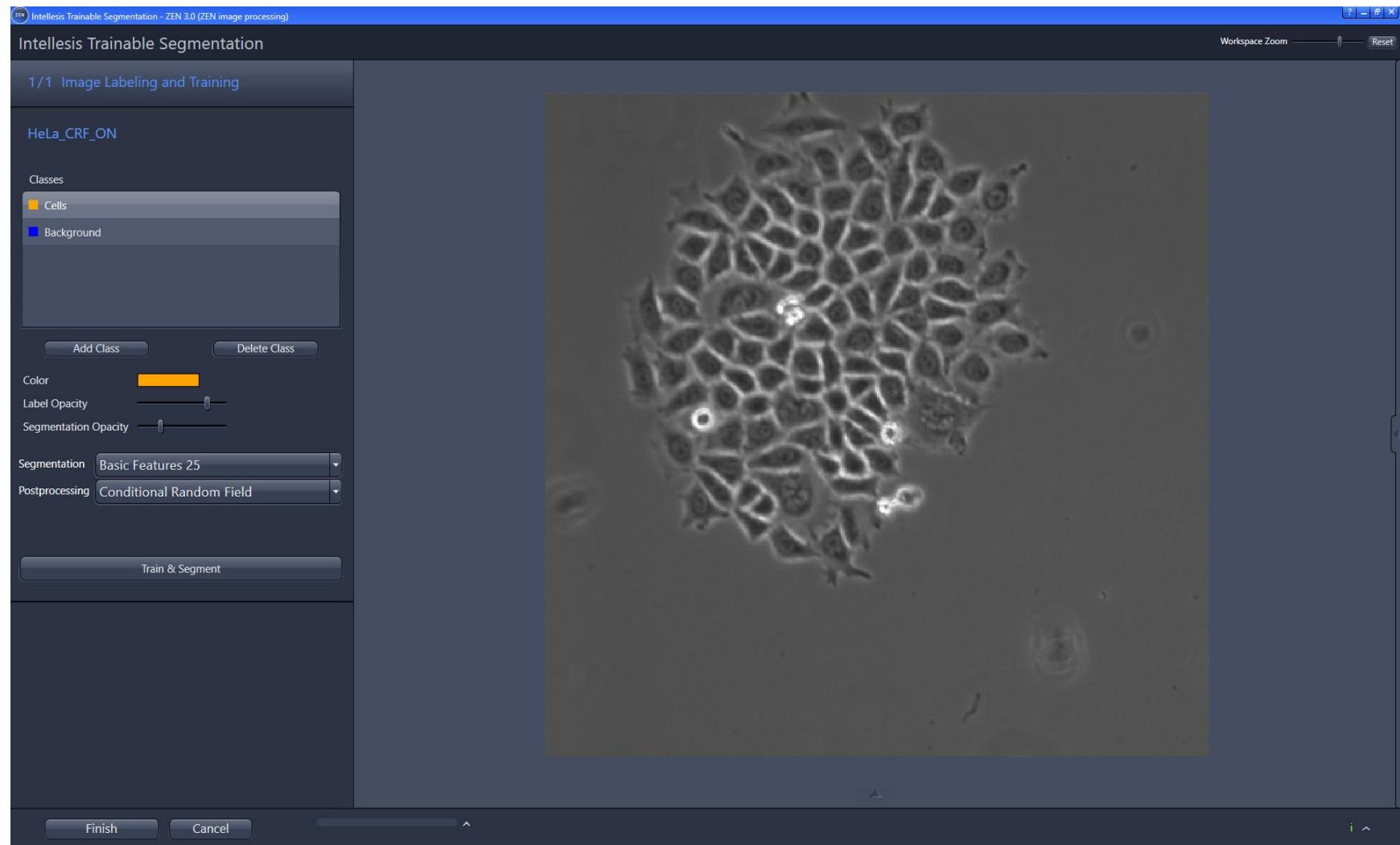
Segmentation is today's biggest problem.

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

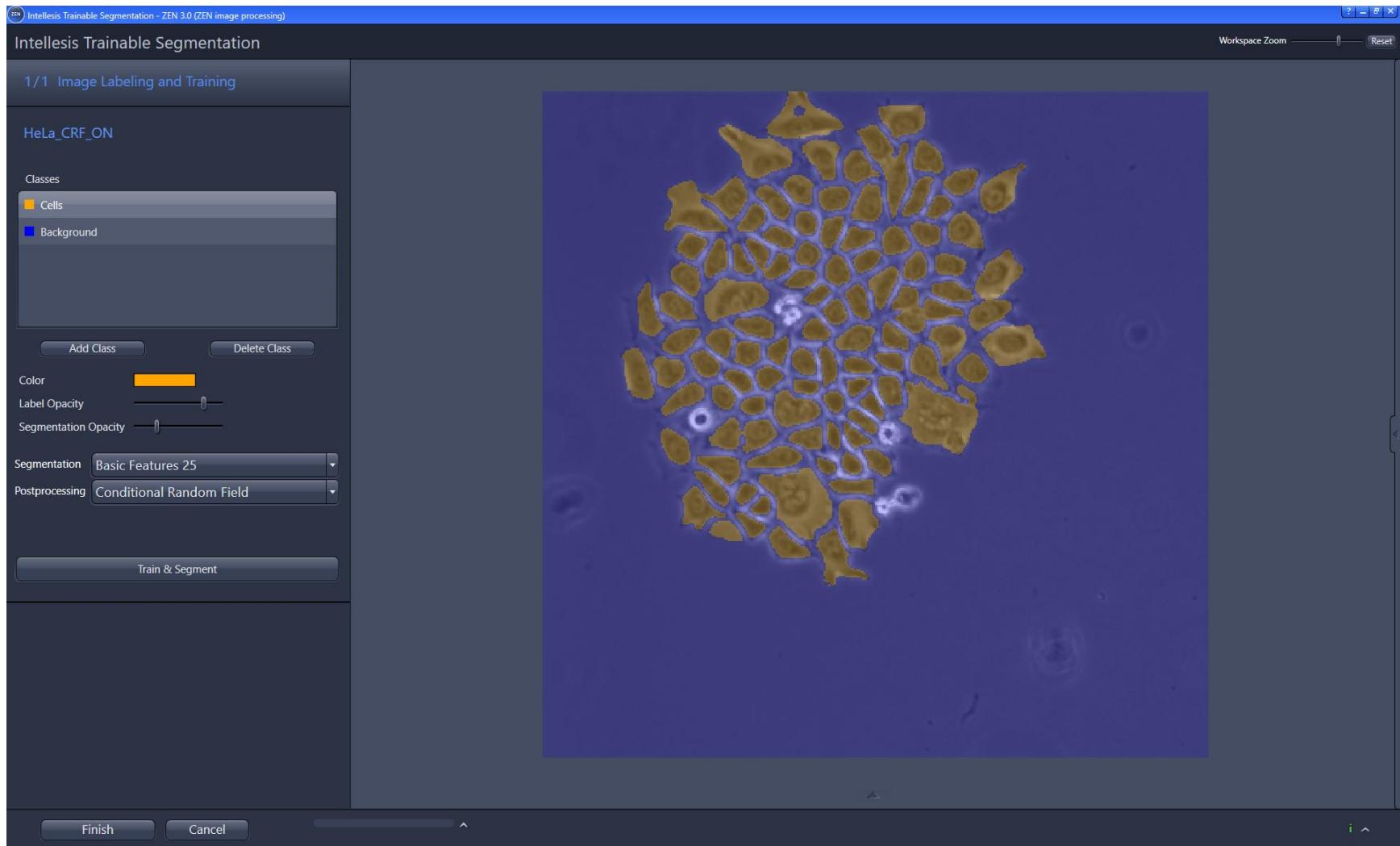
The actual purpose of an image is “to be analyzed”

- 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



Key Points - ZEN Intellesis Trainable Segmentation



- **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



- 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



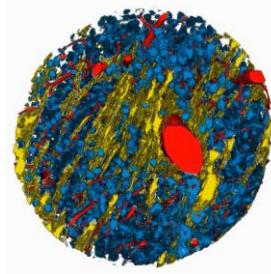
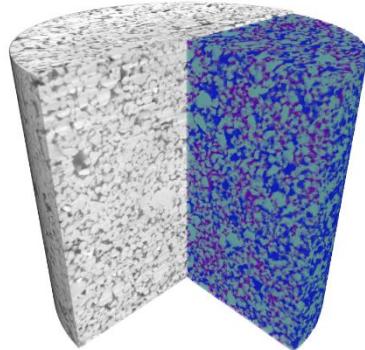
- Machine-Learning Tool for Pixel Classification powered by Python
 - Dask, Scikit-Learn and Tensorflow
- Client / Server Architecture with REST-API
- Engineered Default Feature Sets (CPU)
 - 25 or 33 Features
- Neural Network (vgg19) Layers for Feature Extraction (GPU)
 - 64, 128 (red. 50) or 256 (red. 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 is a **SW module in ZEN Blue and ZEN Core** for image segmentation
- Its 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 **every** pixel of an image.
- Intellesis **extends the 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



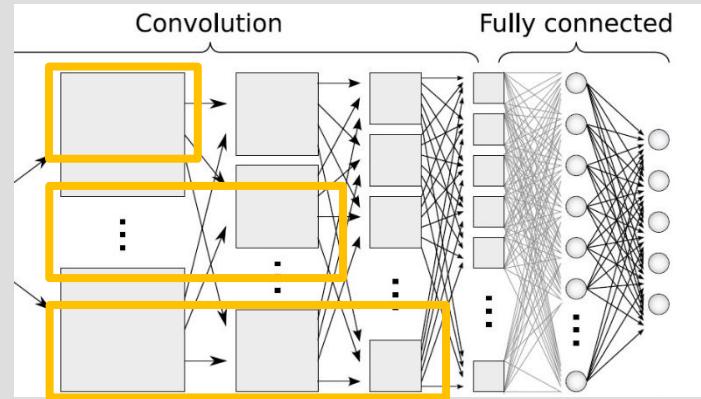
Examples:

- 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

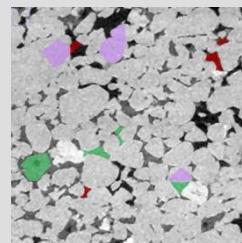
(More data beats clever algorithms)



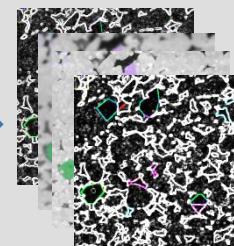
Use **predefined** features

Feature Engineering plus classifier

(Domain knowledge beats more data)



Manual selection of known regions
(Labeled Image)

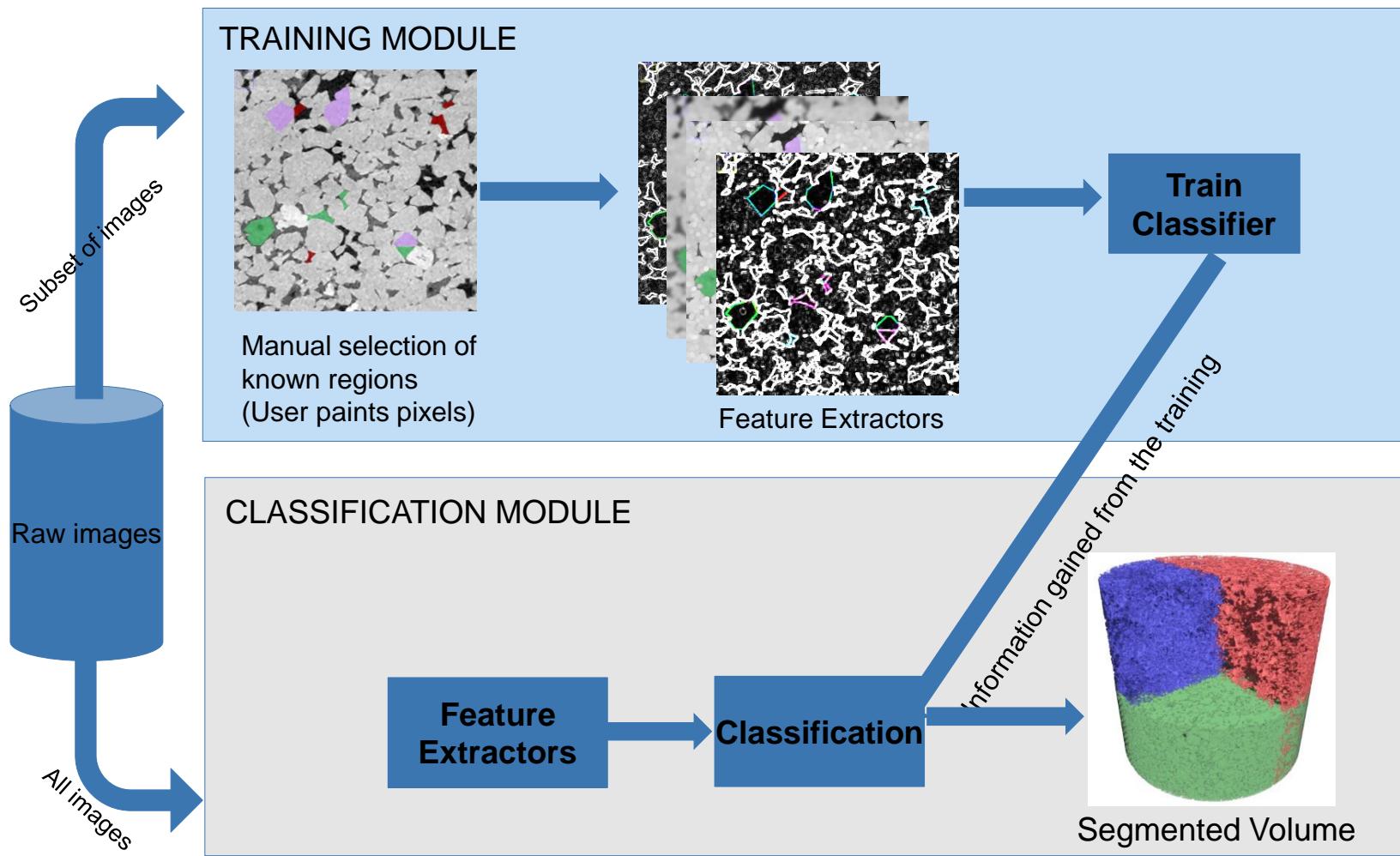


Generation of feature vectors



Segmented Volume

Segmentation workflow

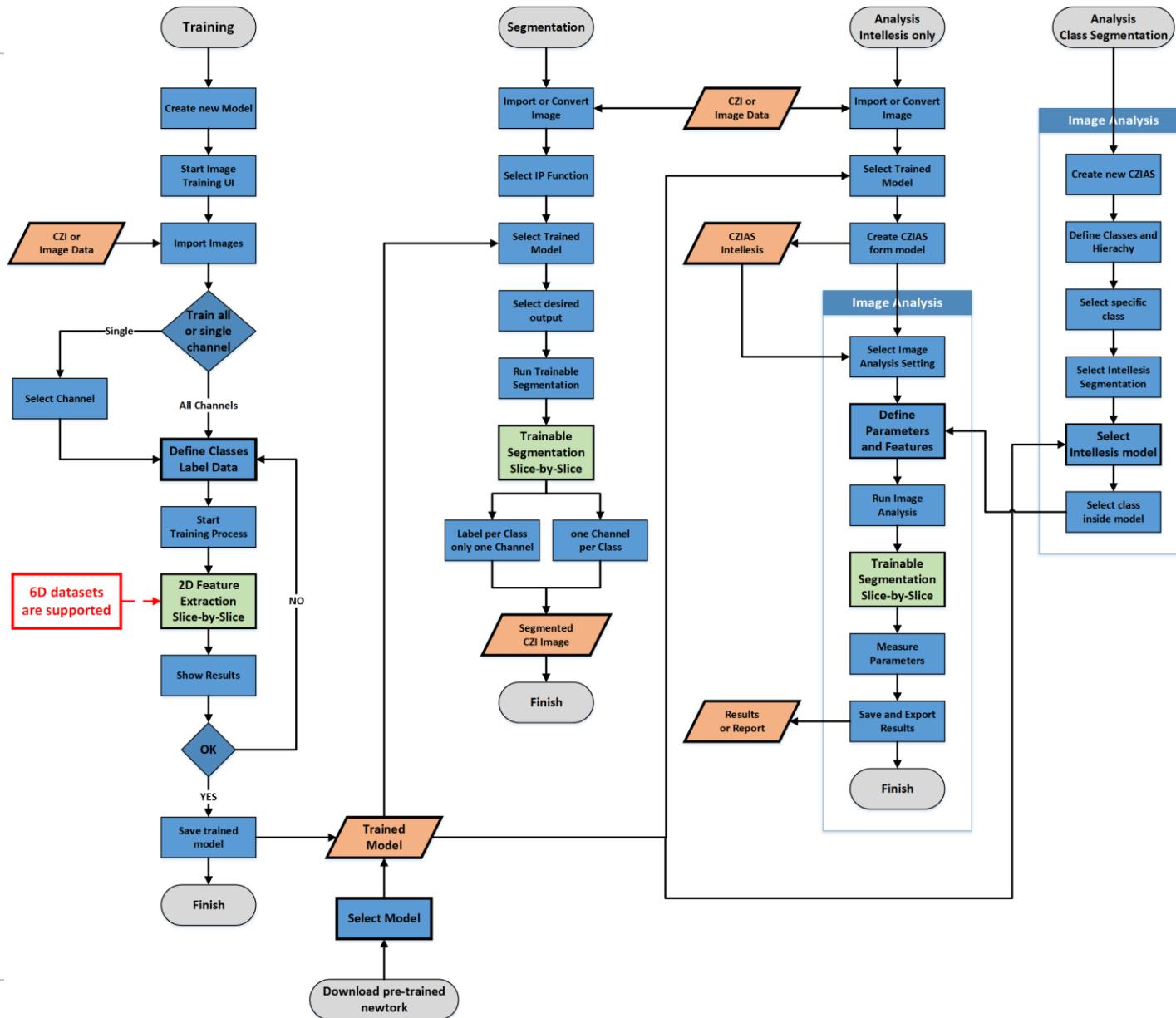


Labeling and Training – Things to know and understand

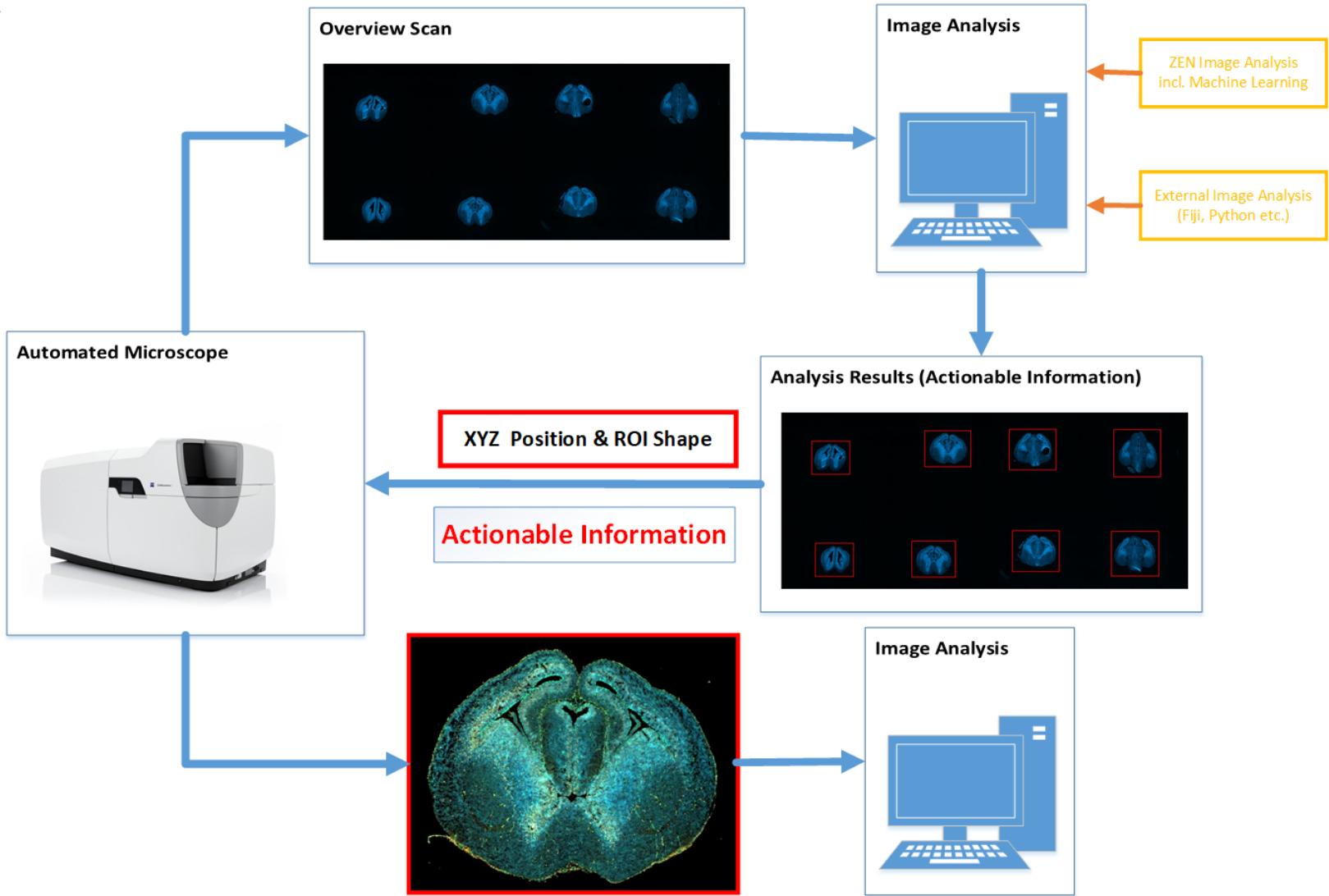


- A label is valid for all channels of an image and the information from all channels will be used to classify that particular pixel (X,Y)
- Label as few pixels as possible and rather do more iterations of training.
- Sometimes it is helpful to more classes in order to get better results for each class.
- Labeling large homogenous areas with many pixels will slow down the training but give not necessarily give better results.
- It is good practice to label “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.

