

**BIO-INSPIRED
MATERIALS**

NATIONAL CENTER OF COMPETENCE
IN RESEARCH

Introduction to ImageJ

Session 3: Thresholding, segmentation and (particle) size analysis

Dimitri Vanhecke

**UNI
FR**

UNIVERSITÉ DE FRIBOURG
UNIVERSITÄT FREIBURG



adolphe merkle institute
excellence in pure and applied nanoscience

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE



**UNIVERSITÉ
DE GENÈVE**

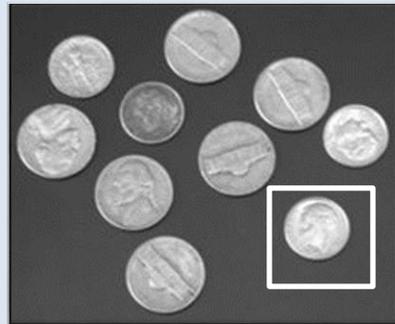


SWISS NATIONAL SCIENCE FOUNDATION

Quantify objects in an image

Lecture 1

Original grayscale



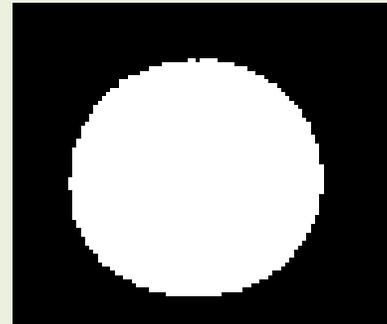
Lecture 2

Image processing (Lecture 2)



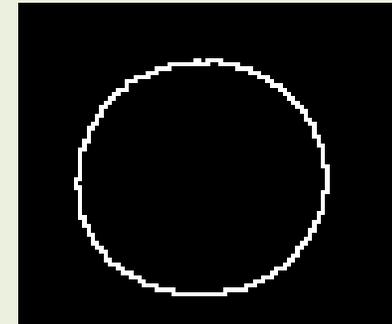
Lecture 3

Thresholding Chapter 3

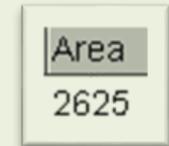


(=segmentation,
binarisation,
masking, ...)

Morphological filtering Chapter 2



Measuring Chapter 1

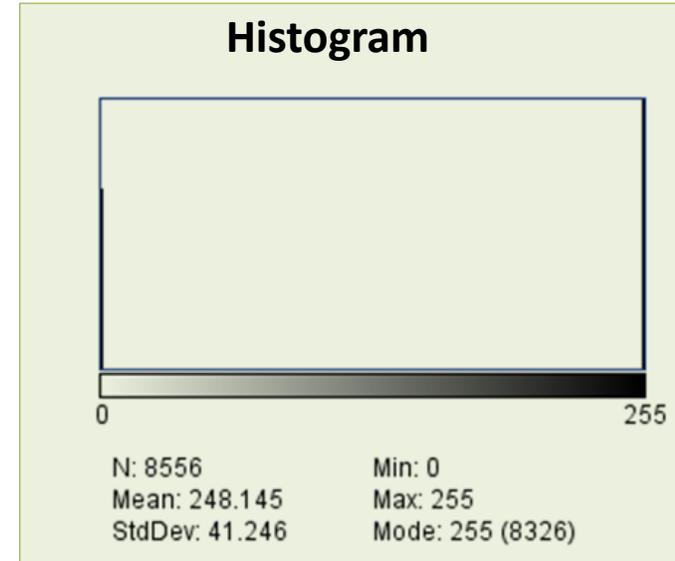
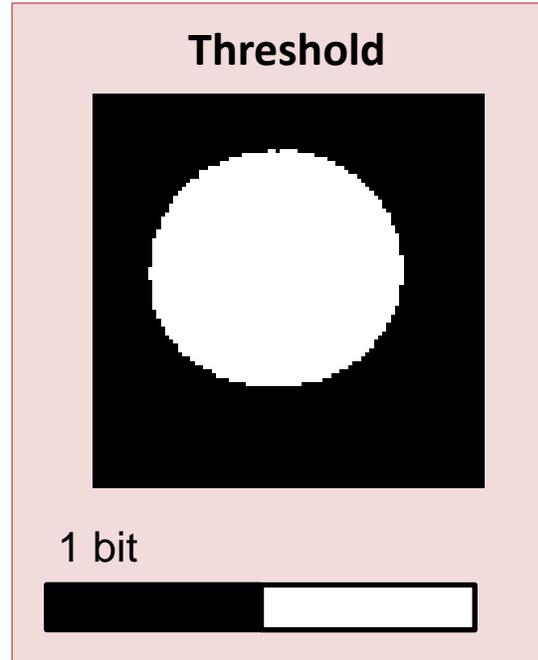
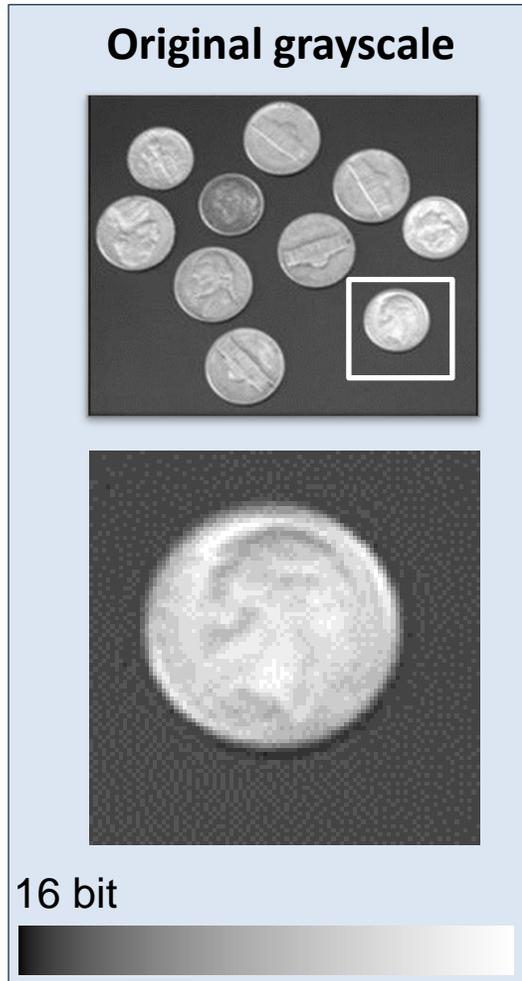




How does software measure objects in images?

How does a software measure images?

Primary units: **Area** of an object



of pixels = area

Analyse > Measure
(= Count pixels)

count	
2625	

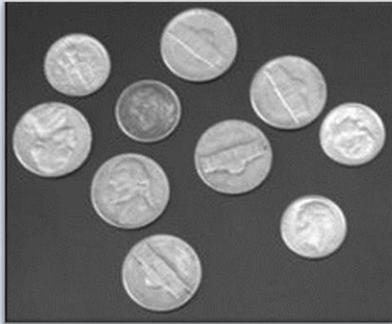
by Analyse > Particles)

Area
2625

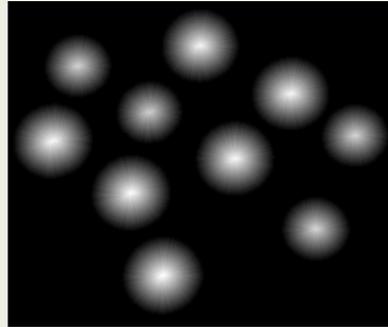
How does a software measure images?

Primary units: **Count** objects

Grayscale

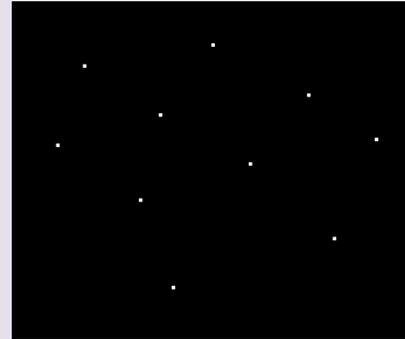


Distance transform



= a derived representation where every foreground pixel takes a value in function of the distance to the nearest background pixel

Max eroded points



= endpoints after iterative erosion until only single-pixel lines remain.

Histogram



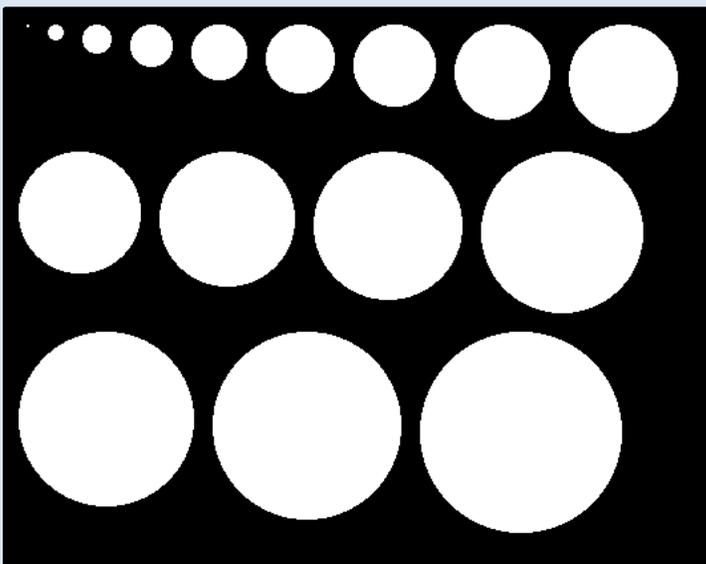
N: 105610 Min: 0
Mean: 254.976 Max: 255
StdDev: 2.481 Mode: 255 (105600)

Get # of black pixels
= # Objects

value	count
255	10

How does a software measure images?

Primary units: **Length** (e.g. **perimeter** of an object) --> tricky (estimates)



Area	Perim.	Circ.
4	5.657	1.000
112	38.042	0.973
384	71.012	0.957
812	103.983	0.944
1396	136.953	0.935
2128	169.924	0.926
3024	202.894	0.923
4060	235.865	0.917
5284	268.836	0.919
6668	304.149	0.906
8184	337.120	0.905
9856	370.090	0.904
11684	403.061	0.904
13692	436.032	0.905
15856	469.002	0.906
18168	501.973	0.906

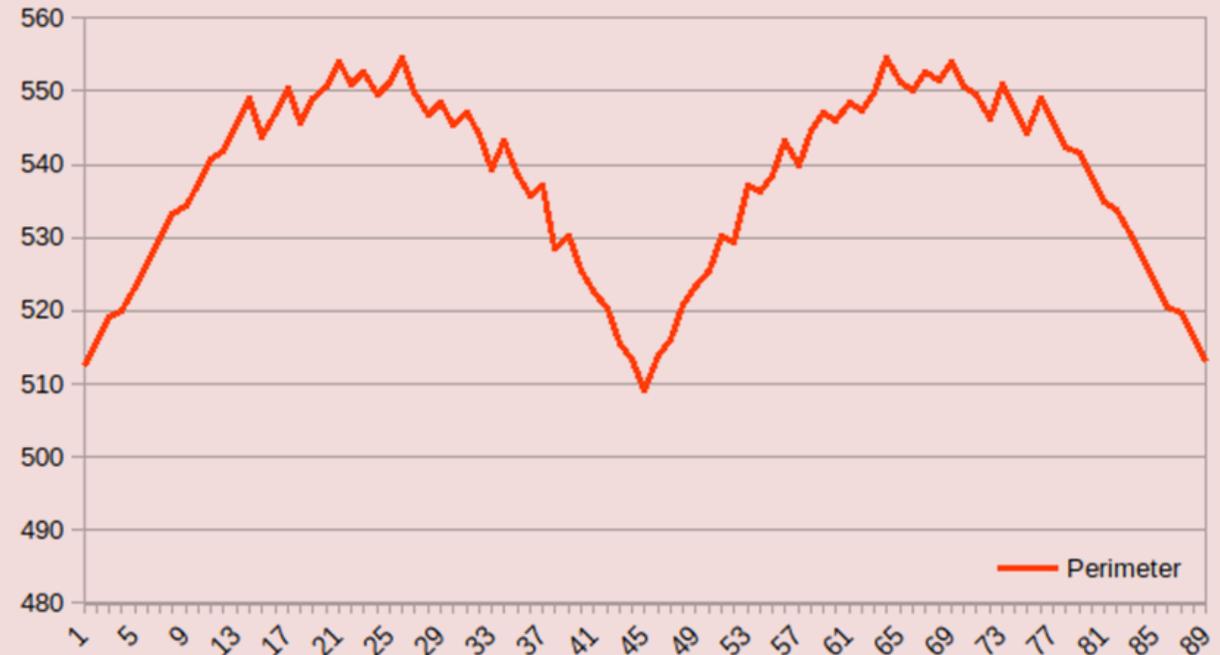
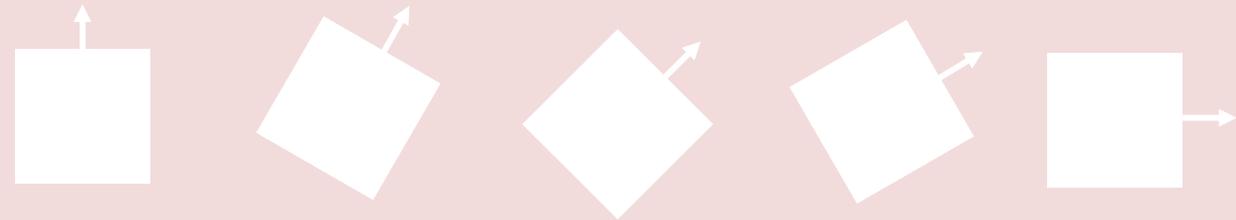
$$Circ. = \frac{4\pi \times Area}{Perimeter^2}$$



“Perfect circles” do not have a circularity of 1

Of course, these are discrete representations of perfect circles (see lecture 1)

The perimeter of an object (a 128x128 square) depends on its angular position.



How does a software measure images?

Primary units: **Length** (e.g. **perimeter** of an object) --> tricky (estimates)



Boundary pixels

0	0	0
0	1	0
0	0	0

Euclidian

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

1.41	1	1.41
1	0	1
1.41	1	1.41

Manhattan

$$|x_1 - x_2| + |y_1 - y_2|$$

2	1	2
1	0	1
2	1	2

Chebyshev
(=Checkerboard king)

$$\max(|x_1 - x_2|, |y_1 - y_2|)$$

1	1	1
1	0	1
1	1	1

of edge pixels \cong perimeter

Perim.
191.823

Threshold + Erosion

232

Skeletonization

163

LoG $\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$

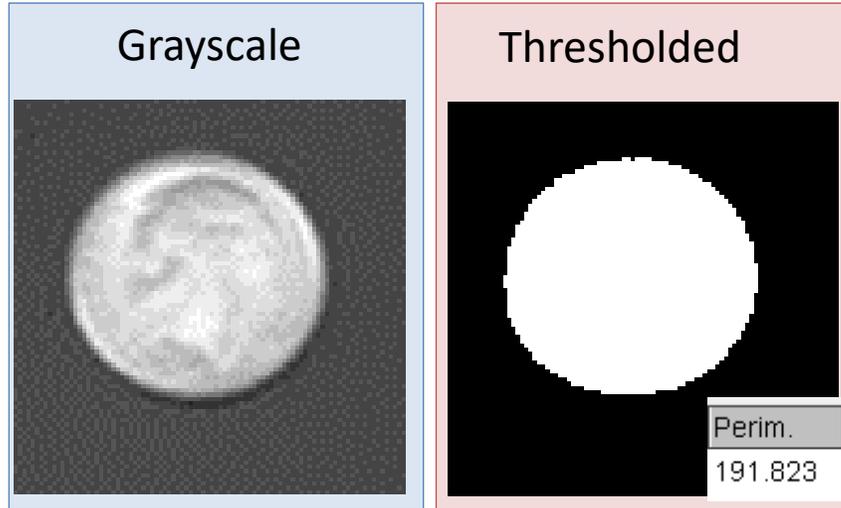
166

LoG $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$

234

How does a software measure images?

Primary units: perimeter of an object. the Crofton estimator

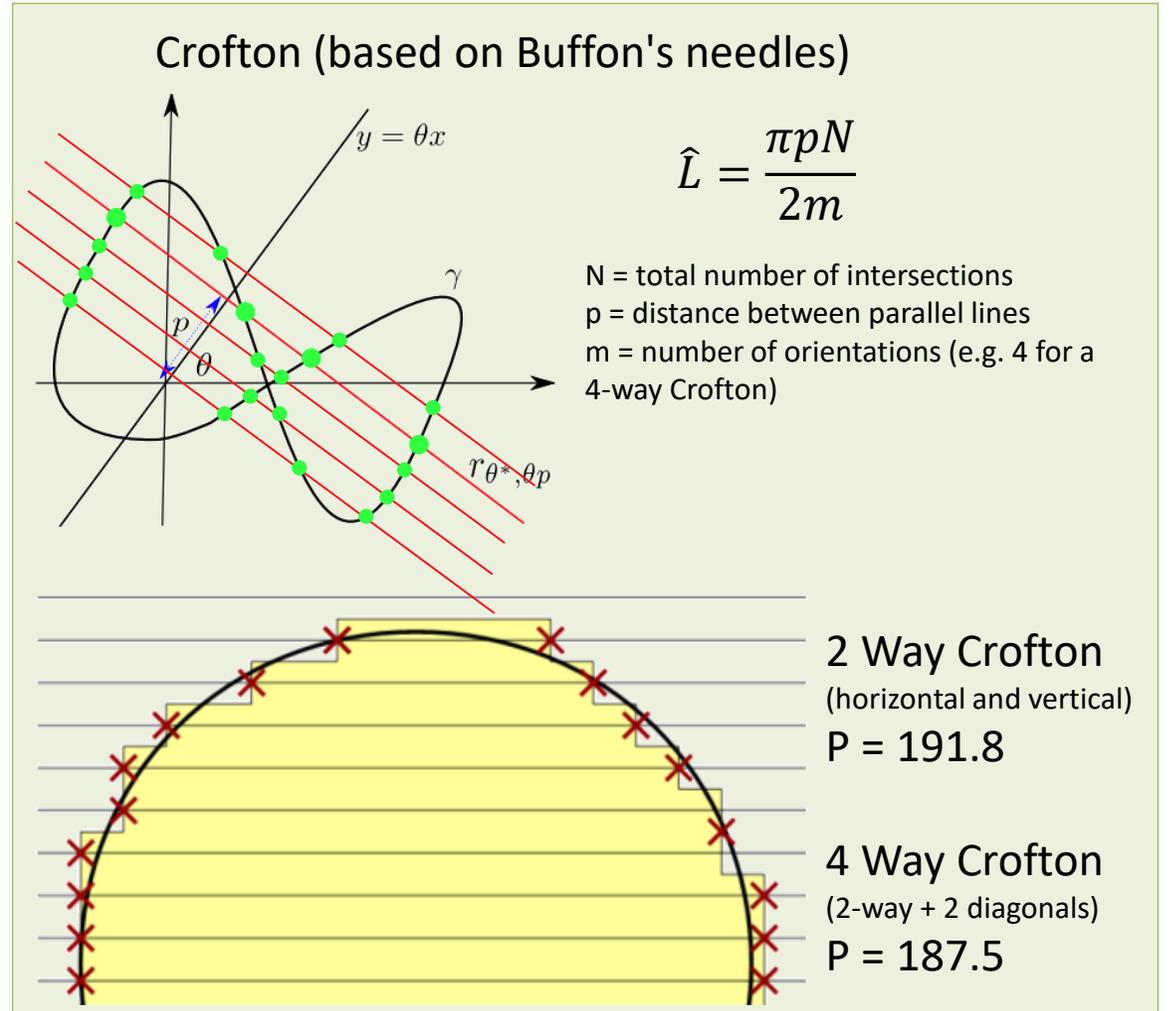


Published in "Nanoscale 9(15): 4918–4927, 2017"
which should be cited to refer to this work.

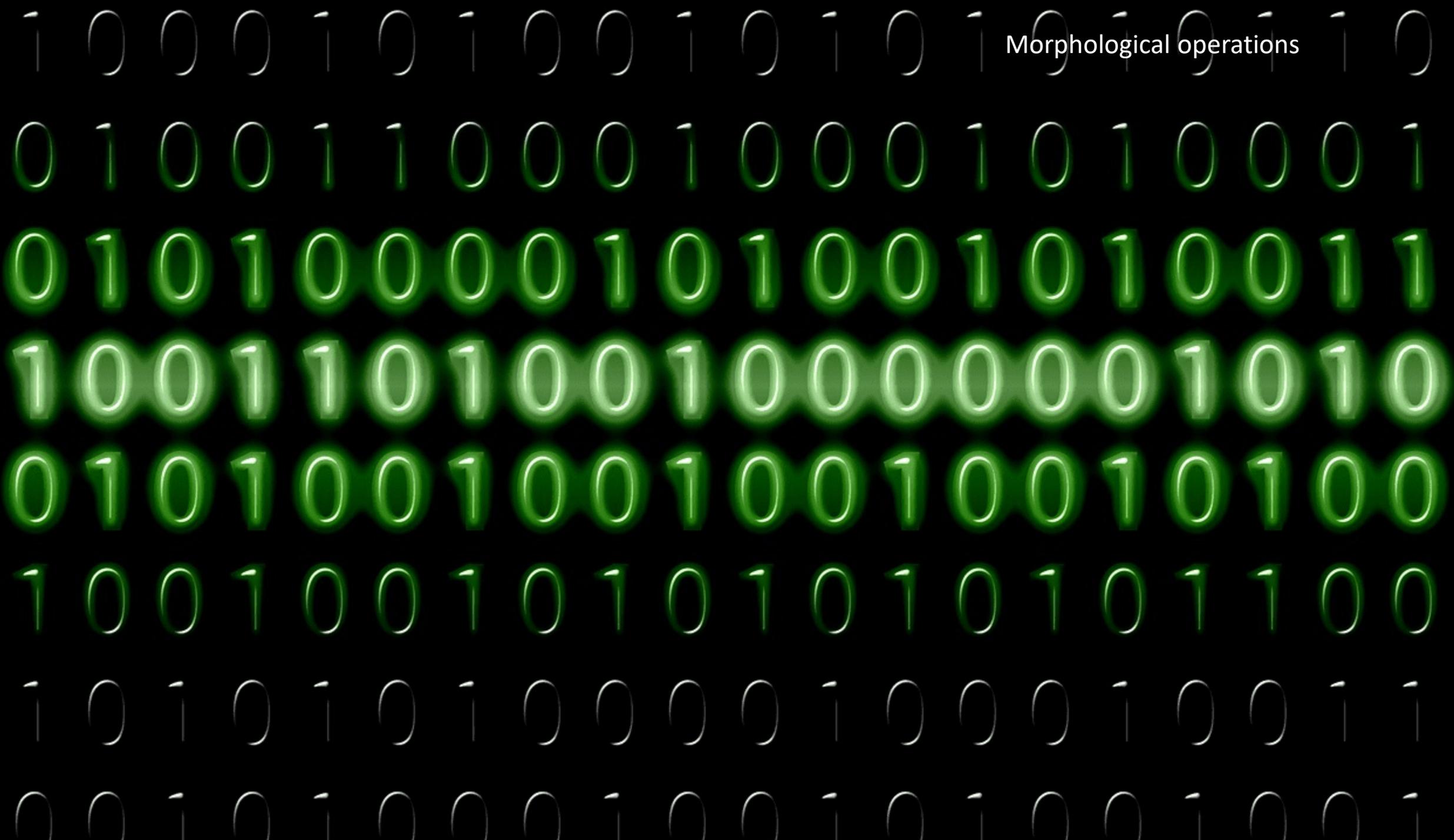
Assumption-free morphological quantification of single anisotropic nanoparticles and aggregates†

Dimitri Vanhecke, ^{*,a} Laura Rodriguez-Lorenzo,^a Calum Kinnear,^b Estelle Durantie,^a Barbara Rothen-Rutishauser^a and Alke Petri-Fink^{a,c}

Characterizing the morphometric parameters of noble metal nanoparticles for sensing and catalysis is a persistent challenge due to their small size and complex shape. Herein, we present an approach to determine the volume, surface area, and curvature of non-symmetric anisotropic nanoparticles using electron tomography and design-based stereology without the use of segmentation tools or modeling of the particles. Finally, we apply these tools to aggregates to estimate their fractal dimension.



Morphological operations

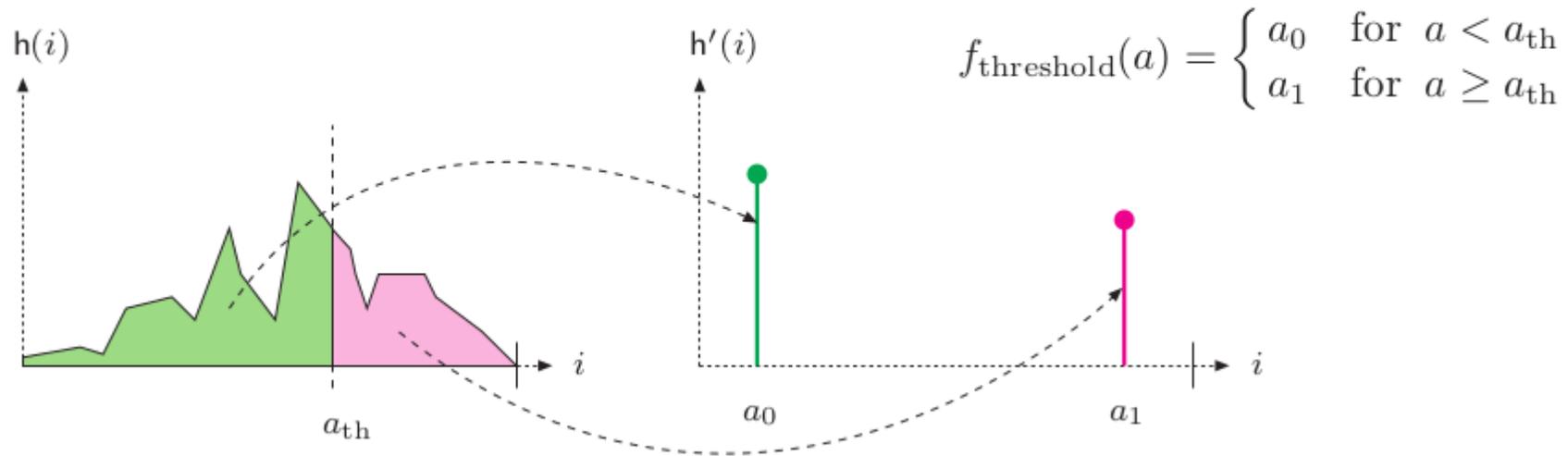


Thresholding / binarization / segmentation

8/12/16 bit



1 bit



Morphological binary operations

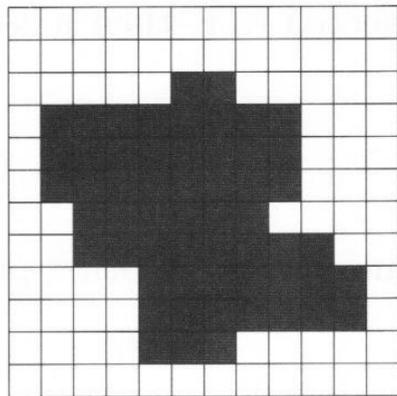
Prerequisite: Binary data

Binary data is the output of thresholding

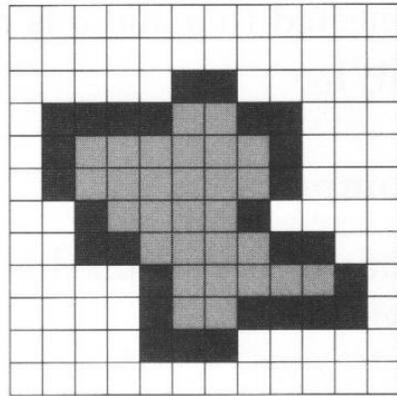
Binary images

are images with only two values: black (usually intensity = 0) and white (intensity = 1, or 255).

It is assumed that objects are black and background is white, but this can vary.



(a) Original image



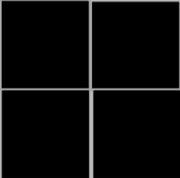
(b) ■ Boundary pixels
■ Interior pixels
□ Surrounds pixels

Morphological operations rely only on the relative ordering of pixel values, not on their numerical values (hence: binary data)

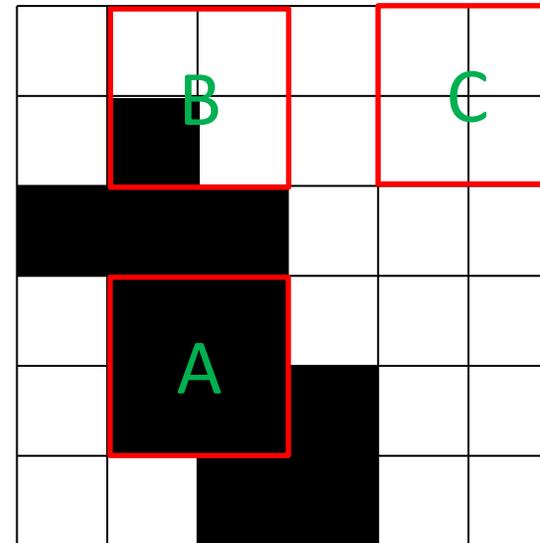
Morphological binary operations – structuring element

Structuring element

Morphological techniques probe an image with a small shape or template called a structuring element. The structuring element is positioned at all possible locations in the image and it is compared with the corresponding neighbourhood of pixels.