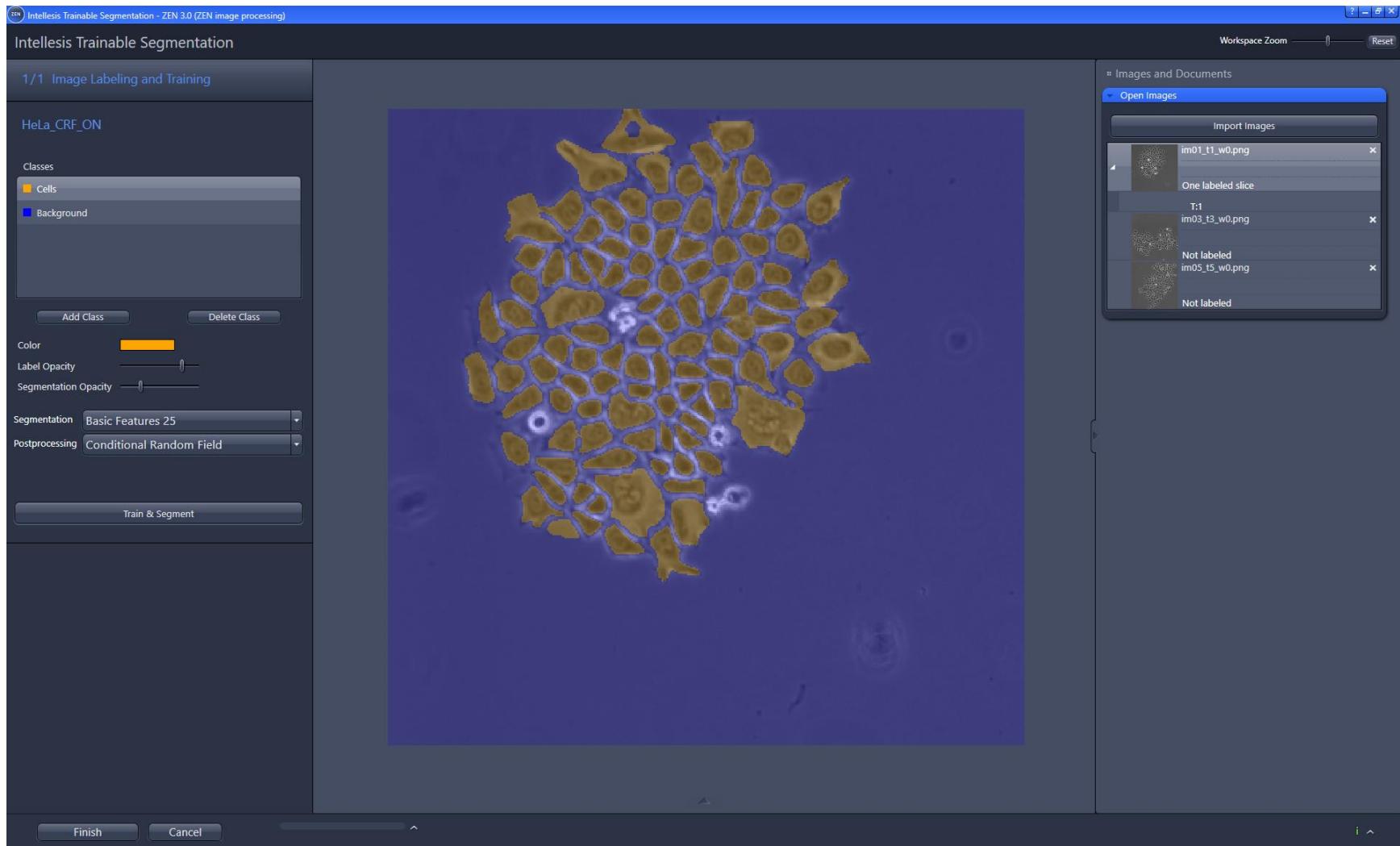
ZEN Intellesis – WorkFlows and Measurement Integration



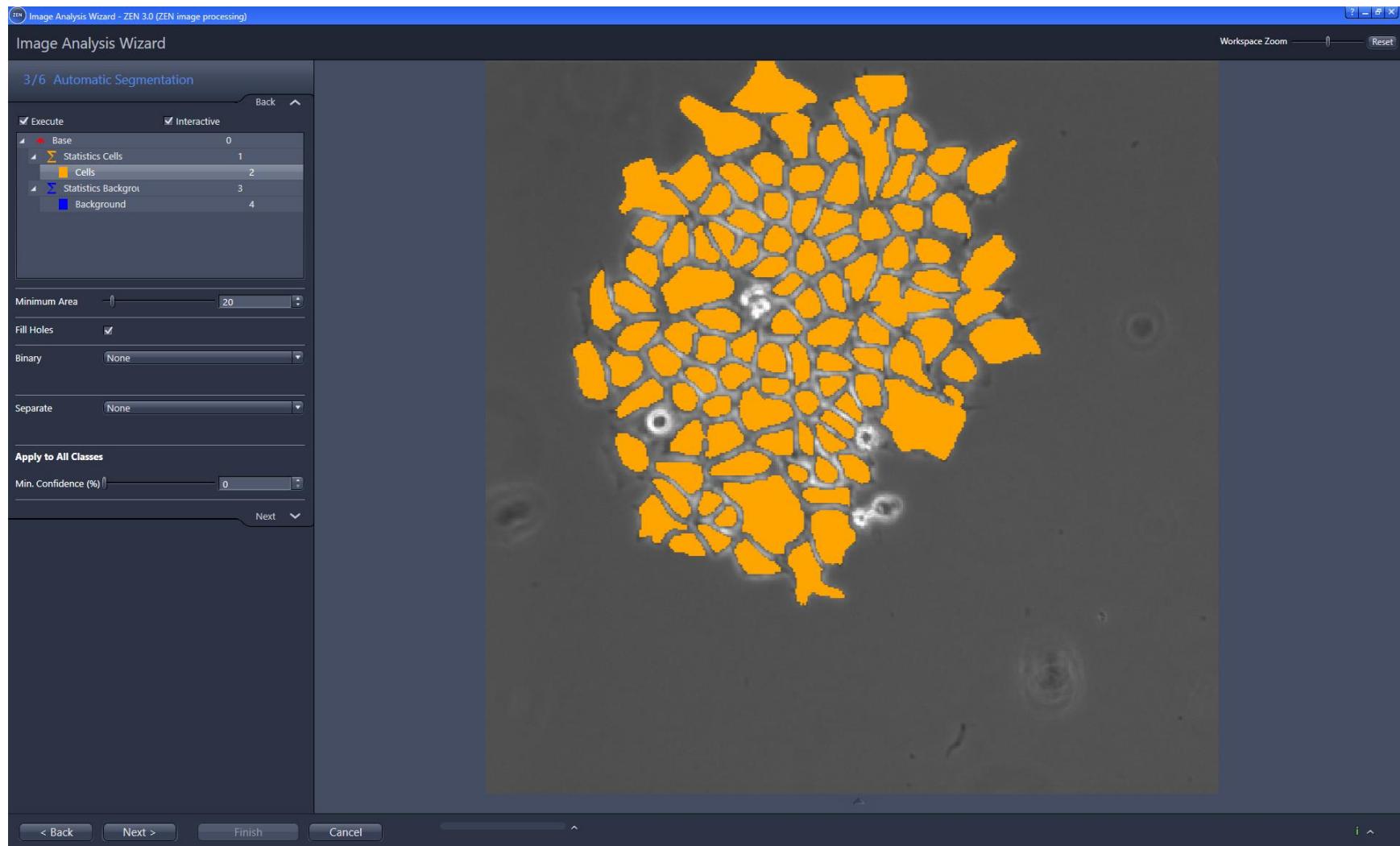
Actionable Information – Where can trained models be useful



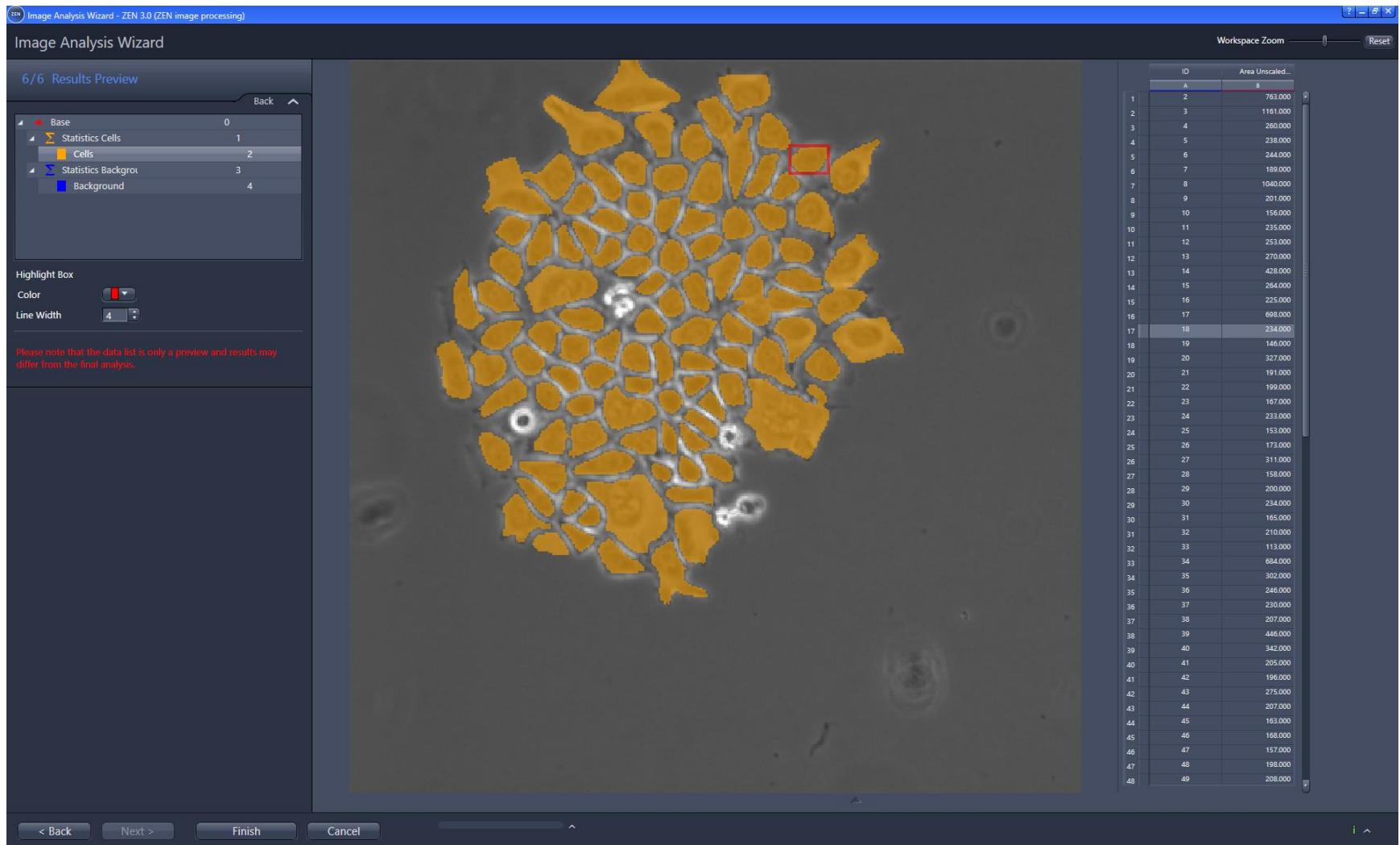
UX Design – Training in ZEN Blue



Integration into Image Analysis Pipelines in ZEN Blue



Integration into Image Analysis Pipelines in ZEN Blue



New UX Design – Intellesis Training in ZEN Core



Manage Templates

Intellesis Trainable Segmentation

1/1 Image Labeling and Training

Grains Intellesis Template Model (1) (1)

Import images

- MET94--03_1Ch.czi
- One labeled slice MET94--01_1Ch.czi
- One labeled slice

Classes

- Grains (Orange)
- Foreign Phase (Teal)

Add Class Delete Class

Color

Label Opacity

Segmentation Opacity

Segmentation: Deep Features 256

Postprocessing: No Postprocessing

Train & Segment

Finish Cancel

DIMENSIONS

Zoom: 100% 104%

Tools: Selection, Zoom, Pan, Hand, Brush

Channels: C1 Labels Seg Conf

DISPLAY

Labeling Options

2D Display

Overlay Options

Undo / Redo

Labeling Mode

Brush Size: 3

All Labels

Single Channel Range Indicator

The screenshot displays the Intellesis Trainable Segmentation interface within the ZEN Core software. On the left, a sidebar shows imported images and class definitions for 'Grains' (orange) and 'Foreign Phase' (teal). The main area shows a microscopy image of tissue with these two classes outlined in dashed lines. Control panels at the bottom allow for zooming, labeling mode selection, and brush size adjustment.

Intellesis – Brush Cursor



Manage Templates

Intellesis Trainable Segmentation

1/1 Image Labeling and Training

Grains Intellesis Template Model (1) (1)

Import Images

- MET94--03_1Ch.czi
- One labeled slice MET94--01_1Ch.czi
- One labeled slice

Classes

- Grains
- Foreign Phase

Add Class Delete Class

Color

Label Opacity

Segmentation Opacity

Segmentation Deep Features 256

Postprocessing No Postprocessing

Your latest changes have not been trained yet.

Train & Segment

Finish Cancel

Brush Cursor

Intellesis – Revised Naming Scheme



Manage Templates

Intellesis Trainable Segmentation

1/1 Image Labeling and Training

Grains Intellesis Template Model (1) (1)

Import images

- MET94--03_1Ch.czi
- One labeled slice MET94--01_1Ch.czi
- One labeled slice

Classes

- Grains
- Foreign Phase

Add Class Delete Class

Color

Label Opacity

Segmentation Opacity

Segmentation

- Deep Features 256
- Basic Features 25
- Basic Features 33
- Deep Features 50
- Deep Features 64
- Deep Features 70
- Deep Features 128
- Deep Features 256

Postprocessing

- C1 Labels Seg Conf
- Single Channel Range Indicator

Revised, more logical naming

DIMENSIONS DISPLAY LABELING OPTIONS 2D DISPLAY OVERLAY OPTIONS

Zoom Tools Channels

Undo / Redo Labeling Mode Brush Size All Labels

Carl Zeiss Microscopy

19

The screenshot shows the Intellesis Trainable Segmentation software interface. On the left, there's a sidebar with 'Import images' (MET94--03_1Ch.czi, One labeled slice MET94--01_1Ch.czi, One labeled slice), 'Classes' (Grains, Foreign Phase), and controls for 'Color', 'Label Opacity', and 'Segmentation Opacity'. Below these are dropdown menus for 'Segmentation' (listing 'Deep Features 256', 'Basic Features 25', etc.) and 'Postprocessing' (listing 'C1', 'Labels', 'Seg', 'Conf'). At the bottom are 'Finish' and 'Cancel' buttons. The main area displays a microscopy image of a tissue sample with blue outlines around cells, representing segmentation results. A callout box at the bottom center contains the text 'Revised, more logical naming'. An orange arrow points from the 'Deep Features 256' entry in the Segmentation list to this callout box.

Intellesis – Confidence map in Training UI



Manage Templates

Intellesis Trainable Segmentation

1/1 Image Labeling and Training

Grains Intellesis Template Model (1) (1)

Import Images

- MET94--03_1Ch.czi
- One labeled slice MET94--01_1Ch.czi
- One labeled slice

Classes

- Grains
- Foreign Phase

Add Class Delete Class

Color

Label Opacity

Segmentation Opacity

Segmentation Deep Features 256

Postprocessing No Postprocessing

Train & Segment

Finish Cancel

Confidence Map [0- 100%]

DIMENSIONS

Zoom 100% Show All 318 %

Tools

Channels C1 Labels Seg Conf

Single Channel Range Indicator

DISPLAY LABELING OPTIONS 2D DISPLAY OVERLAY OPTIONS

Undo / Redo Labeling Mode Brush Size All Labels

Brush Size 3 Clear

Intellesis – Grains Integration



Job Mode

Filter Templates... Filter

All

Material Analysis

Cast Iron Analysis

Comparative Diagrams

Grain Size Analysis

Layer Thickness Measurement

Multiphase Analysis

NMI Analysis

Measurement

Groups

Samples

Generic

TITLE LAST ACCESS CATEGORY

TITLE	LAST ACCESS	CATEGORY
Grain Size Analysis - Intellesis - (Planimetric) Open	18.07.2019 04:42:33	Material Analysis
Grain Size Analysis (Planimetric) Open	18.07.2019 04:42:32	Material Analysis
Grain Size Analysis (Intercept) Open	18.07.2019 04:42:32	Material Analysis
Grain Size Analysis (Comparison) Open	18.07.2019 04:42:32	Material Analysis

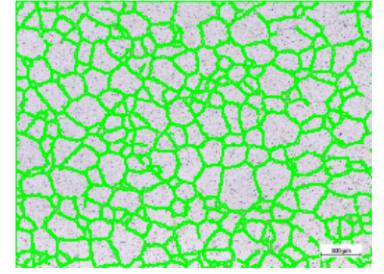
Properties

Name: Grain Size Analysis - Intellesis - (Planimetric)

Description: Job template for grain size analysis using the planimetric method with Intellesis.

Category: Material Analysis

Subcategory: Grain Size Analysis



Intellesis – Grains Integration



Grain Size Analysis - Intellesis - (Planimetric)

Image Segmentation

Add comment here...

Model Name: Grains Model

Model Class: Grains

Min. Confidence: 51

Minimum Area: 300

Min. Hole Area: 200

Fill Holes:

Binary: None

Separate: None

Next

Step 5 of 8

Model and Class Selection

Select Model: Grains Model

Model Class: Grains

Grains

Foreign Phase

New binary post-processing options

DISPLAY DATA ZONE IMAGE ANALYSIS OPTIONS

Show Objects Fill Opacity: 0

Show All Classes

Basis: 0

Grain: 4

Navigator: 100 µm

The screenshot shows the 'Edit Job' interface for a grain size analysis. On the left, the 'Image Segmentation' panel includes a toolbar with various icons, a 'Model Name' dropdown set to 'Grains Model', and sliders for 'Min. Confidence' (51), 'Minimum Area' (300), and 'Min. Hole Area' (200). A 'Fill Holes' checkbox is checked. Below these are dropdowns for 'Binary' (None) and 'Separate' (None). At the bottom of this panel are 'Previous' and 'Next' buttons, and the text 'Step 5 of 8'. In the center, a large image shows a green background with white outlines of grains. A small window titled 'Model and Class Selection' is open, showing 'Grains Model' and 'Grains' selected. Another window titled 'New binary post-processing options' is also visible. At the bottom, there are 'DISPLAY', 'DATA ZONE', and 'IMAGE ANALYSIS OPTIONS' tabs, along with checkboxes for 'Show Objects' (checked), 'Fill' (unchecked), 'Opacity' (set to 0), and 'Show All Classes' (checked). A legend indicates 'Basis' (red square) and 'Grain' (green square). A 'Navigator' window in the bottom right corner shows a 100 µm scale bar. An orange arrow points from the 'Model and Class Selection' window to the main image area, and another orange arrow points from the 'New binary post-processing options' window to the 'Binary' dropdown in the segmentation panel.

Intellesis – Grains Integration



Grain Size Analysis - Intellesis - (Planimetric)

Result View

Original Image 04/18/2019 | 12:13:06

Analyzed Image 04/18/2019 | 12:13:06

Grain Size Analysis (Planimetric) - Results

Minimum recommendation of 700 grains and 3 images for chosen standard not fulfilled

Image Selection

Statistics

Image 01 04/18/2019 | 12:13:06

You reached the required minimum number of loop iterations.

Previous Apply Exit Loop Continue

Step 7 of 8

Edit Job

Result View

Original Image 04/18/2019 | 12:13:06

Analyzed Image 04/18/2019 | 12:13:06

Grain Size Distribution

Grain Size No.	Number of Grains (%)
0.000 - 0.500	0.680 (0.1%)
0.500 - 1.000	0.680 (0.1%)
1.000 - 1.500	0.000 (0.0%)
1.500 - 2.000	0.000 (0.0%)
2.000 - 2.500	0.000 (0.0%)
2.500 - 3.000	0.000 (0.0%)
3.000 - 3.500	0.680 (0.1%)
3.500 - 4.000	0.680 (0.1%)
4.000 - 4.500	2,721 (0.4%)
4.500 - 5.000	4,082 (0.6%)
5.000 - 5.500	14,966 (1.5%)
5.500 - 6.000	12,245 (1.2%)
6.000 - 6.500	15,646 (1.5%)
6.500 - 7.000	9,524 (0.9%)
7.000 - 7.500	8,163 (0.8%)
7.500 - 8.000	8,844 (0.9%)
8.000 - 8.500	10,884 (1.0%)
8.500 - 9.000	3,401 (0.3%)
9.000 - 9.500	2,721 (0.2%)
9.500 - 10.000	4,082 (0.4%)
10.000 - 10.500	1,361 (0.1%)
10.500 - 11.000	0,000 (0.0%)
11.000 - 11.500	0,000 (0.0%)
11.500 - 12.000	0,000 (0.0%)
12.000 - 12.500	0,000 (0.0%)
12.500 - 13.000	0,000 (0.0%)
13.000 - 13.500	0,000 (0.0%)
13.500 - 14.000	0,000 (0.0%)
14.000 - 14.500	0,000 (0.0%)

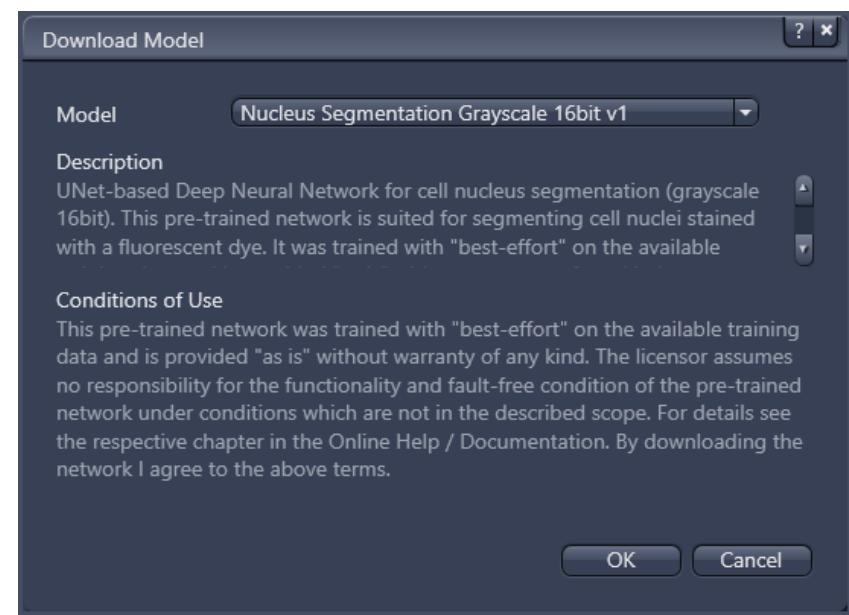
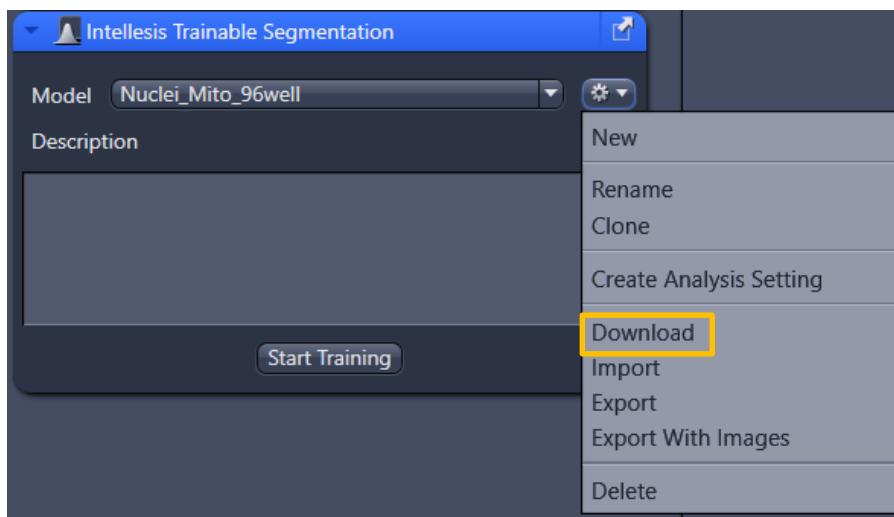
Parameters

Grain Size	Number of Grains	Exact Value
5,5	167,000	5,401

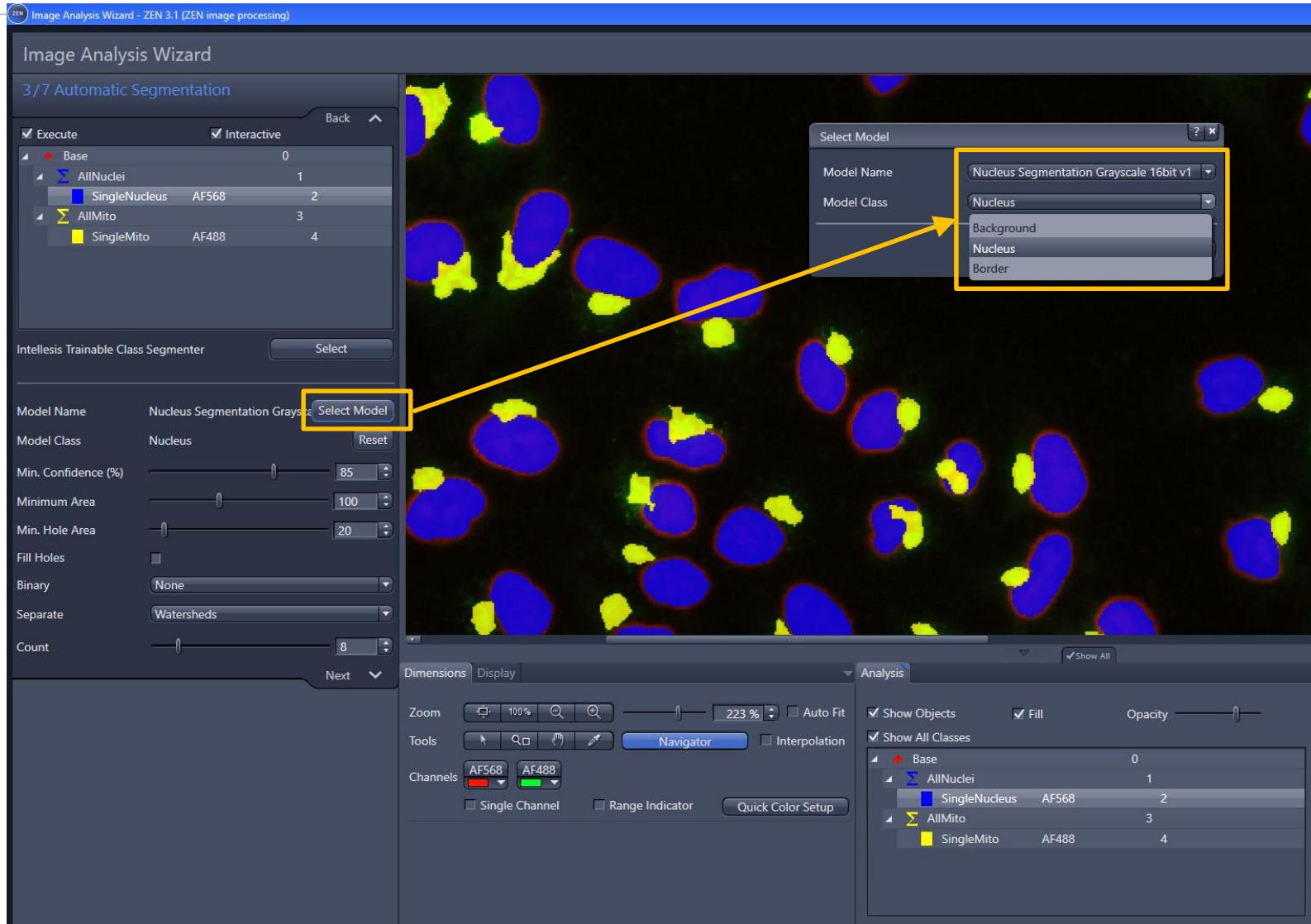
Download Deep Neural Networks for specific application



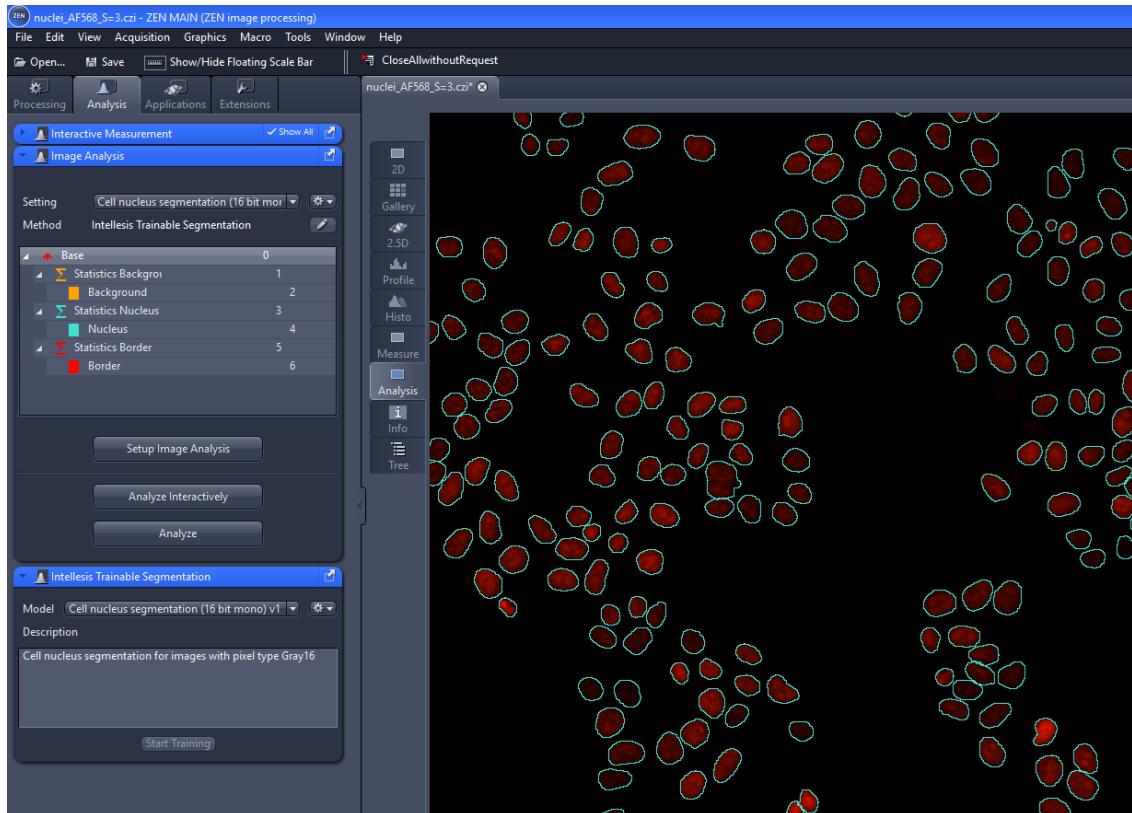
- Download dialog to select networks(s) from ZEISS
- First Application: Pre-trained Nucleus Detection
- Use directly for Segmentation, Image Analysis incl. Class Segmentation



Nucleus Detector - Use inside Class Segmentation

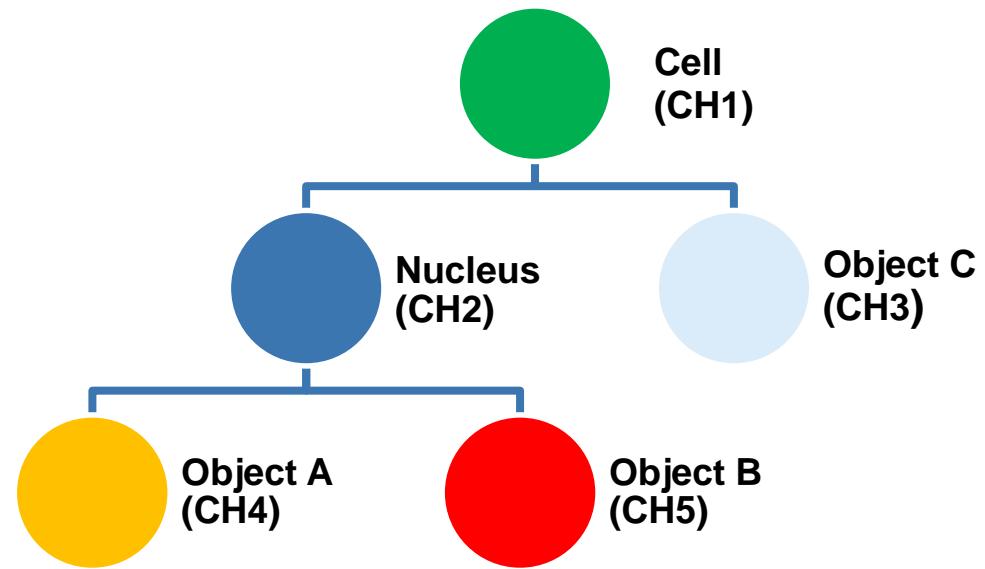
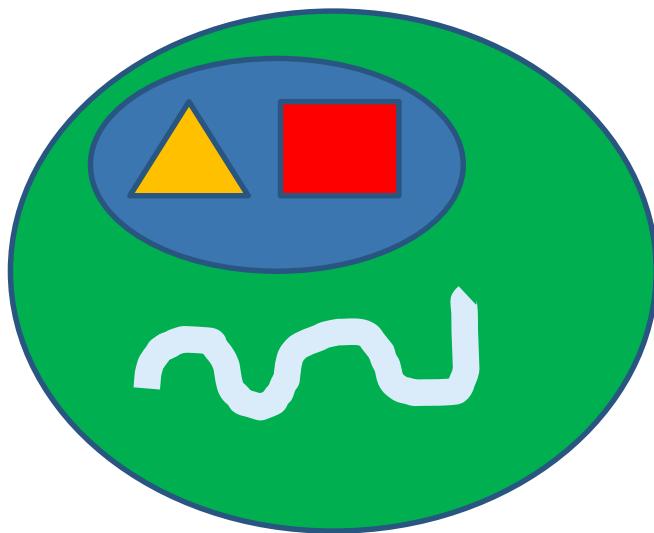


Pretrained Network – Integration with Image Analysis



- 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



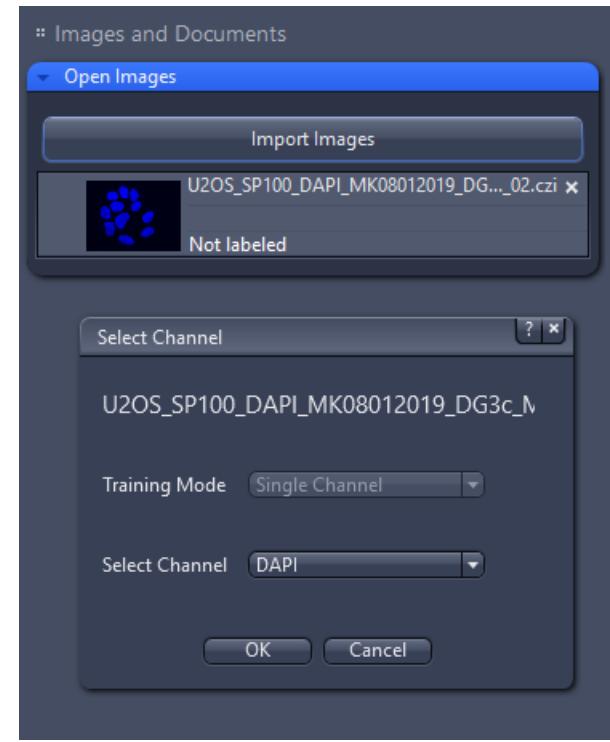
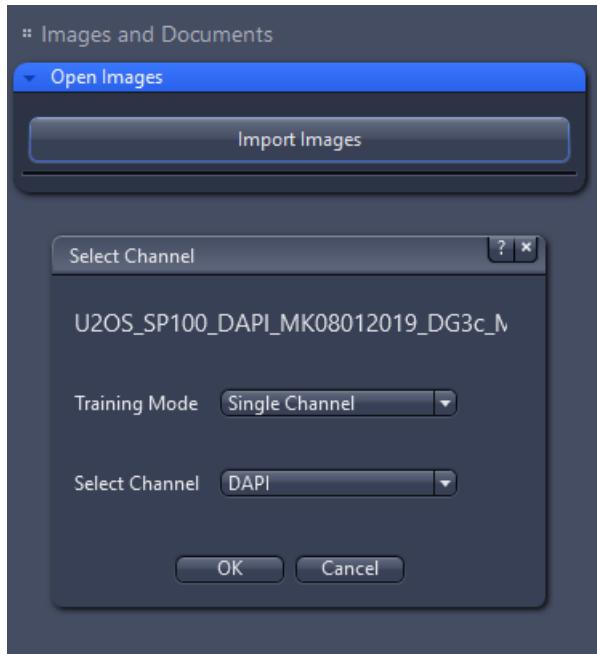
Flexible & Hierarchical Segmentation Scheme (Class Segmentation)

- 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



Image Analysis Wizard - ZEN MAIN (ZEN image processing)

Image Analysis Wizard

3/7 Automatic Segmentation

Back

Execute Interactive

Base 0
Σ AllNuclei 1
 SingleNucleus DAPI 2
 Σ AllSpots 3
 SingleSpots EGFP 4

Apply

Intellesis Trainable Class Segmenter

Select Model

Model Name SP100 Select Model

Model Class Object Reset

Min. Confidence (%) 51

Minimum Area 1

Min. Hole Area 1

Fill Holes

Binary None

Separate None

Next

Workspace Zoom Reset

A microscopy image showing several cells stained with DAPI (blue) and EGFP (yellow). The cells are segmented into individual nuclei, and some spots are identified as single spots.

Dimensions Display

Zoom 100% 37 %

Tools

Channels DAPI EGFP Single Channel Range Indicator

Show All

Analysis

Show Objects Fill Opacity

Show All Classes

Base 0
Σ AllNuclei 1
 SingleNucleus DAPI 2
 Σ AllSpots 3
 SingleSpots EGFP 4

< Back Next > Finish Cancel

Class Segmentation - Hierarchical structures with independent segmenter per class



The screenshot shows the Image Analysis Wizard interface in ZEN MAIN (ZEN image processing). The main window title is "Image Analysis Wizard - ZEN MAIN (ZEN image processing)". The current step is "3/7 Automatic Segmentation".

Left Panel (Segmenter Settings):

- Execute:** Interactive
- Tree View:**
 - Base (0)
 - AllNuclei (1)
 - SingleNucleus DAPI (2)
 - AllSpots (3)
 - SingleSpots EGFP (4)

Buttons: Apply, Select, Model Name (SP100), Model Class (Object), Min. Confidence (%): 51, Minimum Area: 1, Min. Hole Area: 1, Fill Holes (checked), Binary (None), Separate (None).

Bottom Buttons: < Back, Next >, Finish, Cancel.

Center Panel (Image Preview): Displays a fluorescence microscopy image of cells stained with DAPI (blue nuclei) and EGFP (yellow spots). Several cells are shown, each containing multiple yellow spots representing EGFP-positive structures.

Bottom Center Overlay (Select Model Dialog): A modal dialog titled "Select Model" is open, showing the current settings:
 - Model Name: SP100
 - Model Class: Object (highlighted)

Bottom Right Panel (Analysis Tree):

 - Show Objects:**
 - Show All Classes:**
 - Tree View:**
 - Base (0)
 - AllNuclei (1)
 - SingleNucleus DAPI (2)
 - AllSpots (3)
 - SingleSpots EGFP (4)

Bottom Right Buttons: Opacity slider, Show All button.

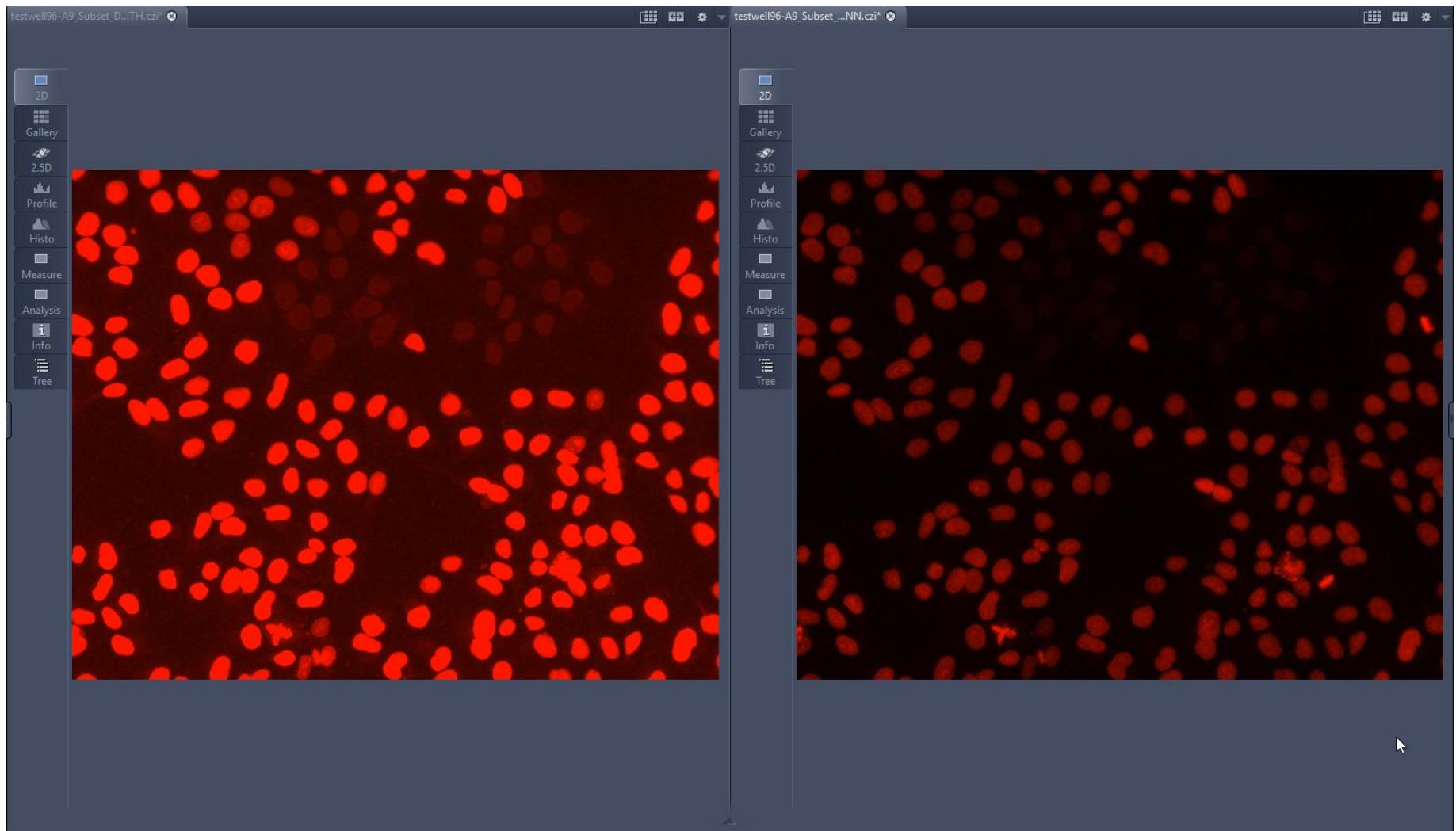
Deep neural Networks versus Threshold



- 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

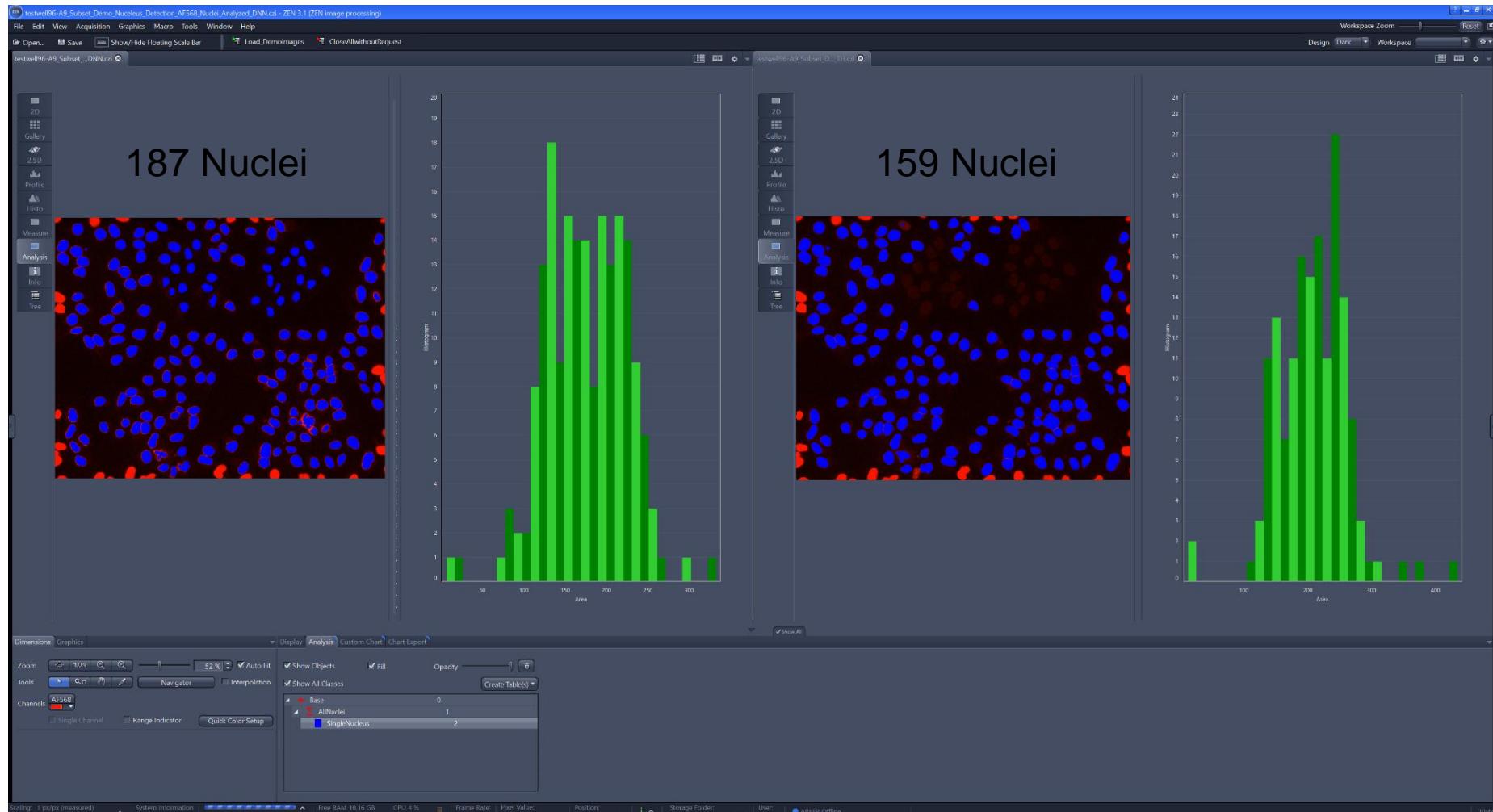
Deep neural Networks versus Threshold

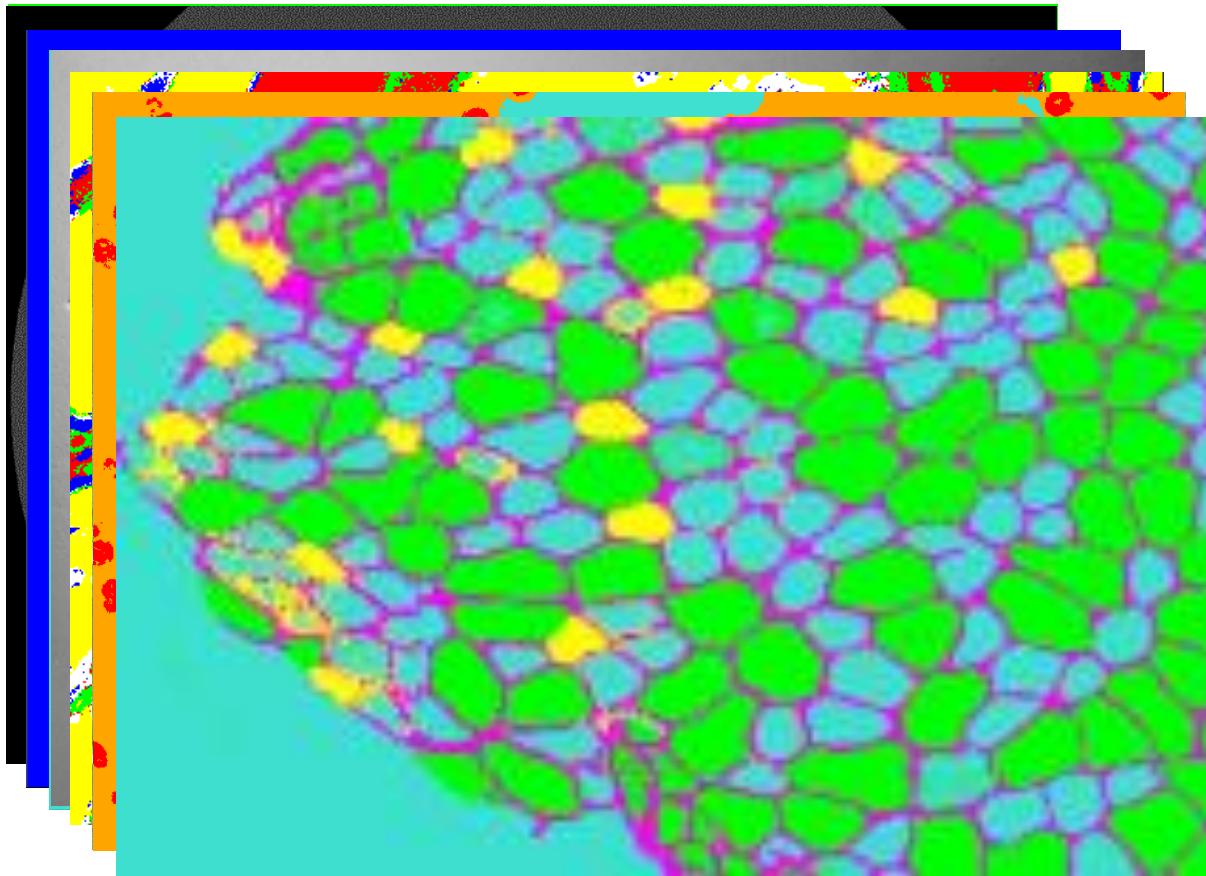
Same Image Data – different Display Curve