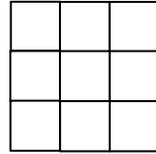
SE	Fits	A	SE fits within the neighbourhood
	Hits	B	SE hits a boundary
	None	C	Neither hits nor fits

Background = 0, black
Foreground = 1, white

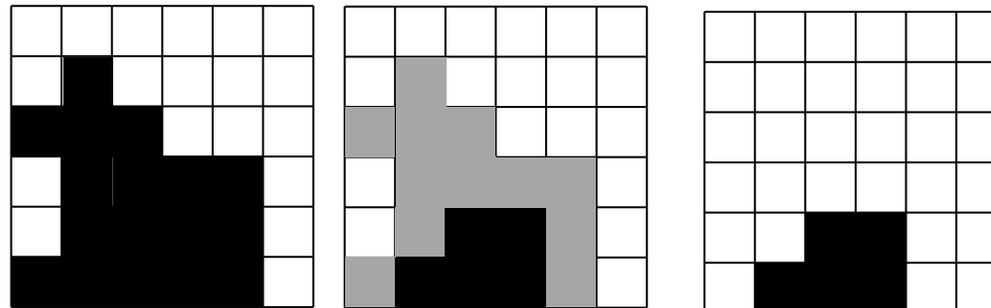
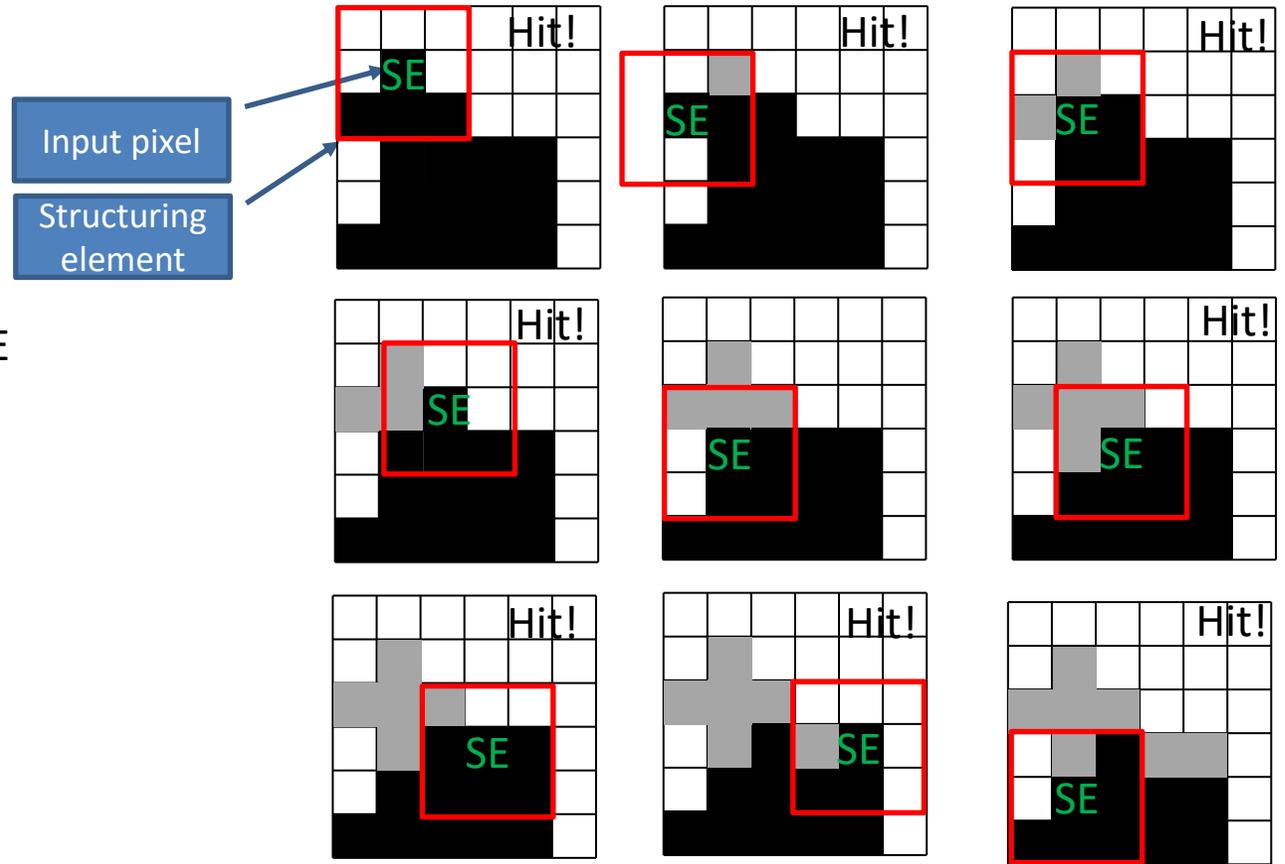


Basic (primary) binary operations: dilation

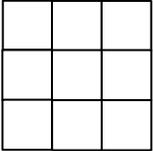
1. Structuring element:

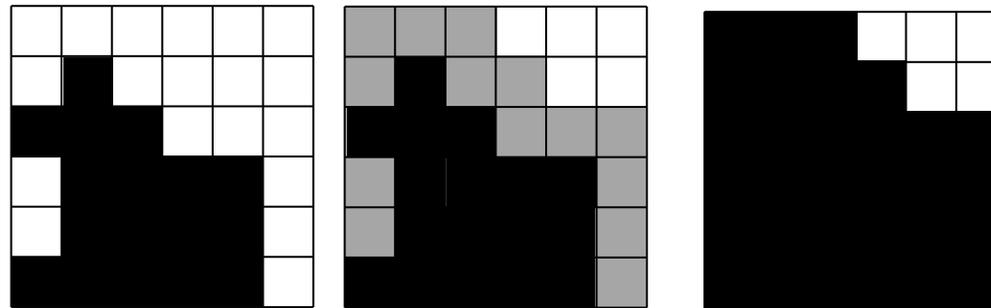
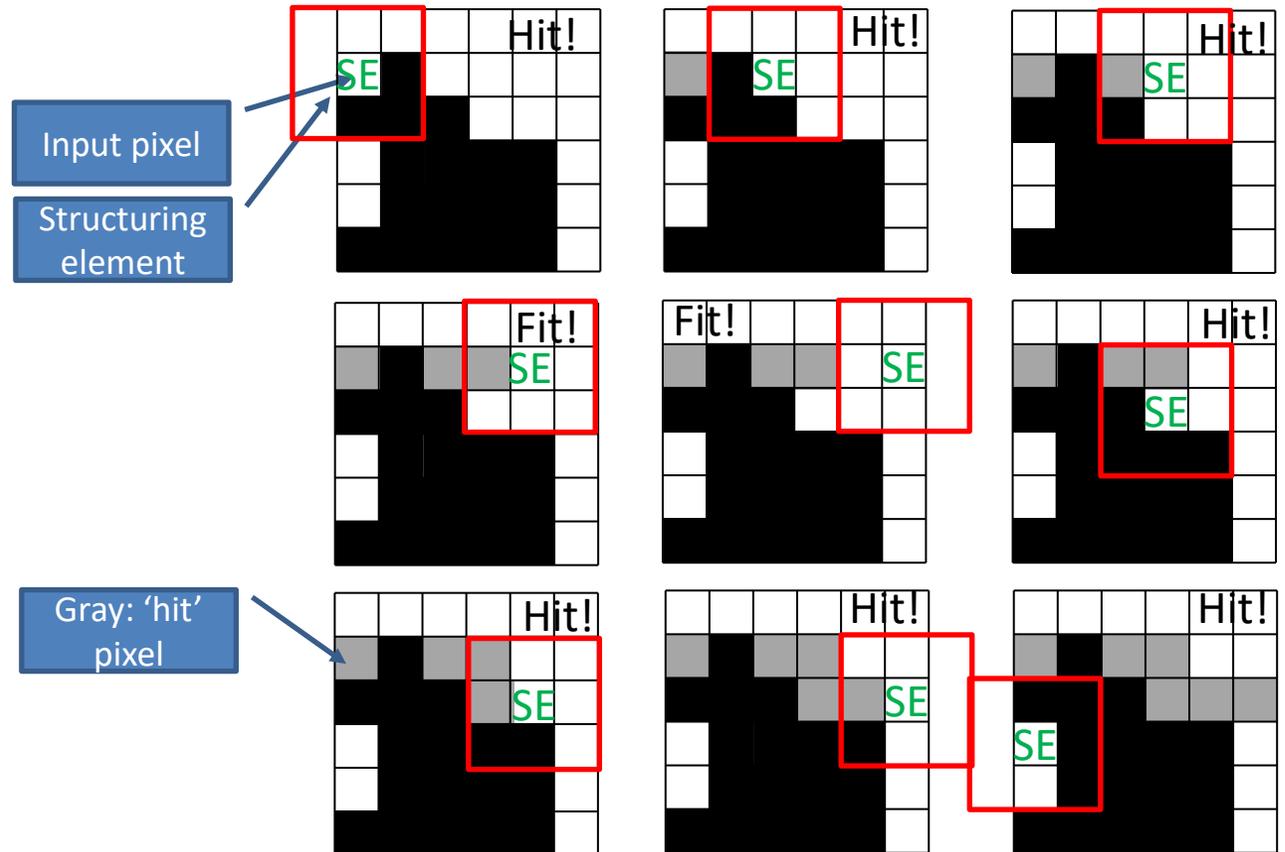


2. Consider each of the **background** (=black) pixels
3. For each background pixel (= *input pixel*) the SE is superimposed. (origin of the SE coincides with the input pixel).
4. **When hit:** input pixel (Black) is changed to a white foreground pixel (=If *at least one* pixel in the structuring element coincides with a foreground pixel in the image underneath)
5. **When fit or none:** do nothing



Basic (primary) binary operations: erosion

1. Structuring element: 
2. Consider each of the *foreground* (=white) pixels
3. For each foreground pixel (= *input pixel*) the SE is superimposed. (origin of the SE coincides with the input pixel).
4. **When hit:** input pixel changed to background (=If *at least one* pixel in the structuring element coincides with a background pixel in the image underneath)
5. **When fit or none:** do nothing



Basic (primary) binary operations: dilation and erosion

Original cameraman



Thresholded cameraman



Dilation



Gradually enlarges the boundaries of the foreground objects (*i.e.* white pixels, typically).

Erosion



Gradually enlarges the boundaries of background regions (*i.e.* black pixels, typically).

Secondary binary operations: close and open

Original cameraman



Thresholded cameraman



Close: First dilation, then erosion



=closing
foreground
gaps

Fills in and connect foreground

- Fill small black holes inside objects
- Bridging small gaps or breaks
- Smoothing object boundaries from the outside
- Removing «Pepper» noise within foreground regions
- Preparing masks for segmentation

Open: First erosion, then dilation



=opening
foreground
connections

Cleans up and separates foreground

- Separating weakly connections objects
- Thinning/trimming small protrusions
- Smoothing boundaries from the inside
- Pre-processing before segmentation

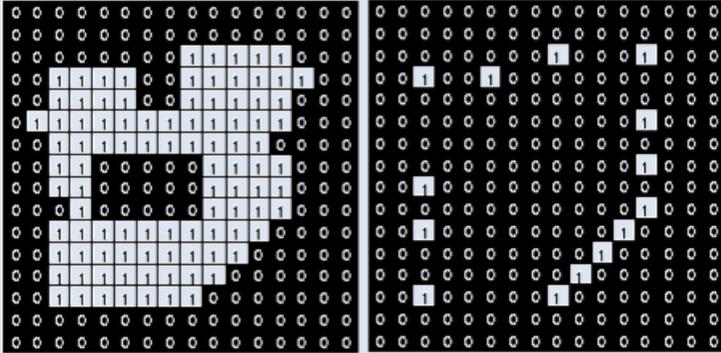
Idempotence

The property of applying more than once does not produces a further change.

Binary operations with SE

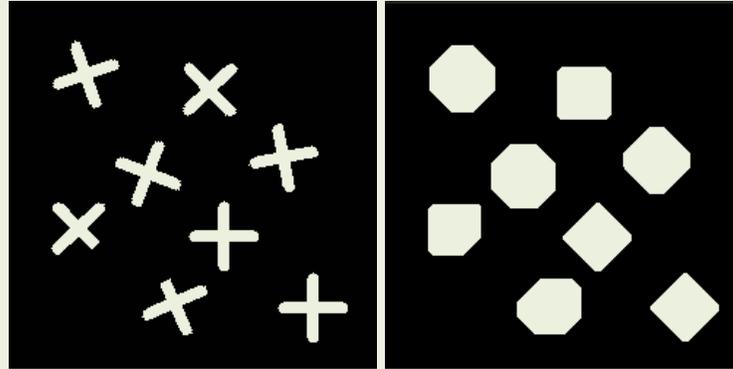
Hit or miss

Finding ends and corners



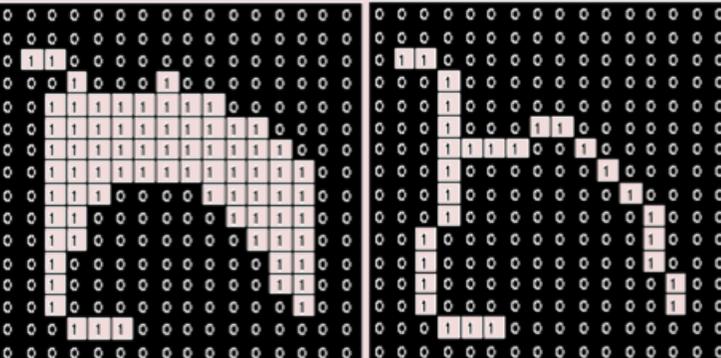
Thickening

Calculate convex hull of object



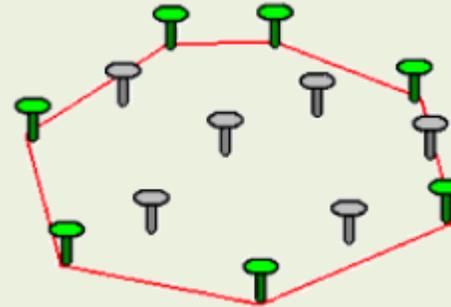
Thinning

Reduces the object to a single pixel line (skeletonization)



Top hat

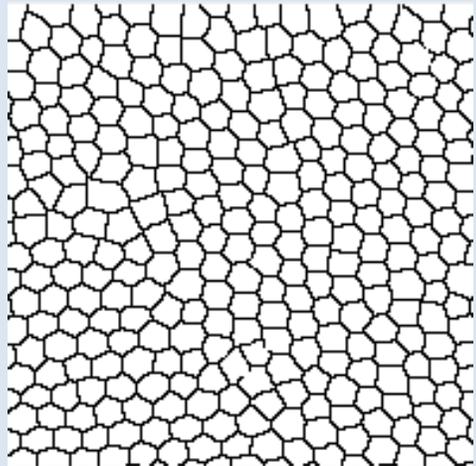
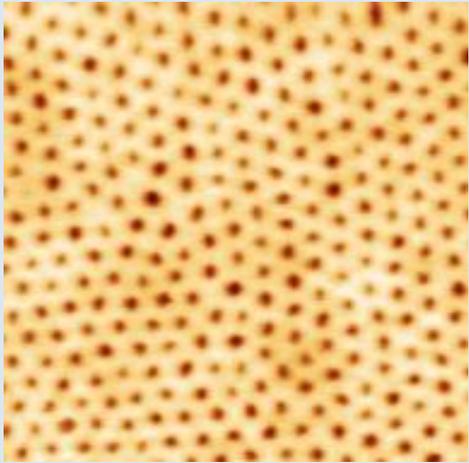
Extracting bright (white top hat) or dark (black top hat) objects



Associated: tessellation

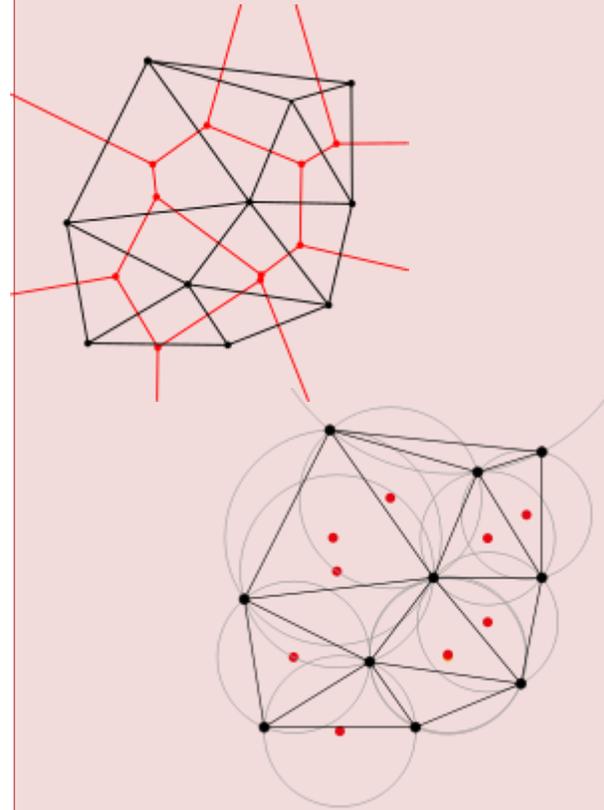
Voronoi tessellation

Boundary maps



Delauney tessellation

= dual graph of voronoi
(=swap of vertices and faces)



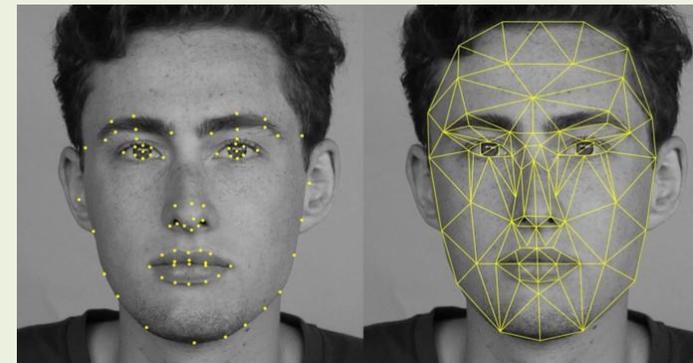
red: Voronoi, black: Delauney

Applications

Fingerprint analysis



Face recognition



- Robotics: path planning, autonomous driving
- Terrain modeling
- Medicine: surgical planning MRI mesh points
- ... (when point clouds need to be converted to mesh)

Binary operations: Euclidean Distance transform

A **distance transform**, is a derived representation, in this case from a **binary digital image**

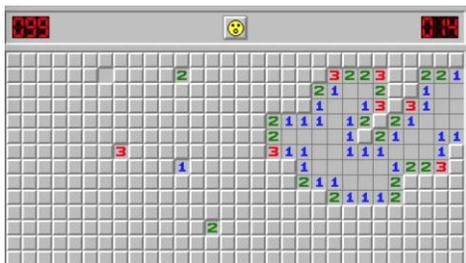
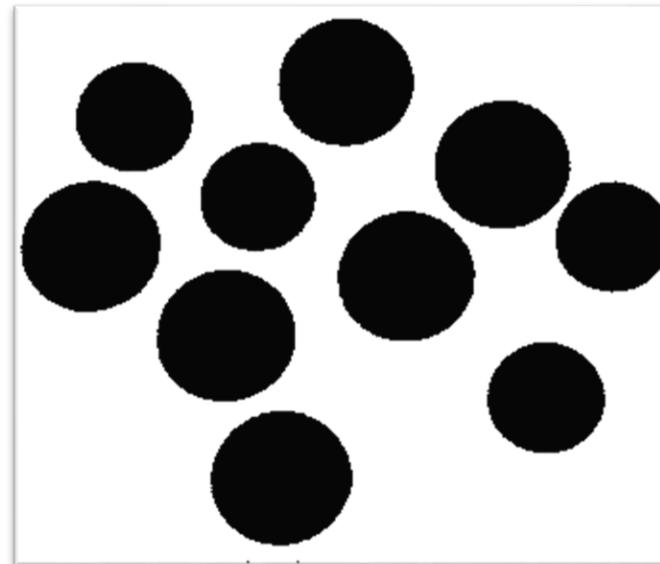
The result: the **Euclidian distance map**.
 Each foreground pixel in the binary image is replaced with a grey value equal to that pixel's distance from the nearest background pixel (for background pixels the EDM is 0)

0	0	0	0	0	0	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	0	0	0	0	0	0

Binary Image

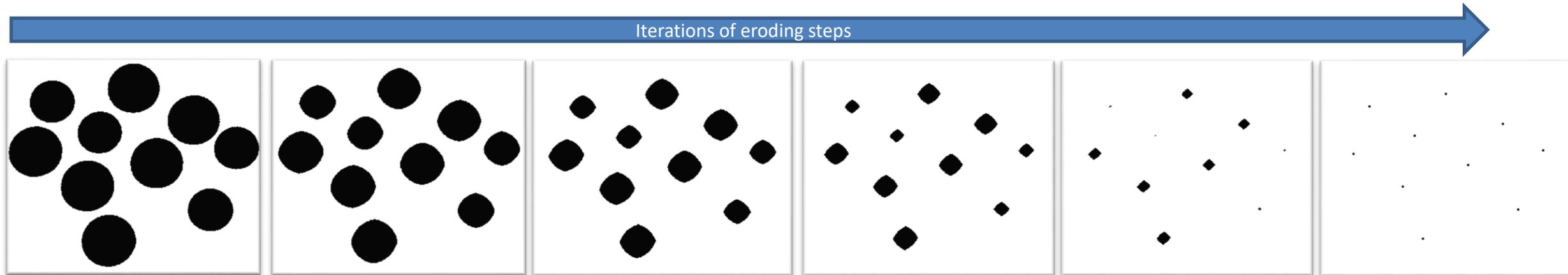
0	0	0	0	0	0	0
0	1	1	1	1	1	0
0	1	2	2	2	1	0
0	1	2	3	2	1	0
0	1	2	2	2	1	0
0	1	1	1	1	1	0
0	0	0	0	0	0	0

Distance transformation



Binary operations: Ultimate eroded points

The **Ultimate Points** extracts the last point that would be removed if the object were eroded to completion.



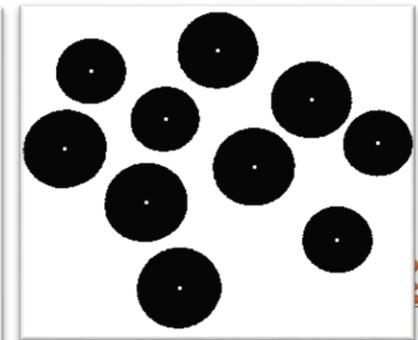
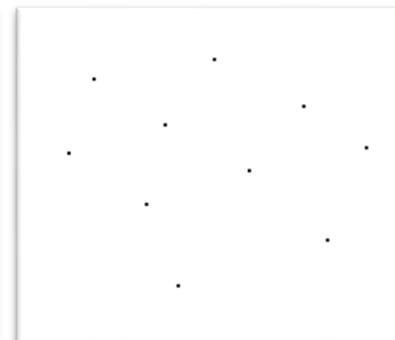
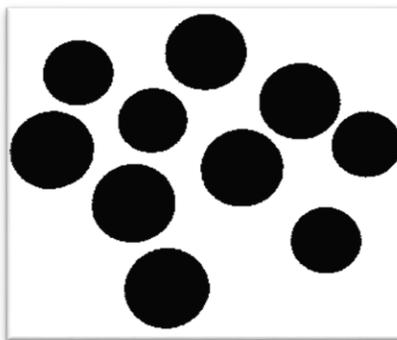
Origin

binary

Euclidian distance
map

Ultimate eroded
points

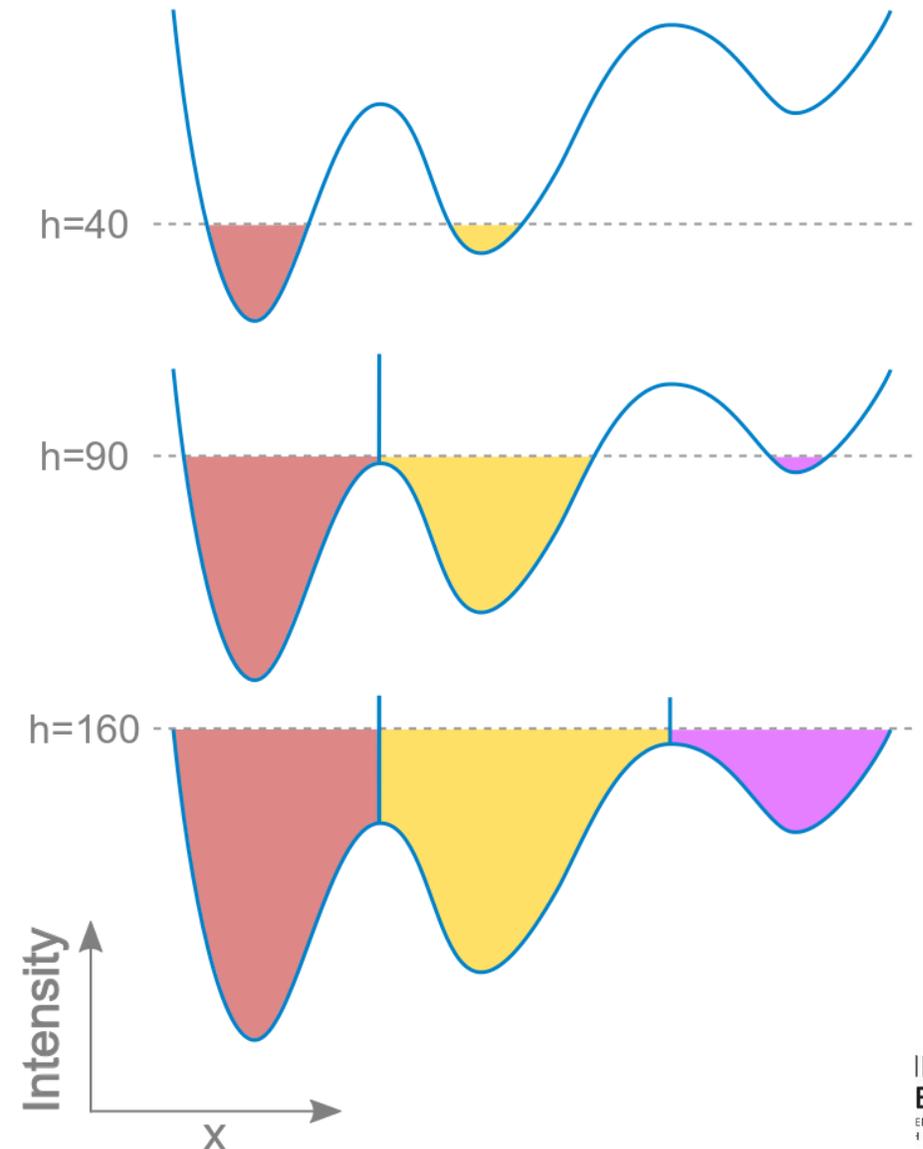
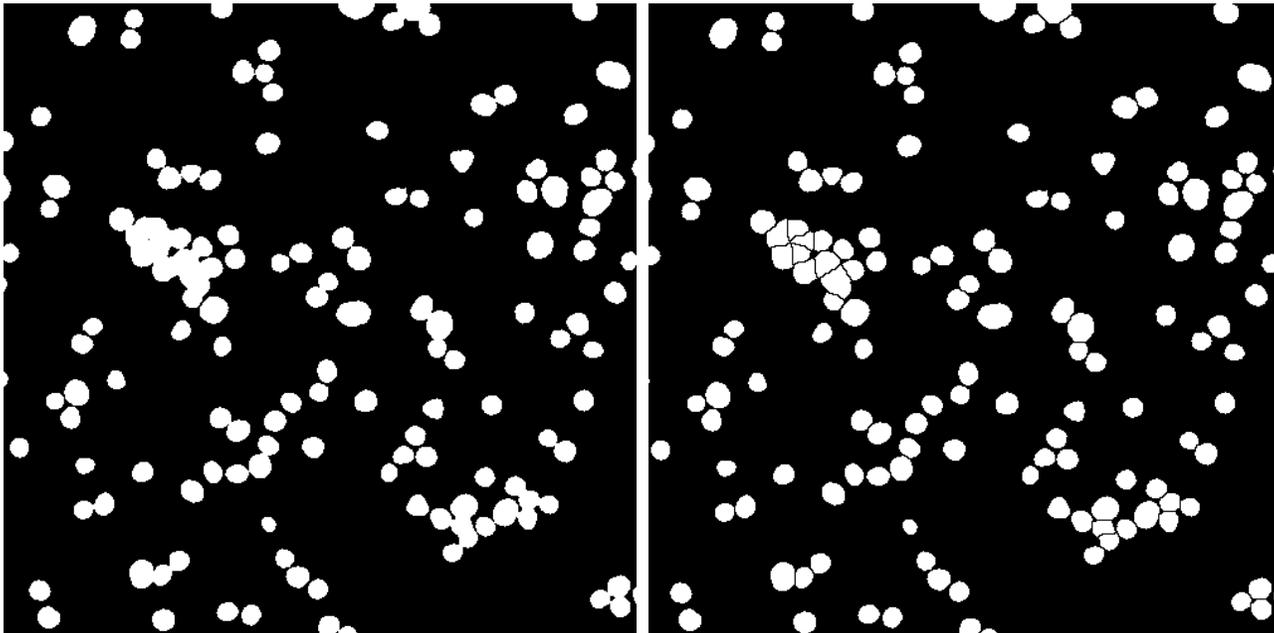
Overlay UEP with
binary



Binary operations: Watershed

Watershed segmentation
Separates touching objects.

1. the Euclidian distance map (EDM) is calculated
2. the ultimate eroded points (UEPs) are calculated .
3. Dilation of each of the UEPs until:
 1. the edge of the original particle is reached
 2. the edge touches a region of another (growing) UEP

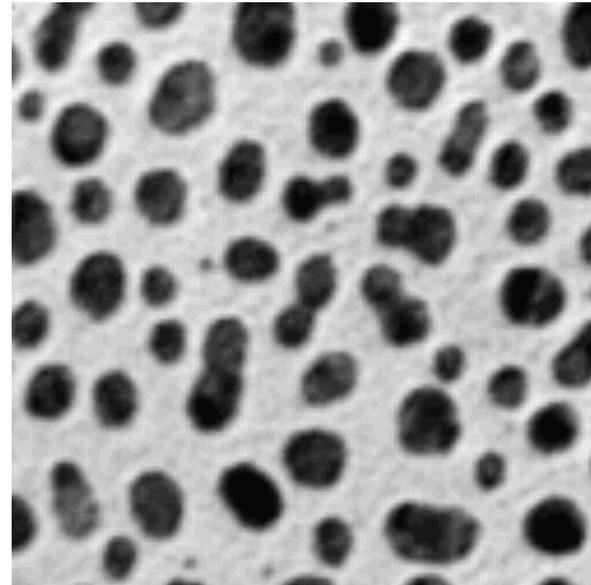
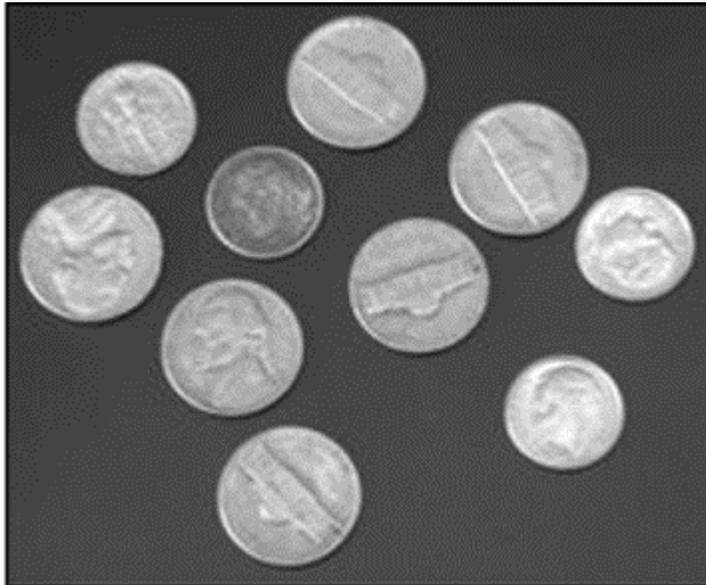


How does a software measure images?

EXERCISE 1

Open example 1A and count the number of coins using eroded points. Repeat for example 1B

Process > Binary > Ultimate points



How does a software measure images?

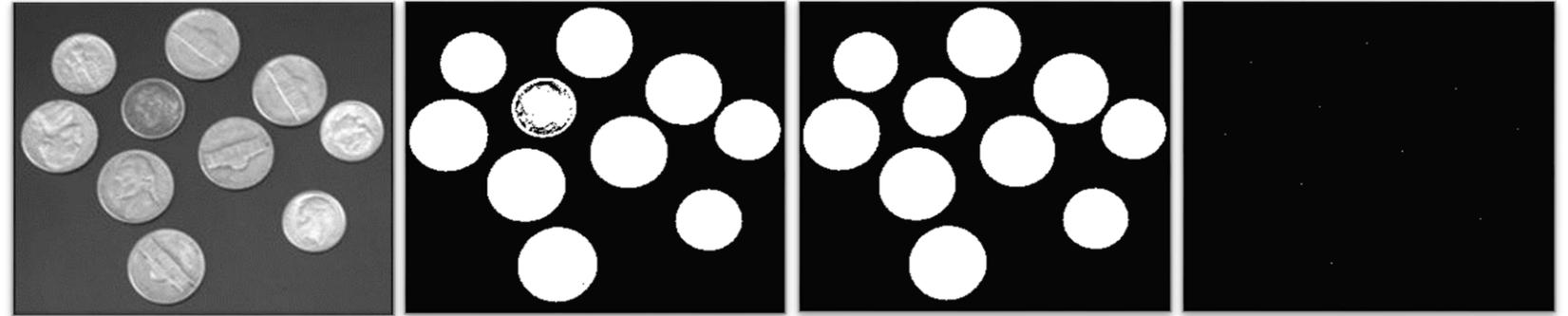
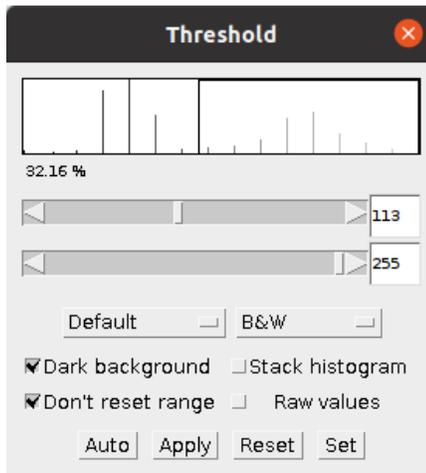
EXERCISE

Open example 1A and count the number of coins using maximum eroded points. Repeat for example 1B

File > Open...

Image > Adjust > Threshold...

113-255



Process > Binary > Fill holes

(Alternative: open / close)

Process > Binary > Ultimate points

(Calculates the EDT and then the UEP)

Process > Histogram

(In Histogram window) > List



How does a software measure images?

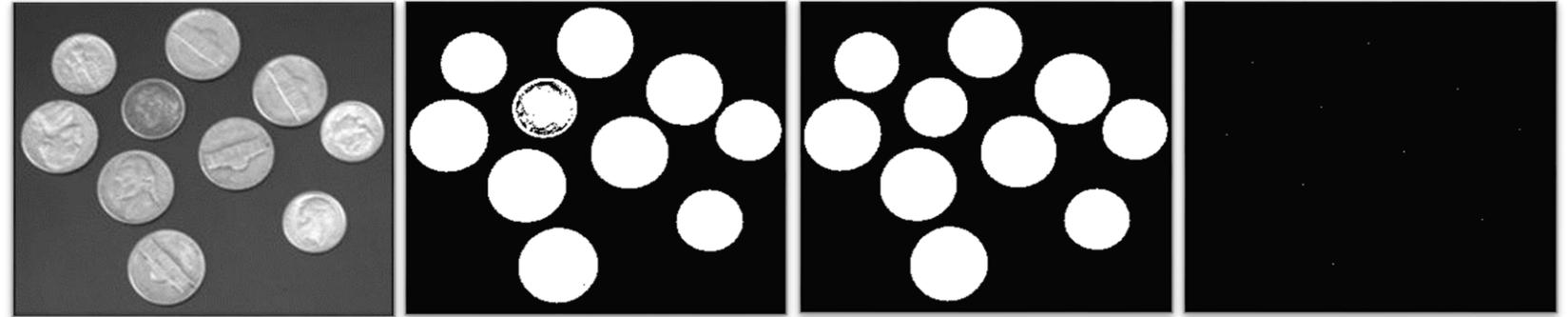
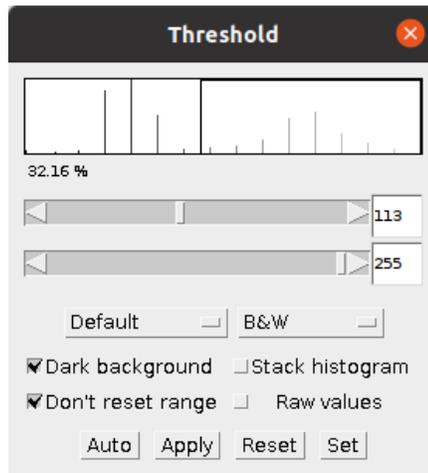
EXERCISE

Open example 1A and count the number of coins using maximum eroded points. Repeat for example 1B

File > Open...