Deep neural Networks versus Threshold

Same Image Data – different results

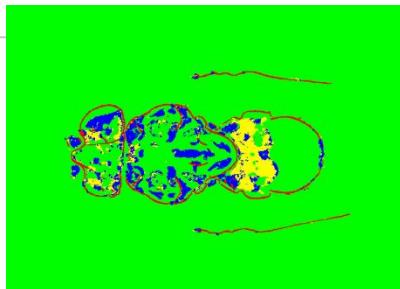
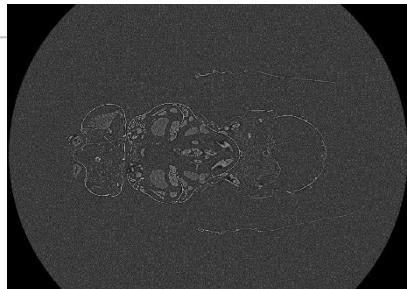




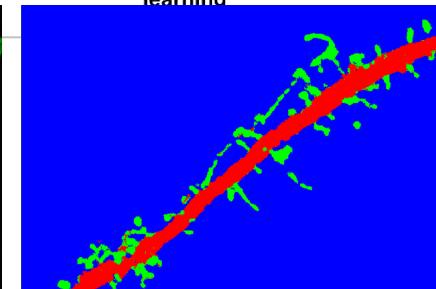
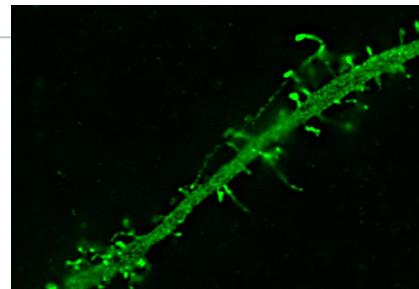
Sample courtesy of DR AKSHAY DIXIT, SANKALP, CSE, IITB, Mumbai, London

- 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

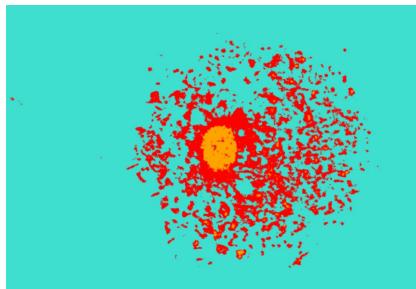
Intellessis Applications: Life Science Examples



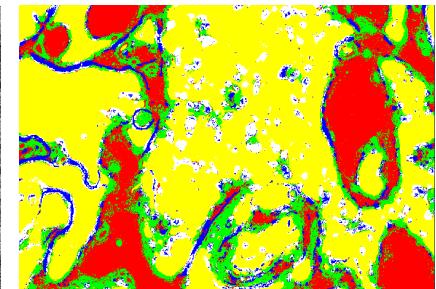
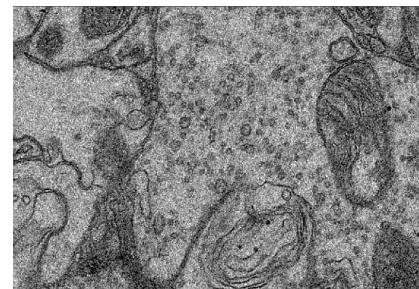
XRM scan: Drosophila



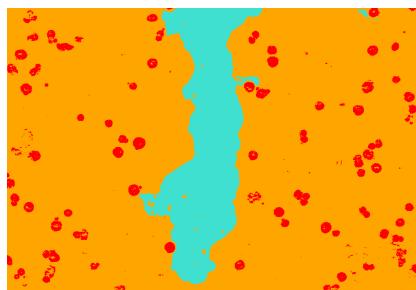
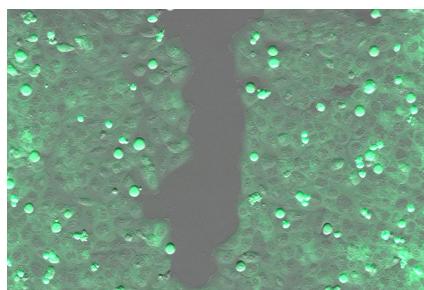
Neuron – spines & dendrite



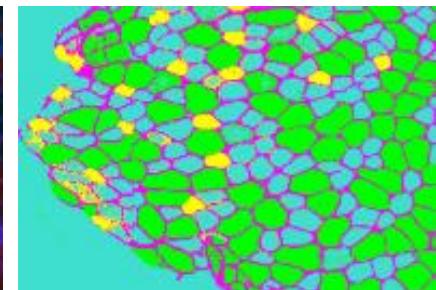
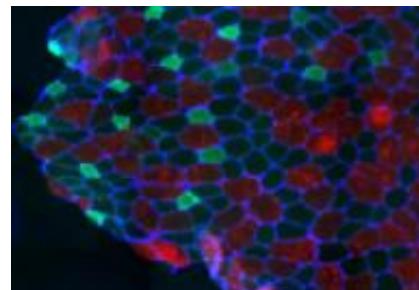
Organoid and migrated cells



Classification of cell compartments



Scratch assay – area and mitotic cells



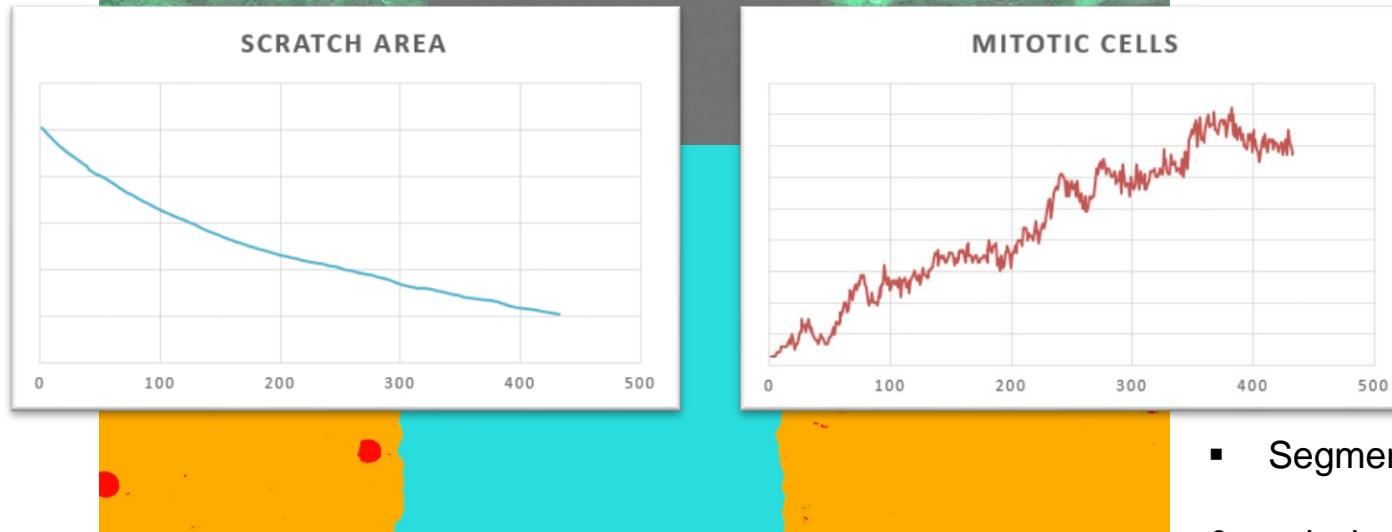
Muscle section – different myosin types

ZEN Intellesis

Scratch Assay



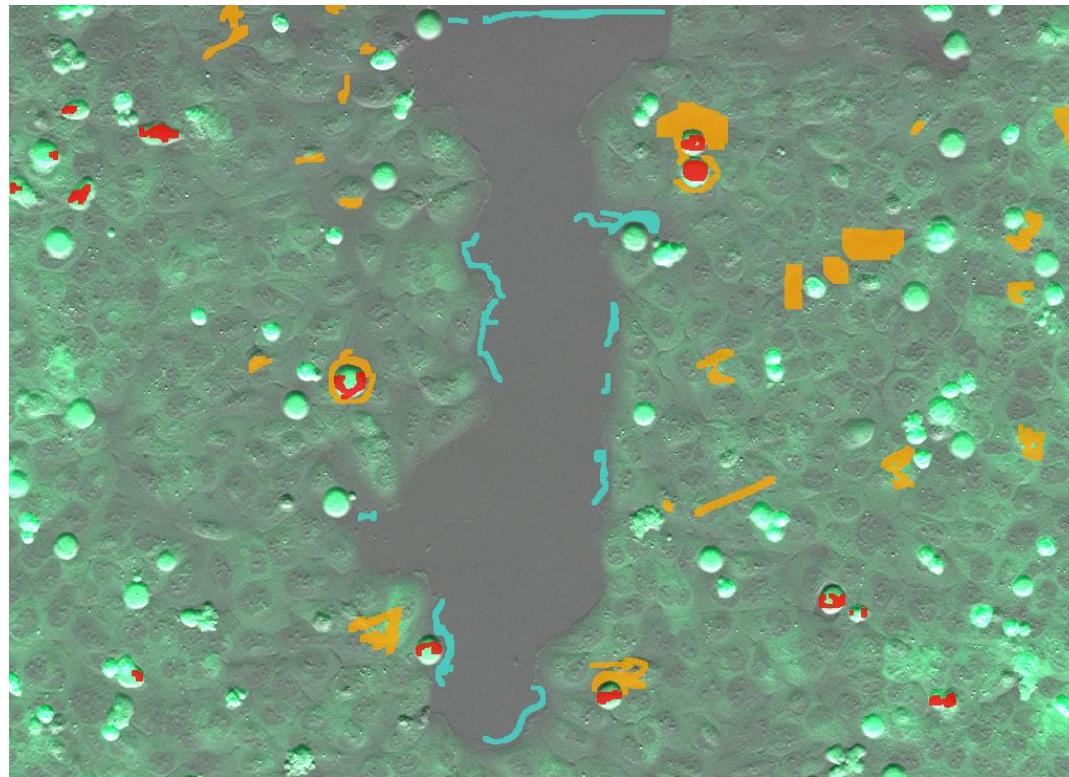
- Time series of GFP expressing HeLa cells
- Scratch area & number of mitotic cells shall be measured



- Segmented data
& analysis results

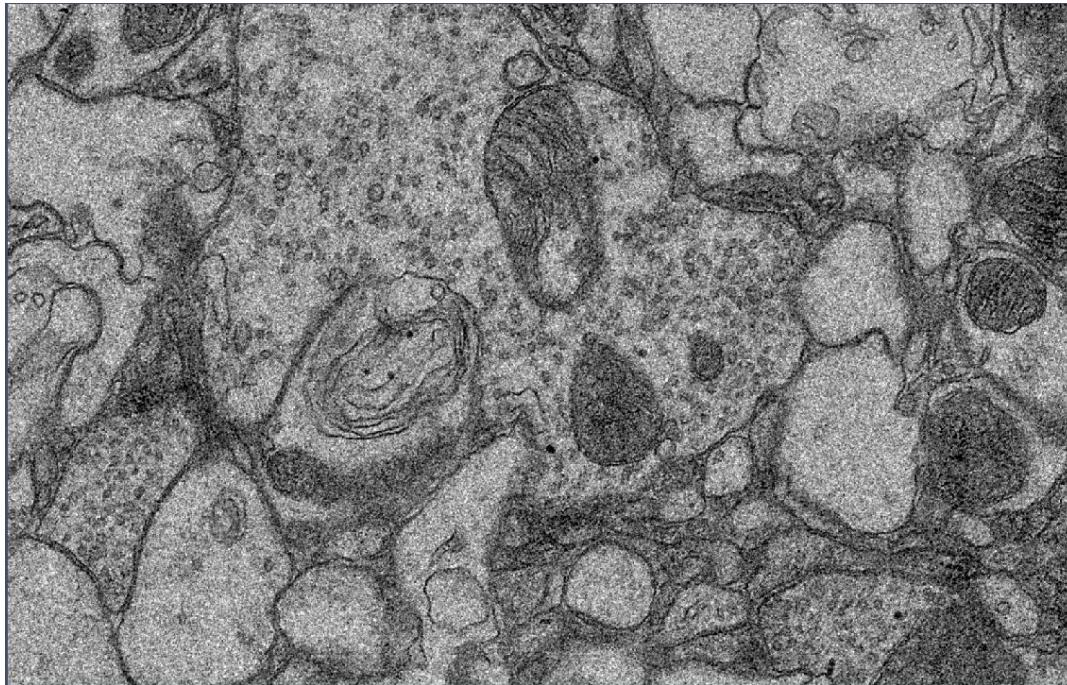
ZEN Intellesis

Scratch Assay



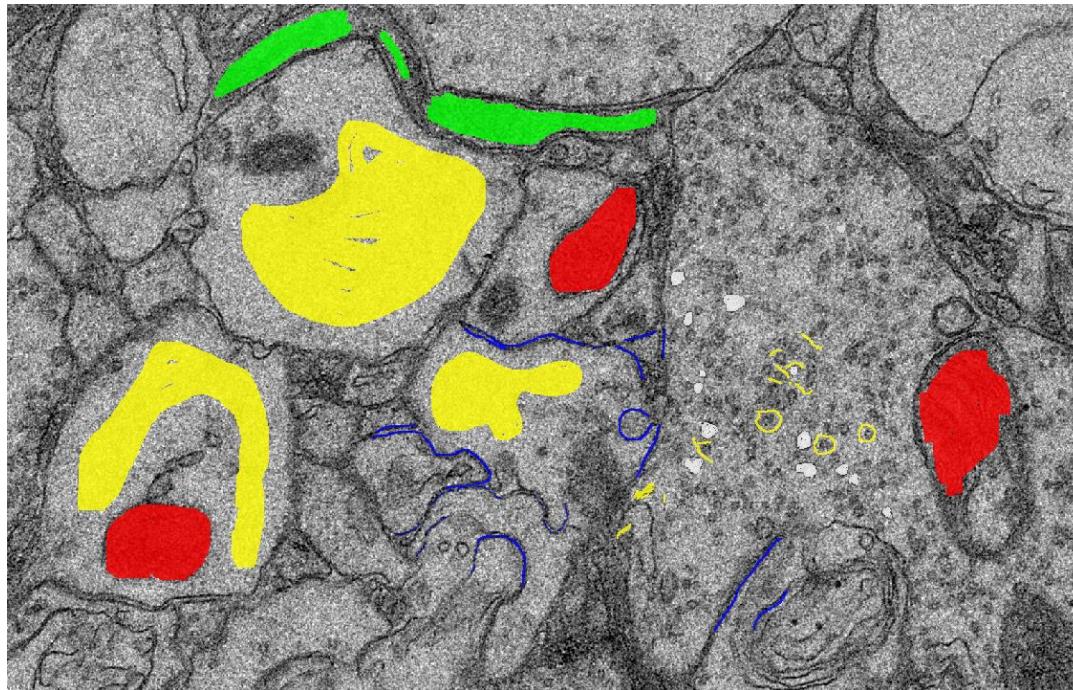
- Time series of GFP expressing HeLa cells
- Scratch area & number of mitotic cells shall be measured
- Training („painted pixels“)

*Images Courtesy of
MPI for infection biology, Berlin*



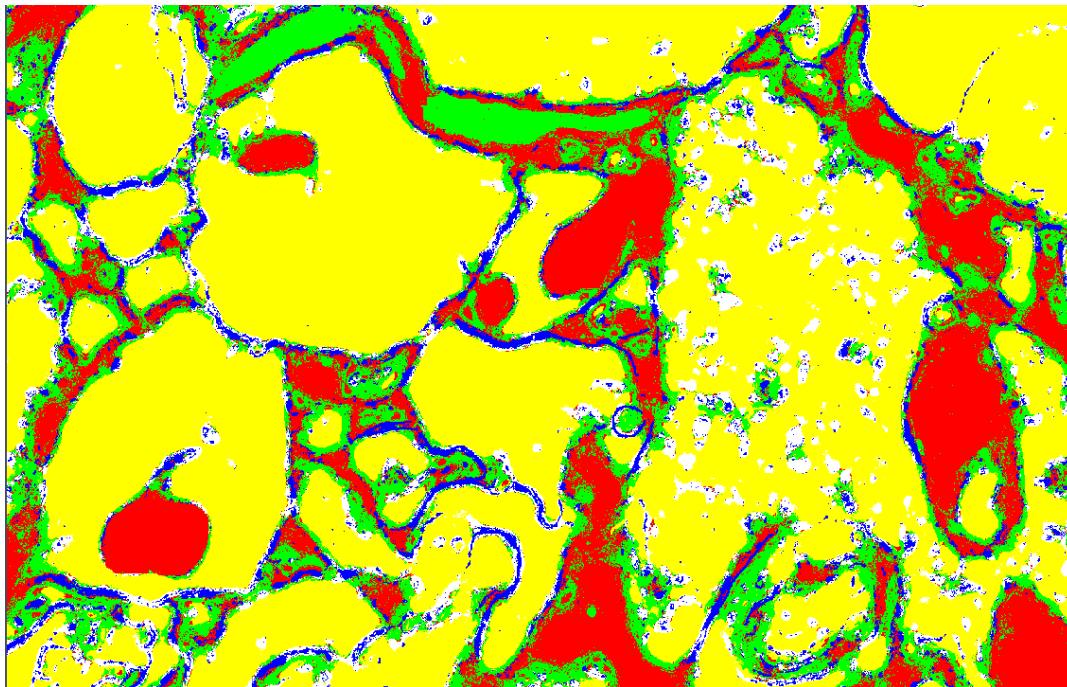
- Section through the calyx region of 30d *Drosophila melanogaster*, aim
- quantification of synaptic vesicles and structure characterization
- Raw data

*Images Courtesy of
MPI for infection biology, Berlin*



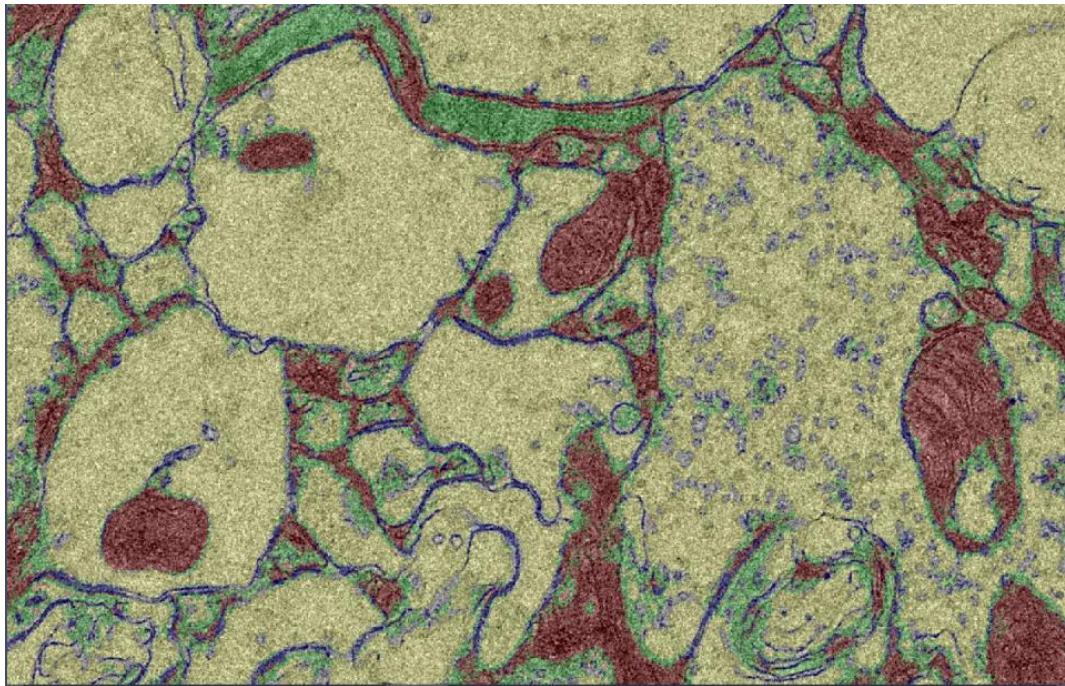
- Section through the calyx region of 30d *Drosophila melanogaster*, aim
- quantification of synaptic vesicles and structure characterization
- Training („painted pixels“)

*Images Courtesy of
MPI for infection biology, Berlin*



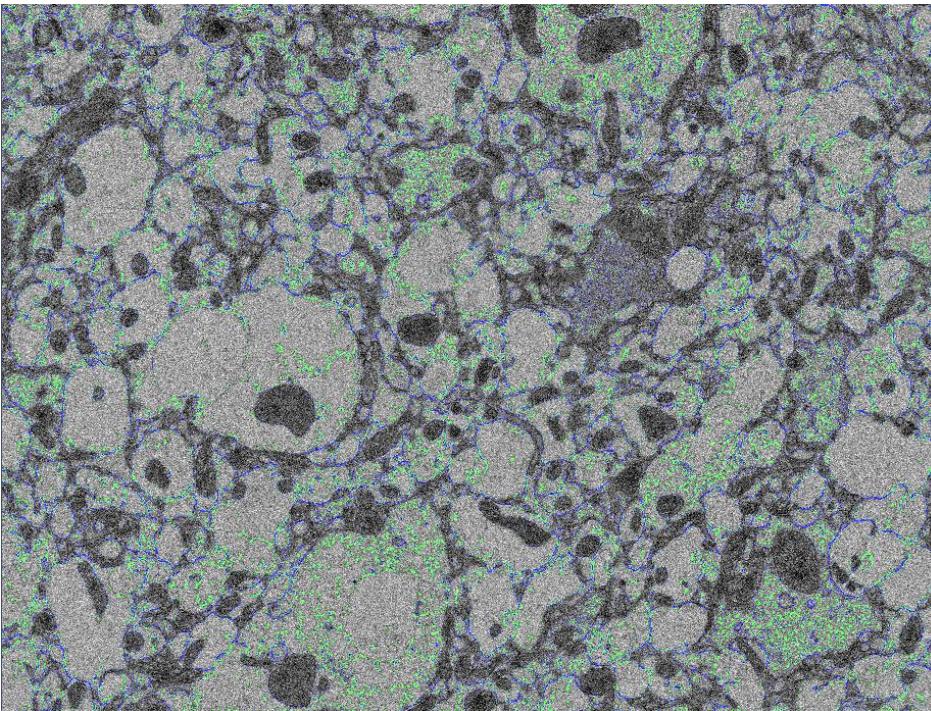
- Section through the calyx region of 30d *Drosophila melanogaster*, aim
- quantification of synaptic vesicles and structure characterization
- segmented data

*Images Courtesy of
MPI for infection biology, Berlin*



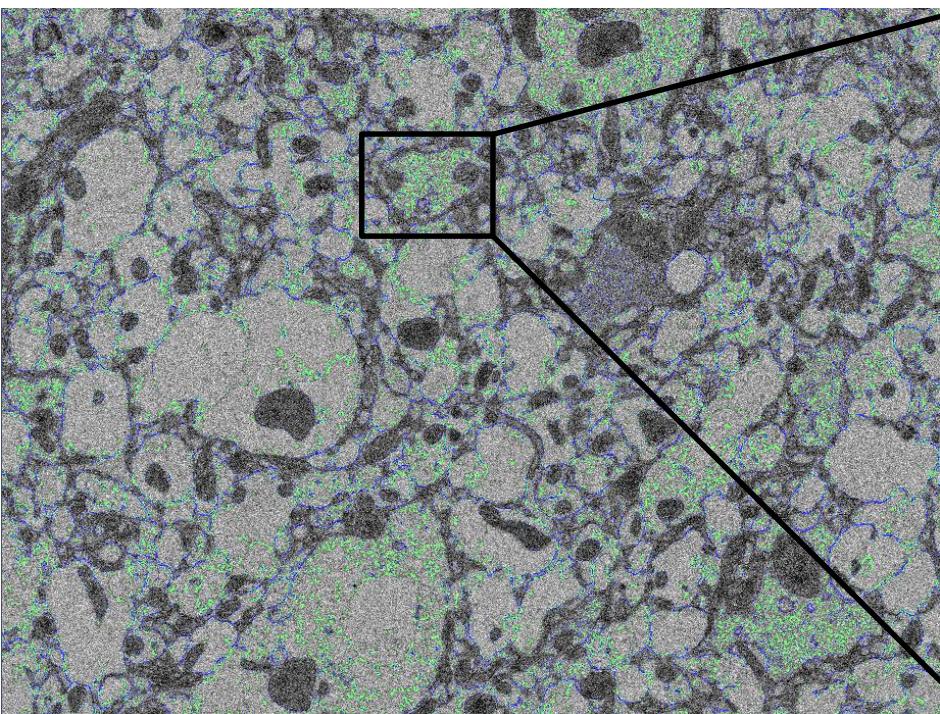
- Section through the calyx region of 30d *Drosophila melanogaster*, aim
- quantification of synaptic vesicles and structure characterization
- segmented data + raw data overlay

*Images Courtesy of
MPI for infection biology, Berlin*

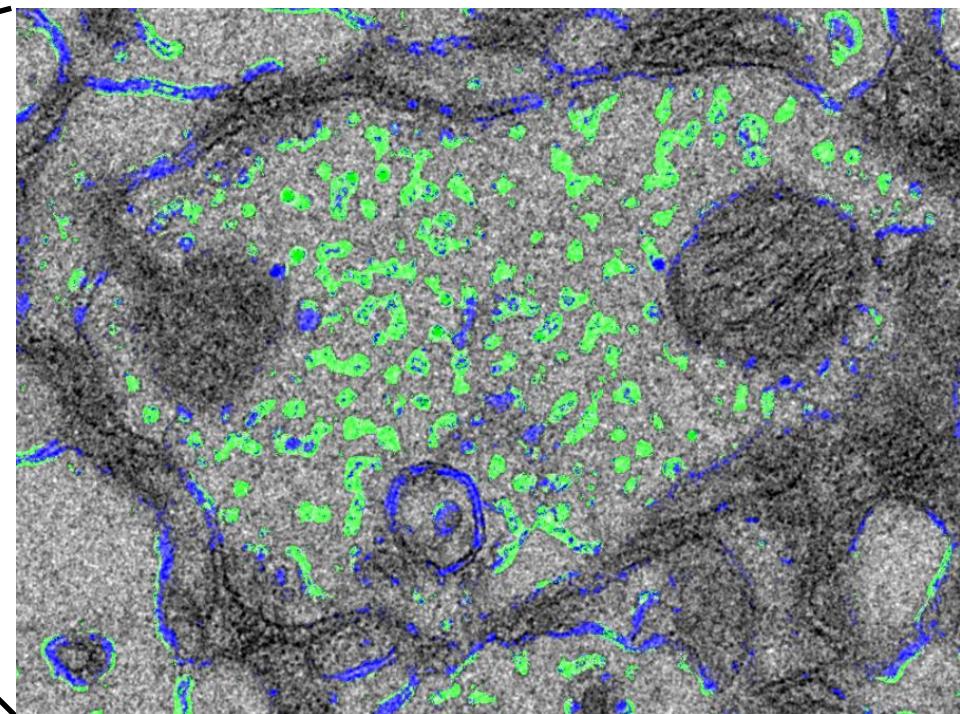


Full Image

*Images Courtesy of
MPI for infection biology, Berlin*

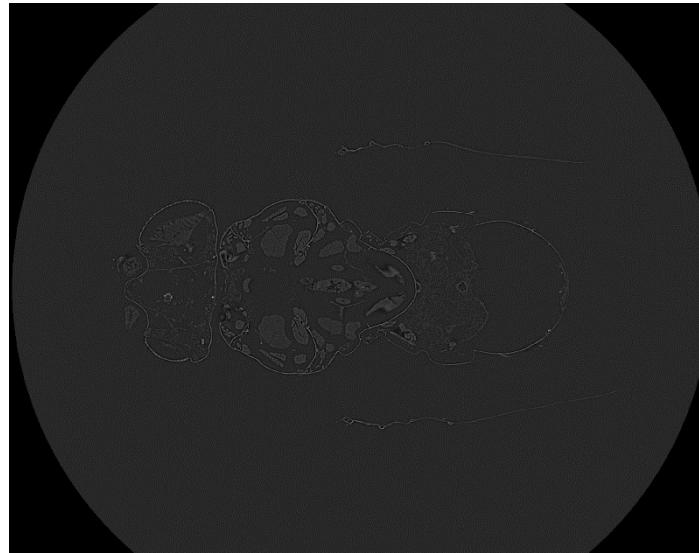


Full Image



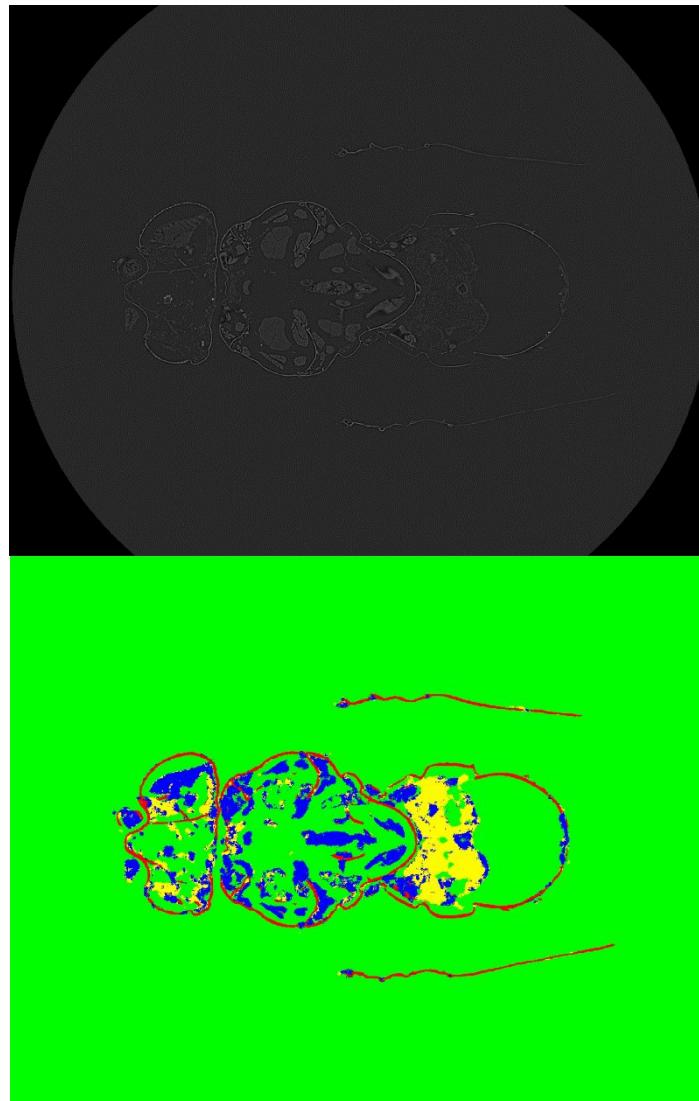
Vesicles and Membranes

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



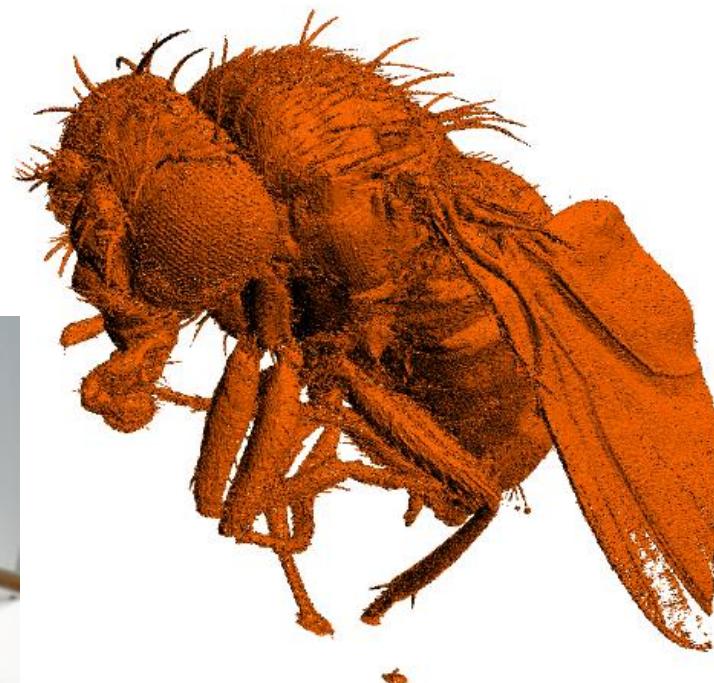
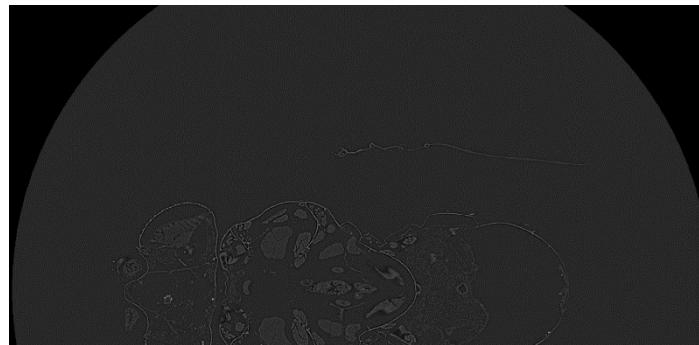
- 3D X-Ray microscope dataset

*Dataset Courtesy of
Francis Crick Institute , London*



- 3D X-Ray microscope dataset
- Segmented data

*Dataset Courtesy of
Francis Crick Institute , London*

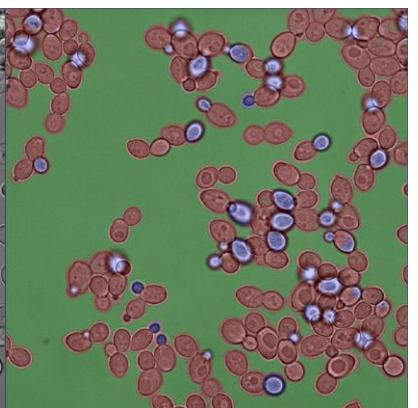
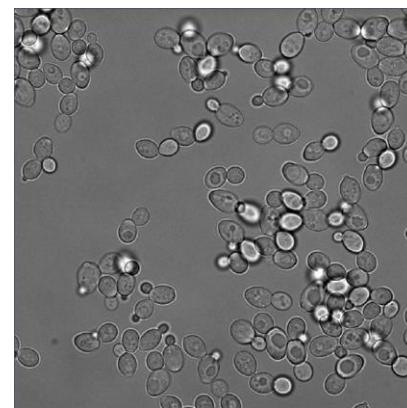
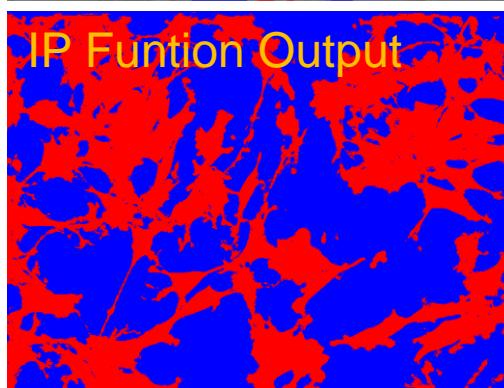
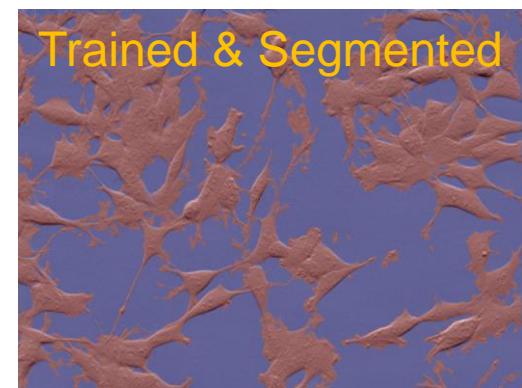
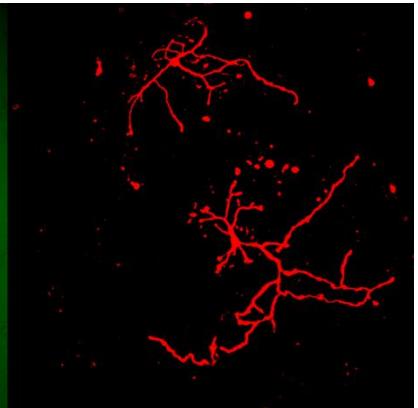
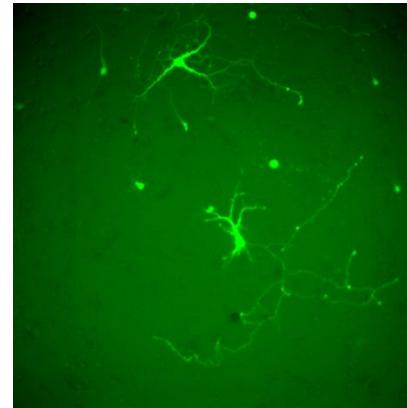
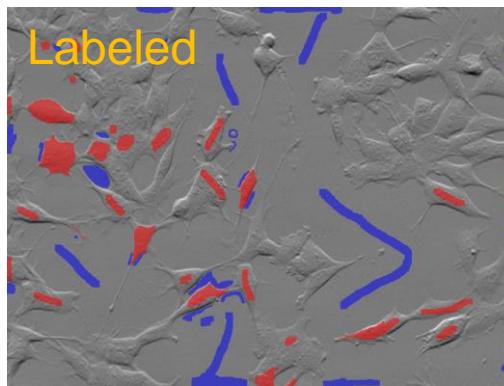
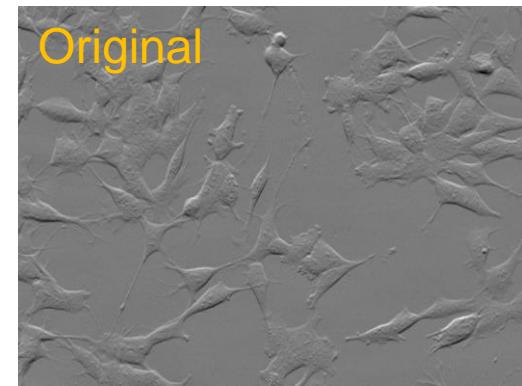


3D rendering

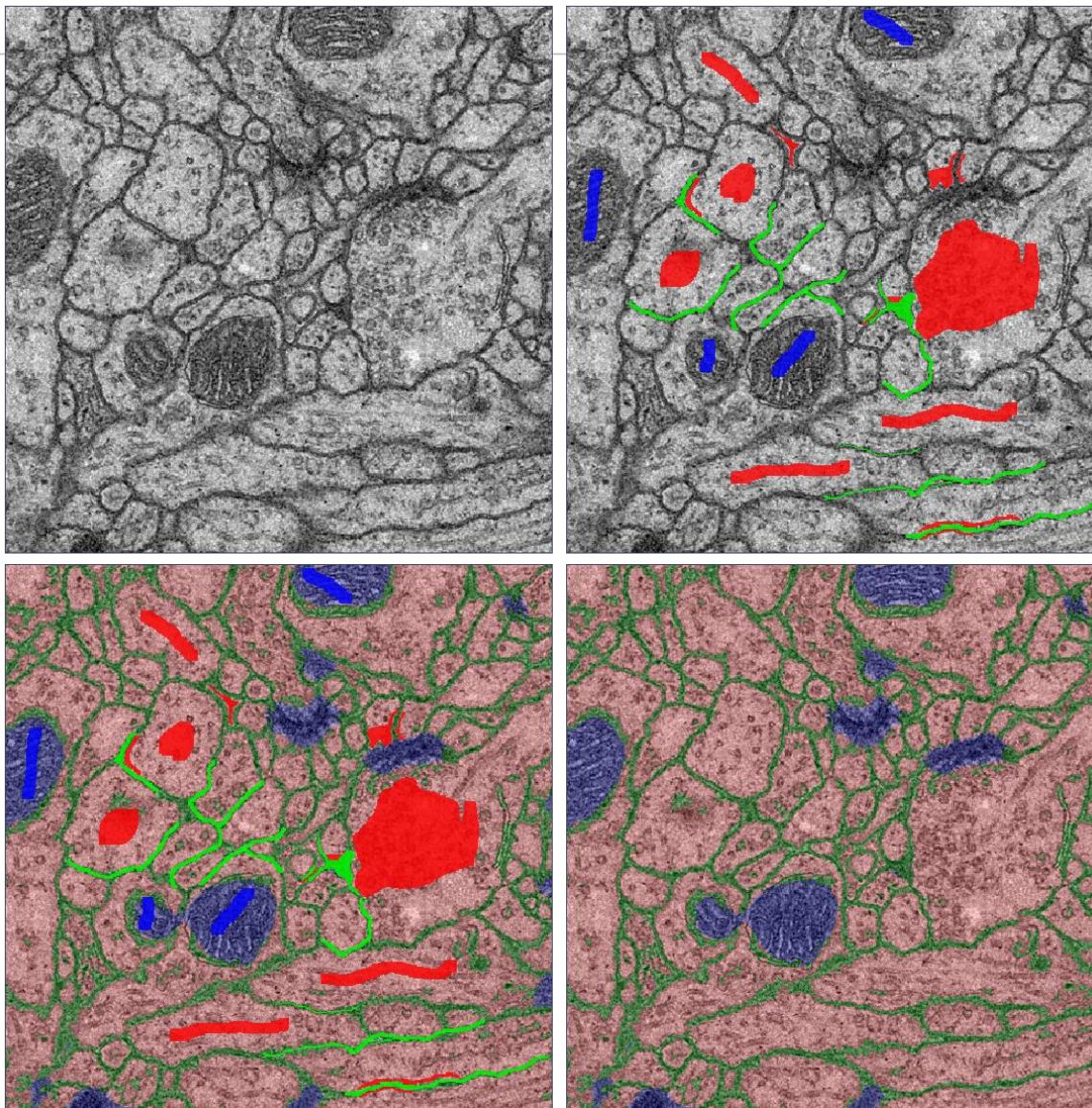
& 3D print...