Image > Adjust > Threshold...

113-255



26	0
27	0
28	3
29	1
30	0
31	0
32	0
33	3
34	2
35	1
36	0
37	0



Process > Binary > Fill holes

(Alternative: open / close)

Process > Binary > Ultimate points

(Calculates the EDT and then the UEP)

Process > Histogram

(In Histogram window) > List (count the non-zero pixels)

How does a software measure images?

EXERCISE

Open example 1B and count the number of blobs using maximum eroded points.

File > Open...

Image > Adjust > Threshold

Edit > Invert (make sure your objects are White)*

Process > Binary > Ultimate Points

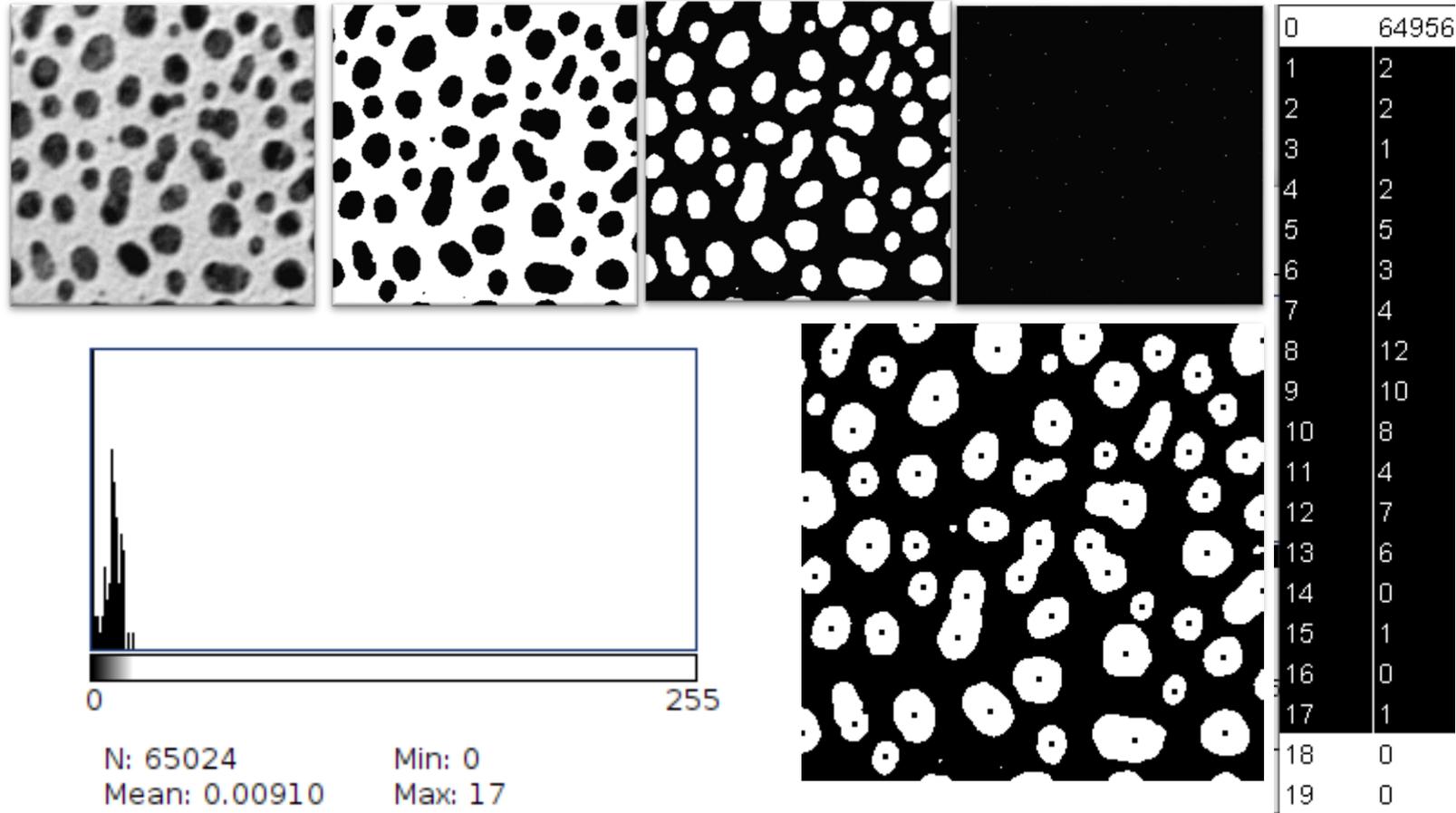
Process Histogram

To count:

Process > Math > Multiply by e.g 1000

(In histogram) > List > check at value 255

255 56



N: 65024 Min: 0
 Mean: 0.00910 Max: 17
 StdDev: 0.301 Mode: 0 (64956)
 Value: 150 Count: 0

* Note: you can also invert the look-up table (Image > Color > Invert LUT), this does not change your objects pixel values. I.e. black => 255, and white => 0, which can be very confusing)

Binary operations: Watershed

EXERCISE

Convert Example 2 – AuNP to a binary image. Compare with and without watershed

File > Open...

Image > Adjust > Threshold...

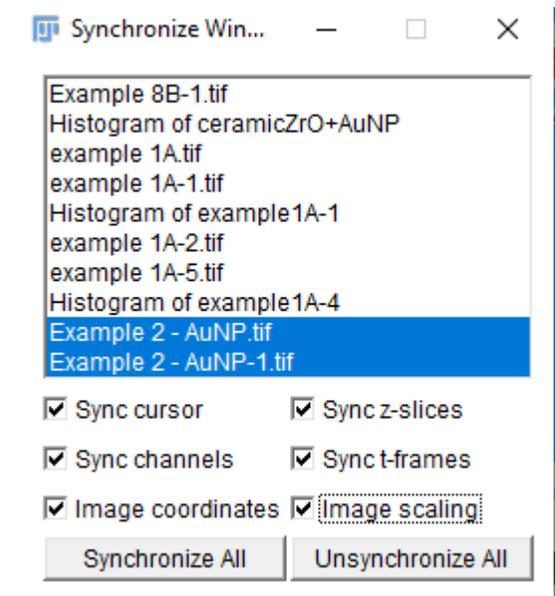
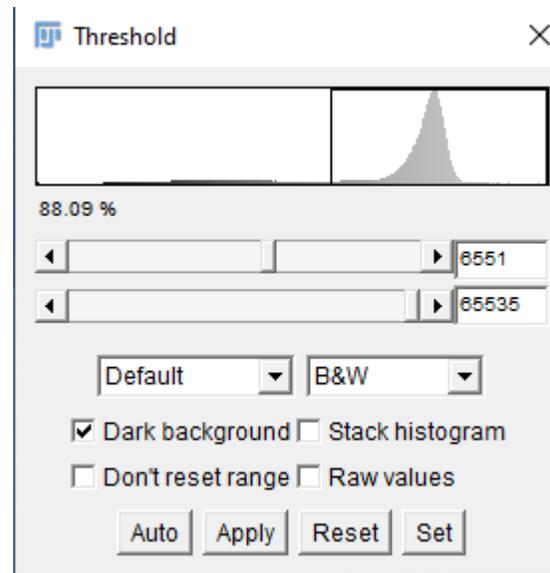
Click 'Auto'

Click 'Apply'

Duplicate the image (ctrl+shift+D)

Process > Binary > Watershed

To compare the two windows: Analyze > tools > Synchronize windows

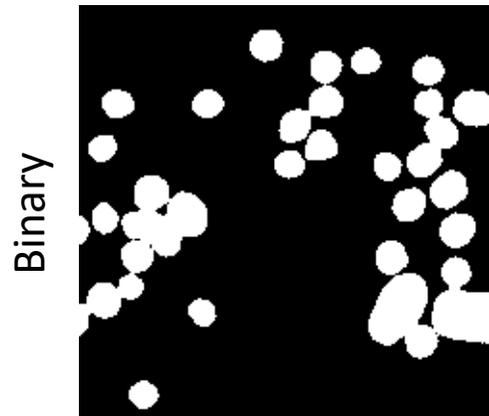
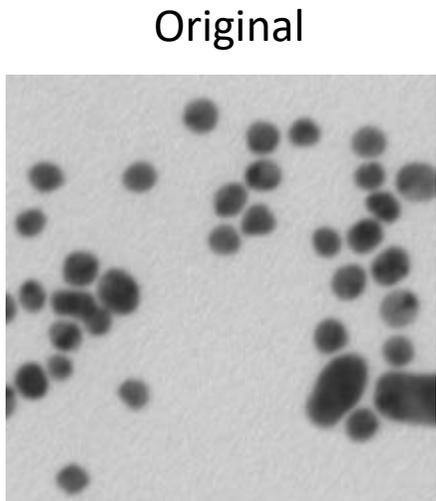


Binary operations: Watershed

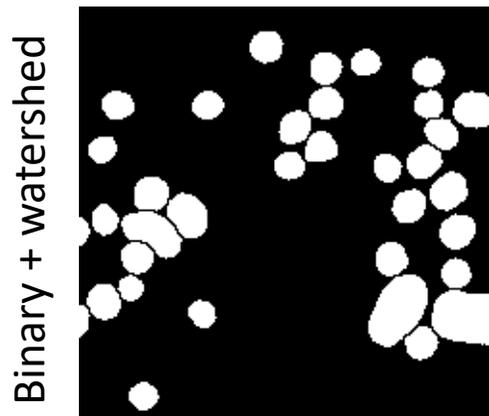
EXERCISE

Convert Example 2 – AuNP to a binary image. Compare with and without watershed

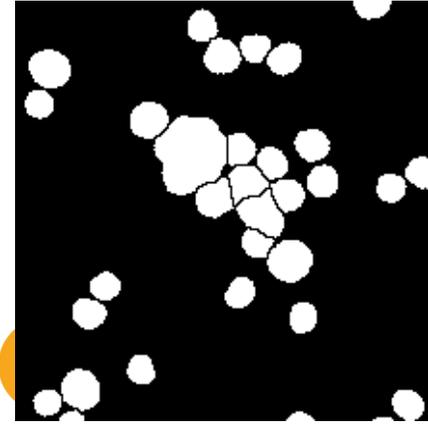
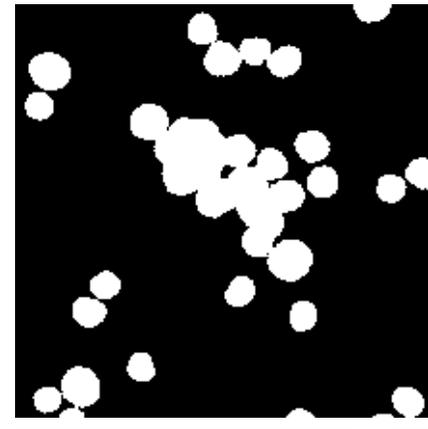
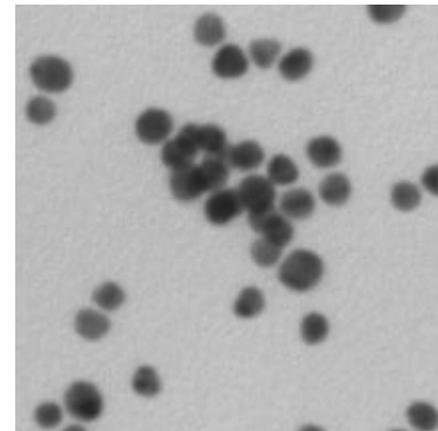
Process > Binary > Watershed



17 objects
(not touching the edge)



31 objects
(not touching the edge)



How does a software measure images?

Given:

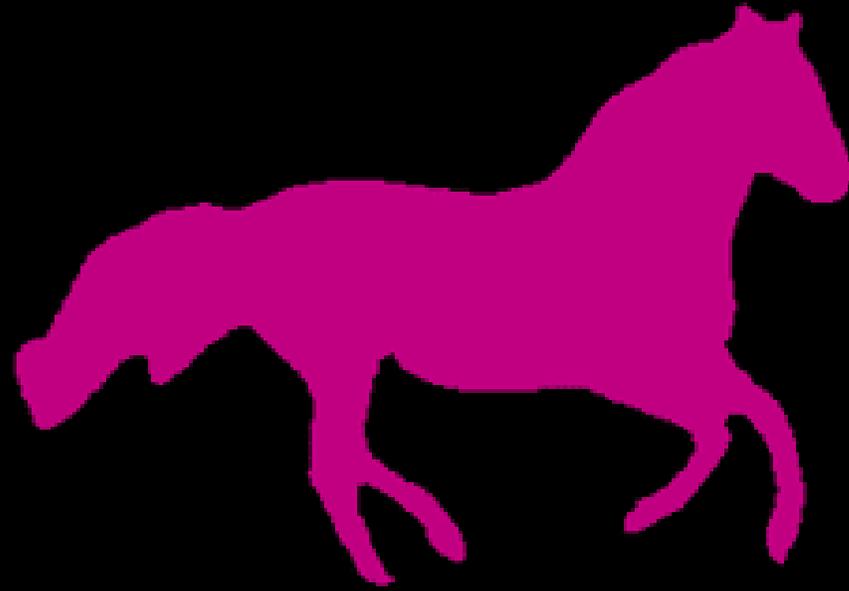
- The primary units (area, perimeter, number)
- The position of all foreground pixels (array of X and Y)

Secondary units:

Centroid	Average of all x and y within each object
Bounding Rectangle	The smallest rectangle enclosing the object
Fit Ellipse	Fit an ellipse to the object
Circularity	$\frac{4 \cdot \pi \cdot \text{area}}{\text{perimeter}^2}$, for each object
Aspect ratio	$\frac{\text{Minor axis}}{\text{Major axis}}$, for each object
Roundness	$\frac{4 \cdot \text{area}}{\pi \cdot \text{major axis}^2}$, for each object
Solidity	area/convex area.
Feret's Diameter	Longest distance between any two pixels in an object.
...	

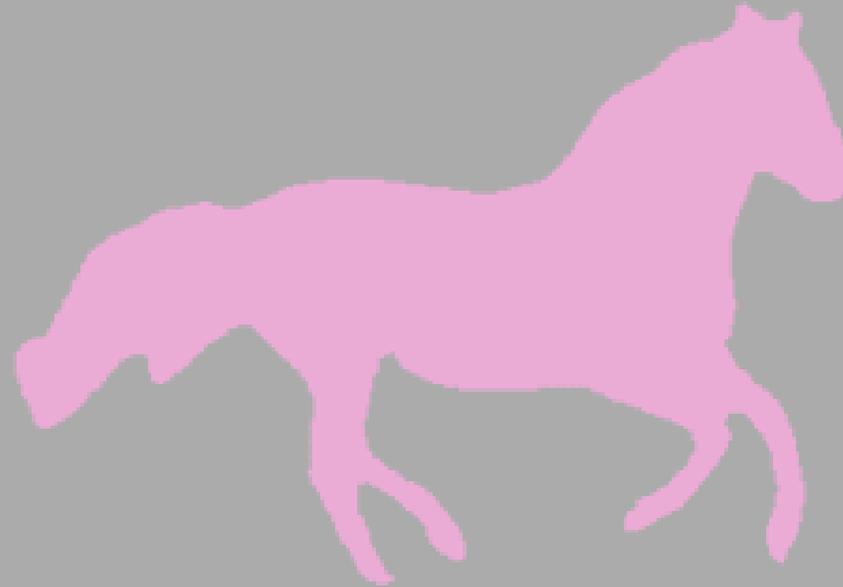
Everything relies on the thresholding step...

Thresholding, classification and segmentation



Level 1

Thresholding, classification and segmentation

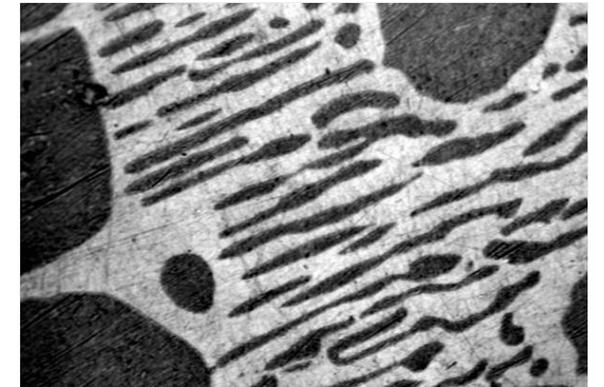
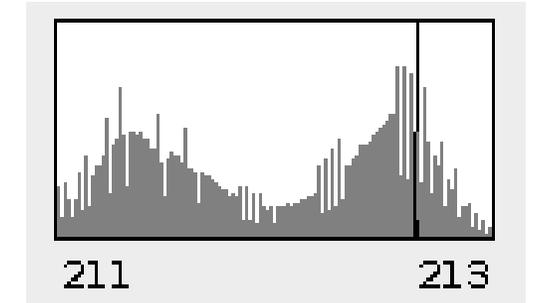
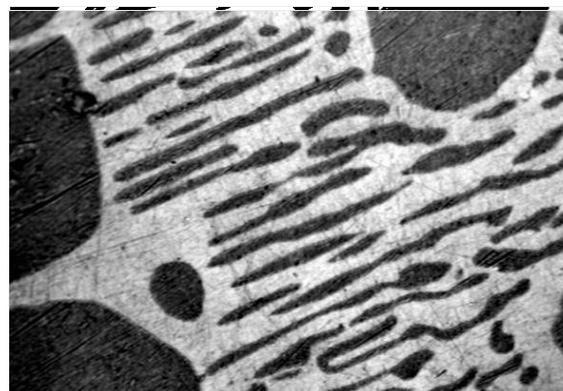
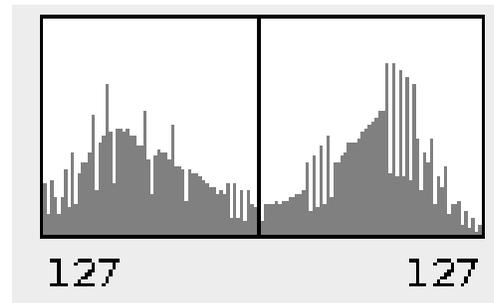
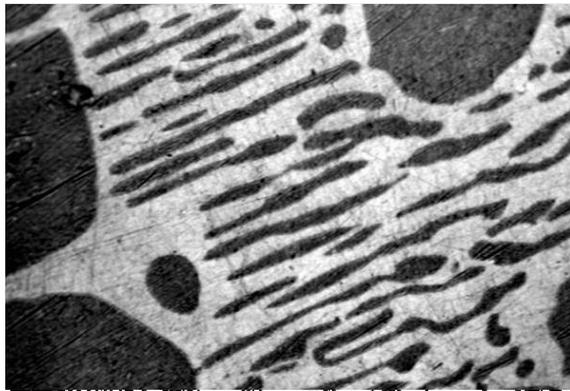
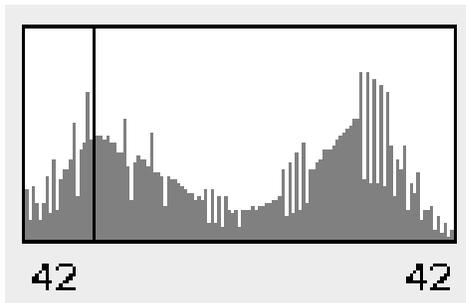


Histogram-based

Thresholding

How?

By setting the transfer function to a **vertical asymptote** (=infinite contrast), preferably automatic (=non-subjective)



Two concepts for unsupervised pixel thresholding (a.k.a. automatic thresholding):

Histogram shape based

Image entropy based

(there are more, but these two classes are the most common)

Thresholding

Some thoughts:

- Use **16-bit data (or 32 bit)**. Not 8 bit
- **Global thresholding** is preferred over local thresholding (=last resort)
- Try to go for easy, straightforward and **known thresholding algorithms** (ISOdata, Otsu, ...), which are described in the scientific literature (references)
- **Auto-thresholding** is preferred over manual thresholding (reproducibility)
- There is no «correct» solution, just models that try to simplify the complexity of nature.

Continuous function > Discrete function > Binary function

Image processing to the rescue (pre-processing):

- Gradient background: Mean filter with large kernel
- Fireflies/hot pixels/dead pixels: Bin your data, Median filter with a kernel as small as possible,
- Shot noise / poisson noise: GIGO, non-local filters (maybe)
- Undersampling aliasing: low pass filters
- ...

post thresholding: Morphological filters (open/close)
Touching objects: Watershed

Auto – thresholding

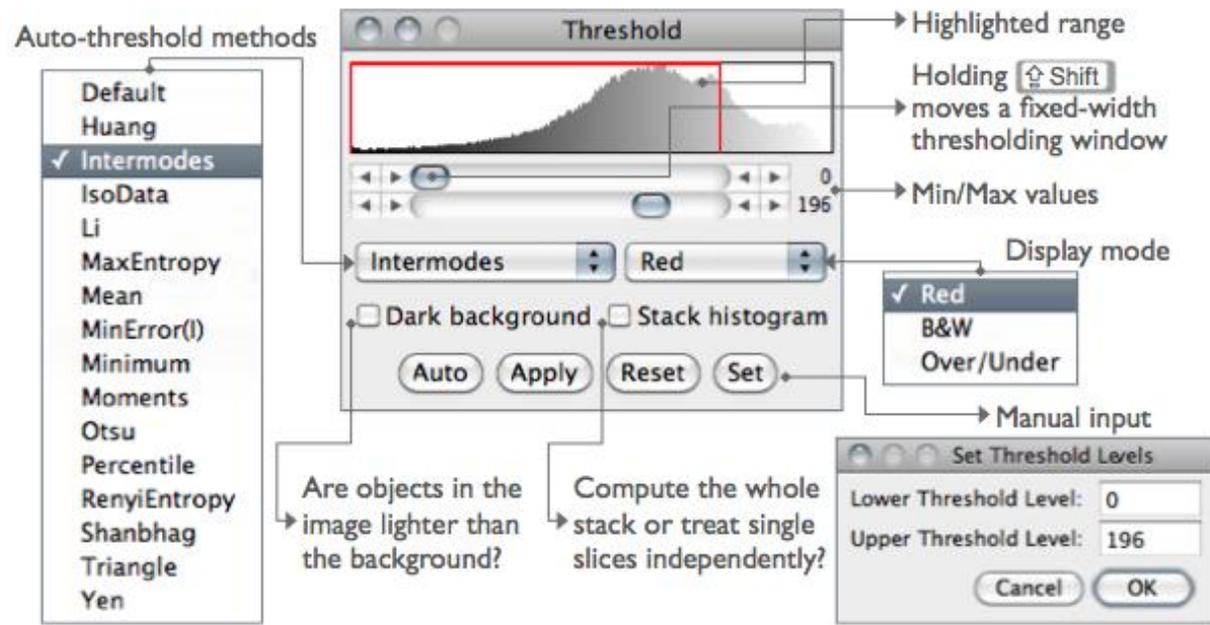
Clustering

- ISOdata
- Otsu
- Intermodes (assumes equal bimodal histogram)
- Minimum
- Mean (Mean of grayscale as threshold, initiates ISOdata)
- Percentile (assumes foreground pixels fraction of 0.5)
- Yen

Entropy

- Huang and Huang 2 (faster)
- Shannon's entropy
- Li
- MaxEntropy
- RenyiEntropy
- Shanbhag

Image > Adjust > Threshold...



Metric

Triangle

Moments

Tsai

Unsupervised thresholding: clustering

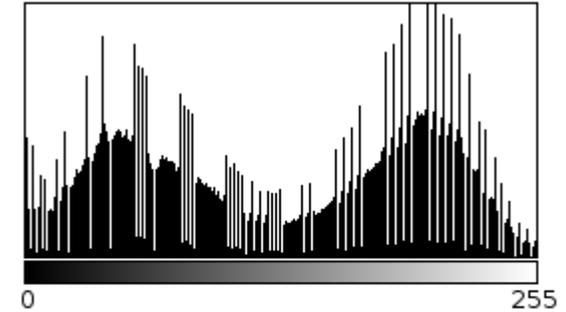
All pixels values are randomly assigned into two clusters (foreground and background)

The **standard deviation of the values** within each cluster, and the distance between cluster centers (between-class) is calculated

Pixel values are moved between clusters to minimize the deviations

Ideal for bimodal histograms!

You do not have a bimodal histogram?
Use entropic thresholding

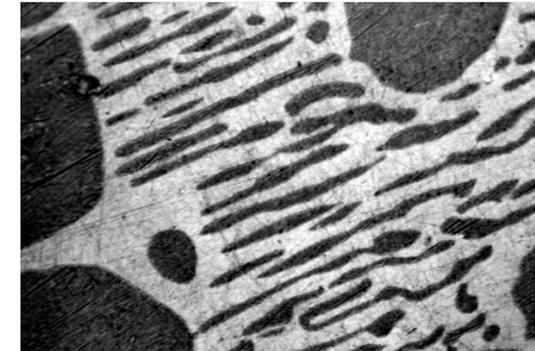


Count: 247200 Min: 0
Mean: 125.901 Max: 255
StdDev: 73.247 Mode: 201 (3164)

Iterate!

Until...

- the average intercenter distance between the clusters falls below a threshold,
- the average change in the intercenter distance between iterations is less than a preset threshold, or
- the maximum number of iterations is reached



Unsupervised thresholding: entropy

All pixels values are randomly assigned into 2 clusters (foreground and background)

The **entropy** within each cluster is calculated:

$$H = - \sum_{i=i_0}^{i_M} p_i \log_2(p_i)$$

Clusters are re-arranged to minimize entropy

Iterate!

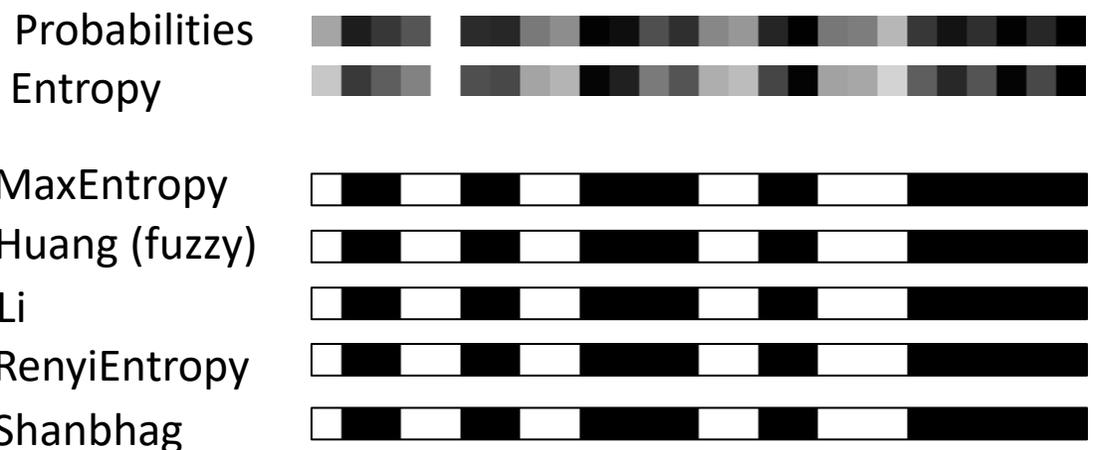
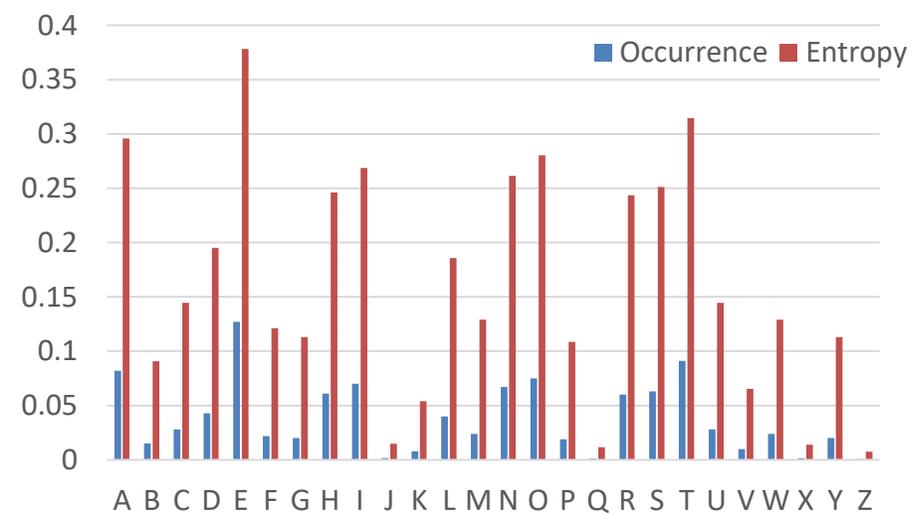


Until...

- Entropy difference is maximized (MaxEntropy)
- Entropy difference is minimized (MinEntropy)
- Fuzziness is minimized (Huang)

Information entropy:
Quantification for Surprise
 e.g.: flip a coin. The "surprise" factor is 1/2

Relative occurrence of letters in the english language



Thresholding algorithms

EXERCISE

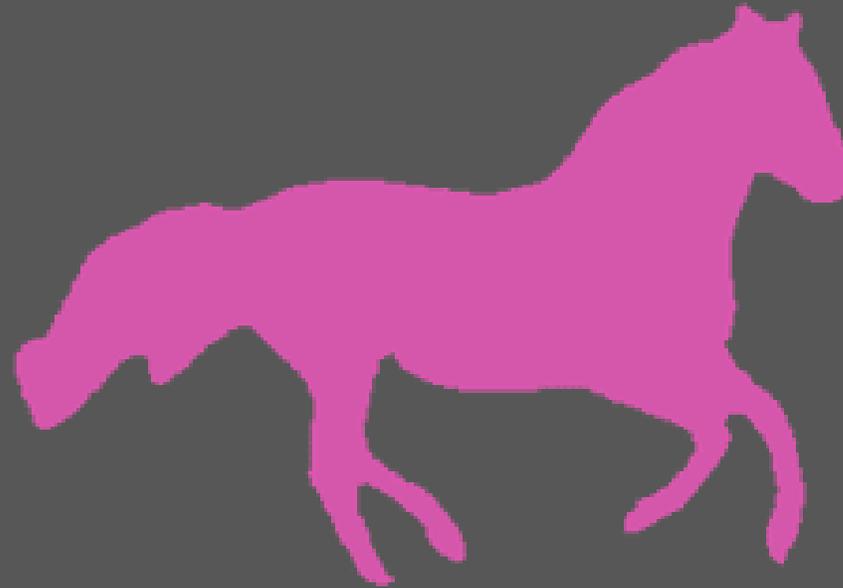
Open Example 3 (A/B/C). Run a threshold and check the differences between the algorithms. Try it also on your own data.

Image > adjust > Threshold...

Note the difference between different pixel classification algorithms

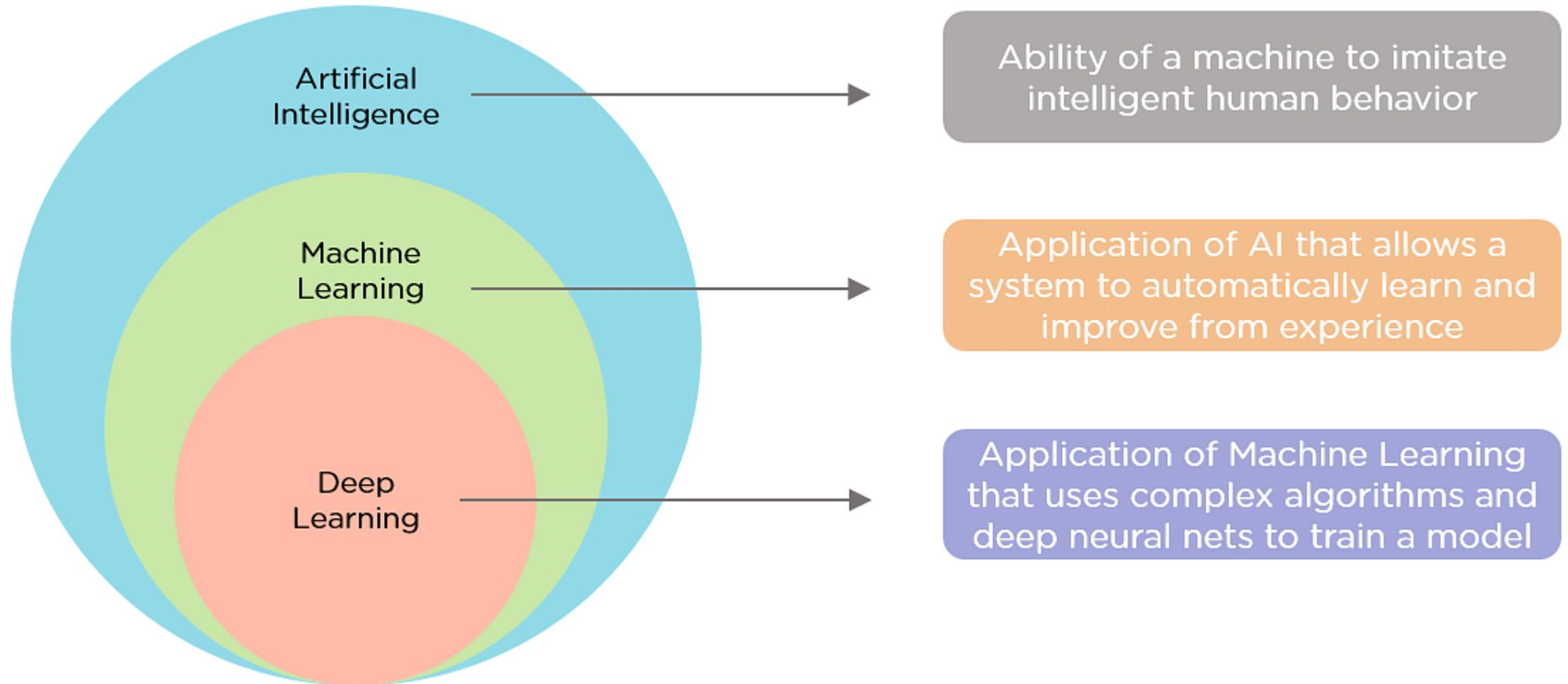
Level 2

Thresholding, classification and segmentation

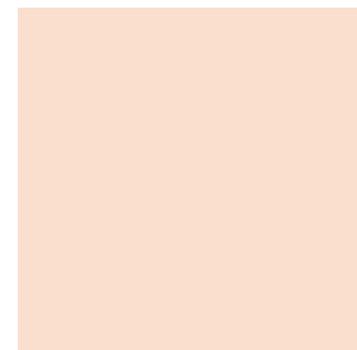
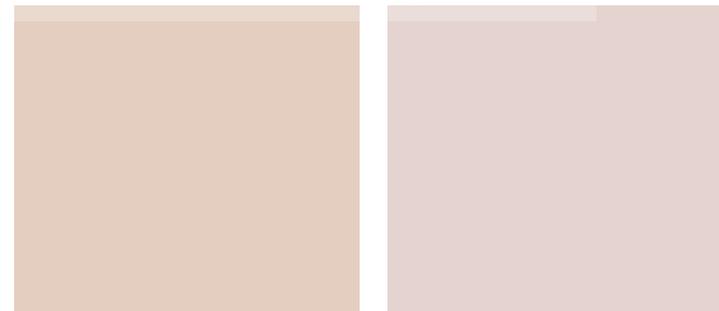
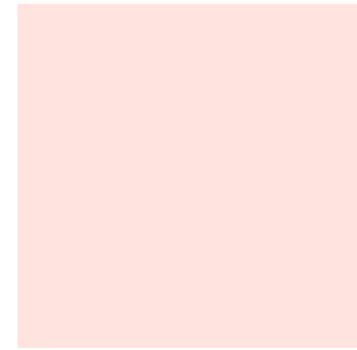
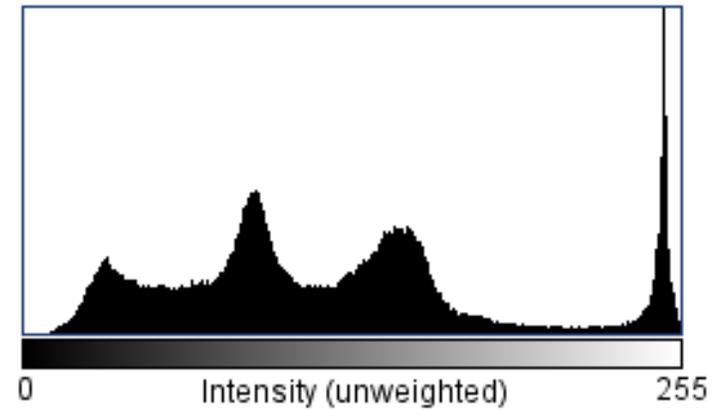
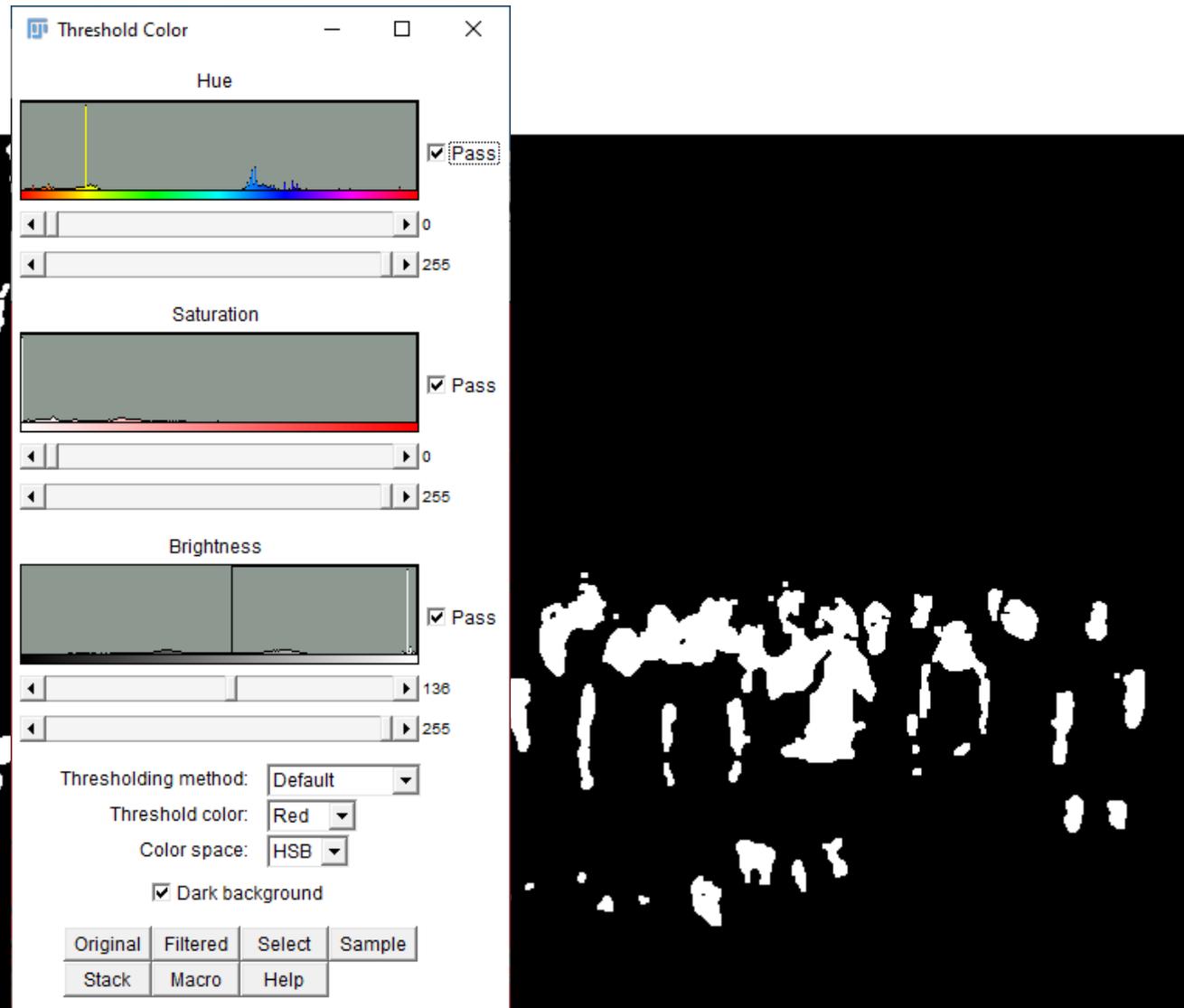


Machine learning

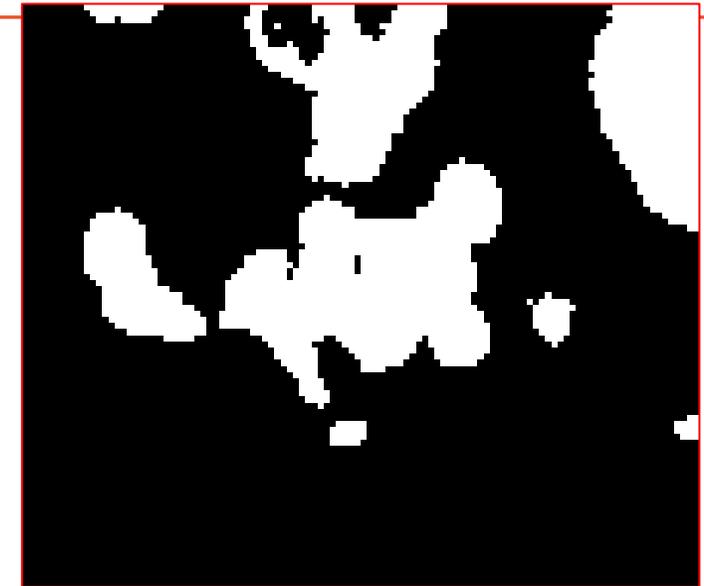
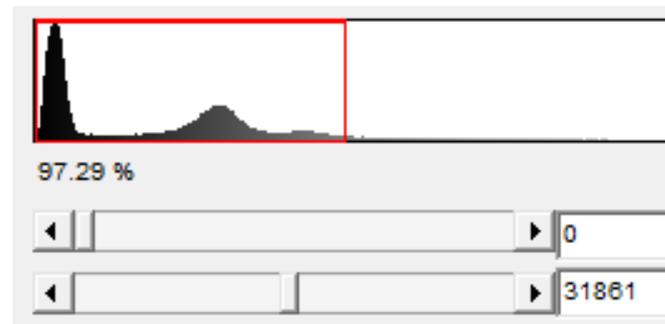
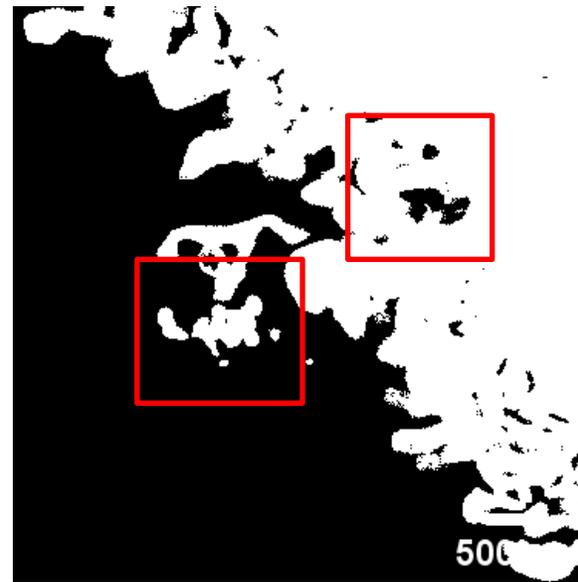
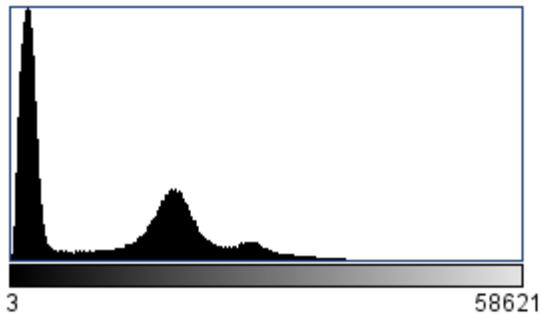
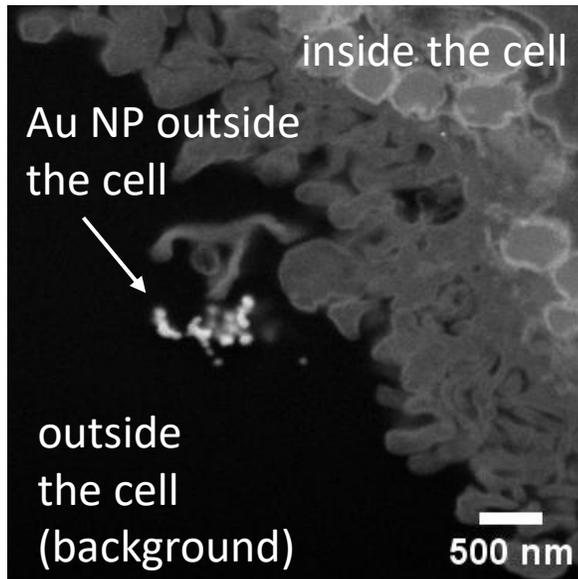
Thresholding: human vs machine



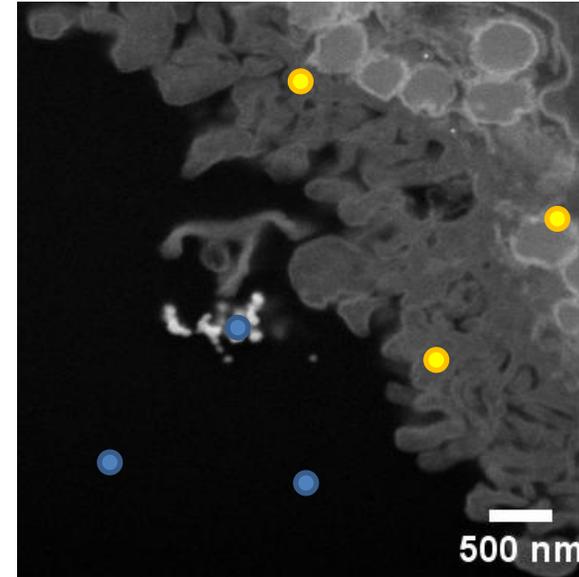
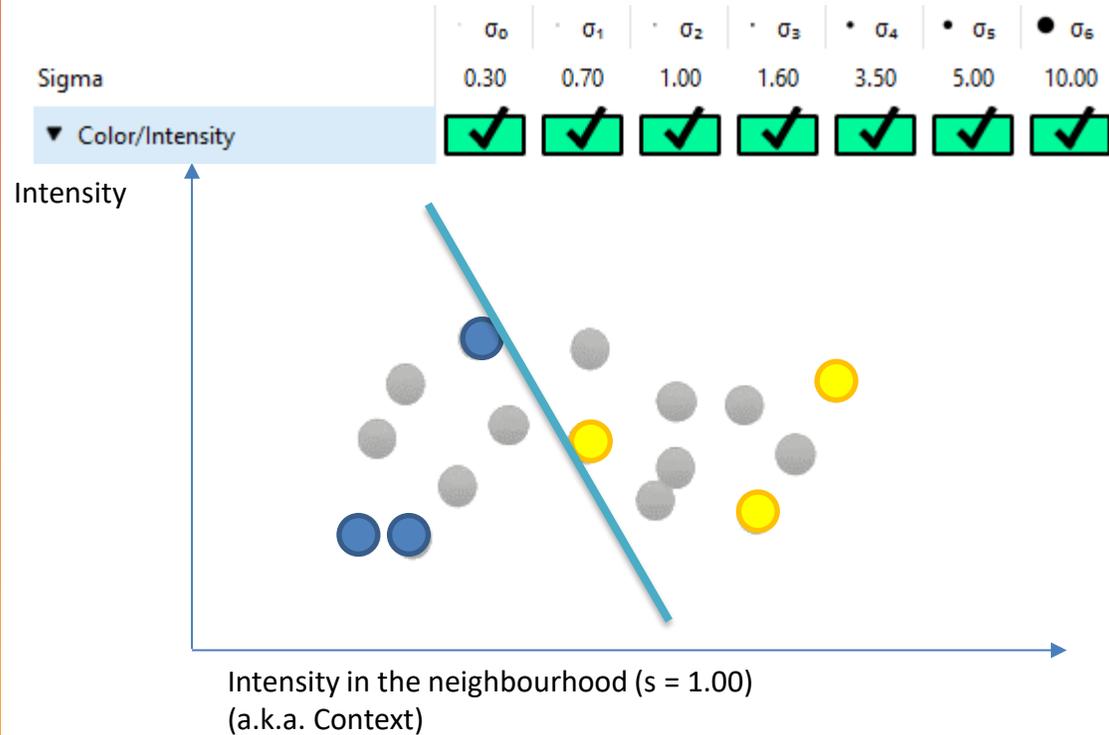
Thresholding: human vs machine



From thresholding to classification



From thresholding to classification



Classification features

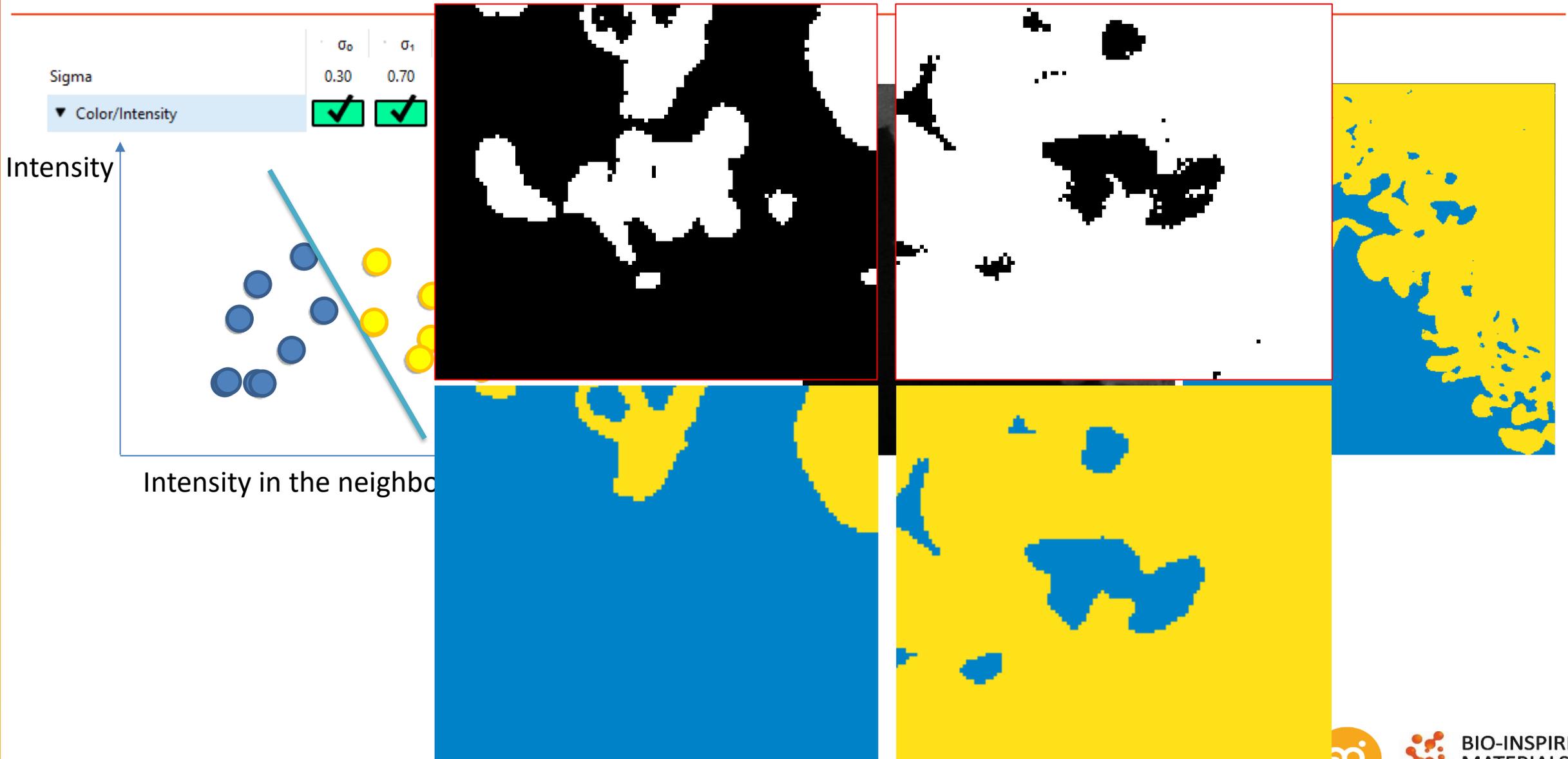
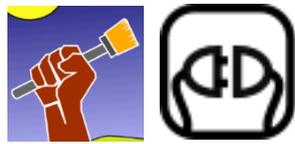
- Color/Intensity
- Gradient magnitude
- Distance to a local edge
- Texture
- Pixel isotropy
- Local curvature

- ▼ Color/Intensity
 - Gaussian Smoothing
- ▼ Edge
 - Laplacian of Gaussian
 - Gaussian Gradient Magnitude
 - Difference of Gaussians
- ▼ Texture
 - Structure Tensor Eigenvalues
 - Hessian of Gaussian Eigenvalues

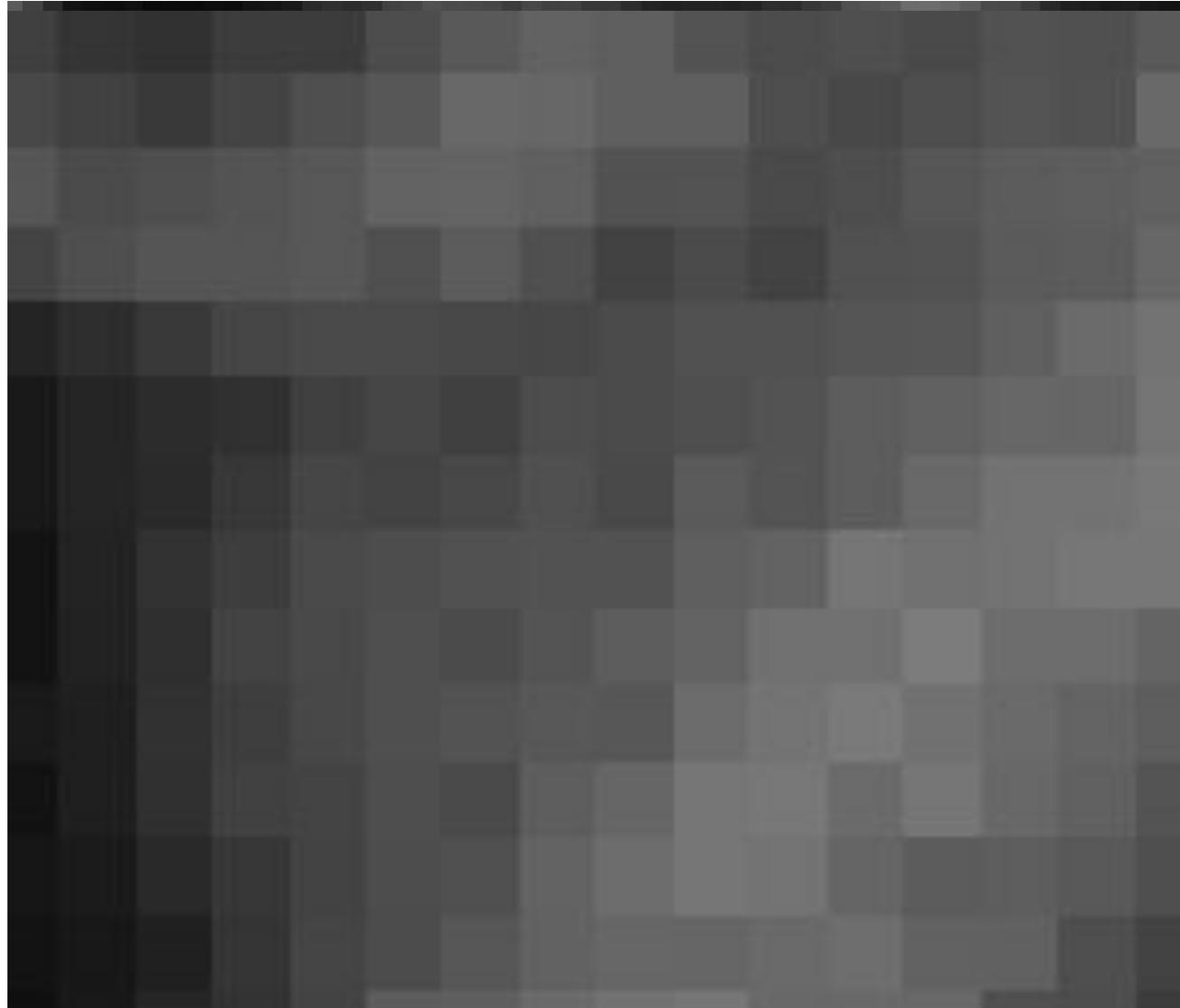
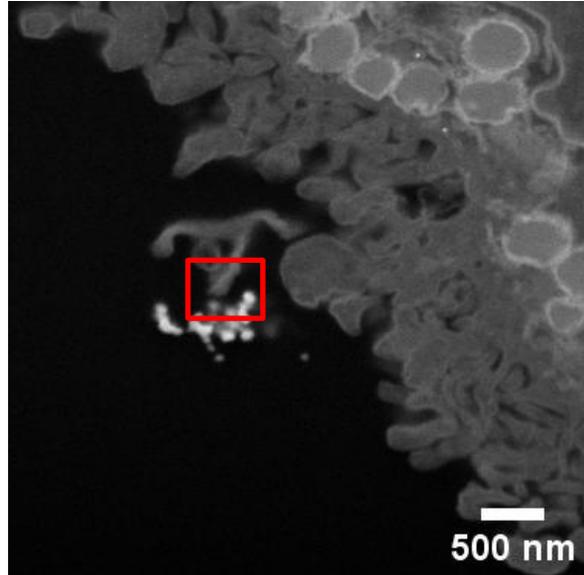
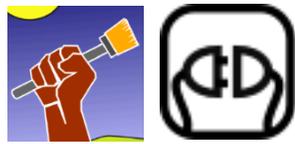
Statistical classification methods

- Artificial neural networks
- Decision tree learning e.g. Random forest
- Kernel estimation e.g. k-nearest neighbour
- Linear classifier e.g. Bayes classifier
- Least squares support vector machine
- ... And many many more

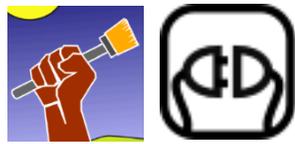
From thresholding to classification



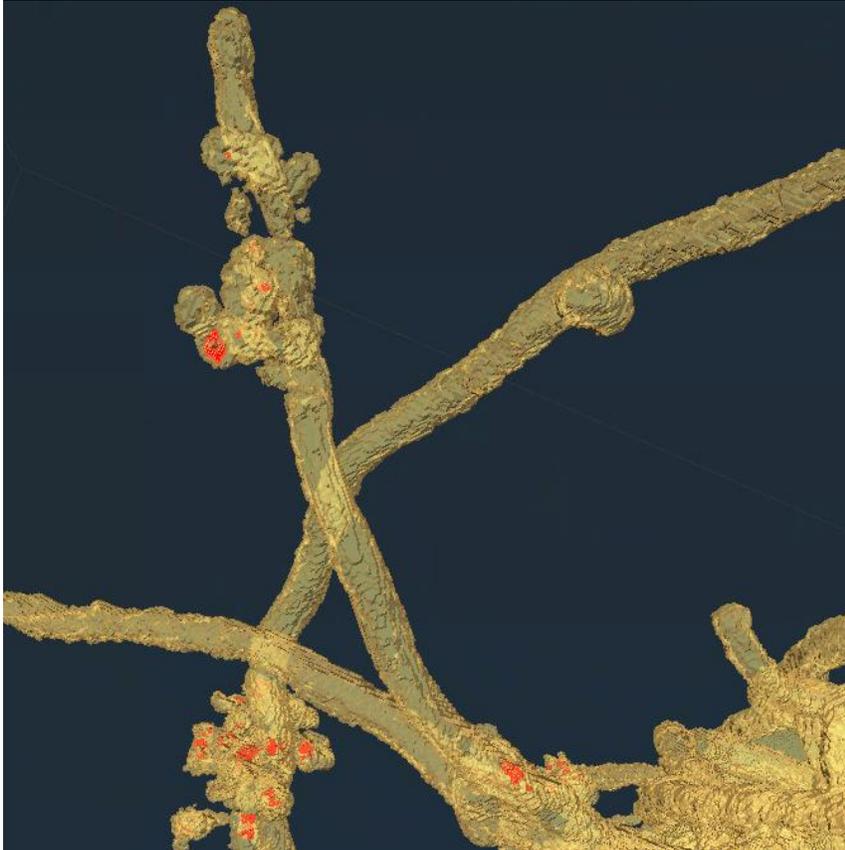
From thresholding to classification



From thresholding to classification to segmentation



- Use random forest ML to create a model
- Use the model to decide on other pixels in your sample (~1 000 000 pixel classifications / s on the Bionano workstation)
- (batch) Export the resulting data as probabilities or segmentations...and in case of 3D data: input them in 3D surface rendering software
- Or quantify



Cell volume:	1871 μm^3
NP inside volume:	25.82 μm^3
NP outside volume:	0.7842 μm^3

(assuming spheres with a diameter of 50 nm)

Number of NP inside the cell 387815

NP per volume cell: 207 NP / μm^3 cell



iLastik

Standalone software iLASTIK
www.ilastik.org

About ilastik



ilastik version 1.4.1rc2

ilastik development has started in 2010 in the [group of Prof. Fred Hamprecht](#) at University of Heidelberg.

In 2018 the ilastik development team has moved with Anna Kreshuk to her [newly established lab](#) at European Molecular Biology Laboratory Heidelberg. More information can be found at www.ilastik.org.