*Dataset Courtesy of
Francis Crick Institute, London*

CD7 Phase Gradient Contrast, Neurons, Yeast

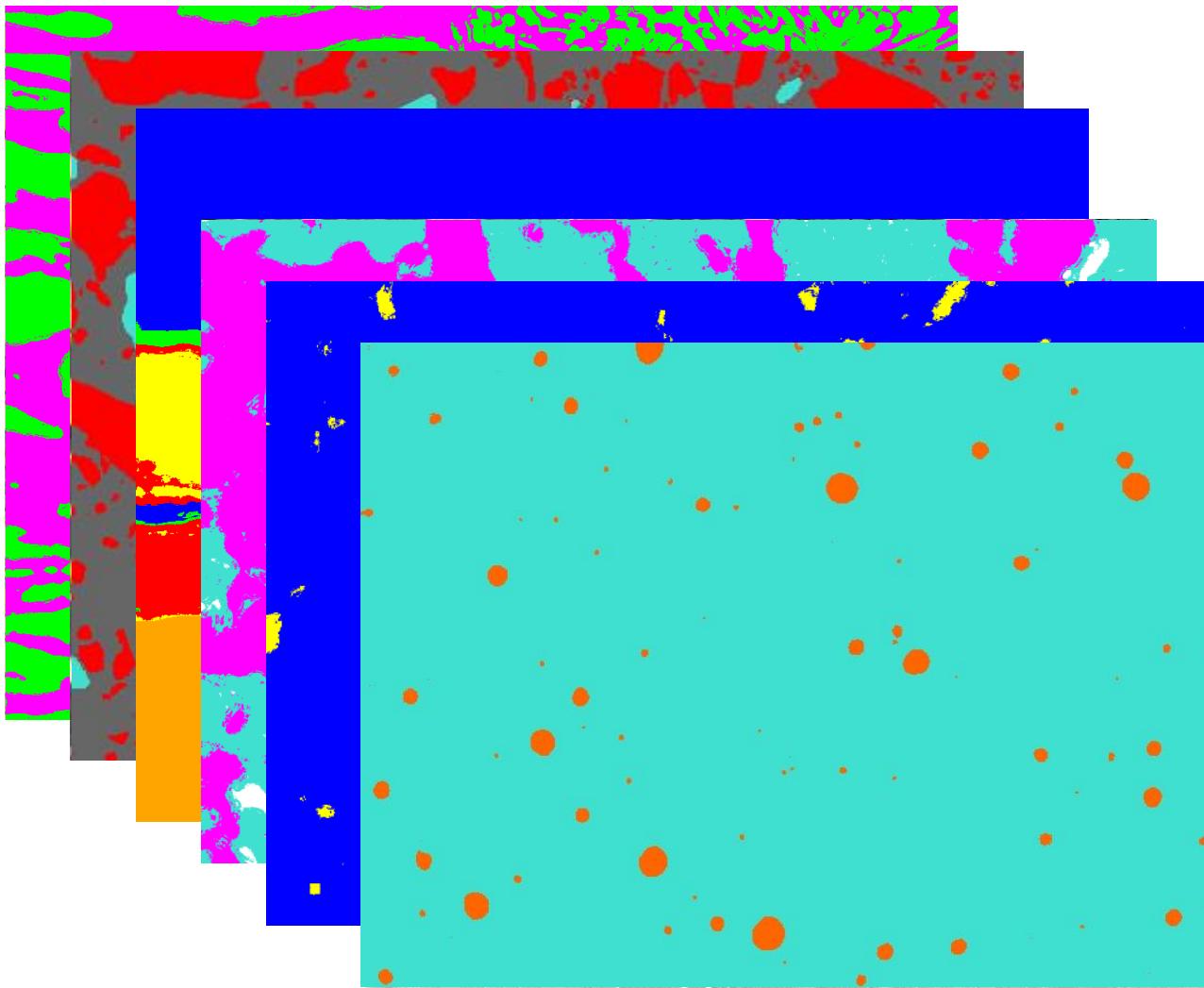


EM-Neurons



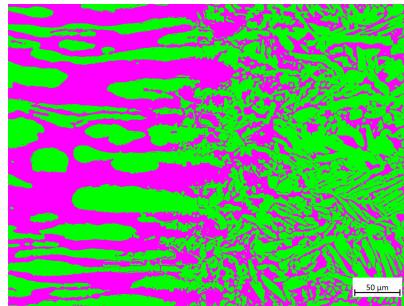
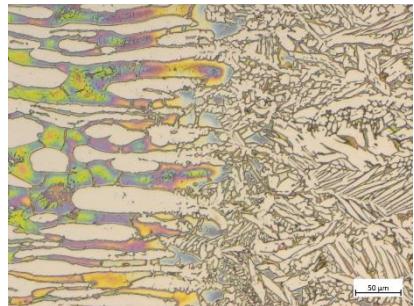
Applications:

Light and electron microscopy

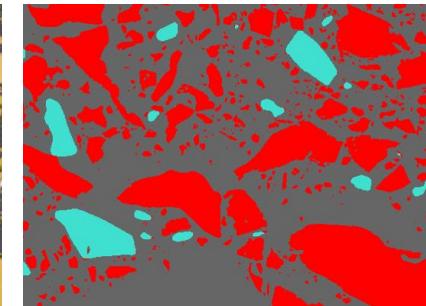
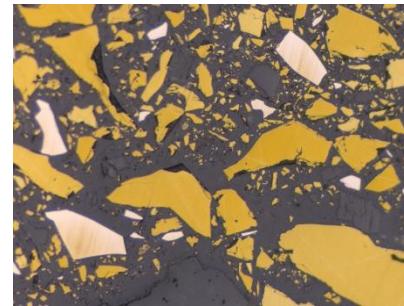


- Multiphase steels
- Ore and mineral samples
- Corrosion scale mapping
- Surface corrosion mapping
- Impurities or inclusions
- Porosity in ceramics

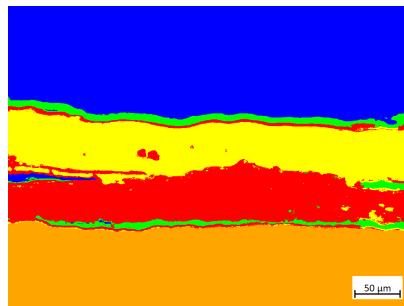
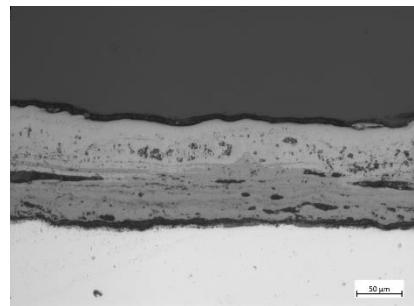
Intellessis Applications: Light and Electron Microscopy



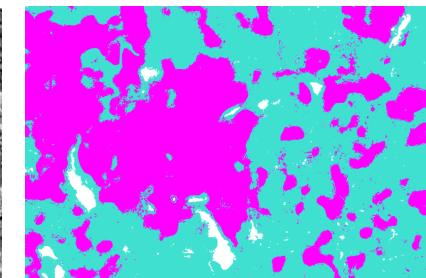
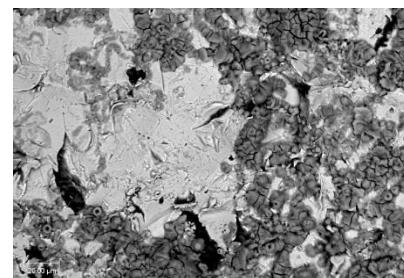
Multiphase steels



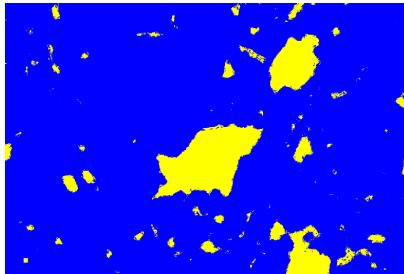
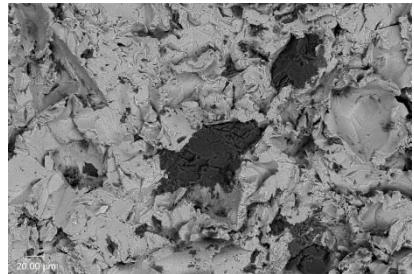
Ore and mineral samples



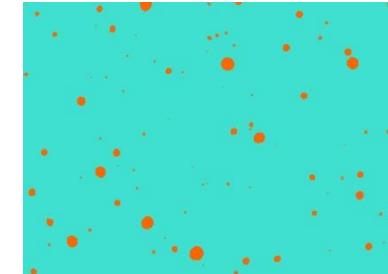
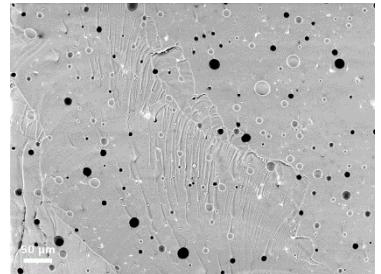
Corrosion scale mapping



Surface corrosion mapping



Impurities or inclusions



Porosity in ceramics

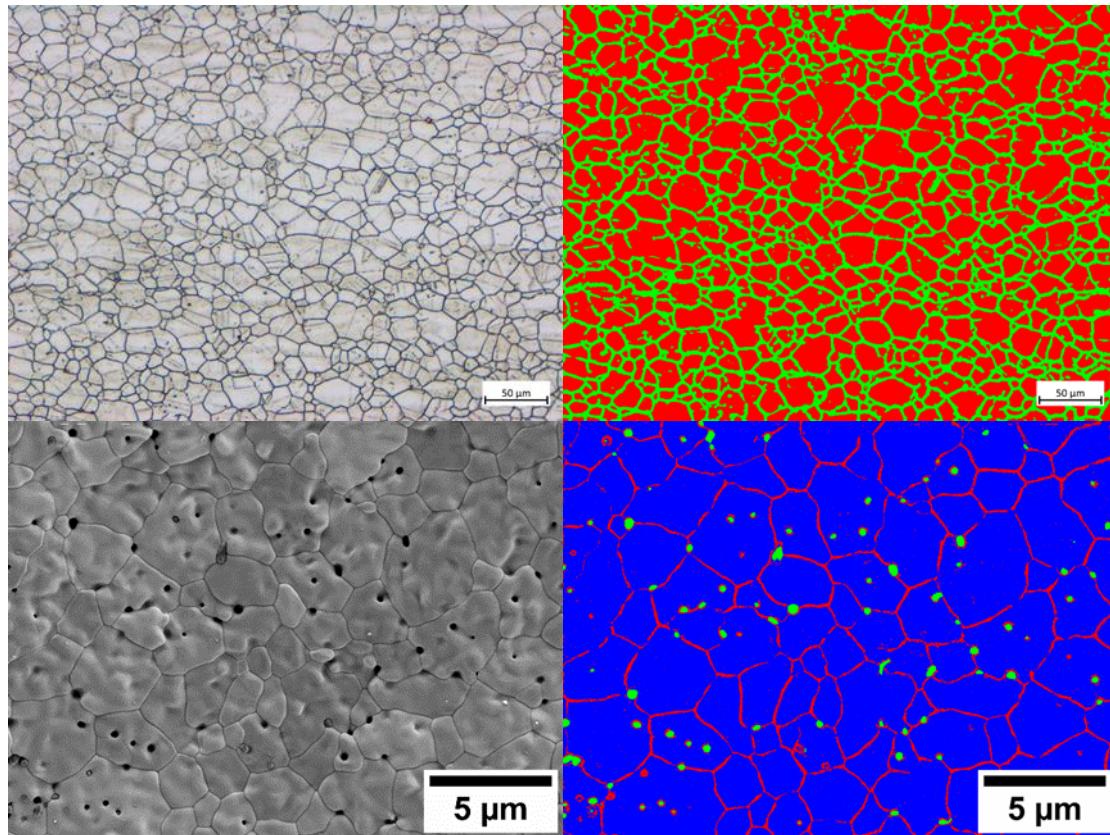
Applications:

Light and electron microscopy



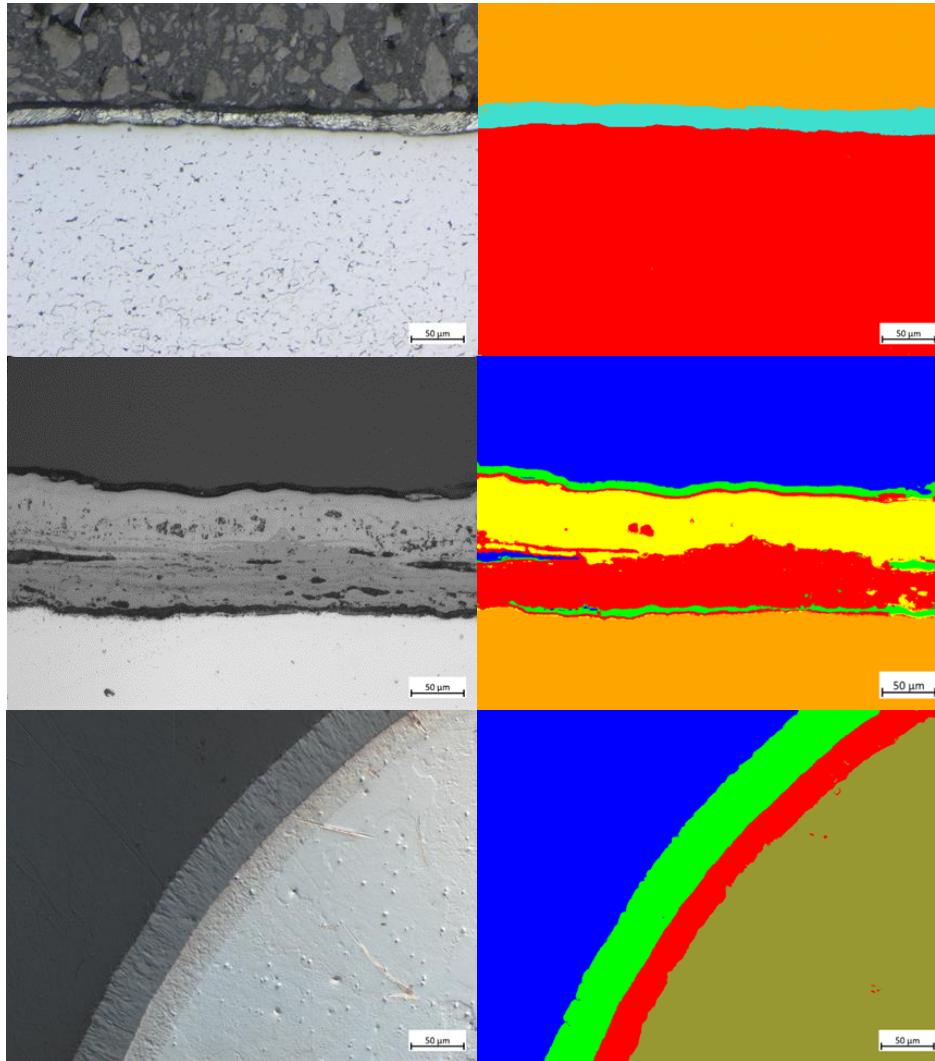
ZEISS ZEN Intellesis segmentation of metallographic samples.

- (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

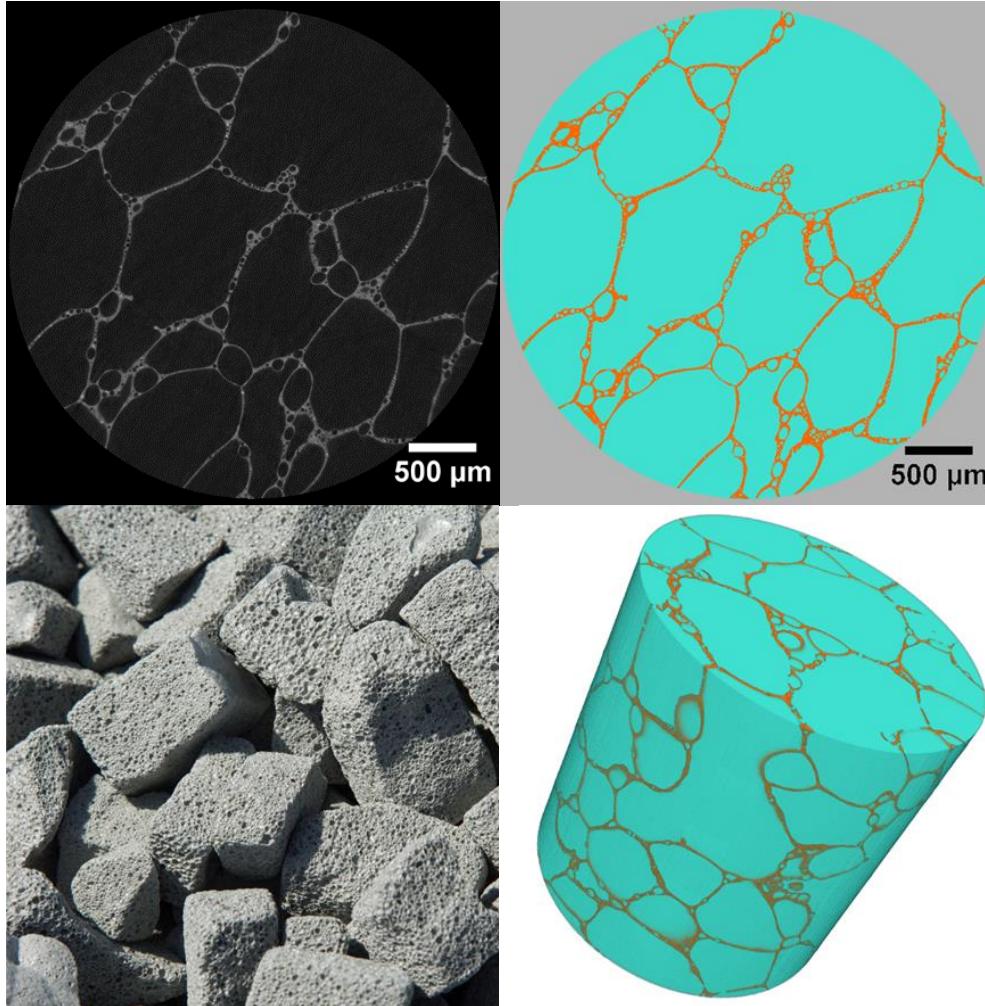


ZEISS ZEN Intellesis segmentation of coating cross-sections.

- 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

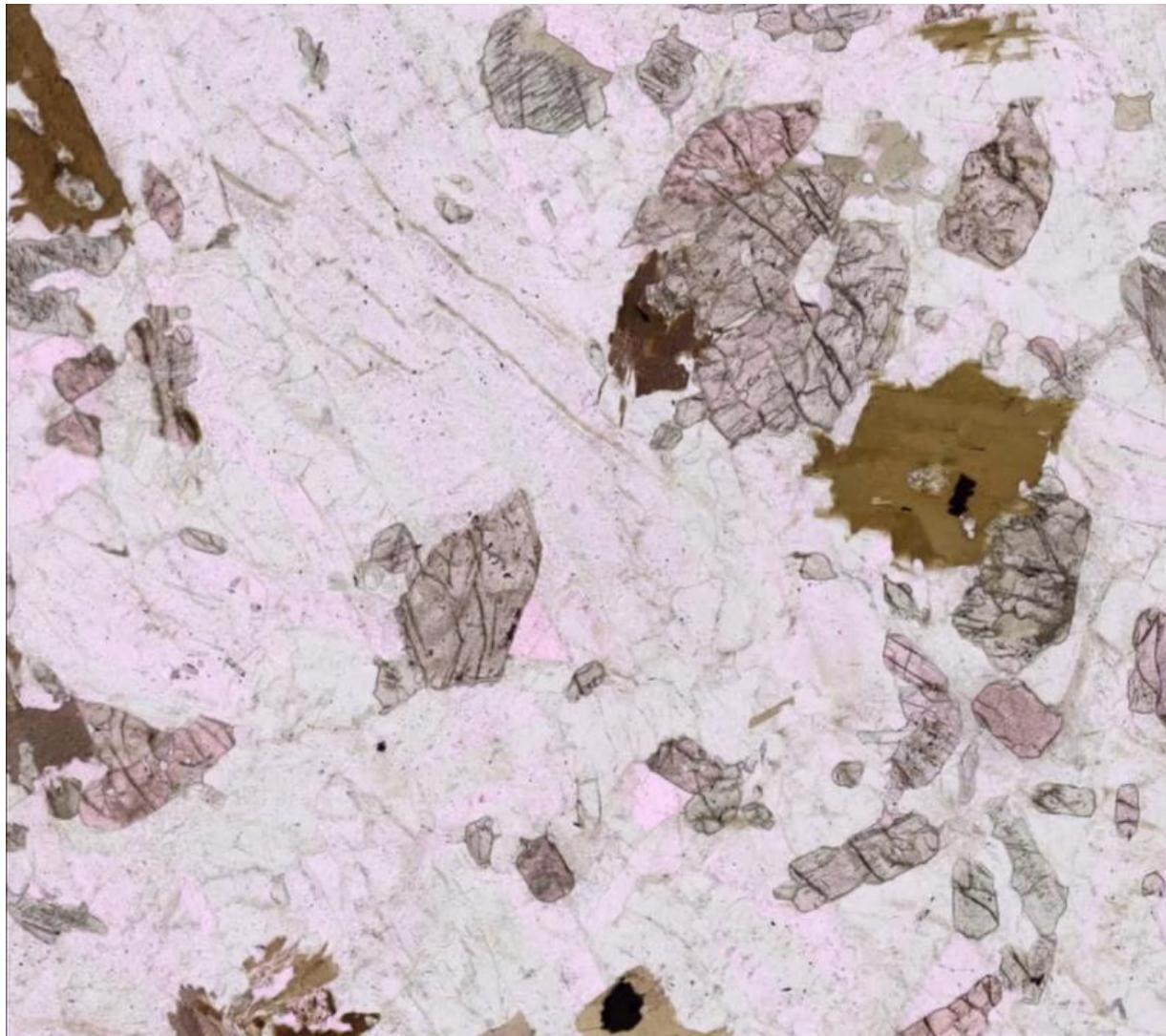


ZEISS ZEN Intellesis segmentation of a foam glass

- 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

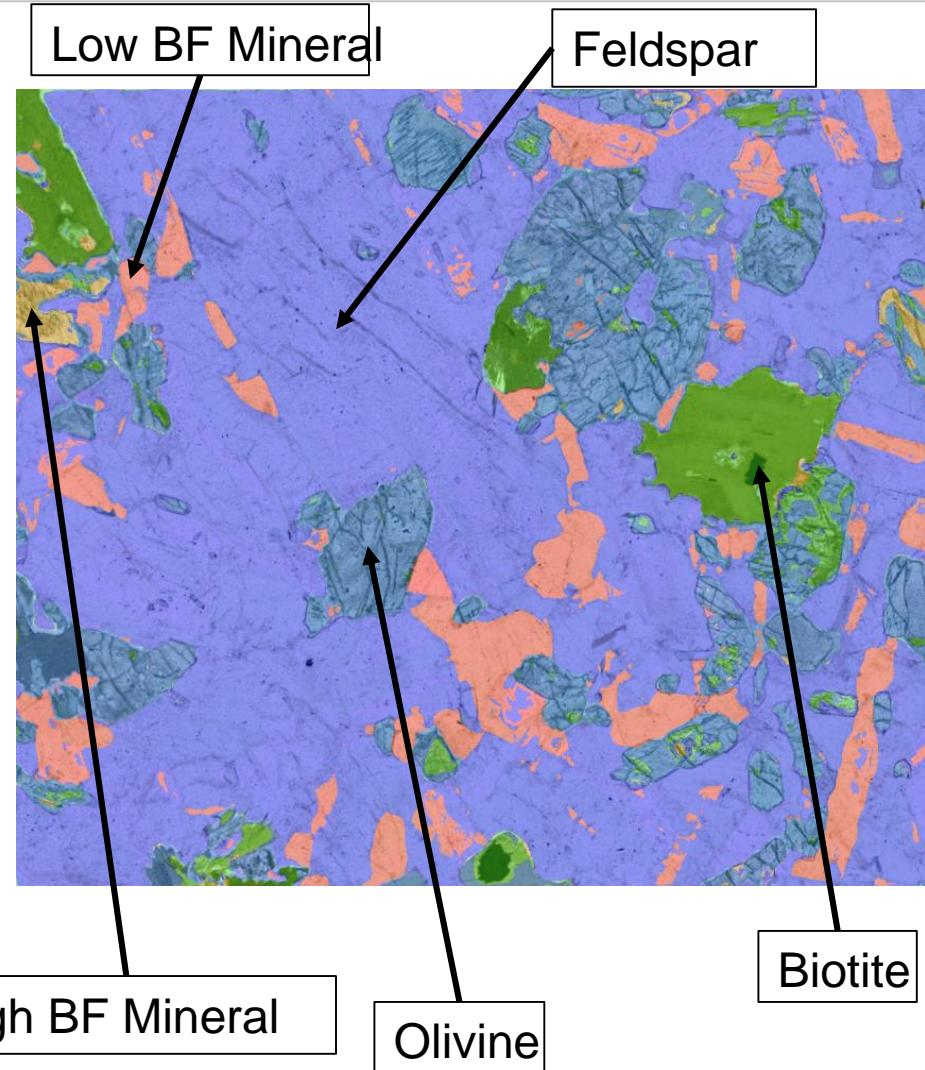
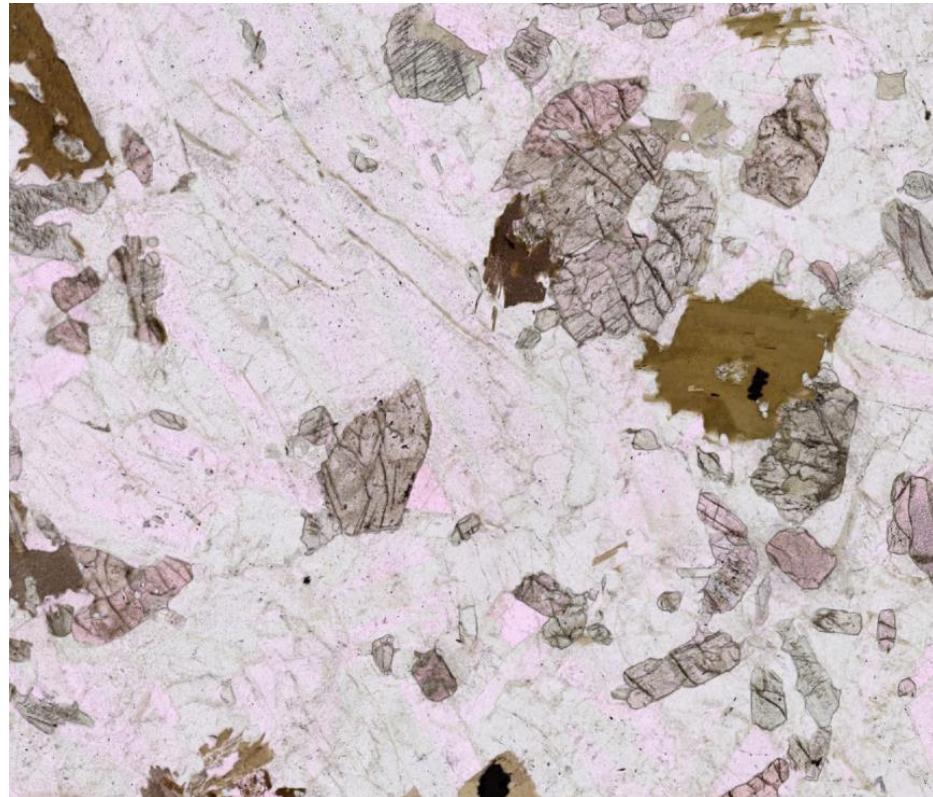
Applications: Multipolarised light microscopy