The full list of contributors over time can be found at github:

- [ilastik contributors](#)
- [lazyflow contributors](#)
- [volumina contributors](#)



ilastik - No Project Loaded

Project Settings Help

ilastik



www.ilastik.org

[@ilastik-team](#)

[Get in Touch](#)

Create New Project

- Segmentation Workflows
 - Pixel Classification**
 - Autocontext (2-stage)
 - Neural Network Classification (Local)
 - Neural Network Classification (Remote)
 - Trainable Domain Adaptation (Local) (beta)
 - Carving
 - Boundary-based Segmentation with Multicut
- Object Classification Workflows
 - Object Classification [Inputs: Raw Data, Pixel Prediction Map]
 - Object Classification [Inputs: Raw Data, Segmentation]
- Tracking Workflows

Open Project

- Browse Files

Open Recent Project

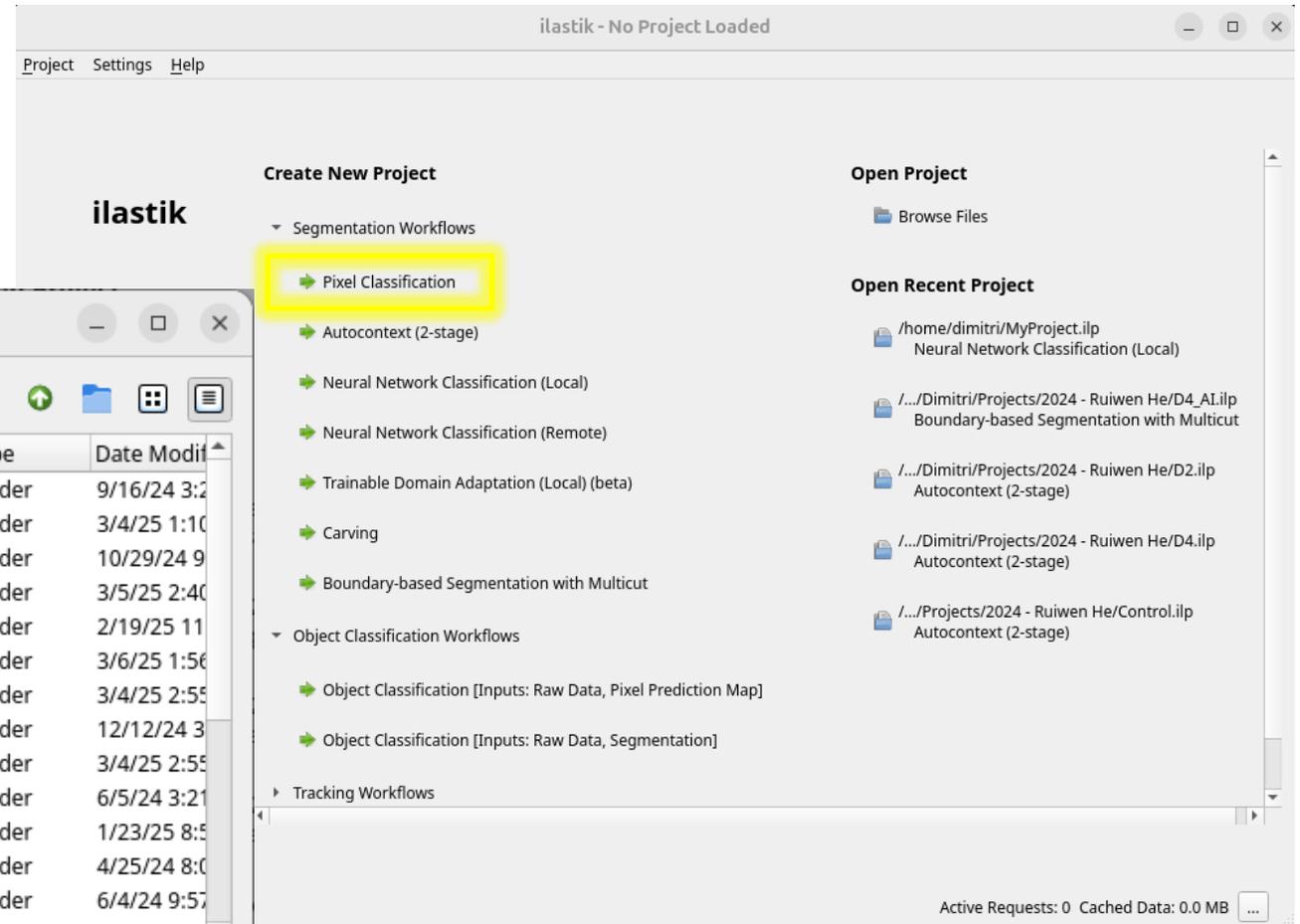
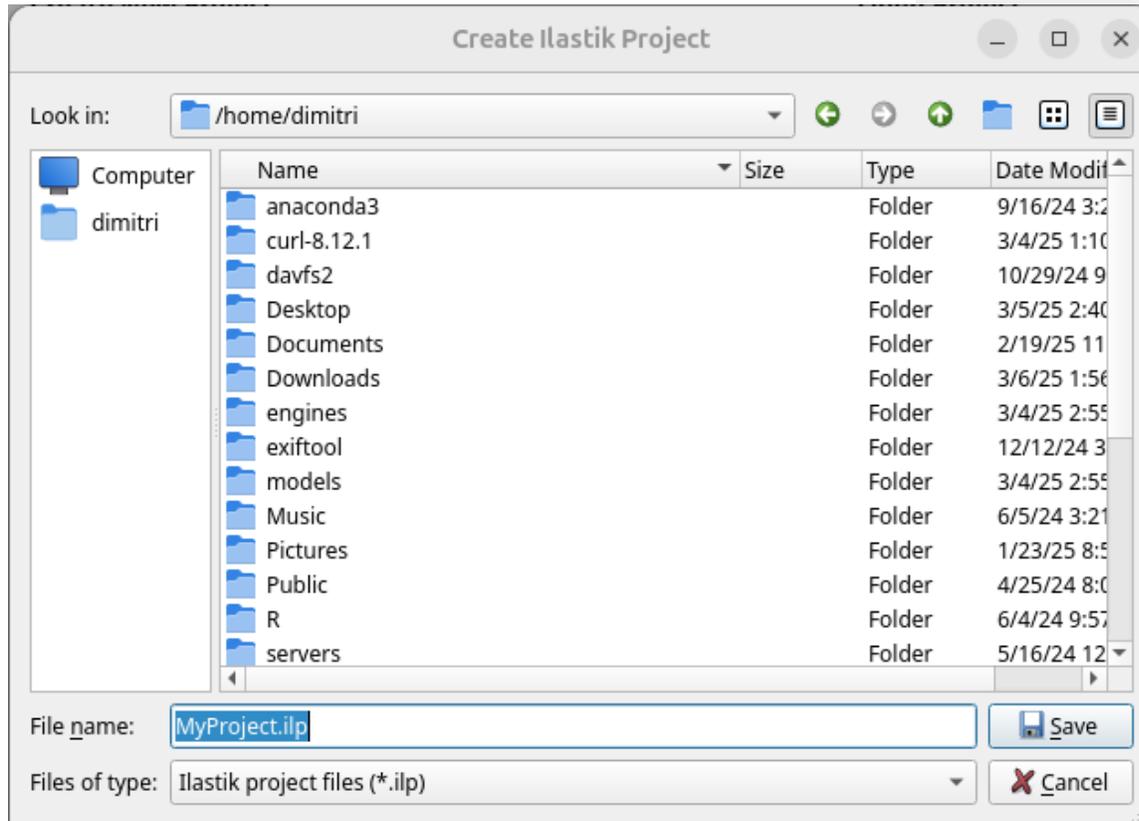
- /home/dimitri/MyProject.ilp
Neural Network Classification (Local)
- /.../Dimitri/Projects/2024 - Ruiwen He/D4_AI.ilp
Boundary-based Segmentation with Multicut
- /.../Dimitri/Projects/2024 - Ruiwen He/D2.ilp
Autocontext (2-stage)
- /.../Dimitri/Projects/2024 - Ruiwen He/D4.ilp
Autocontext (2-stage)
- /.../Projects/2024 - Ruiwen He/Control.ilp
Autocontext (2-stage)

Active Requests: 0 Cached Data: 0.0 MB



iLastik

Standalone software iLASTIK
www.ilastik.org





iLastik: 1. Input data

Process

ilastik - C:/Users/vanheckd/Desktop/test ilastik/TestProject.ilp - Pixel Classification

Project Settings Help

1. Input Data

Raw Data Prediction Mask Summary

Nickname	Location	Internal Path	Axes	Shape	Data Range
+ Add New..					

Select your input data using the 'Raw Data' tab shown on the right

2. Feature Selection

3. Training

4. Prediction Export

5. Batch Processing

Active Requests: 0 Cached Data: 0.0 MB

1. Add new > Add separate images... >
 - Example4-stack.h5 (for a fast PC)
 - Example4-single.h5 (for a normal PC)
2. Click in the left process menu Feature selection

make sure your training images have

- Grayscale LUT
- No burnt-in scalebar



iLastik: 1. Input data

The screenshot displays the iLastik software interface, which is used for image segmentation. It features three 2D image windows and a 3D view area.

- Top Left Window (Z):** Shows a 2D image with a toolbar at the top. The toolbar includes a zoom level of 24/48, a refresh icon, and a 100% zoom indicator. The image content is mostly black.
- Top Right Window (X):** Shows a 2D image with a toolbar at the top. The toolbar includes a zoom level of 256/511, a refresh icon, and a 100% zoom indicator. The image content is mostly black with some faint white spots.
- Bottom Left Window (Y):** Shows a 2D image with a toolbar at the top. The toolbar includes a zoom level of 256/511, a refresh icon, and a 100% zoom indicator. The image content is mostly black with some faint white spots.
- Bottom Right Area:** A large, empty 3D view area. At the bottom right of this area, there is a toolbar with icons for X, Y, Z, and a "Show 3D" checkbox.

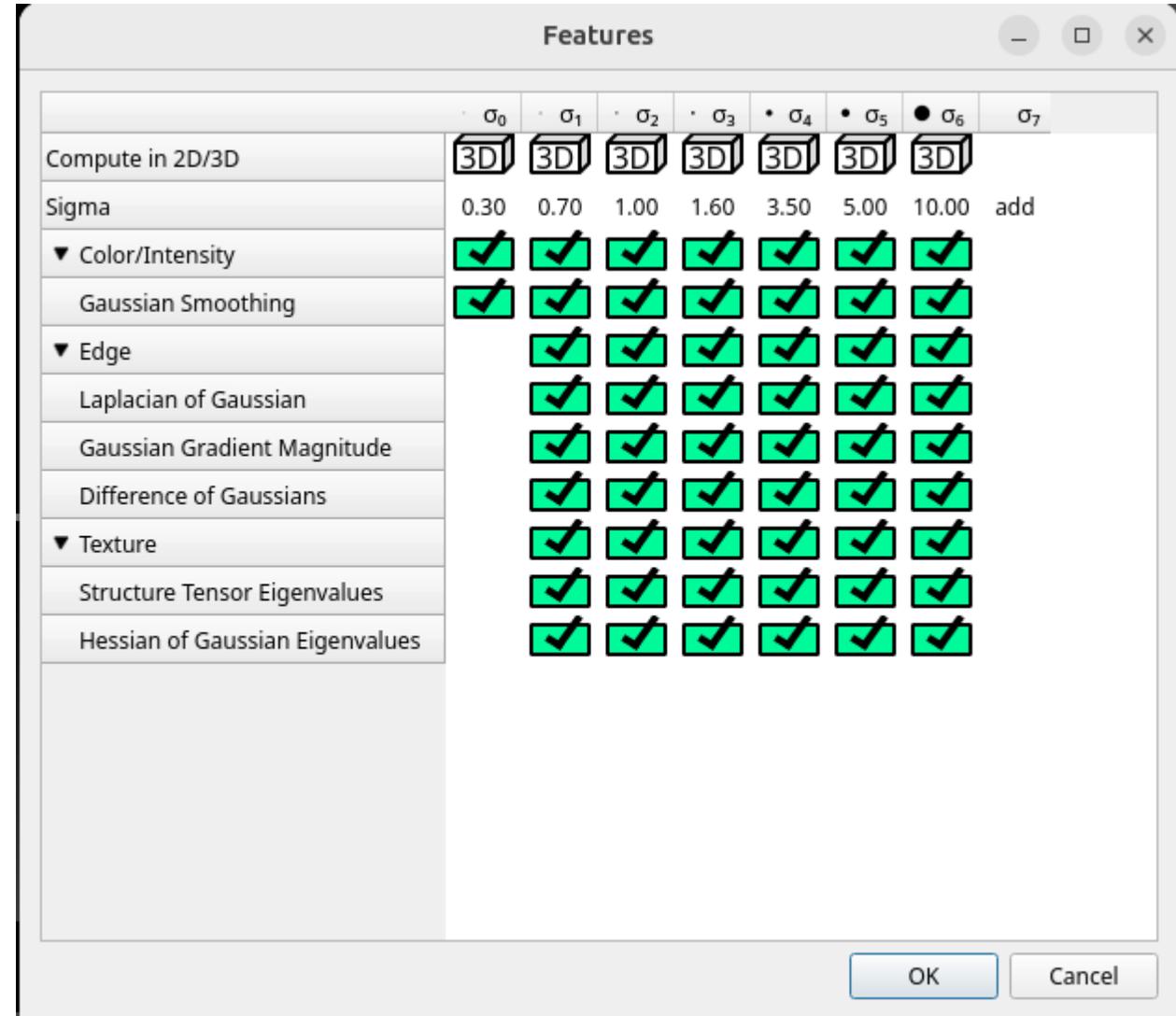


iLastik: 2 Feature selection

Process



Select features...
(select all) > click OK



Click 3. Training



iLastik: 3. Training the model

Process

1. Input Data
2. Feature Selection
3. Training
4. Prediction Export
5. Batch Processing

Cell
Background

Live Update

Group Visibility:
 Probability Segmentation
Labels $\alpha=100.0$
Uncertainty $\alpha=100.0$
Segmentation (Cell) $\alpha=100.0$
Segmentation (Background) $\alpha=100.0$
Prediction for Cell $\alpha=25.0$
Prediction for Background $\alpha=25.0$

X 330 Y 294 Z 40 Crosshairs

Show 3D

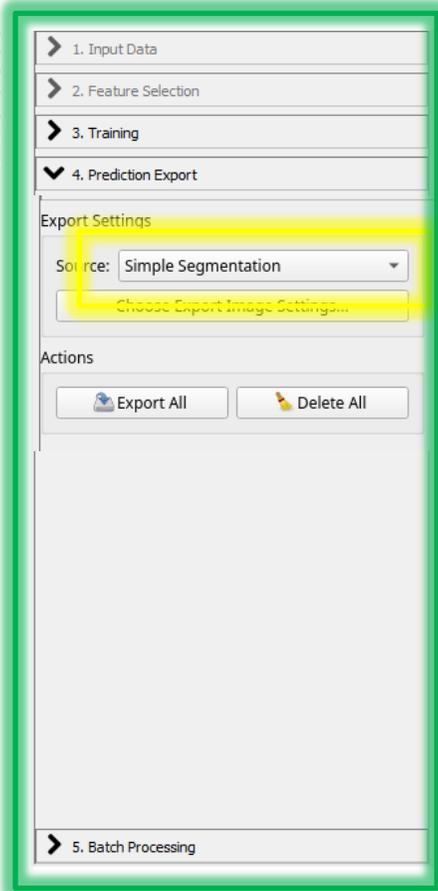
Scroll button: change plane in 3D
Ctrl + Scroll button = zoom in/out
1, 2 ... = label select
I = Image overlay
S = segmentation
U = Uncertainty / probability

Iterate and improve your result



iLastik: 4. Save the data

Process



Single Image:

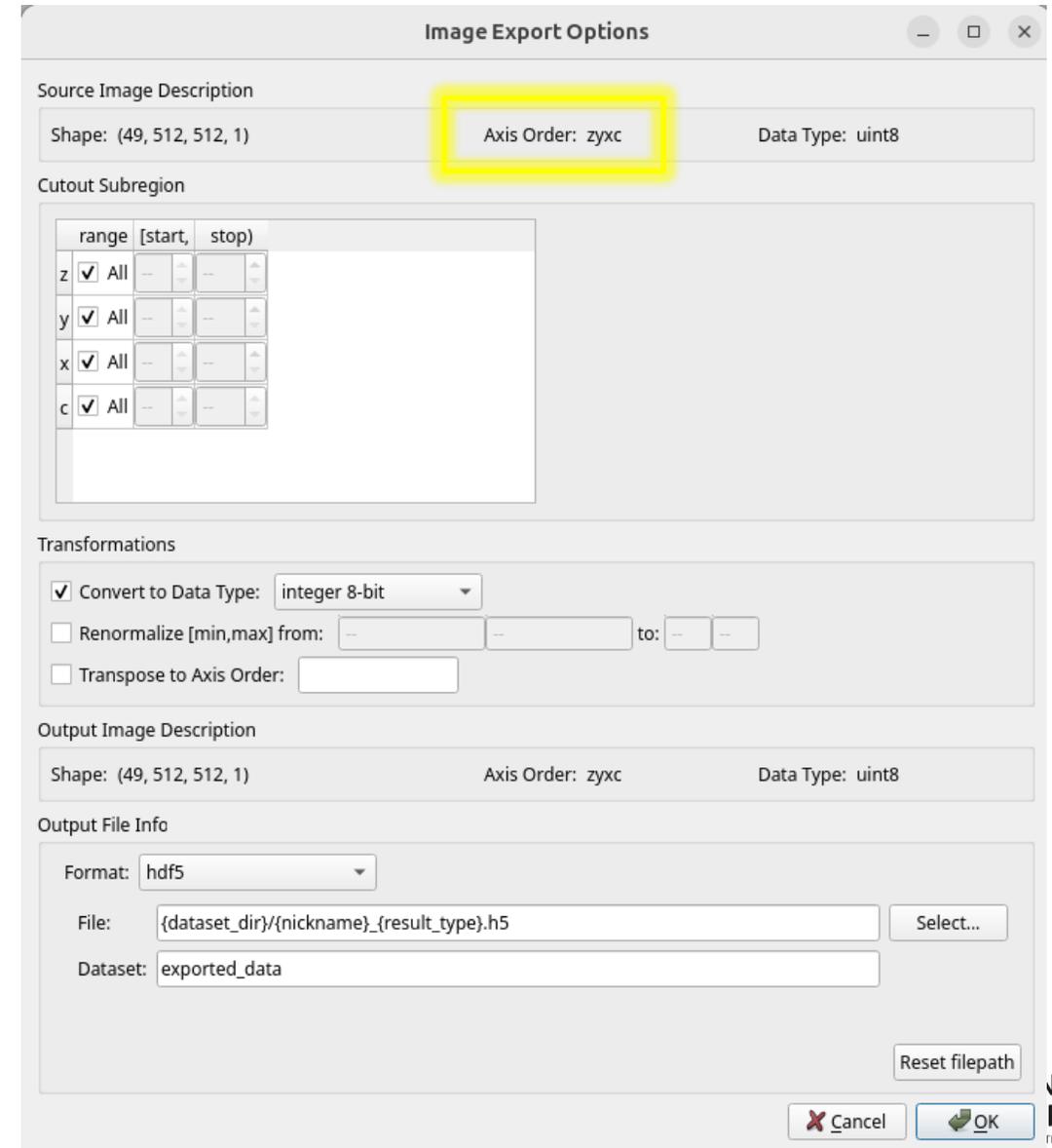
Tick: convert to unsigned 8-bit
Format: .hdf5 (tif also possible)
Click OK

Stack:

Tick: convert to unsigned 8-bit
Format: .HDF5
Click OK

----- OR -----

Tick: convert to unsigned 8-bit
Format: TIF sequence
Click OK





Back to ImageJ: Open the segmented data

Stack saved as HDF

File > Import > HDF5...

- Click on the data (/exported_data)
- Select «Individual hyperstacks (custom layout)»
- Data set layout: change to «zyxt»

Select data sets

data set path	size	type	element size [um]
/exported_data	49×512×512×1	uint8	unknown

Load as ...

individual stacks

individual hyperstacks (custom layout)

- data set layout:

Combine to ...

hyperstack (multichannel)

hyperstack (time series)

hyperstack (multichannel time series)

- Number of channels:

Stack saved as tif sequence

File > Import > Image sequence

Filter: "Simple Segmentation"

(→ "Count" should then become 49)

Import Image Sequence

Dir:

drag and drop target

Type:

Filter:

enclose regex in parens

Start:

Count:

Step:

Scale: %

Sort names numerically

Use virtual stack

Open as separate images

Single saved as tif

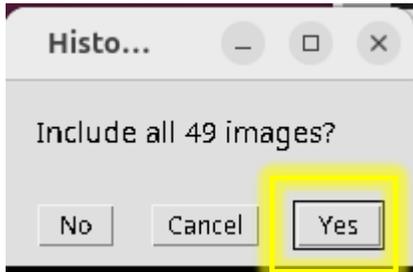
File > Open...





Ilastik output

Process > Histogram...



Min = 1, Max = 2



(i.e. all values are either 1 or 2)

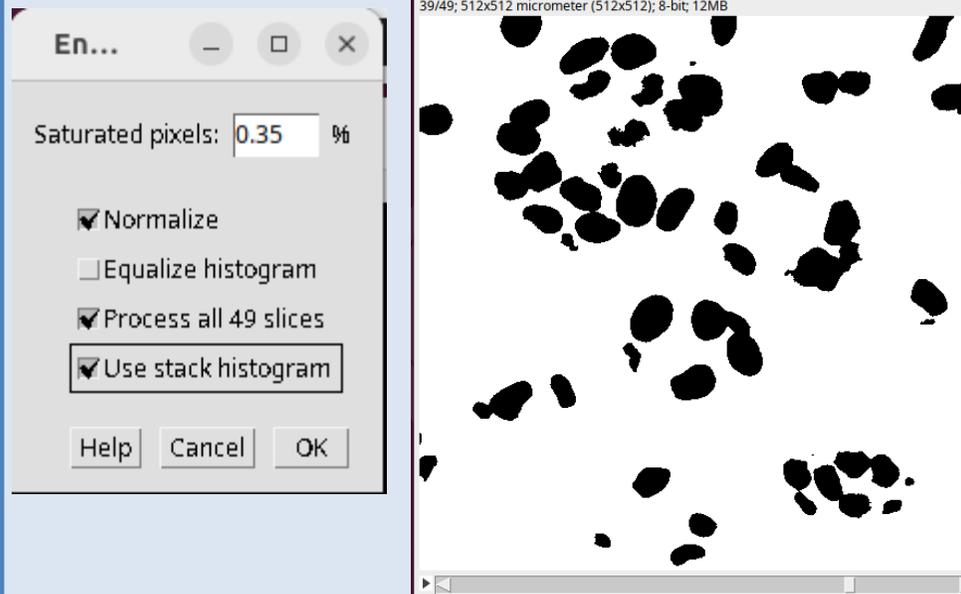
Histog... - □ ×

File Edit Font

value	count
0	0
1	653839
2	12191217
3	0

Normalize

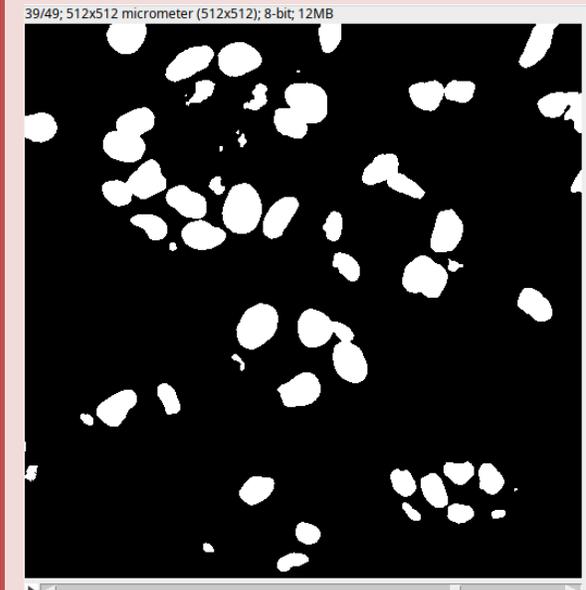
Process > enhance contrast



Invert

(Foreground objects = white)

Edit > Invert





Back to ImageJ: Loss of metadata

Before iLastik thresholding

36/49; 184.52x184.52 micrometer

Channels (c): 1
Slices (z): 49
Frames (t): 1
Note: c*z*t must equal 49

Pixel width: 0.3603981 microm
Pixel height: 0.3603981 -
Voxel depth: 0.6059462 -

Frame interval: 0 sec
Origin (pixels): 0,0,0

Invert Y coordinates
 Global

Cancel OK

After iLastik Thresholding

36/49; 512x512 micrometer

Channels (c): 1
Slices (z): 49
Frames (t): 1
Note: c*z*t must equal 49

Pixel width: 1.0000 microm
Pixel height: 1.0000 -
Voxel depth: 1.0000 -

Frame interval: 0 sec
Origin (pixels): 0,0,0

Invert Y coordinates
 Global

Cancel OK

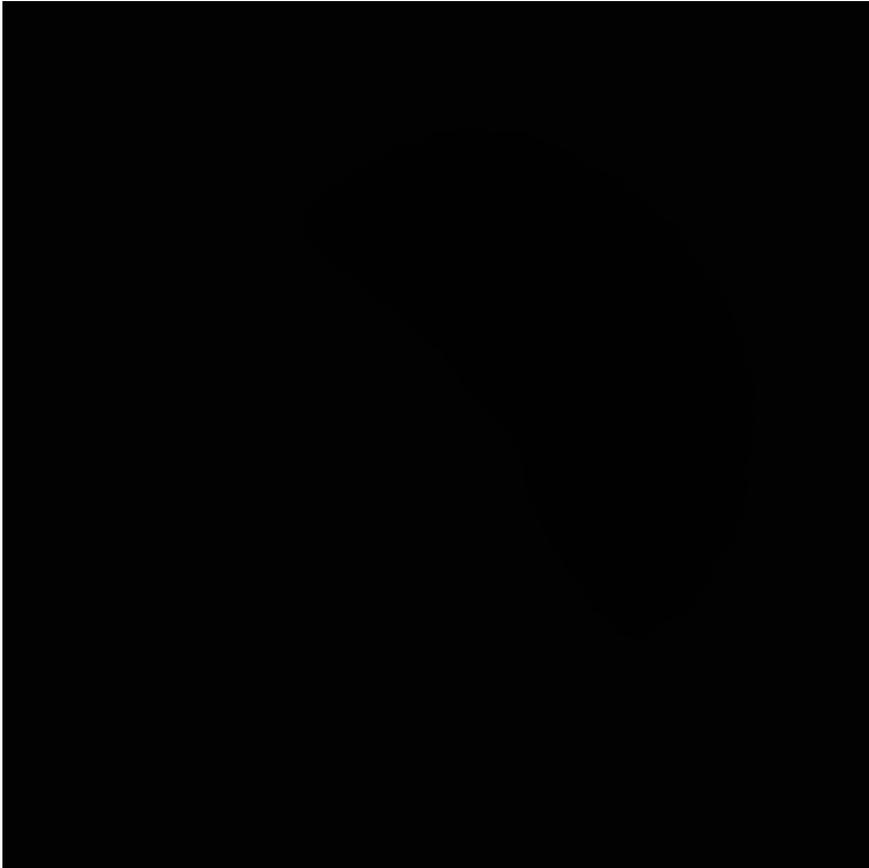
Image > Properties ...



Ilastik output

EXERCISE

Open the segmentation result.





iLastik: 5. Batch processing

Process

1. Input Data
2. Feature Selection
3. Training
4. Prediction Export
5. Batch Processing

Select the input files for batch processing using the controls on the right.
The results will be exported according to the same settings you chose in the interactive export page above.

Process all files

Select raw datafiles ...

Run «process all files» (can take a while)

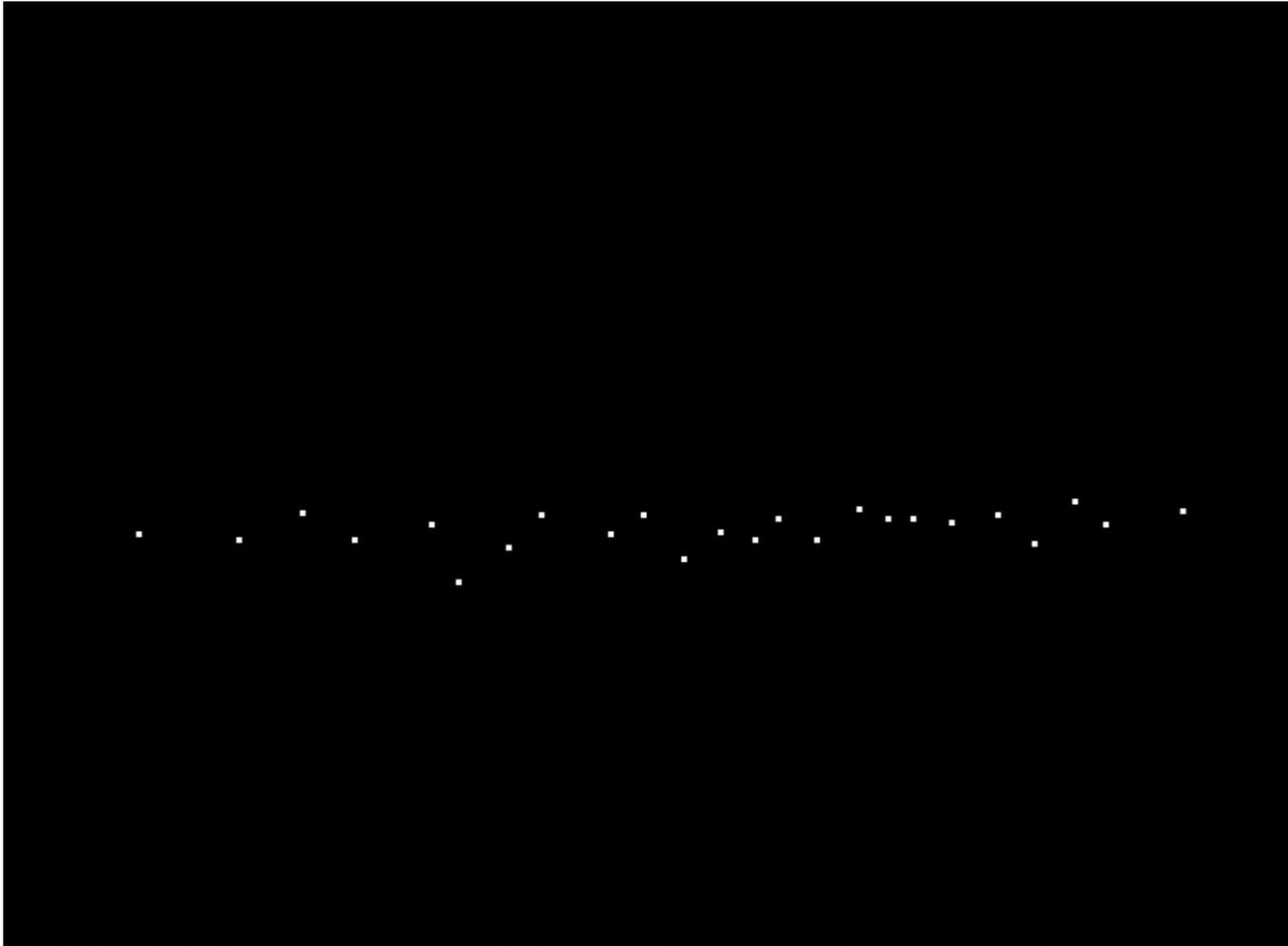
Select Raw Data Files...

Z:\Teaching\ImageJ course\ImageJ basics\Thresholding\Example4 stack\iLastiktest00.tif
Z:\Teaching\ImageJ course\ImageJ basics\Thresholding\Example4 stack\iLastiktest01.tif
Z:\Teaching\ImageJ course\ImageJ basics\Thresholding\Example4 stack\iLastiktest02.tif
Z:\Teaching\ImageJ course\ImageJ basics\Thresholding\Example4 stack\iLastiktest03.tif
Z:\Teaching\ImageJ course\ImageJ basics\Thresholding\Example4 stack\iLastiktest04.tif
Z:\Teaching\ImageJ course\ImageJ basics\Thresholding\Example4 stack\iLastiktest05.tif
Z:\Teaching\ImageJ course\ImageJ basics\Thresholding\Example4 stack\iLastiktest06.tif
Z:\Teaching\ImageJ course\ImageJ basics\Thresholding\Example4 stack\iLastiktest07.tif
Z:\Teaching\ImageJ course\ImageJ basics\Thresholding\Example4 stack\iLastiktest08.tif
Z:\Teaching\ImageJ course\ImageJ basics\Thresholding\Example4 stack\iLastiktest09.tif
Z:\Teaching\ImageJ course\ImageJ basics\Thresholding\Example4 stack\iLastiktest10.tif
Z:\Teaching\ImageJ course\ImageJ basics\Thresholding\Example4 stack\iLastiktest11.tif
Z:\Teaching\ImageJ course\ImageJ basics\Thresholding\Example4 stack\iLastiktest12.tif
Z:\Teaching\ImageJ course\ImageJ basics\Thresholding\Example4 stack\iLastiktest13.tif
Z:\Teaching\ImageJ course\ImageJ basics\Thresholding\Example4 stack\iLastiktest14.tif
Z:\Teaching\ImageJ course\ImageJ basics\Thresholding\Example4 stack\iLastiktest15.tif
Z:\Teaching\ImageJ course\ImageJ basics\Thresholding\Example4 stack\iLastiktest16.tif
Z:\Teaching\ImageJ course\ImageJ basics\Thresholding\Example4 stack\iLastiktest17.tif
Z:\Teaching\ImageJ course\ImageJ basics\Thresholding\Example4 stack\iLastiktest18.tif
Z:\Teaching\ImageJ course\ImageJ basics\Thresholding\Example4 stack\iLastiktest19.tif
Z:\Teaching\ImageJ course\ImageJ basics\Thresholding\Example4 stack\iLastiktest20.tif
Z:\Teaching\ImageJ course\ImageJ basics\Thresholding\Example4 stack\iLastiktest21.tif
Z:\Teaching\ImageJ course\ImageJ basics\Thresholding\Example4 stack\iLastiktest22.tif
Z:\Teaching\ImageJ course\ImageJ basics\Thresholding\Example4 stack\iLastiktest23.tif
Z:\Teaching\ImageJ course\ImageJ basics\Thresholding\Example4 stack\iLastiktest24.tif





iLastik

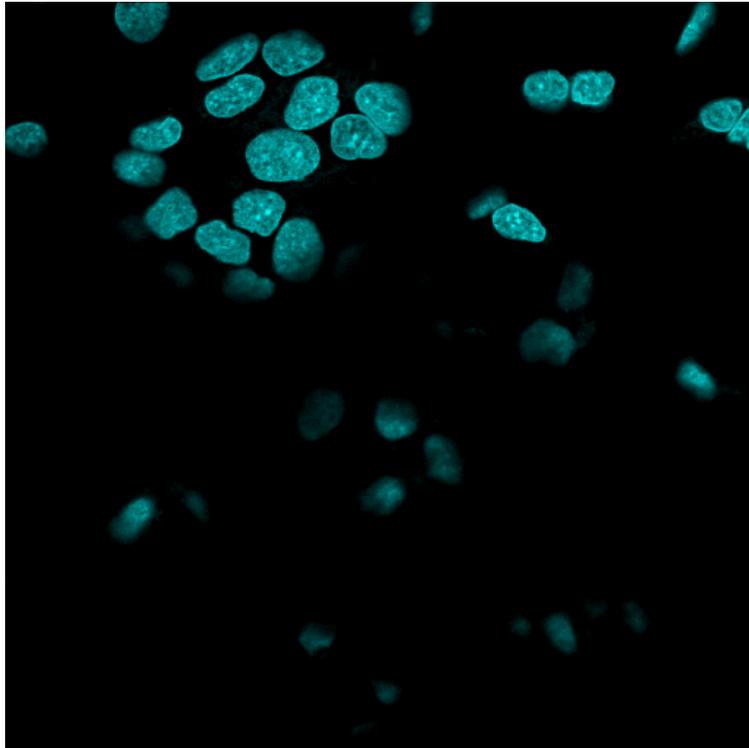


Value	# of Pixels
255	24

Labkit: example and hands-on workflow

EXCERCISE

Open Example 4. Duplicate 1 image of the stack. Train a model using LabKit

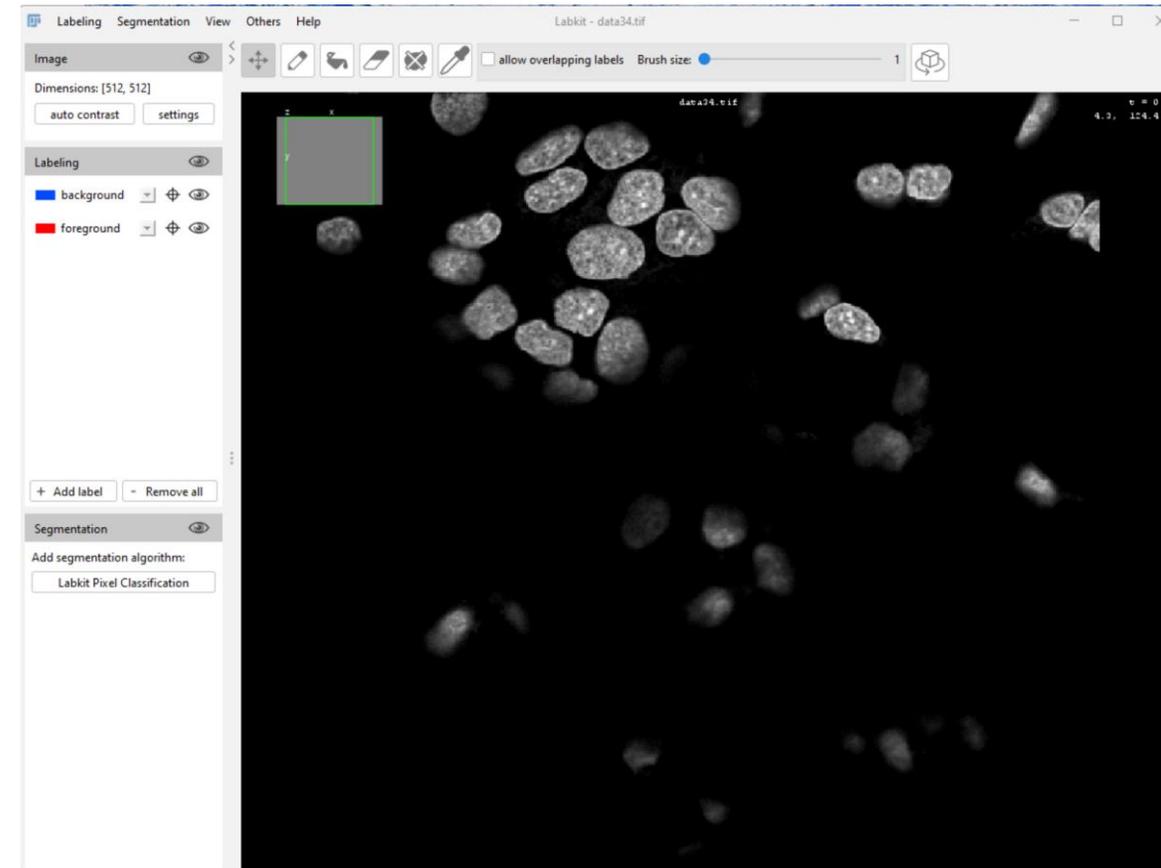
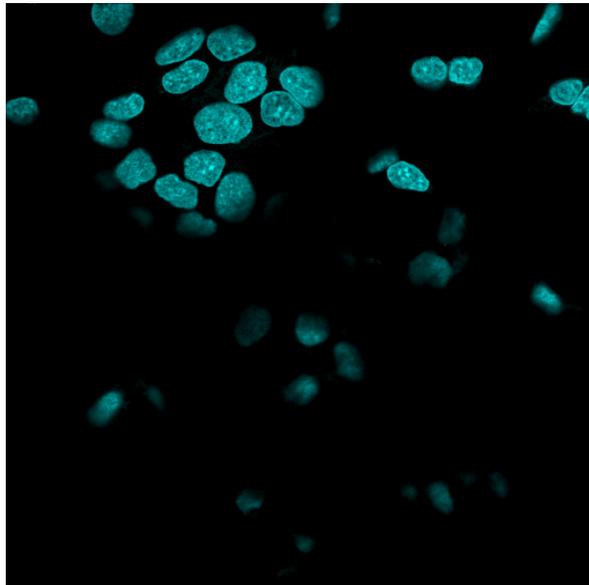