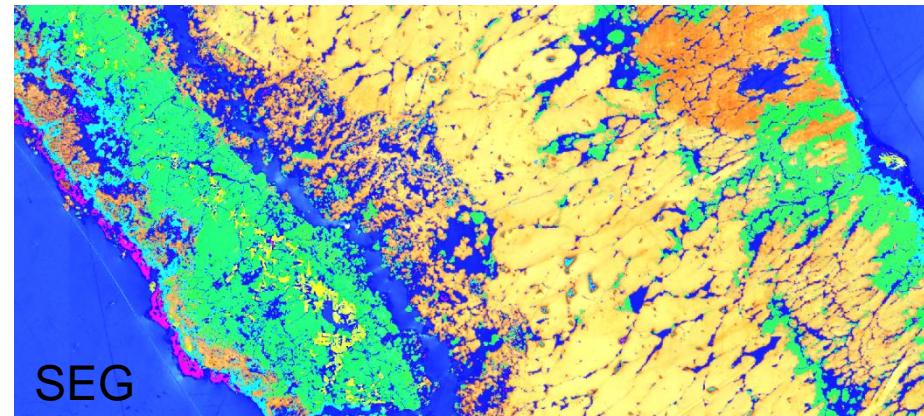
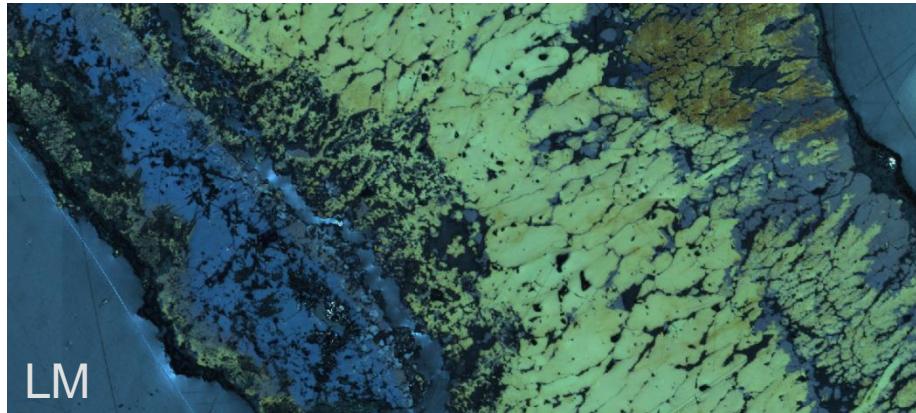
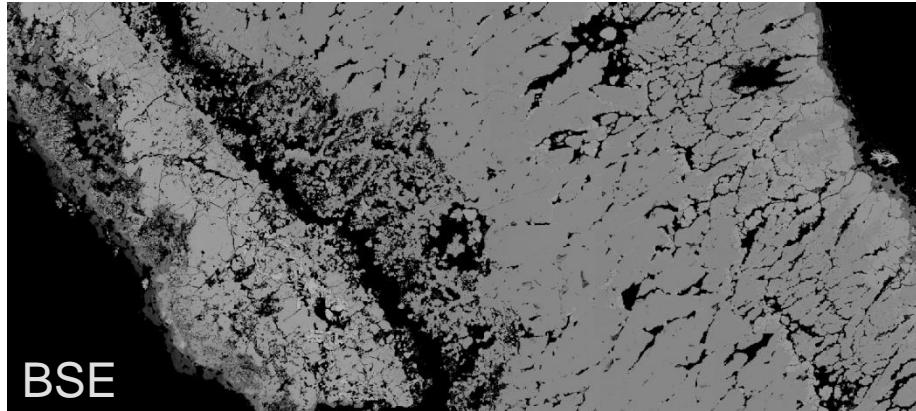
Each individual image can be considered a single channel

Applications: Multipolarised light microscopy

Segment structures not visible in one channel alone

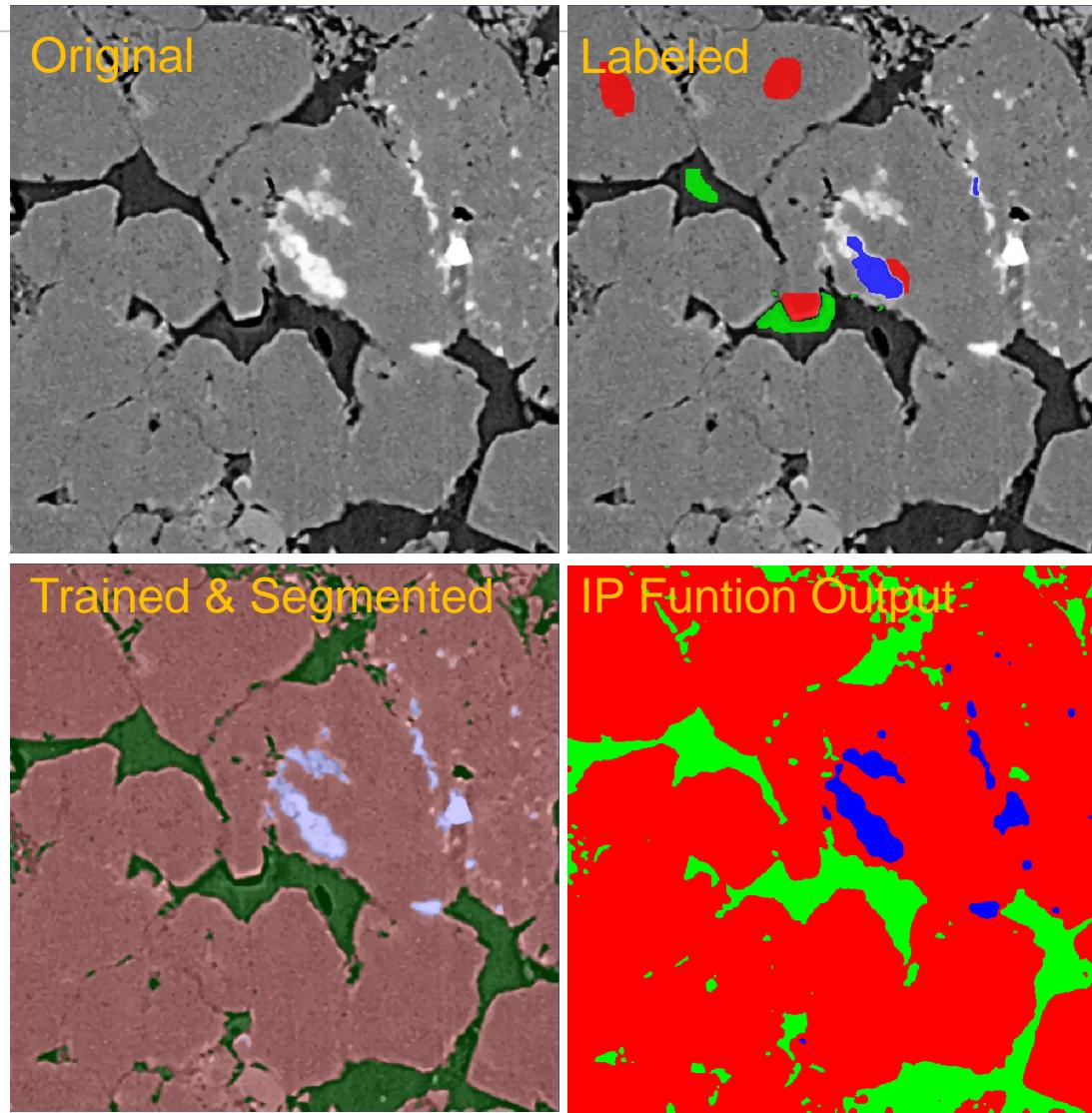


Correlative Data & ZEN Intellesis

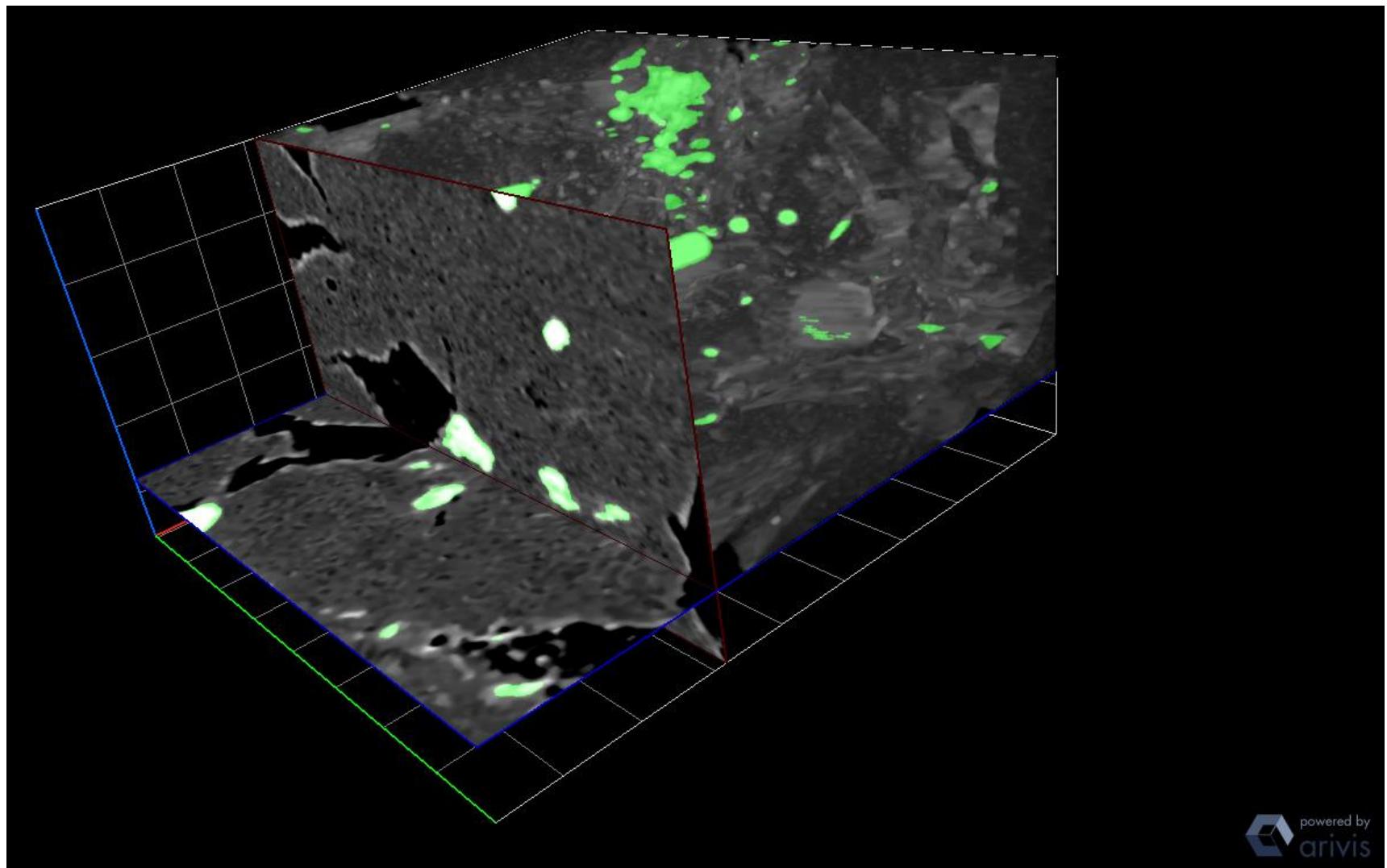


- Acquire BSE & LM data
- Correlate
- Use Correlative Data in Intellesis
- Train & Segment

XRM Sandstone – Segmentation of 3D dataset

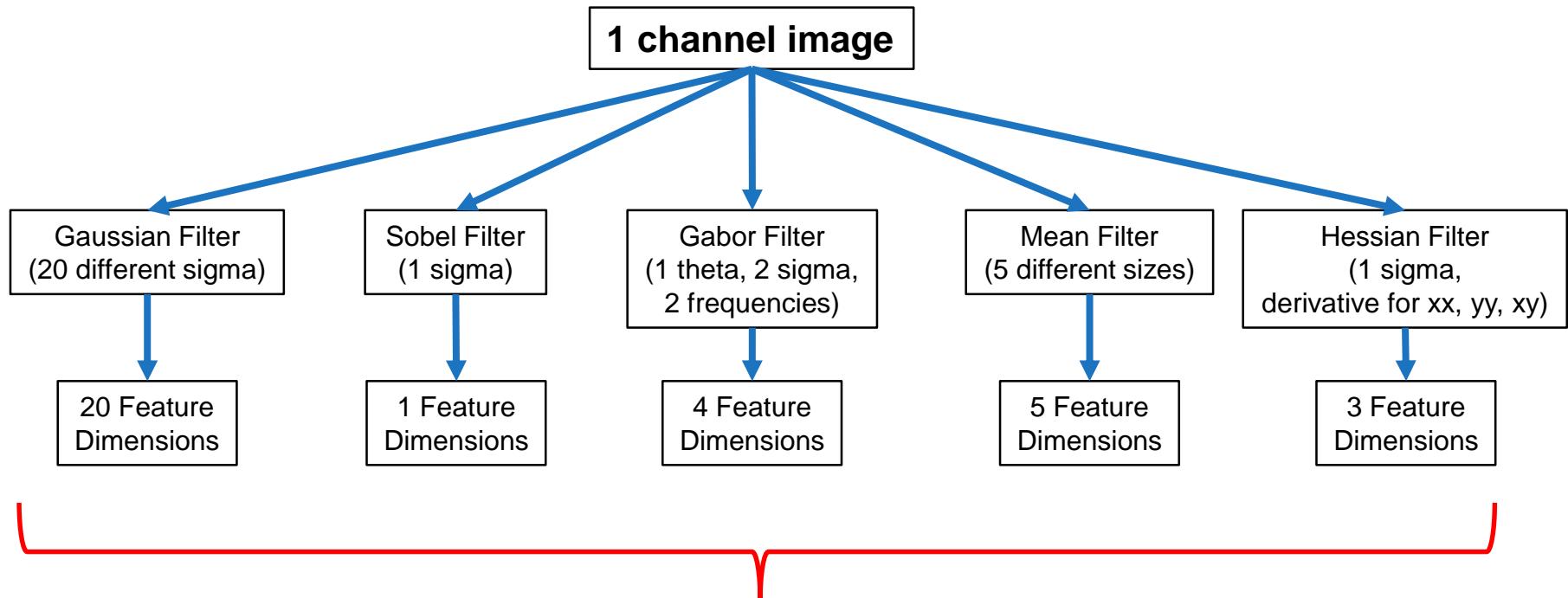


3D Segmentation Example: XRM – Sandstone inside 3DxL



powered by
arivis

- 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

Basic Features 33 (in ZEN 2.6 still called Default)

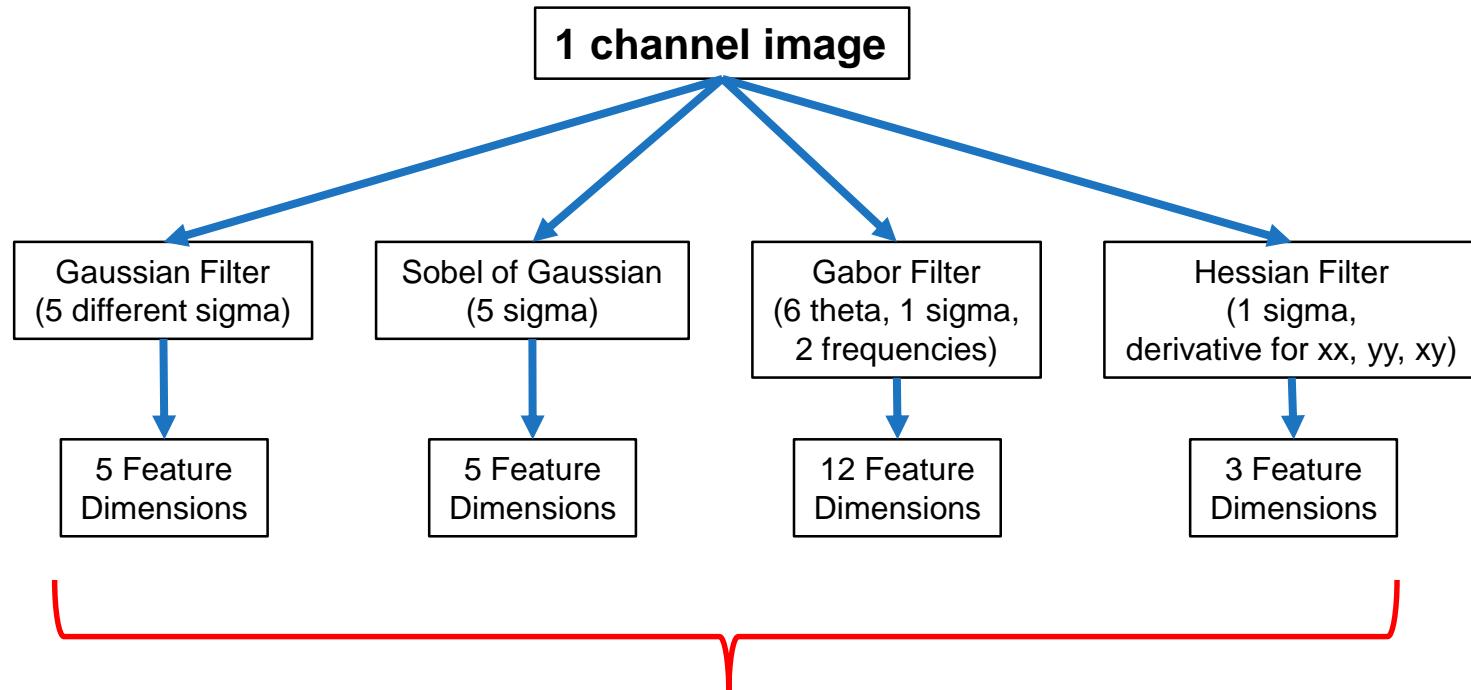
The user labels a number of pixels in the image as part of a Class. Classes could be anything that you are interested in – structures, holes, individual cells/grains etc. Multiple different Classes can be present in one image.

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.

- 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.



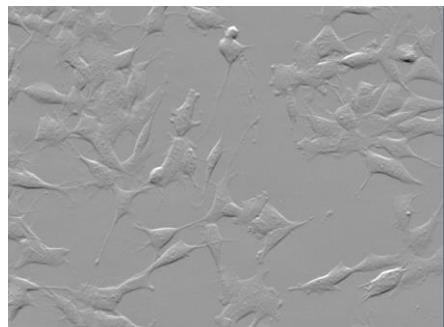
25 pieces of information per image pixel
= feature vector with 25 dimensions

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).

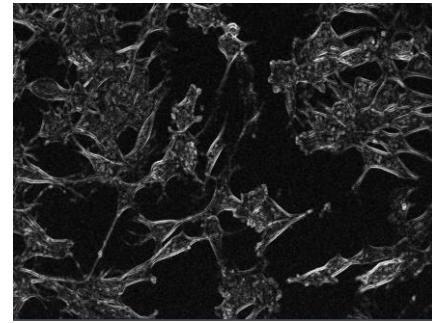
Feature Extractor Information



Original brightfield image

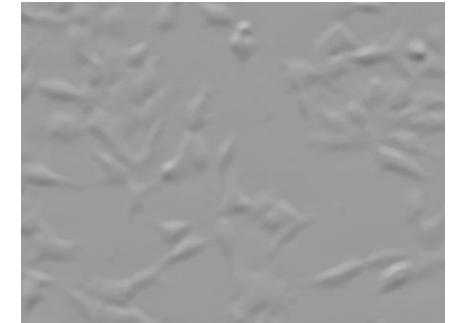


Gaussian Filter (Sigma 10)

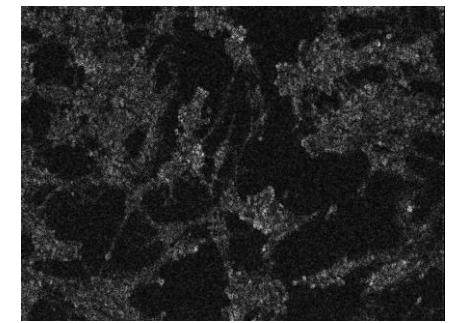


Sobel Filter (Clip)

...

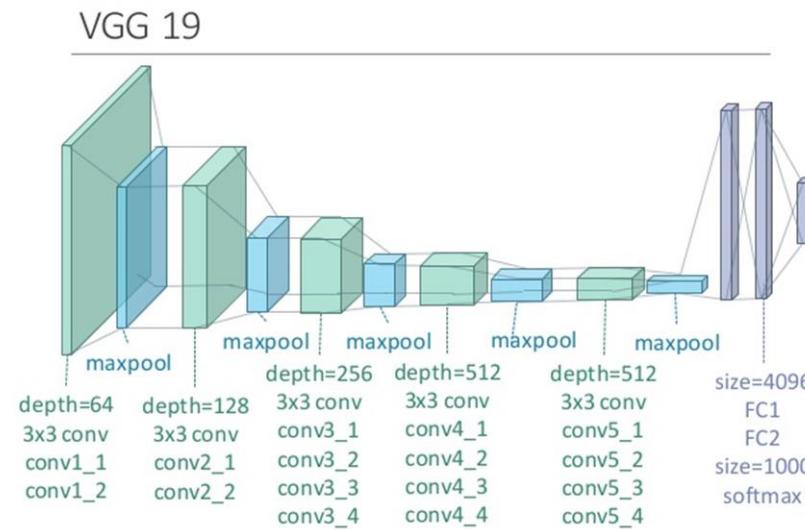


Mean Filter (10 Pixels)



Hessian Filter (smallest)

- 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 Features 50 - 256

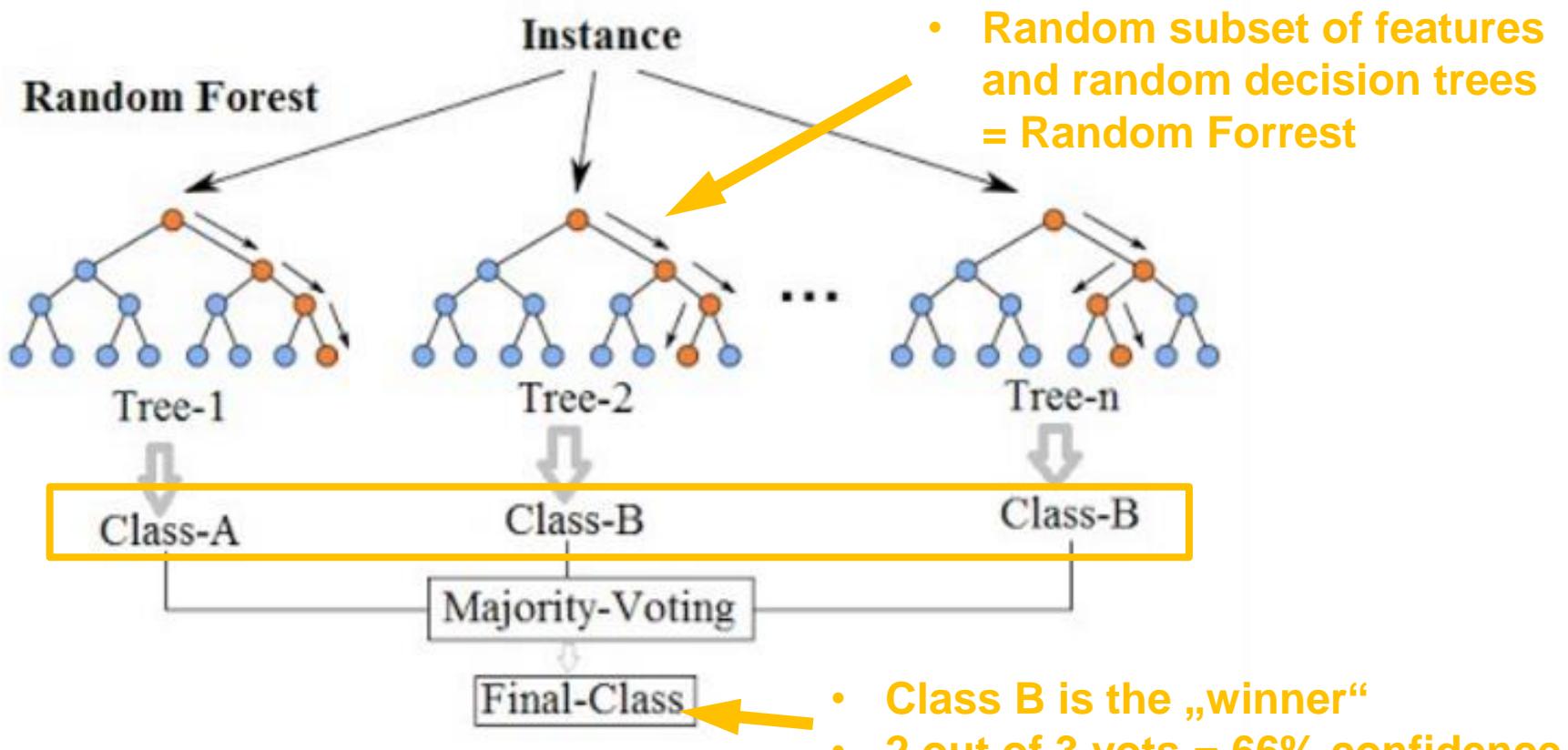
Deep Features 50 - 256 use the entire image as an input for the pre-trained network (VGG19). Each layer of this network becomes the input for the next layer as a feature vector, doubling the number of pieces of information per pixel. e.g. the output from Deep Features 3 will have been processed by the 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 Simplified



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

Random Forest Classifier approach



- 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](#)

CRF tries to create smoother and sharper 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

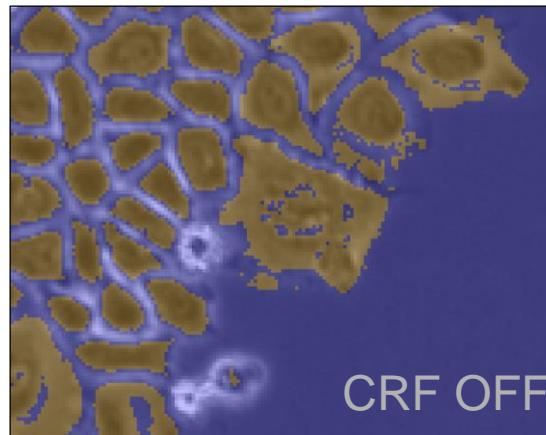
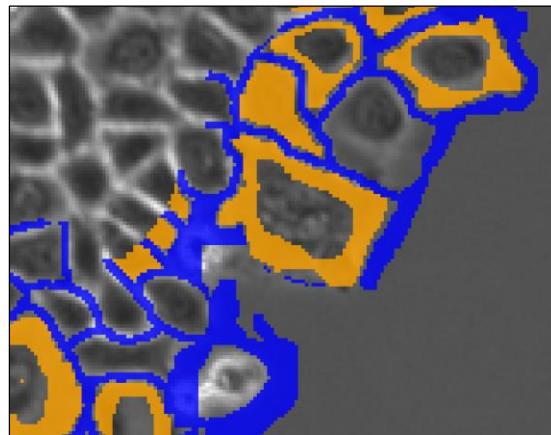
Conditional random fields (CRFs) are a class of statistical modeling method often applied in pattern recognition and machine learning and used for structured prediction. CRFs fall into the sequence modeling family. Whereas a discrete classifier predicts a label for a single sample without considering "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](#)

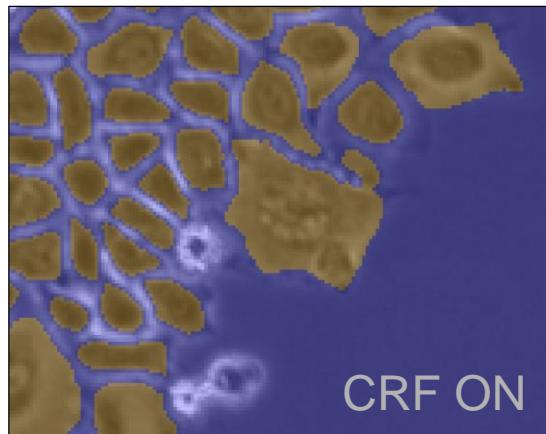
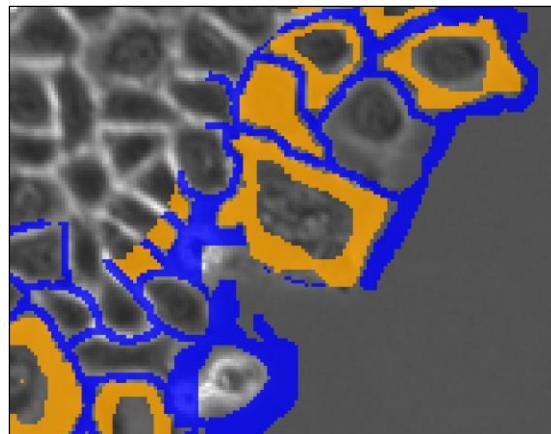
Postprocessing – Conditional Random Fields (CRF)



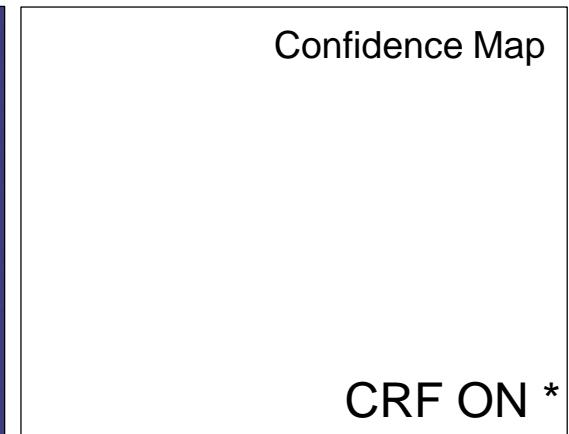
CRF OFF



Confidence Map



CRF ON



Confidence Map

CRF ON *

* - 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



The screenshot displays the ZEN 2.5 software interface for image analysis. The main window shows a grayscale image of nanoparticles with red outlines indicating segmented regions. A detailed table on the right lists 33 individual particles with their geometric and morphological parameters. The left sidebar contains various analysis tools and a trainable segmentation model named 'Nanoparticle Segmentation ZEN25'. The bottom navigation bar includes system information like RAM usage and storage details.

ID	Area [pixel ²]	Circularity	Convexity	Feret Ratio	Perimeter [pixel]	Roundness	
1	A	B	C	D	E	F	G
2	463.000	0.787	0.958	0.775	96.669	0.619	
3	520.000	0.859	0.948	0.901	97.497	0.737	
4	315.000	0.833	1.010	0.832	72.184	0.694	
5	1,684.000	0.920	0.755	0.938	186.652	0.846	
6	1,061.000	0.871	0.803	0.877	147.681	0.759	
7	105.000	0.802	0.994	0.868	45.213	0.643	
8	226.000	0.839	1.050	0.792	59.355	0.704	
9	952.000	0.760	0.819	0.705	144.953	0.578	
10	765.000	0.864	0.729	0.830	128.782	0.746	
11	349.000	0.756	0.938	0.735	80.912	0.571	
12	300.000	0.822	1.044	0.799	68.770	0.676	
13	838.000	0.795	0.849	0.787	134.711	0.631	
14	1,030.000	0.856	0.938	0.847	132.125	0.733	
15	286.000	0.692	0.903	0.615	77.012	0.479	
16	396.000	0.721	0.700	0.741	108.326	0.520	
17	611.000	0.898	1.007	0.901	97.255	0.806	
18	250.000	0.748	0.963	0.659	69.012	0.559	
19	627.000	0.644	0.874	0.558	127.640	0.415	
20	413.000	0.829	0.875	0.861	90.326	0.687	
21	2,225.000	0.883	0.898	0.865	197.480	0.779	
22	1,137.000	0.909	0.966	0.898	135.640	0.826	
23	111.000	0.591	0.968	0.443	51.496	0.350	
24	1,699.000	0.854	0.976	0.801	165.296	0.729	
25	1,085.000	0.686	0.675	0.603	183.439	0.471	
26	660.000	0.890	0.859	0.897	111.397	0.793	
27	1,145.000	0.760	0.944	0.670	149.782	0.577	
28	682.000	0.915	0.932	0.928	107.154	0.837	
29	1,115.000	0.868	0.865	0.845	144.368	0.753	
30	517.000	0.740	0.895	0.609	102.912	0.548	
31	297.000	0.783	0.793	0.789	86.083	0.613	
32	374.000	0.841	0.947	0.790	79.497	0.708	
33	296.000	0.674	0.747	0.592	95.497	0.455	

Complete Automation of Workflow



Size Distribution.png - ZEN 2.5 system

File Edit View Acquisition Graphics Macro Tools Window Help

Processing Analysis Applications Extensions

Interactive Measurement Image Analysis

Nanoparticles.tif* Size Distribution.png

Setting Nanoparticle Segmentation ZEN25

Base 0
Statistics Nanopar 1
Nanoparticles 2
Statistics Backgroun 3
Background 4
Statistics Boundar 5
Boundaries 6

Setup Image Analysis

Analyze Interactively

Analyze

Intelless Trainable Segmentation

Model Nanoparticle Segmentation ZEN25

Description

Start Training

Dimensions Graphics Custom Graphics

Zoom 100% Auto Fit

Tools Navigator Interpolation

Channels C1 Spline Mode

Single Channel Range Indicator Quick Color Setup

Display 2D Display Overlay Options ROI Tracing

C1 Spline Mode

Auto Min/Max Best Fit 2.00 0.01 Current Reset

Black 0 Gamma 1.00 White 255

Scaling: 1 px(px) (measured)

System Information Idle

Free RAM 4,06 GB Free HD 41,21 GB CPU 61 % Frame Rate: fps

Pixel Value: Position: Storage Folder: User:

Parameter: -f "C:\Users\MSR\OneDrive - Carl Zeiss AG\Python_Notebooks\RML\Nanoparticles\Data\Size Distribution.csv"

Workspace Zoom Reset Design Dark Workspace

Images and Documents

Container 1

Nanoparticles.tif* 550.62 kB / 158.86 kB

Size Distribution.png 1.17 MB / 16.68 kB

Macro Editor

New Macro Record Run Debug Help

RML_Nanoparticle_Analysis

```
72 parameter2display = 'Area::AreaIR'
73 title = 'Size Distribution'
74 xlabel = 'Area [micron**2]'
75 bins = 10
76 savename = 'Size-Distribution.png'
77 savename_complete = Path.Combine(Path.GetDirectoryName(csvfile), savename)
78 print 'Savename: ', savename_complete
79
80 # construct the command line string
81 params = '-f "' + csvfile + '" -p ' + parameter2display + ' -sp Fa'
82 print 'Parameter: ', params
83
84 # start the data display script as an external application
85 app = Process();
86 app.Start();
87 app.StartInfo.Arguments = script + params;
88 app.Start();
89 app.WaitForExit();
90
91 print 'Trying to show figure in ZEN.'
92
93 if File.Exists(savename_complete):
94     plotfigure = Zen.Application.LoadImage(savename_complete, False)
95     Zen.Application.Documents.Add(plotfigure)
96 else:
97     print 'Saved figure not found.'
98
99 print 'Done and Exit.'
```

Watch Message

Clear all

('Saving table for class: ' Nanoparticles.csv')
('Saving table for class: ' Statistics Nanoparticles.csv)
('Saving table for class: ' Background.csv')
('Saving table for class: ' Statistics Background.csv)
('Saving table for class: ' Boundaries.csv)
('Saving table for class: ' Statistics Boundaries.csv)

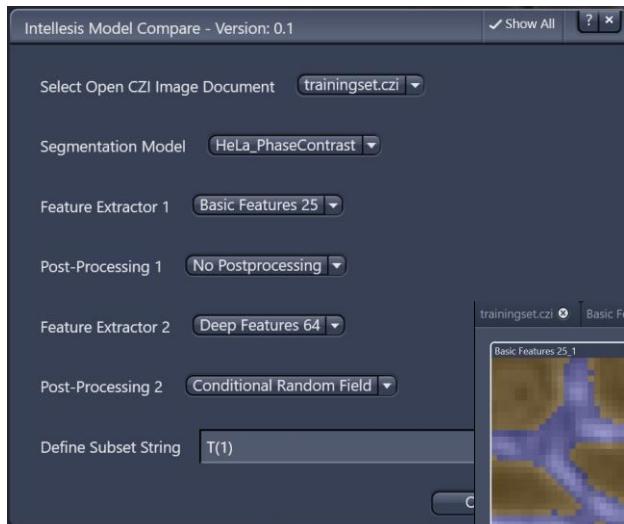
Savename: c:\Users\MSR\OneDrive - Carl Zeiss AG\Python_Notebooks\RML\Nanoparticles\Data\Size Distribution.csv

Parameter: -f "C:\Users\MSR\OneDrive - Carl Zeiss AG\Python_Notebooks\RML\Nanoparticles\Data\Size Distribution.csv"

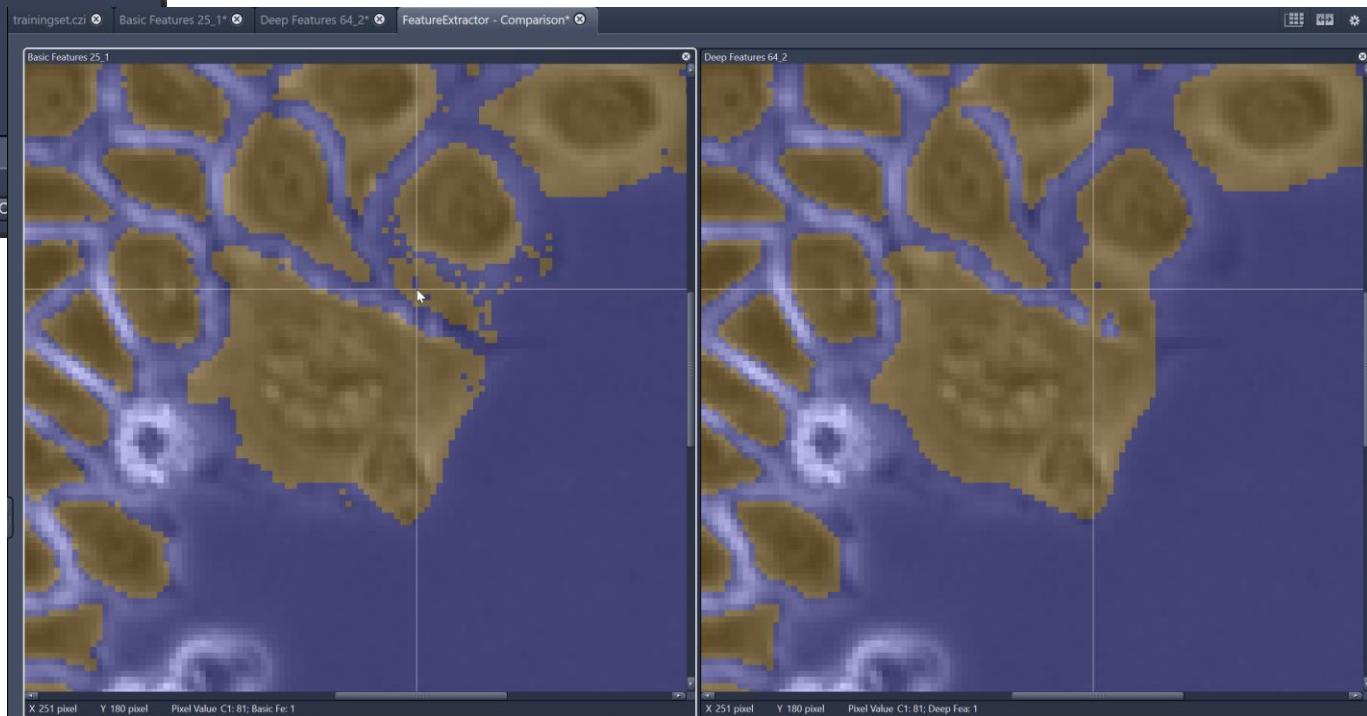
Try to show figure in ZEN.

Done and Exit.

Compare Feature Extractors



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



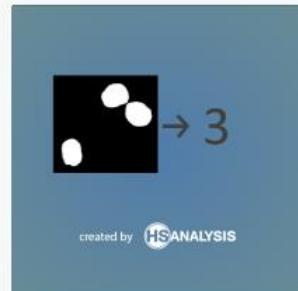
<https://www.zeiss.com/microscopy/int/website/landingpages/zen-intellesis.html>

Get the trial licence and try it out!

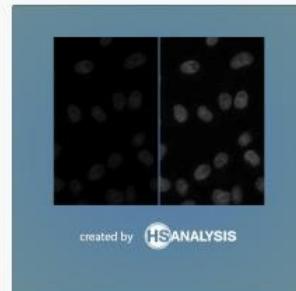
Integration in APEER platform

[Workflows](#)[Modules](#)[My Workspace](#)**Categories**

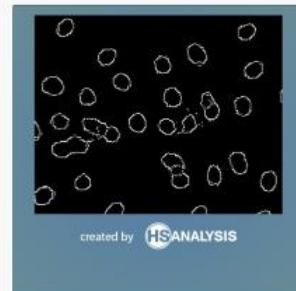
- Visualization (1)
- Analysis (8)
- Processing (8)
- Input (6)
- Tools (0)



Version 1.2
Blob Counter



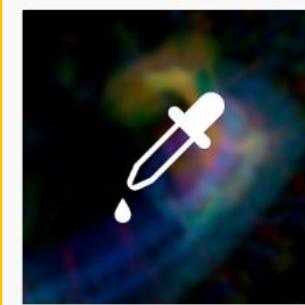
Version 1.1
Adaptive Histogram Equalization



Version 1.0.0
Preprocessing

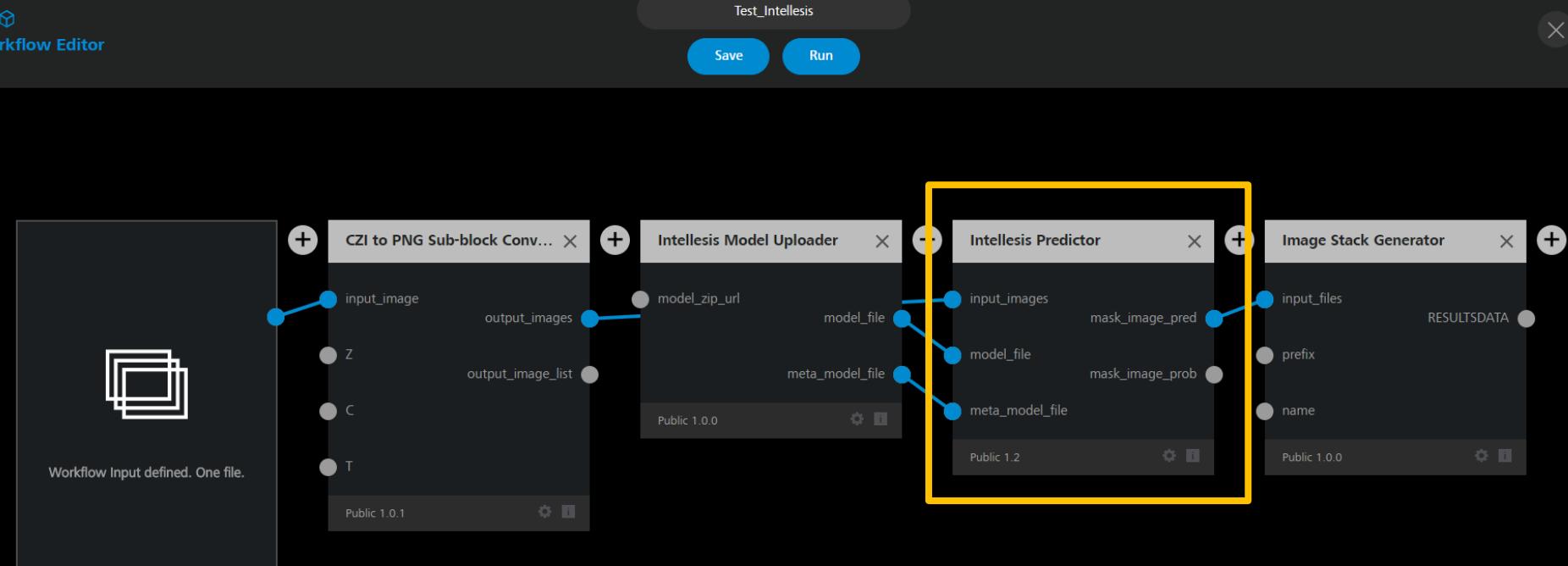


Version 1.1
Thresholding



Version 1.0.0
Tint Image

Intellesis - Integration in APEER platform



Intellesis - Integration in APEER platform



APEER

Sebastian Rhode ▾

Workflows

Modules

My Workspace

11

My Workflows

5

My Modules

13

My Results

Created on Mar 22, 2018 | Created By Sebastian Rhode

Private



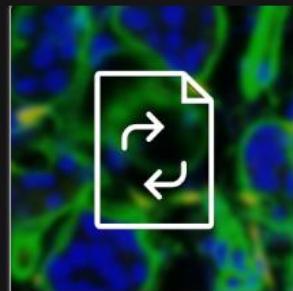
Test_Intellesis

Run Workflow

Publish Workflow

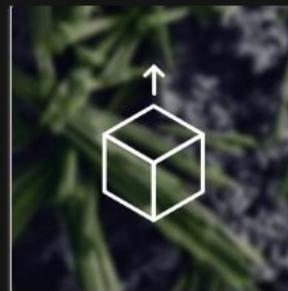
Edit Workflow

4 Modules in this Workflow



Version 1.0.1
CZI to PNG Sub-block Converter

Created by Alexander Urich



Version 1.0.0
Intellesis Model Uploader

Created by Christian Kungel



Version 1.2
Intellesis Predictor

Created by Christian Kungel



Version 1.0.0
Image Stack Generator

Created by Robert Kirmse



We make it visible.