Labkit: example and hands-on workflow

EXCERCISE

Open Example 4. Duplicate 1 image of the stack. Train a model using LabKit

- Install labkit from the repository: Help > Update ... > Manage update sites > Labkit
- Restart FIJI
- Open Example 4
- Duplicate 1 image (e.g. # 34)
- Start Labkit: Plugins > Open current image with Labkit



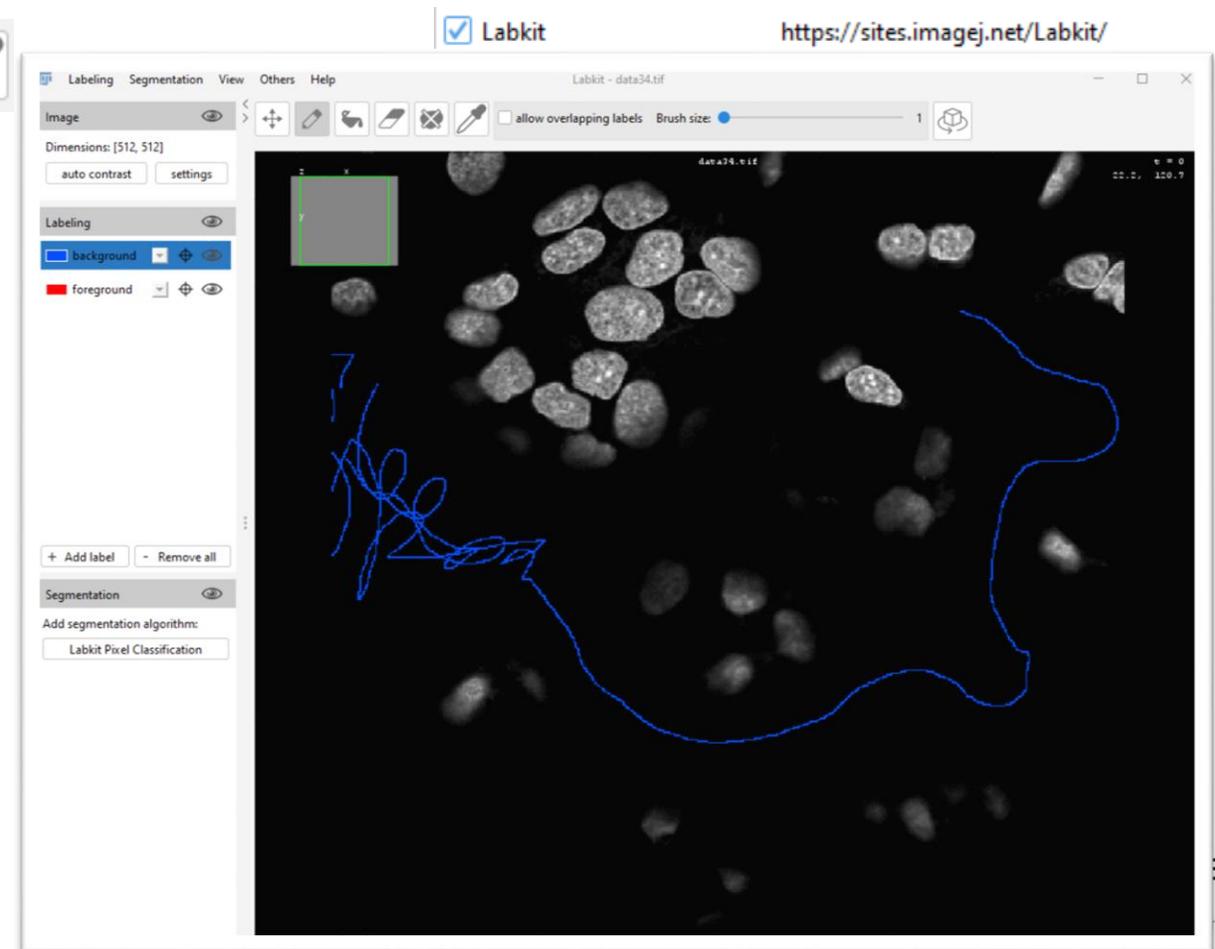
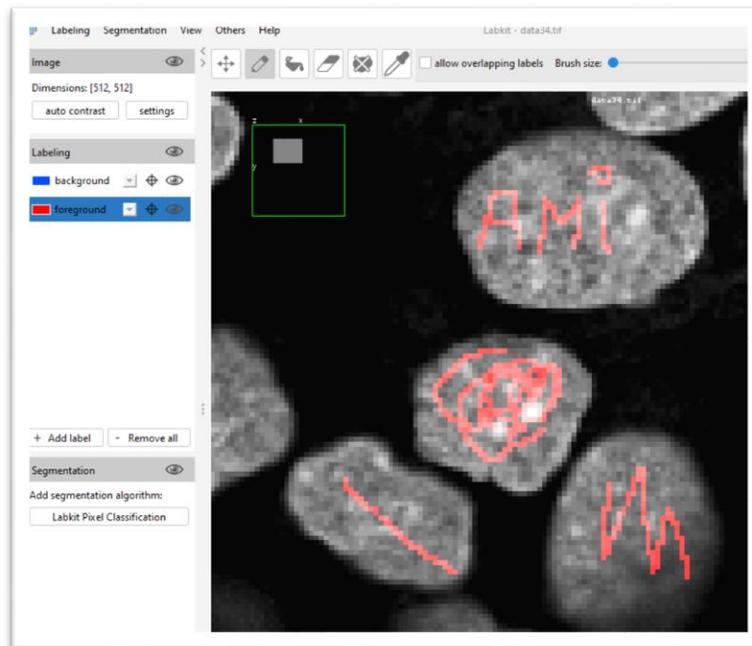
Labkit: example and hands-on workflow

EXERCISE

Open Example 4. Duplicate 1 image of the stack. Train a model using LabKit

1. Train the model

- Select Draw in the top menu
- Select background in the left menu
- paint some background pixels blue
- Repeat for foreground pixels (nuclei)



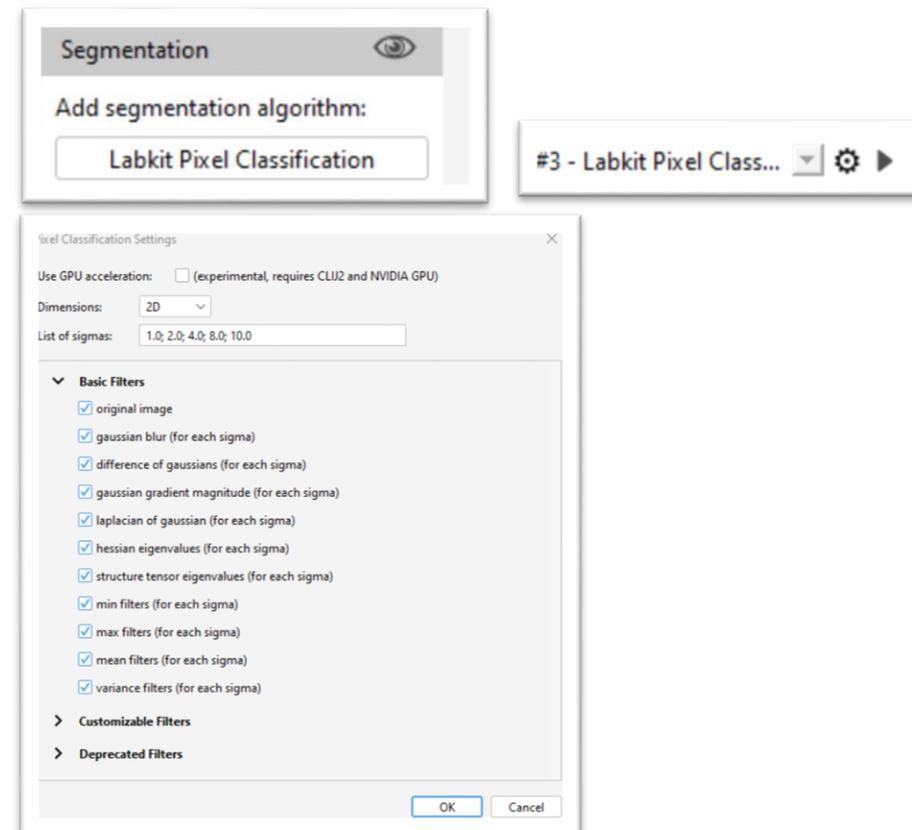
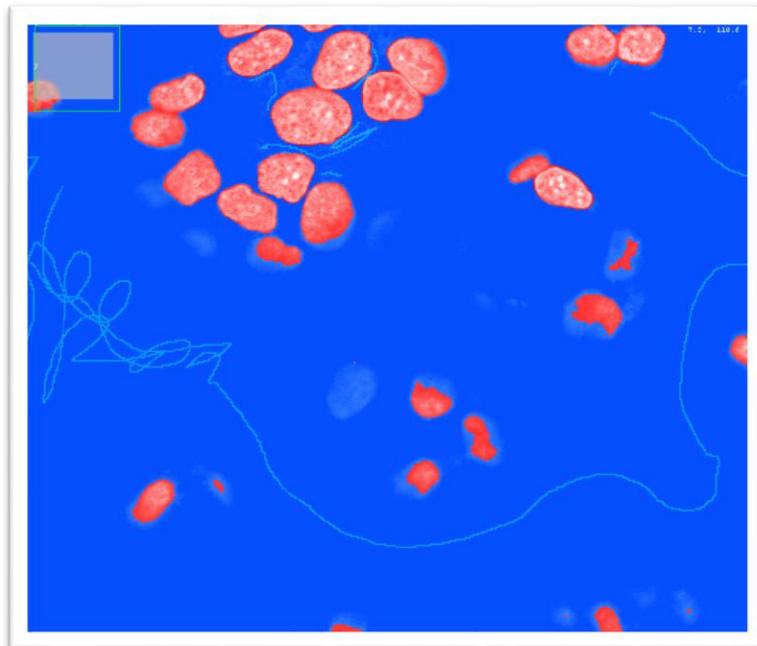
Labkit: example and hands-on workflow

EXERCISE

Open Example 4. Duplicate 1 image of the stack. Train a model using LabKit

2. Add a classifier

- In the left menu, click “Labelkit Pixel classification”
- Click the cog wheel, check all basic filters
- Click the play button  (or CTRL+SHIFT+T)
- Repeat step 1 to optimize the model



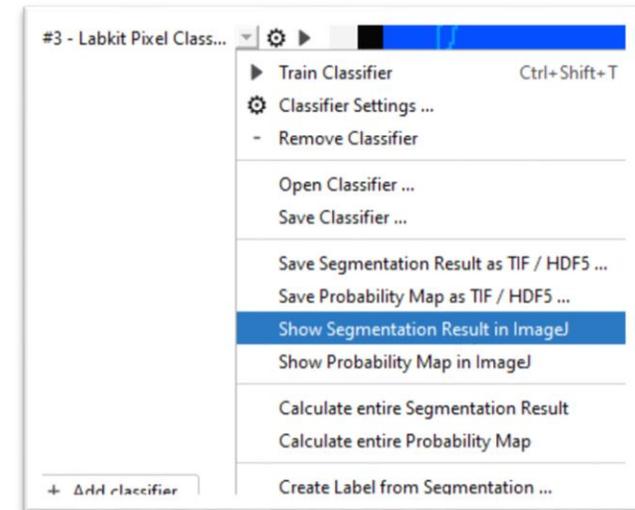
Labkit: example and hands-on workflow

EXCERCISE

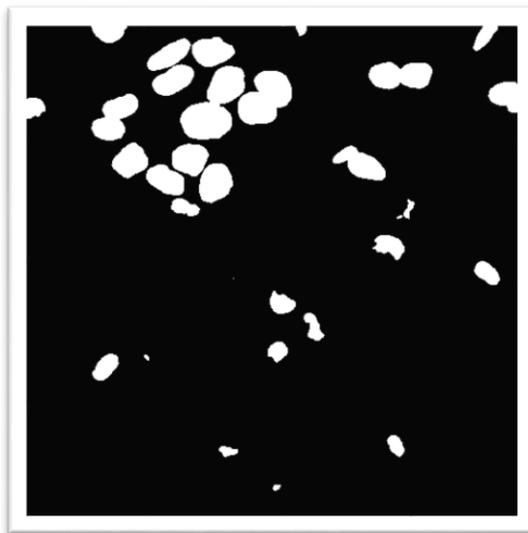
Open Example 4. Duplicate 1 image of the stack. Train a model using LabKit

3. Check model uncertainty and segmentation

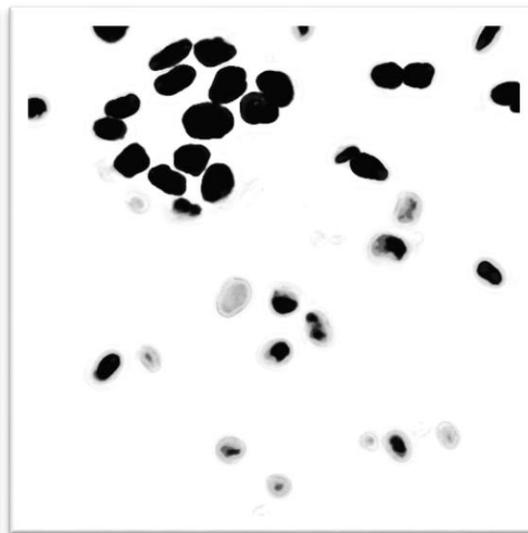
- In the Labkit pixel classifier, click the down arrow 
- Select "Show segmentation Result in ImageJ"
- Repeat with Show Probability Map in ImageJ



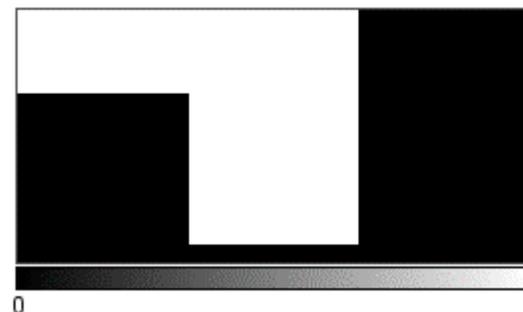
Segmentation



Probability



3 bin histogram



index	bin start	count
0	0.000	19835
1	0.333	2165
2	0.667	240144

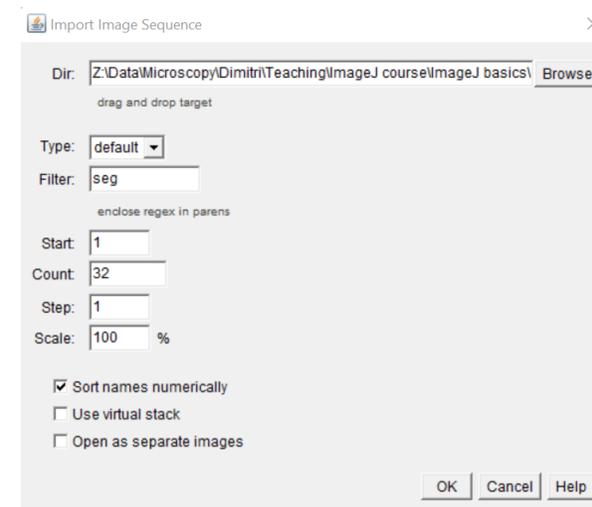
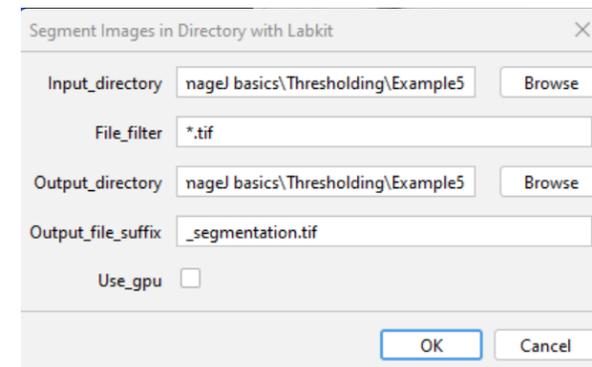
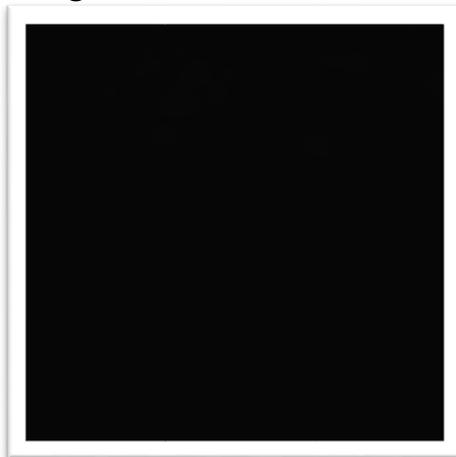
Labkit: example and hands-on workflow

EXERCISE

Open Example 4. Duplicate 1 image of the stack. Train a model using LabKit

4. Batch export: apply the model to all images in the folder

- Save the stack as a list of files: File > save as... > Image sequence...
- In Labkit: Others > Batch segment images...
- Select the folder with the separate images Example 4 (also as output)
- Do not use the GPU
- Run the batch (progress can be followed in the FIJI info bar)
- File import > Image sequence: point to the folder
- Filter: use 'seg' to filter for file names that contain segmentation
- The images are black!



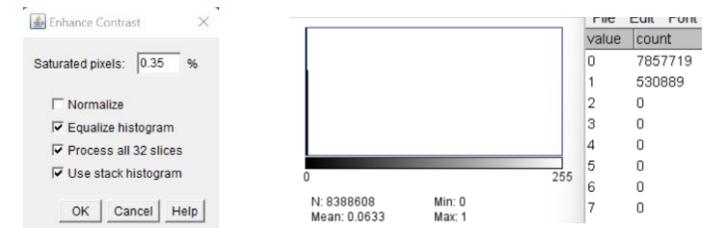
Labkit: example and hands-on workflow

EXERCISE

Open Example 4. Duplicate 1 image of the stack. Train a model using LabKit

5. Equalise the histogram of the segmented data

- With the segmented data stack open: Process > enhance contrast
- Check all except normalize
- Click OK

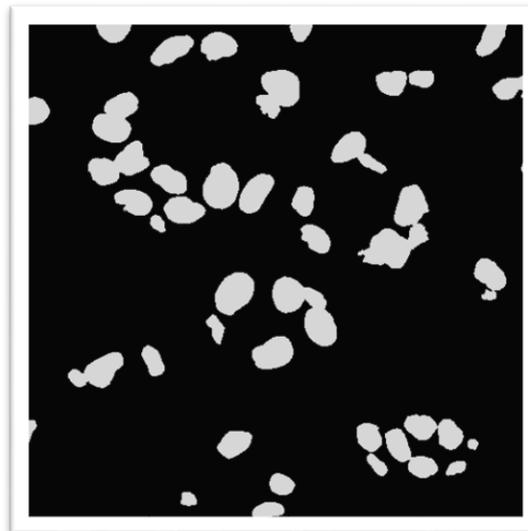


(alternative: Process > Math > Multiply: 255)

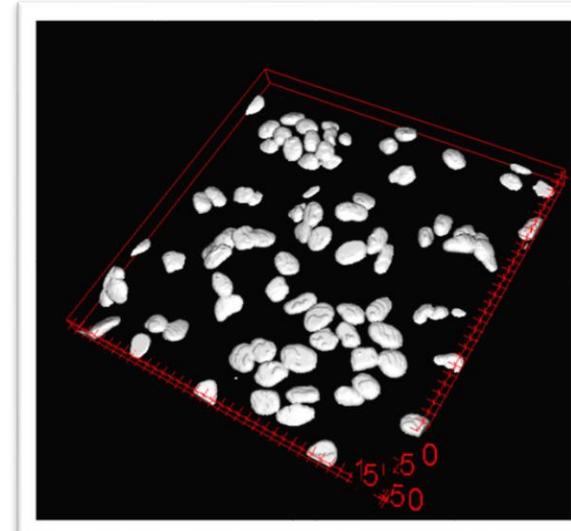
Before equalization



After equalization

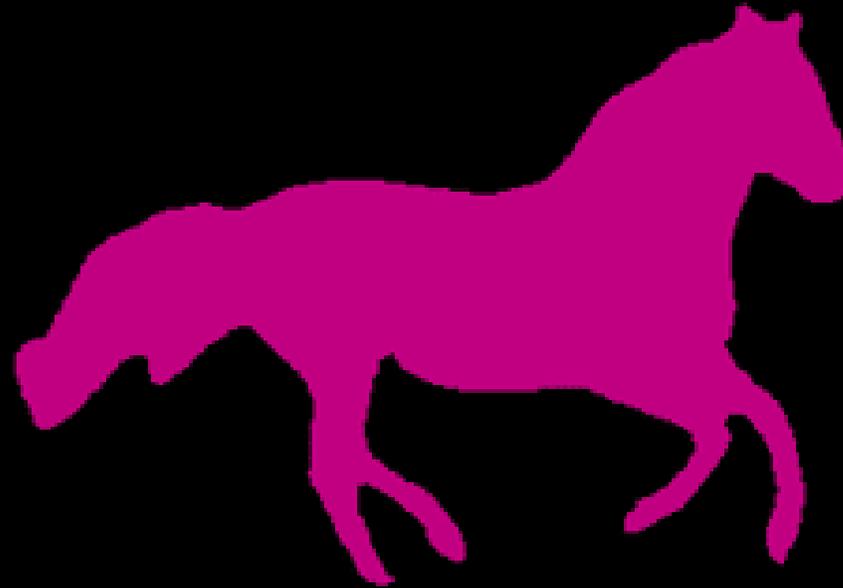


3D rendering



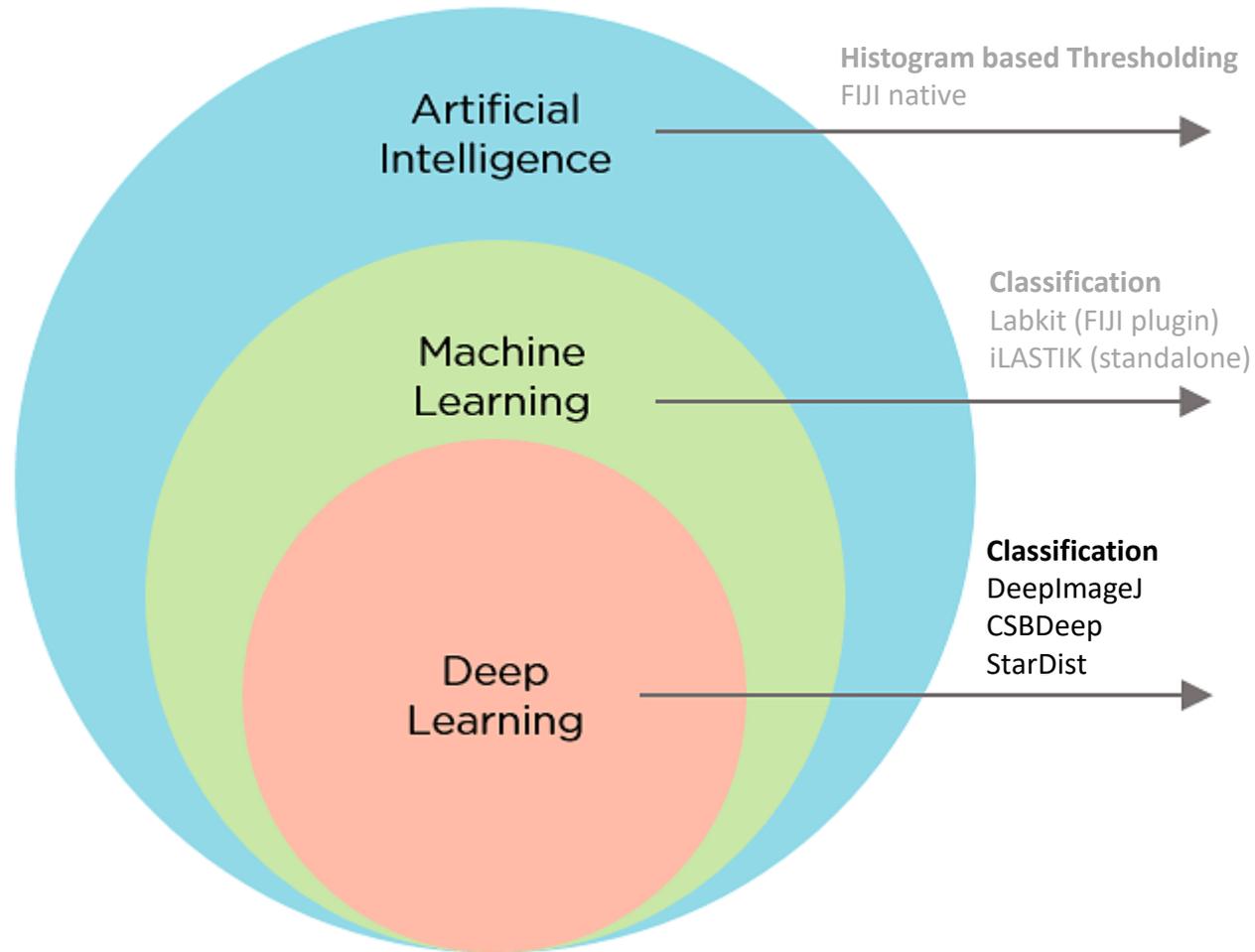
Level 3

Thresholding, classification and segmentation



Deep learning

Thresholding: human vs machine



Ability of a machine to imitate intelligent human behavior

Application of AI that allows a system to automatically learn and improve from experience

Application of Machine Learning that uses complex algorithms and deep neural nets to train a model

Deep Learning

have a look at www.bioimage.io

The screenshot shows the BioImage Model Zoo website. At the top, there is a navigation bar with the BioImage.IO logo and links for Models, Datasets, Applications, Docs, and About. There are also buttons for Upload and Login. Below the navigation bar, the title "BioImage Model Zoo" is displayed in purple. Underneath, a text line reads "Supported by our amazing community partners in AI-powered bioimage analysis". A row of 12 partner logos is shown, including DL4MicEverywhere, CAREamics, ImJoy, SpotMAX, HPA, BiaPy, Fiji, StarDist, QuPath, Icy, ZeroCostDL4Mic, and DeepMa. Below this, a purple heading says "Discover, explore, and deploy cutting-edge bioimage analysis models". A search bar with the placeholder text "Search models, datasets, applications..." is present. To the right of the search bar is a filter icon. Below the search bar, there is a decorative illustration of a savanna scene with a giraffe, an elephant, and a lion. The main content area displays four model cards. Each card features a thumbnail image, a title, an ID, a brief description, and a list of tags.

BioImage.IO Models Datasets Applications Docs About Upload Login

BioImage Model Zoo

Supported by our amazing community partners in AI-powered bioimage analysis

DL4MicEverywhere CAREamics ImJoy SpotMAX HPA BiaPy Fiji StarDist QuPath Icy ZeroCostDL4Mic DeepMa

Discover, explore, and deploy cutting-edge bioimage analysis models

Search models, datasets, applications...

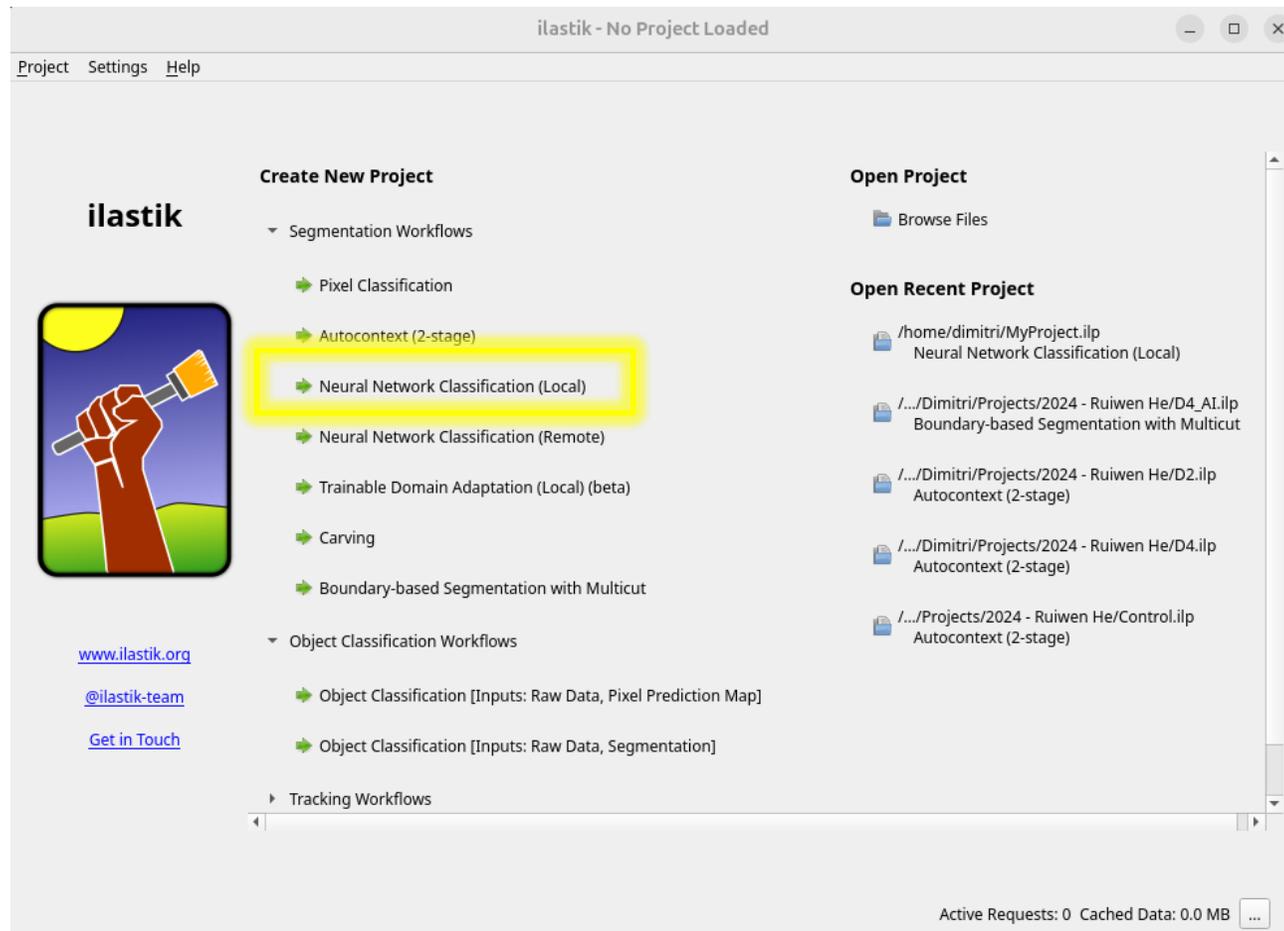
SEM_n2v
ID: humorous-fox
This model is a UNet trained using the Noise2Void algorithm to denoise...
denoising restoration UNet

2D U-NeXt V2 for mitochondri...
ID: sensible-cat
Mitochondria segmentation for electron microscopy
electron-microscopy mitochondria 2d

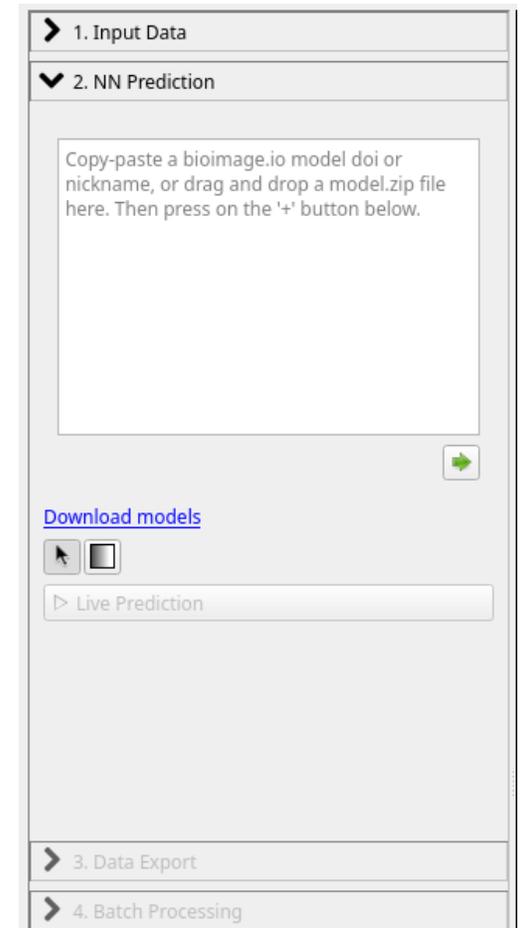
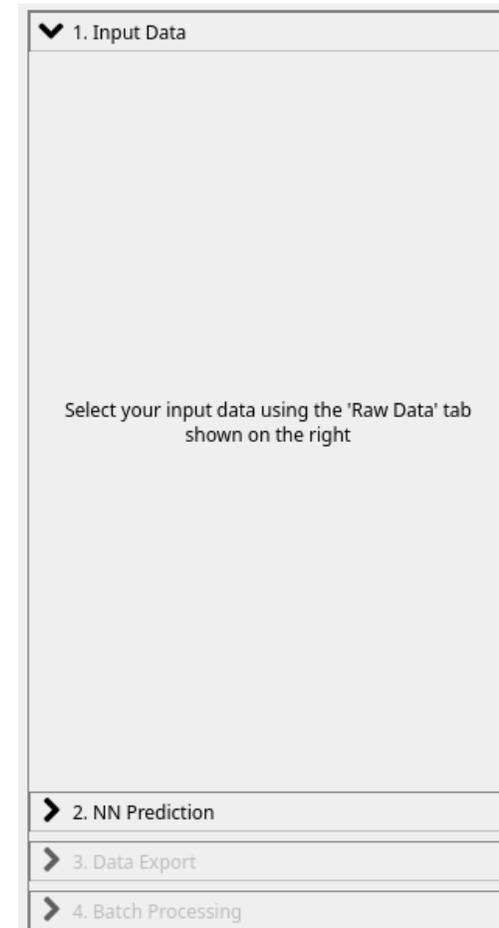
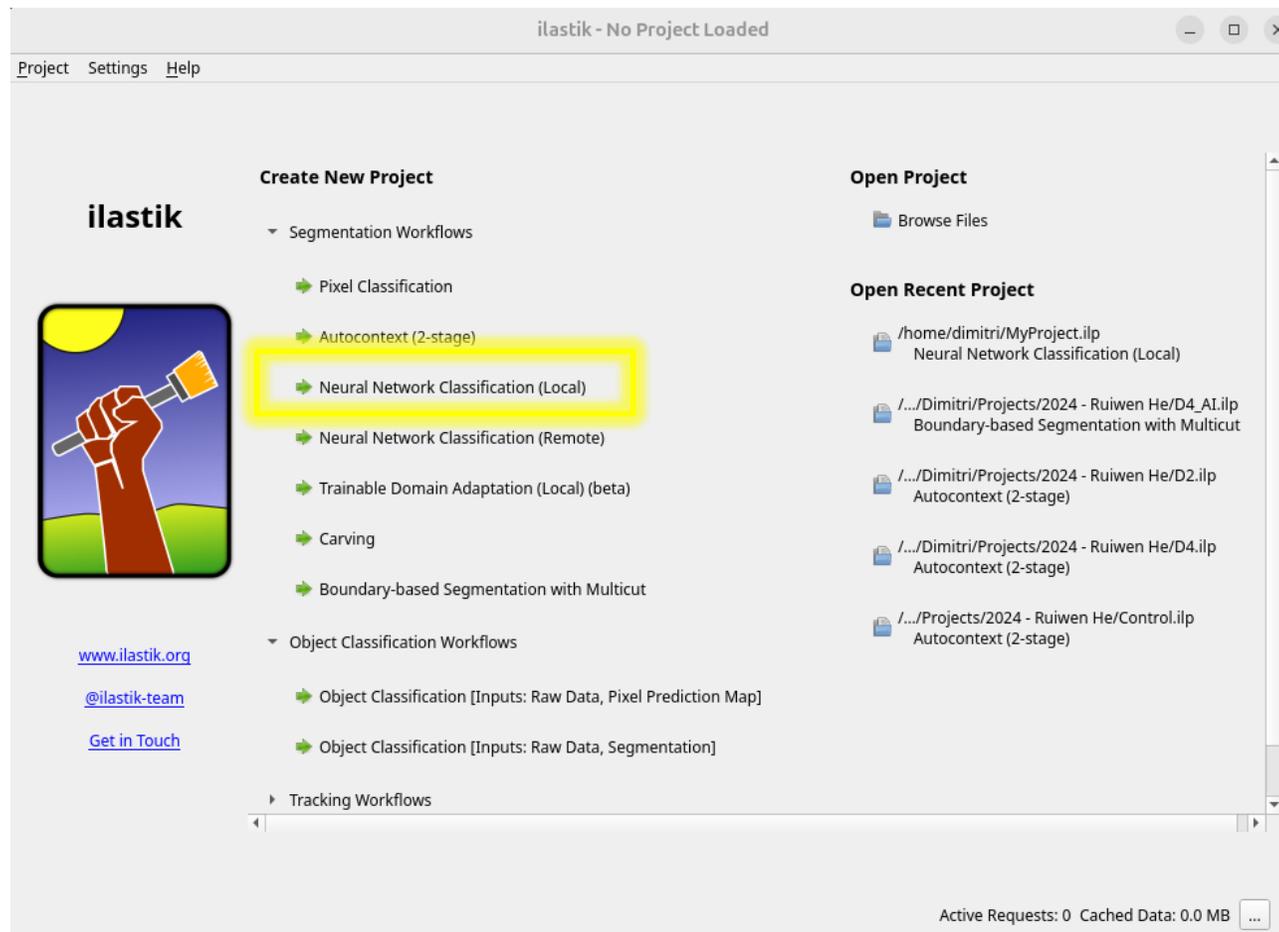
2D MultiResUNet for mitocho...
ID: fear less-deer
Mitochondria segmentation for electron microscopy
electron-microscopy mitochondria 2d

OC1 Project 11 Cellpose
ID: happy-elephant
Segmentation of Epithelial Cells
nuclei segmentation epithelial cells
fluorescence microscopy

Deep Learning with Ilastik



Deep Learning with DeepImageJ



1. Add new > Add separate images... > Example4-ilastik-2D.h5

Deep Learning with DeepImageJ

1. Input Data

2. NN Prediction

Copy-paste a bioimage.io model doi or nickname, or drag and drop a model.zip file here. Then press on the '+' button below.

[Download models](#)

Live Prediction

3. Data Export

4. Batch Processing

NucleiSegmentationBoundaryModel

10.5281/zenodo.5764892 | downloads: 70601 | license

10.5281/zenodo.5764892 | affable-shark

1. Input Data

2. NN Prediction

Copy-paste a bioimage.io model doi or nickname, or drag and drop a model.zip file here. Then press on the '+' button below.

[Download models](#)

Live Prediction

3. Data Export

4. Batch Processing

Downloading model

Downloading model: 0%

weights-torchscript.pt: 0%

Cancel

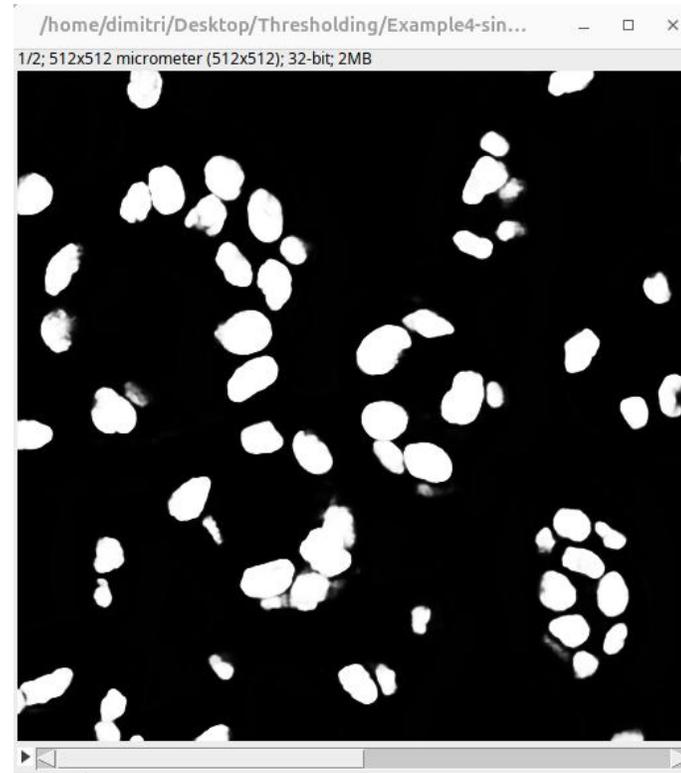
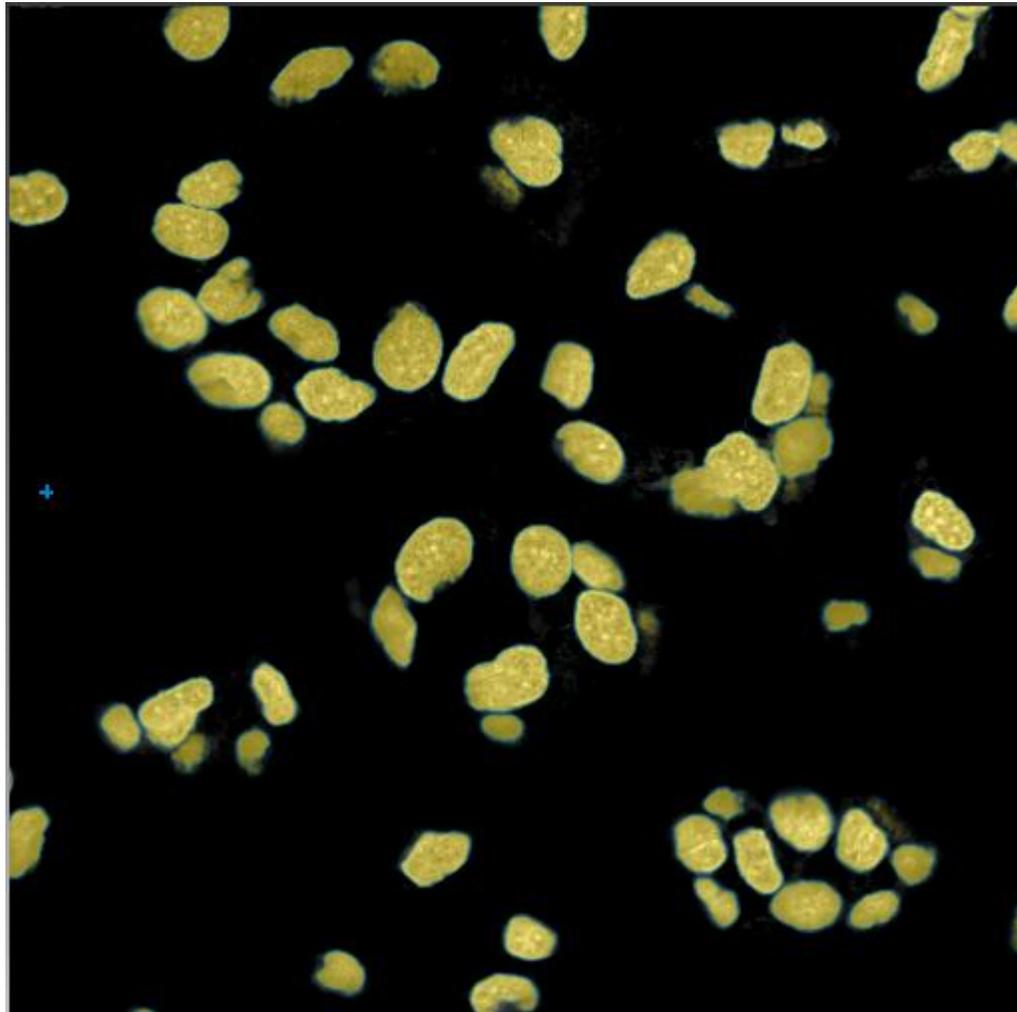
Initializing model

Initializing model: 54%

Cancel

Finally: click Live prediction

Deep Learning with DeepImageJ

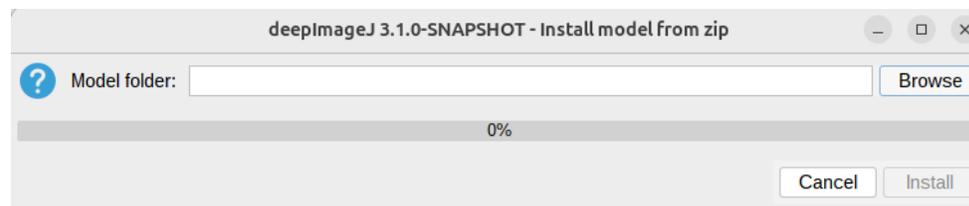
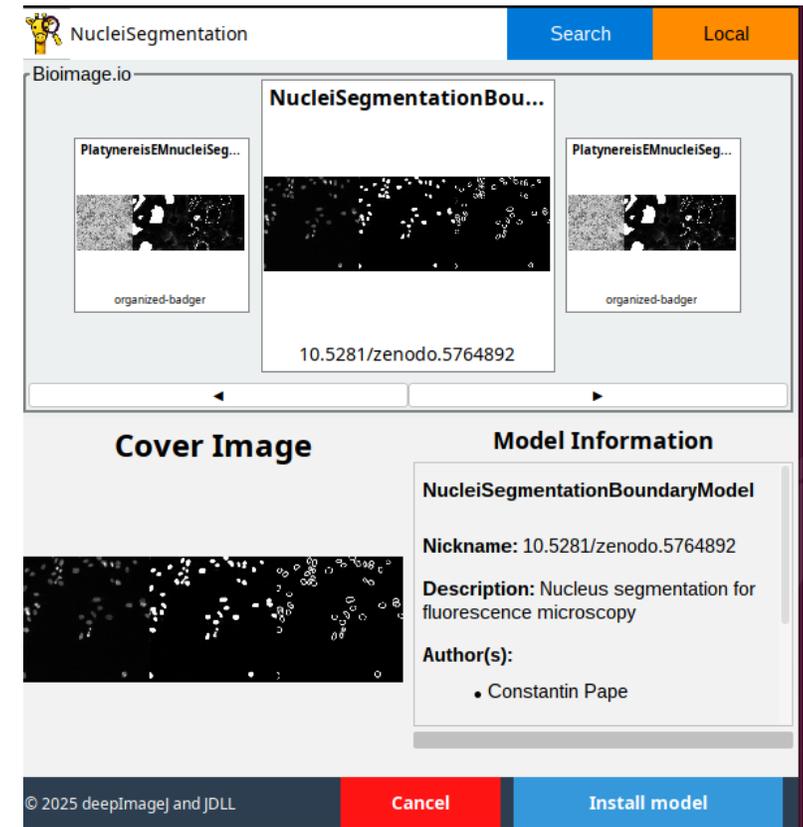
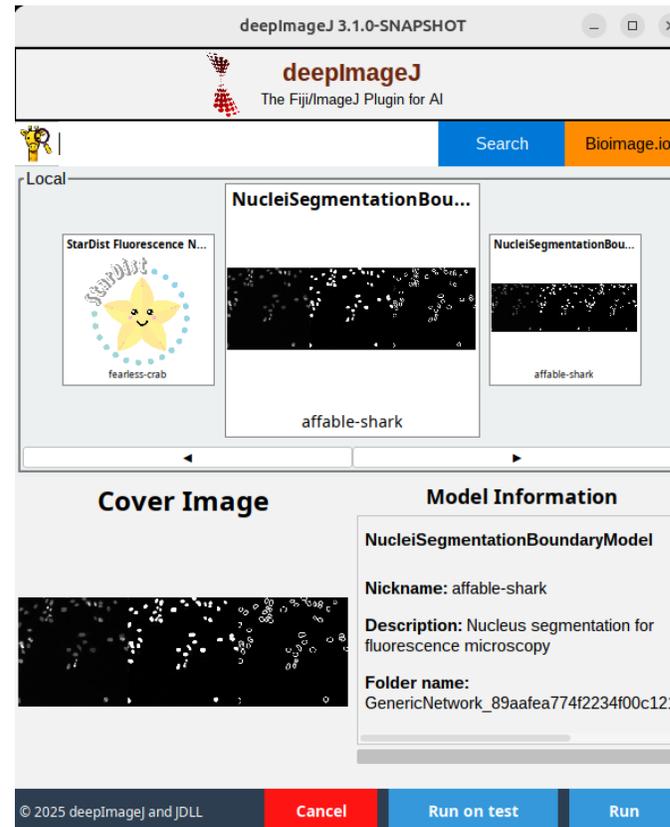


Can we do this in ImageJ, too?

Deep Learning with DeepImageJ

Running a model from the model zoo

- Open the Cell nucleus image (Example4-ilastik-2D.h5)
- Go to Plugins > DeepImageJ > DeepImageJ Run
- Click the orange button «Bioimage.io» (toggles between locally installed and online models)
- The search for: NucleiSegmentation
- Click «Install model»
- Click «Local». You will now see the locally installed models (i.e. 1)
- Click «Run»



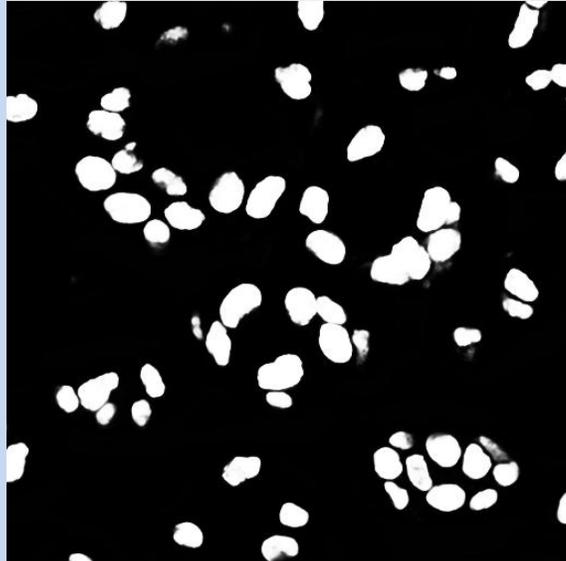
Deep Learning with DeepImageJ

Result: 2 images

Image1 = probabilities

32-bit image, with values between 0 and 1

c:/2 (1); 512x512 pixels; 32-bit; 2MB



N: 262144
Mean: 0.132
StdDev: 0.325
Bins: 256

Min: 0.0000692
Max: 1.000
Mode: 0.00202 (199445)
Bin Width: 0.00391

Image 1: convert to segmentation

Process > Binary > make Binary

"1"; 512x512 pixels; 8-bit; 256K

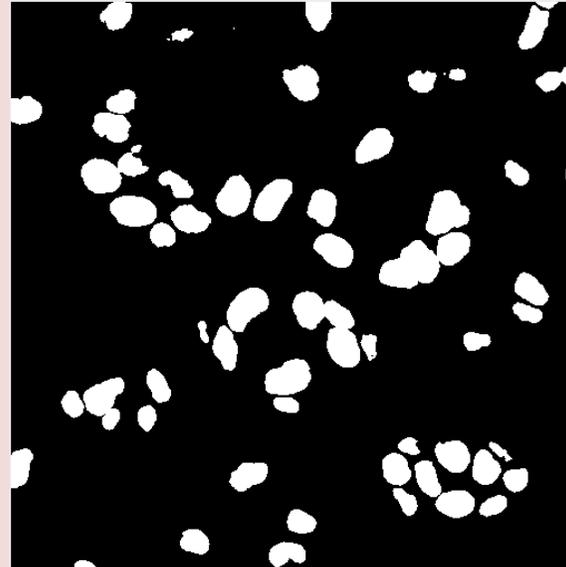
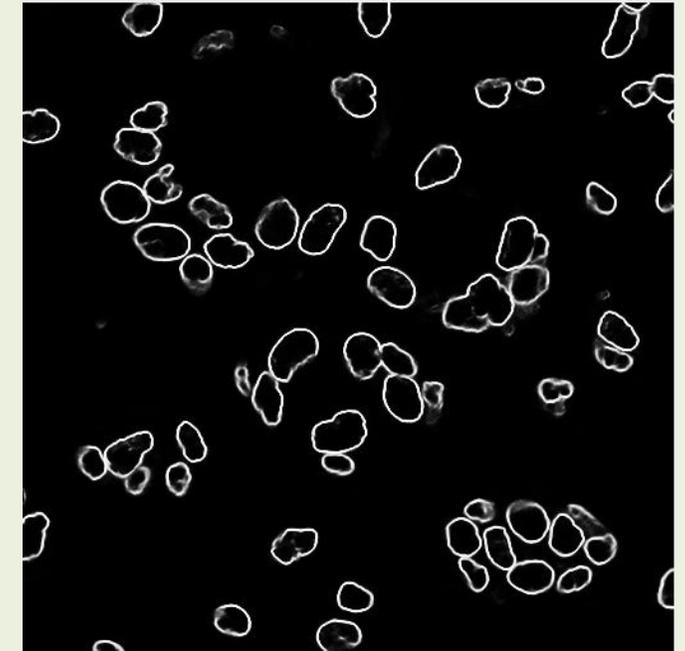


Image 2: Uncertainties

The brighter the pixel, the more uncertain the segmentation

c:/2 (2); 512x512 pixels; 32-bit; 2MB

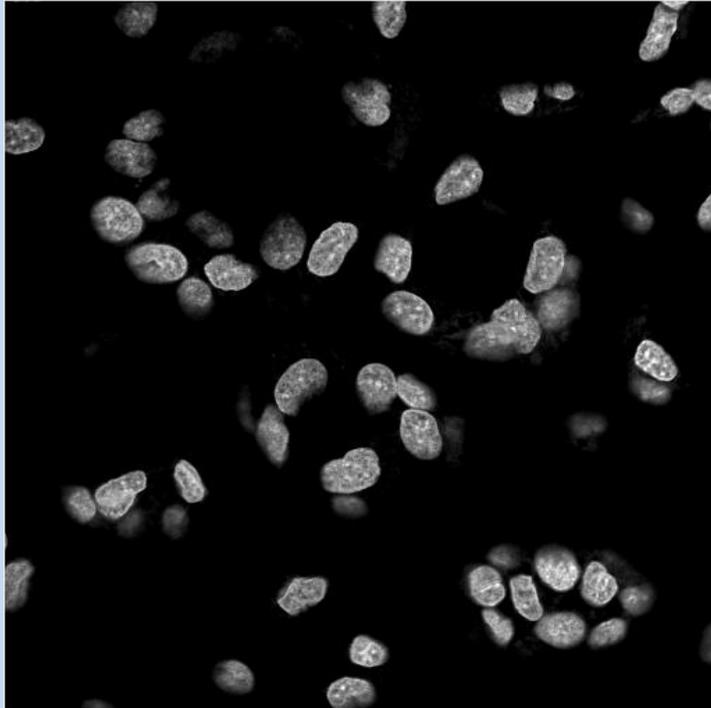


Deep Learning with StarDist

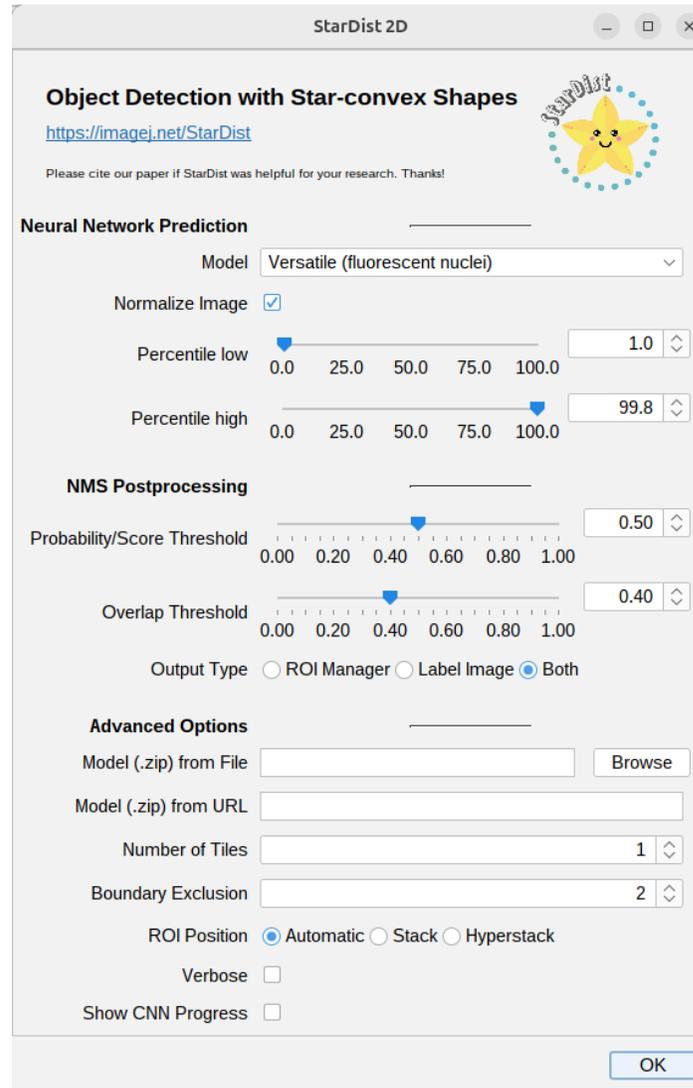
Specific applications: nuclei and cell bodies, 2D only but scriptable

Original dataset

184.52x184.52 micrometer (512x512); 8-bit; 256K

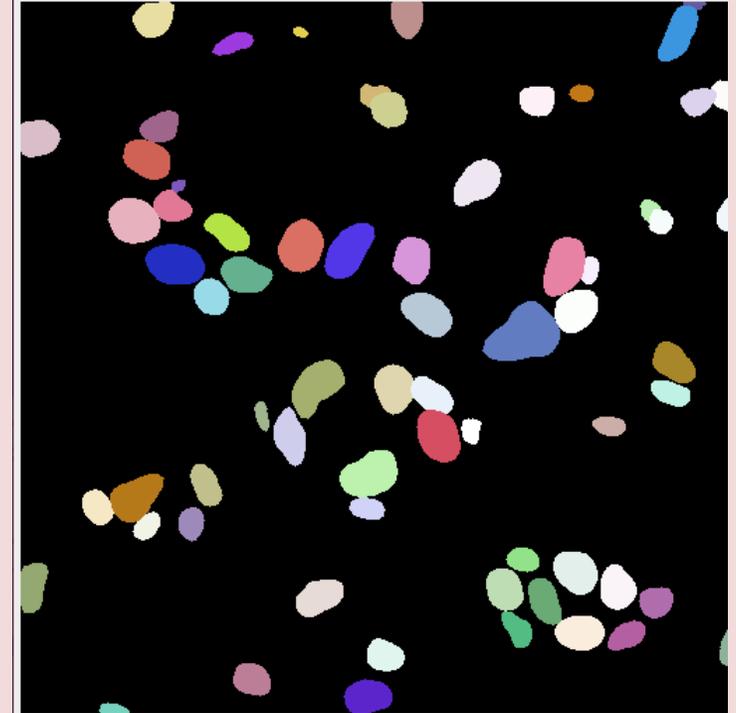


Plugins > StartDist 2D



Output:

"1"; 512x512 pixels; 16-bit; 512K



Note: the colours are a false-colour LUT. Change it to Grays if needed

Deep Learning with CSBDeep: : training your own model

How to train a model yourself?

> Install CSBDeep

Plugins > CSBDeep > DenoiSeg > Train

DenoSeg train

Folder containing training raw images Browse

Folder containing training labeling images Browse

Folder containing validation raw images Browse

Folder containing validation labeling images Browse

Number of epochs ▾

Number of steps per epoch ▾

Batch size ▾

Patch shape ▾
16 128 256 384 512

Neighborhood radius ▾

OK Cancel

Training data:

1. Raw datasets
2. Masked (manually) segmented datasets (e.g. 100 2D images at 512x512 px)

Training: use about 80% of your dataset, 20% for validation (e.g. 80 images for training)

Number of Epochs: the more the better

Steps per Epoch: the more the better

Batch/Patch size: do not change

Then: wait...

Deep Learning with CSBDeep: training your own model

Training DenoiSeg model: Plugins > CSBDeep > DenoiSeg > DenoiSeg Train

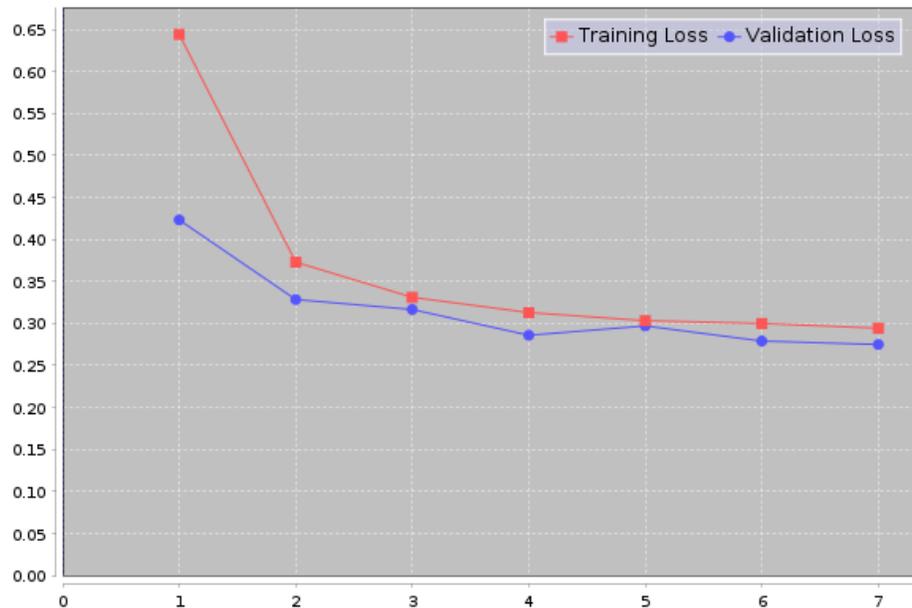
→ Data in folder: Example 7

□ Preparation

* Training

Epoch 8/50

Step 199/200

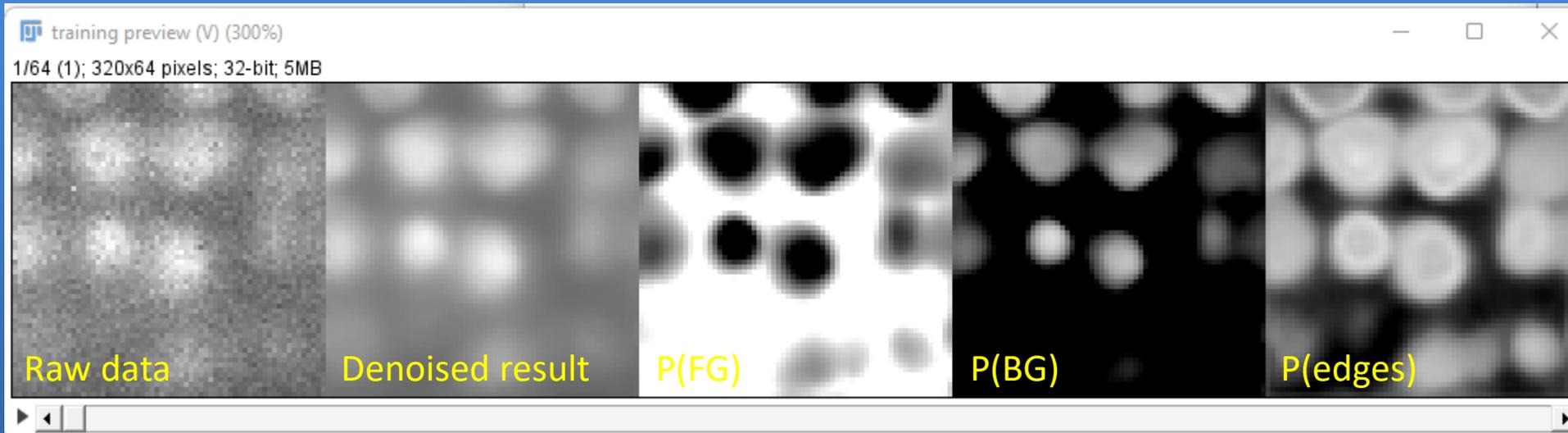


```
164 / 200 [*****--] - loss: 0.307055 seg loss: 0.517235 denoise loss: 0.096875 lr: 0.000400
165 / 200 [*****--] - loss: 0.248698 seg loss: 0.405297 denoise loss: 0.092099 lr: 0.000400
166 / 200 [*****--] - loss: 0.292199 seg loss: 0.486998 denoise loss: 0.097401 lr: 0.000400
167 / 200 [*****--] - loss: 0.284221 seg loss: 0.470408 denoise loss: 0.098034 lr: 0.000400
168 / 200 [*****--] - loss: 0.270557 seg loss: 0.444086 denoise loss: 0.097028 lr: 0.000400
169 / 200 [*****--] - loss: 0.333659 seg loss: 0.553356 denoise loss: 0.113962 lr: 0.000400
170 / 200 [*****--] - loss: 0.302505 seg loss: 0.506758 denoise loss: 0.098252 lr: 0.000400
171 / 200 [*****--] - loss: 0.297332 seg loss: 0.498418 denoise loss: 0.096245 lr: 0.000400
172 / 200 [*****--] - loss: 0.304597 seg loss: 0.499291 denoise loss: 0.109903 lr: 0.000400
173 / 200 [*****--] - loss: 0.293135 seg loss: 0.488004 denoise loss: 0.098265 lr: 0.000400
174 / 200 [*****--] - loss: 0.284084 seg loss: 0.467967 denoise loss: 0.100200 lr: 0.000400
175 / 200 [*****--] - loss: 0.276361 seg loss: 0.453689 denoise loss: 0.099033 lr: 0.000400
176 / 200 [*****--] - loss: 0.271480 seg loss: 0.440373 denoise loss: 0.102587 lr: 0.000400
177 / 200 [*****--] - loss: 0.272924 seg loss: 0.452161 denoise loss: 0.093687 lr: 0.000400
178 / 200 [*****--] - loss: 0.263199 seg loss: 0.422120 denoise loss: 0.104279 lr: 0.000400
179 / 200 [*****--] - loss: 0.271619 seg loss: 0.446509 denoise loss: 0.096729 lr: 0.000400
180 / 200 [*****--] - loss: 0.296185 seg loss: 0.492147 denoise loss: 0.100224 lr: 0.000400
181 / 200 [*****--] - loss: 0.292469 seg loss: 0.486170 denoise loss: 0.098768 lr: 0.000400
182 / 200 [*****--] - loss: 0.323374 seg loss: 0.540127 denoise loss: 0.106621 lr: 0.000400
183 / 200 [*****--] - loss: 0.301137 seg loss: 0.488686 denoise loss: 0.113588 lr: 0.000400
184 / 200 [*****--] - loss: 0.307349 seg loss: 0.477888 denoise loss: 0.136809 lr: 0.000400
185 / 200 [*****--] - loss: 0.308576 seg loss: 0.518423 denoise loss: 0.098728 lr: 0.000400
186 / 200 [*****--] - loss: 0.302579 seg loss: 0.499741 denoise loss: 0.105417 lr: 0.000400
187 / 200 [*****--] - loss: 0.305442 seg loss: 0.512856 denoise loss: 0.098029 lr: 0.000400
188 / 200 [*****--] - loss: 0.270236 seg loss: 0.432321 denoise loss: 0.108152 lr: 0.000400
189 / 200 [*****--] - loss: 0.311597 seg loss: 0.520181 denoise loss: 0.103013 lr: 0.000400
190 / 200 [*****--] - loss: 0.324790 seg loss: 0.527424 denoise loss: 0.122156 lr: 0.000400
191 / 200 [*****--] - loss: 0.287673 seg loss: 0.472959 denoise loss: 0.102386 lr: 0.000400
192 / 200 [*****--] - loss: 0.291883 seg loss: 0.487375 denoise loss: 0.096391 lr: 0.000400
193 / 200 [*****--] - loss: 0.288857 seg loss: 0.479057 denoise loss: 0.098658 lr: 0.000400
```

Deep Learning with CSBDeep

Training DenoiSeg model: Plugins > CSBDeep > DenoiSeg > DenoiSeg Train
> Data in folder: Example 7

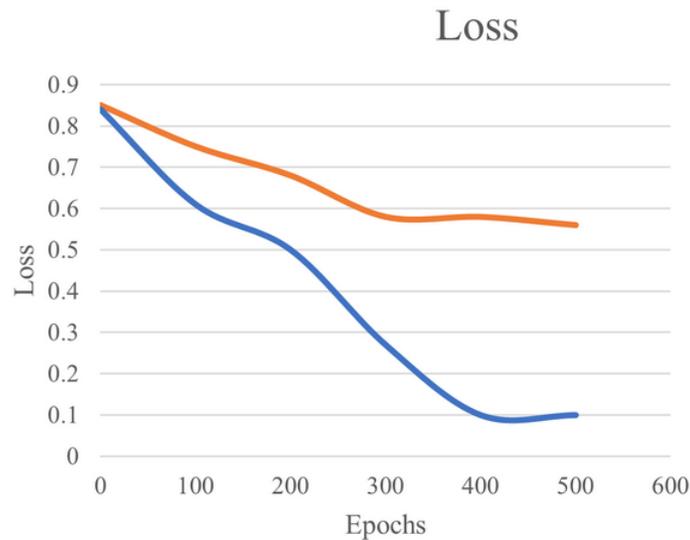
Preview window



Deep Learning with CSBDeep

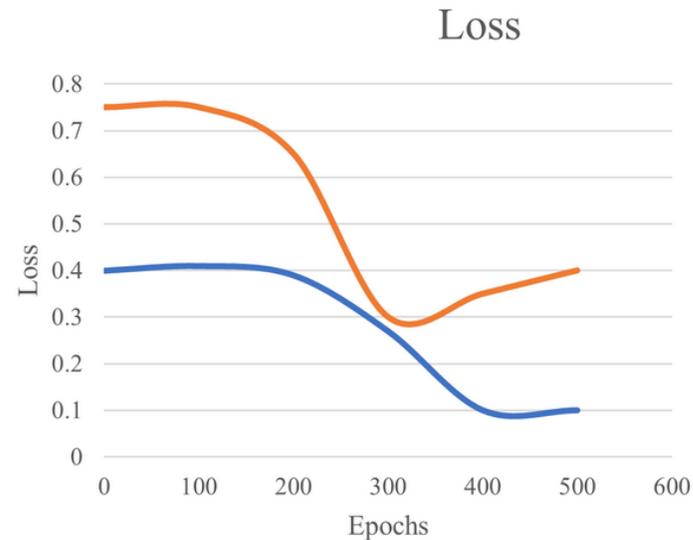
Underfitting

the model is unable to accurately model the training data, and hence generates large errors

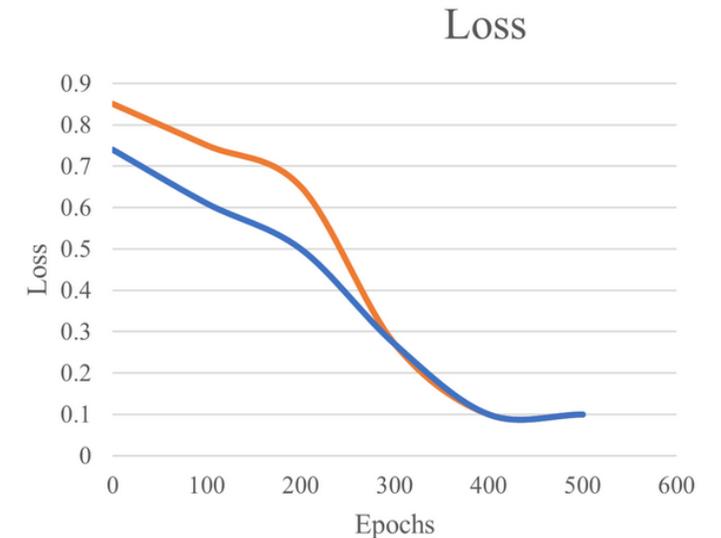


Overfitting

the model performs well on training data but poorly on the new data in the validation set.



Good fit



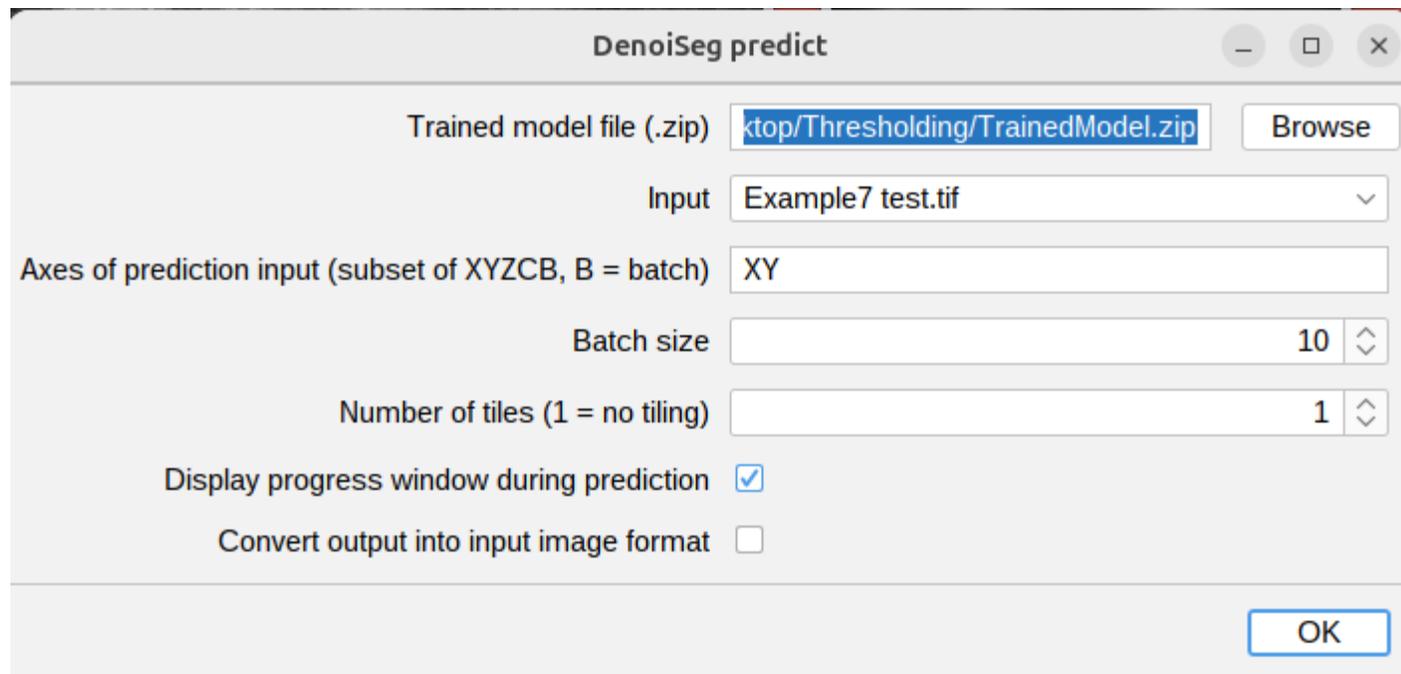
— Validation Loss
— Training Loss

Deep Learning with CSBDeep

EXERCISE

Use the trained model on data from Example 7

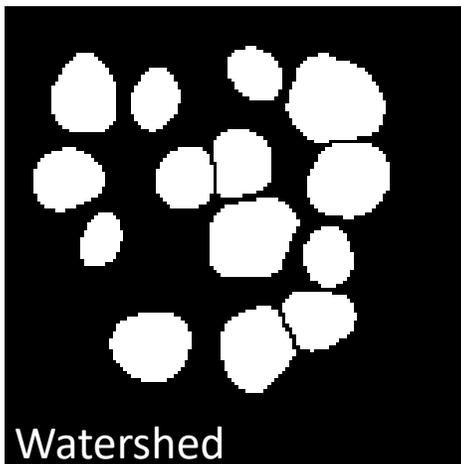
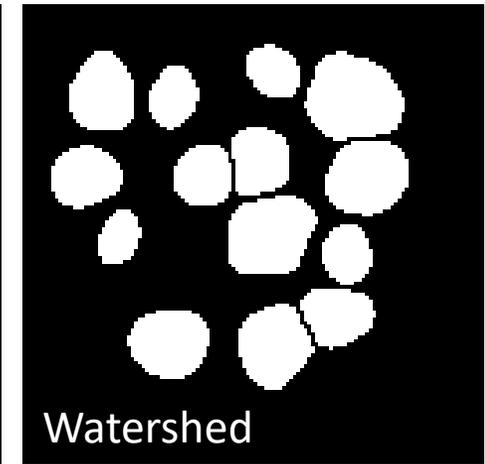
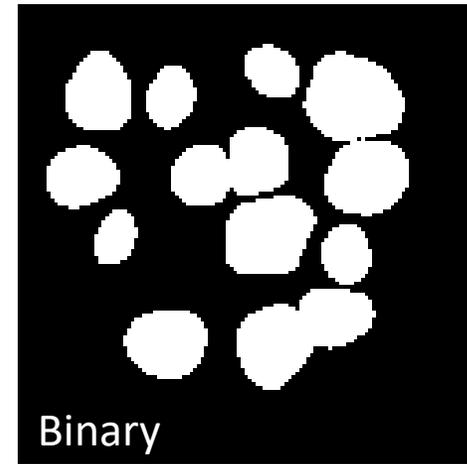
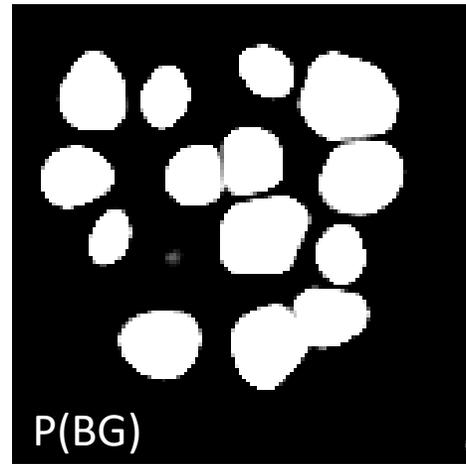
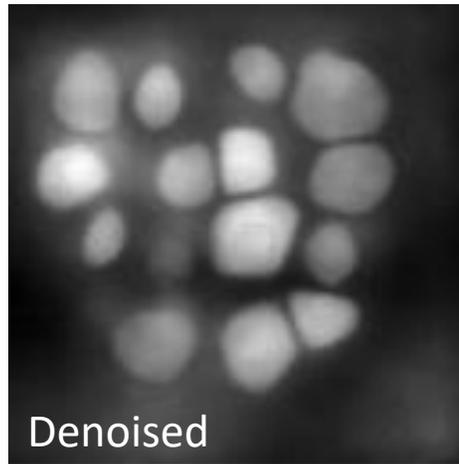
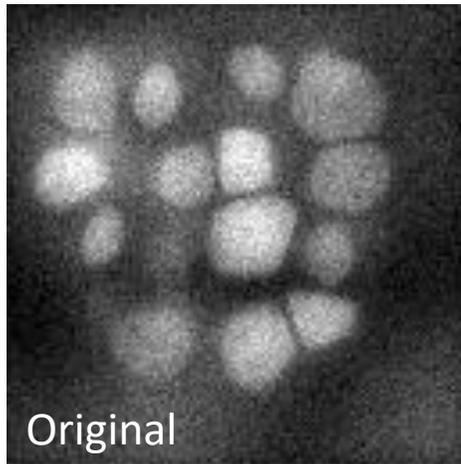
- Open a dataset from the trained images (e.g. Images > test > Example7 test.tif)
- Duplicate 1 image
- Plugins > CSBDeep > DenoiSeg > DenoiSeg predict



Deep Learning with CSBDeep

EXERCISE

Use the trained model on data from Example 7



Number extraction



Blob analysis aka particle counting

Before you start:

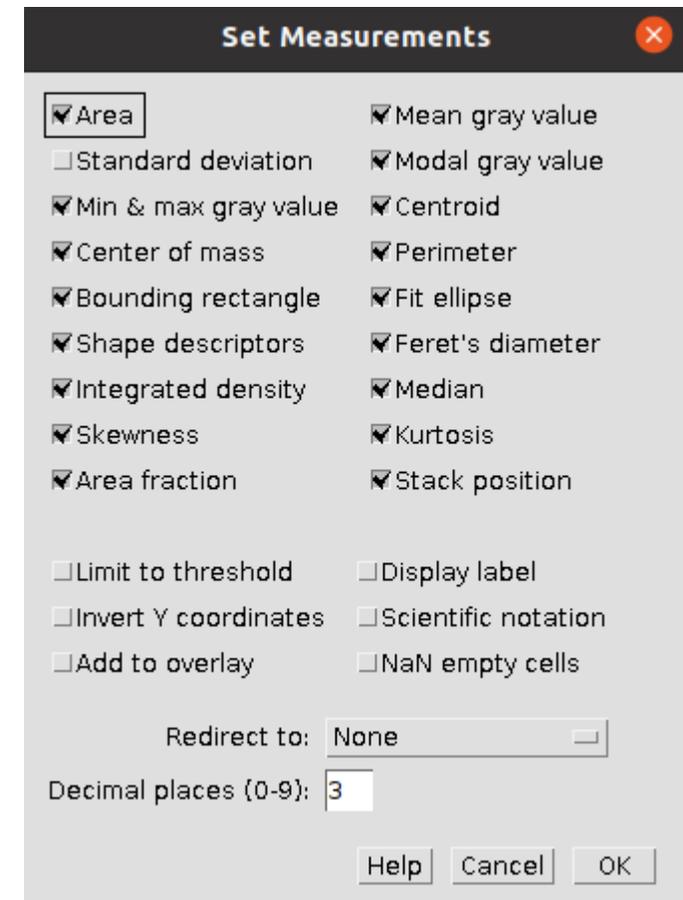
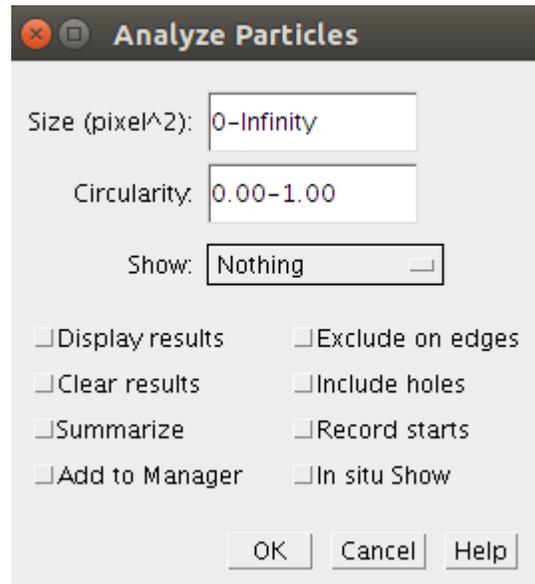
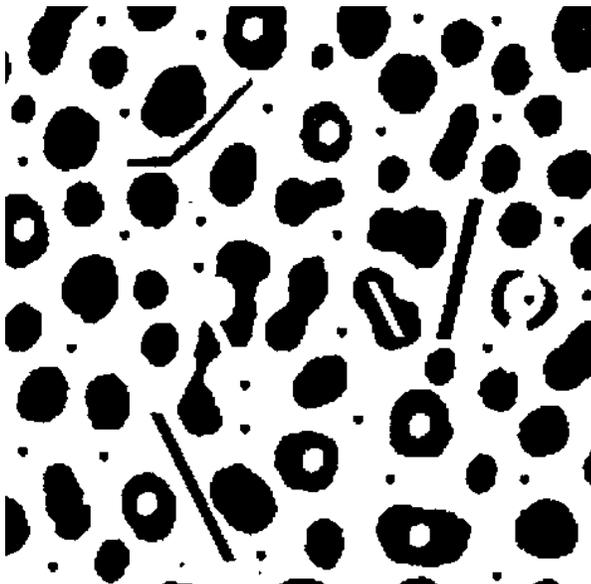
- Can you trust your binary image?
- Is the scale properly set? (Analyze > set scale)
- Is the foreground particle white (if not: invert: ctrl+i)
- What do you want to measure (Analyze > Set Measurements)

Assumption

Your data is binary (or at least segmented)

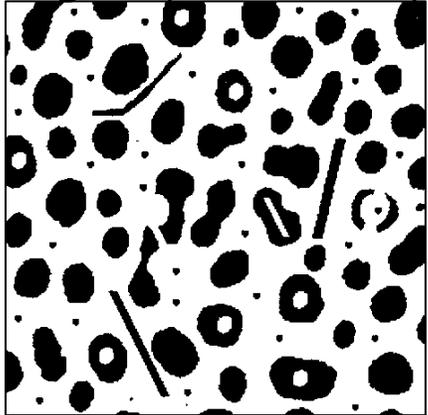
Two step procedure:

1. Binarization (=threshold)
2. Measurement: Analyze > Measure particles

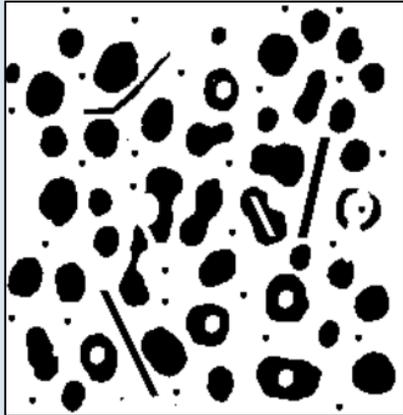


Size measurements: filters

Original (thresholded)

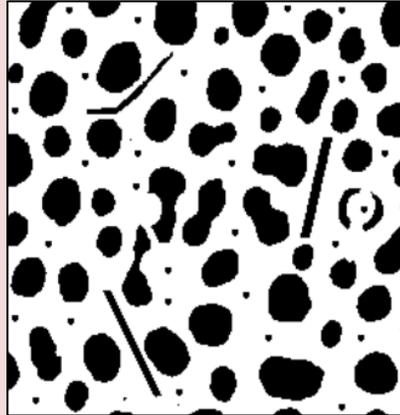


Edge filter



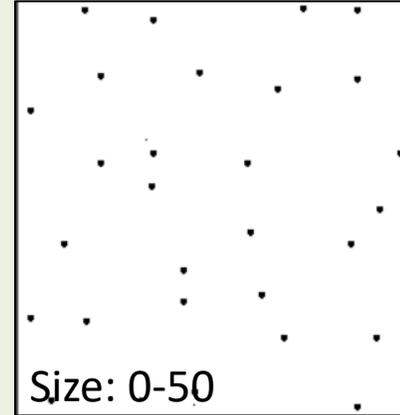
particles touching the edge will be ignored

Include holes

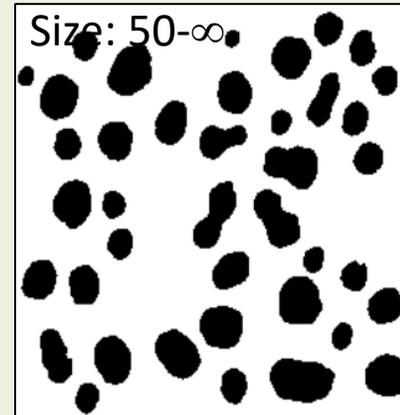


Interior holes will be included

Size filter



Size: 0-50



Size: 50-∞

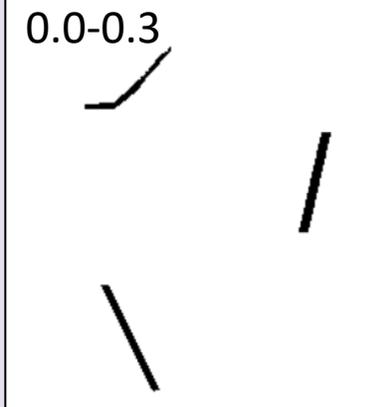
Particles with size (=area) outside the range specified in this field are ignored.

Circularity filter

$$\text{Circ.} = 4\pi \times \frac{\text{Area}}{\text{Perimeter}^2}$$



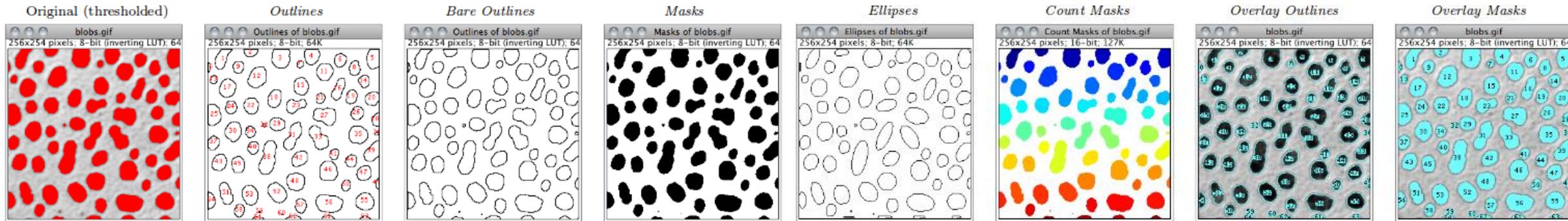
0.3-0.5



0.0-0.3

Ranges from 0 (infinitely elongated polygon) to 1 (perfect circle).

Size measurements: Outlines, masks and overlays



Nothing: Neither Outlines, masks nor Overlays will be displayed.

Outlines: 8-bit image containing numbered outlines of the measured particles.

Bare Outlines: 8-bit image containing simple outlines of the measured particles without labels.

Masks: 8-bit binary image containing filled outlines of the measured particles

Ellipses: 8-bit binary image containing the best fit ellipse (cf. Edit>Selection>Fit Ellipse)

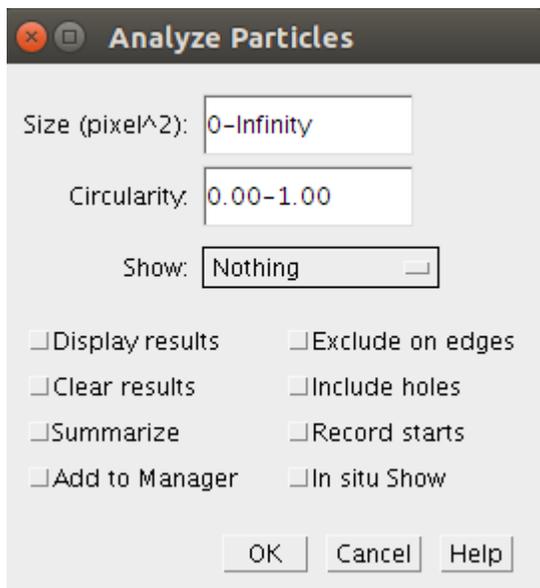
Count Masks: 16-bit image containing filled outlines of the measured particles painted with a grayscale value corresponding to the particle number.

Overlay Outlines: Displays numbered outlines of the measured particles in the image overlay.

Overlay Masks: Displays numbered and filled outlines of the measured particles in the image overlay.

If *In situ Show* is checked, the original image will be replaced by this image.

Size measurements: Results



Display results

The measurements for each particle will be displayed in the Results Table.

Clear Results

If checked, any previous measurements listed in the Results Table will be cleared

Summarize

If checked, the particle count, total particle area, average particle size, area fraction and the mean of all parameters listed in the Set Measurements. . . dialog box will be displayed in a separate Summary table (useful for “stacks”).

Note that while single images ‘Summaries’ are output to the same Summary table, stack Summaries are printed in dedicated tables (named Summary of [stack title]). Also, note that descriptive statistics on Results measurements can be obtained at any time using the Summarize command.

Add to Manager

If checked, the measured particles masks will be added to the ROI Manager. . .

File	Edit	Font	Results								
Label	Area	Mean	StdDev	Mode	Min	Max	XM	YM	Perim.	BX	
47	Example 7 - blobs.tif-1	0.005	255	0	255	255	255	0.809	0.641	0.250	0.769
48	Example 7 - blobs.tif-1	0.007	255	0	255	255	255	0.453	0.685	0.308	0.405
49	Example 7 - blobs.tif-1	8.280E-4	255	0	255	255	255	0.879	0.696	0.135	0.866
50	Example 7 - blobs.tif-1	0.002	255	0	255	255	255	0.715	0.705	0.165	0.693
51	Example 7 - blobs.tif-1	0.006	255	0	255	255	255	0.092	0.748	0.309	0.059
52	Example 7 - blobs.tif-1	0.007	255	0	255	255	255	0.358	0.742	0.326	0.308
53	Example 7 - blobs.tif-1	0.007	255	0	255	255	255	0.217	0.755	0.300	0.177
54	Example 7 - blobs.tif-1	0.002	255	0	255	255	255	0.015	0.784	0.218	0.000
55	Example 7 - blobs.tif-1	0.007	255	0	255	255	255	0.813	0.789	0.302	0.766
56	Example 7 - blobs.tif-1	0.010	255	0	255	255	255	0.626	0.798	0.394	0.558
57	Example 7 - blobs.tif-1	0.003	255	0	255	255	255	0.480	0.810	0.217	0.450
58	Example 7 - blobs.tif-1	0.003	255	0	255	255	255	0.161	0.835	0.189	0.135
59	Example 7 - blobs.tif-1	3.600E-5	255	0	255	255	255	0.266	0.841	0.020	0.263
60	Example 7 - blobs.tif-1	1.200E-5	255	0	255	255	255	0.383	0.854	0.010	0.381
61	Example 7 - blobs.tif-1	9.721E-4	255	0	255	255	255	0.621	0.872	0.147	0.589
62	Example 7 - blobs.tif-1	0.001	255	0	255	255	255	0.444	0.872	0.172	0.402
63	Example 7 - blobs.tif-1	6.360E-4	255	0	255	255	255	0.814	0.873	0.123	0.786

Results table

File > save as...

Saves the table as comma separated values (CSV)
Which can be imported in Excel, R, Stata, ...

	K	L	M	N	O	P	Mii
Perim.	BX	BY	Width	Height	Major	Minor	
0	0.322	0.035	0	0.09	0.104	0.117	0.
7	0.191	0.184	0	0.073	0.038	0.071	(
5	0.337	0.329	0	0.094	0.097	0.104	0.
5	0.272	0.499	0	0.08	0.08	0.084	0.
9	0.296	0.821	0	0.066	0.1	0.107	0.
5	0.215	0.655	0.021	0.062	0.073	0.071	0.
7	0.112	0.461	0.059	0.031	0.038	0.039	0.
3	0.222	0.731	0.059	0.059	0.076	0.077	0.
2	0.193	0.128	0.062	0.055	0.062	0.063	0.
3	0.106	0	0.069	0.01	0.048	0.045	(
7	0.286	0.561	0.073	0.087	0.09	0.091	0.
5	0.341	0.204	0.09	0.097	0.107	0.116	0.
7	0.124	0.01	0.135	0.035	0.042	0.044	0.
5	0.193	0.779	0.135	0.055	0.062	0.063	0.
L	0.272	0.443	0.145	0.076	0.09	0.091	0.
7	0.303	0.637	0.149	0.073	0.114	0.124	(
L	0.293	0.059	0.152	0.083	0.097	0.097	0.
5	0.27	0.308	0.208	0.069	0.094	0.097	0.
5	0.215	0.714	0.208	0.059	0.076	0.075	0.
5	0.238	0.814	0.218	0.073	0.073	0.079	0.
4	0.16	0.561	0.225	0.045	0.055	0.055	0.
2	0.262	0.184	0.252	0.076	0.082	0.082	0.

Size measurements: Results

EXERCISE

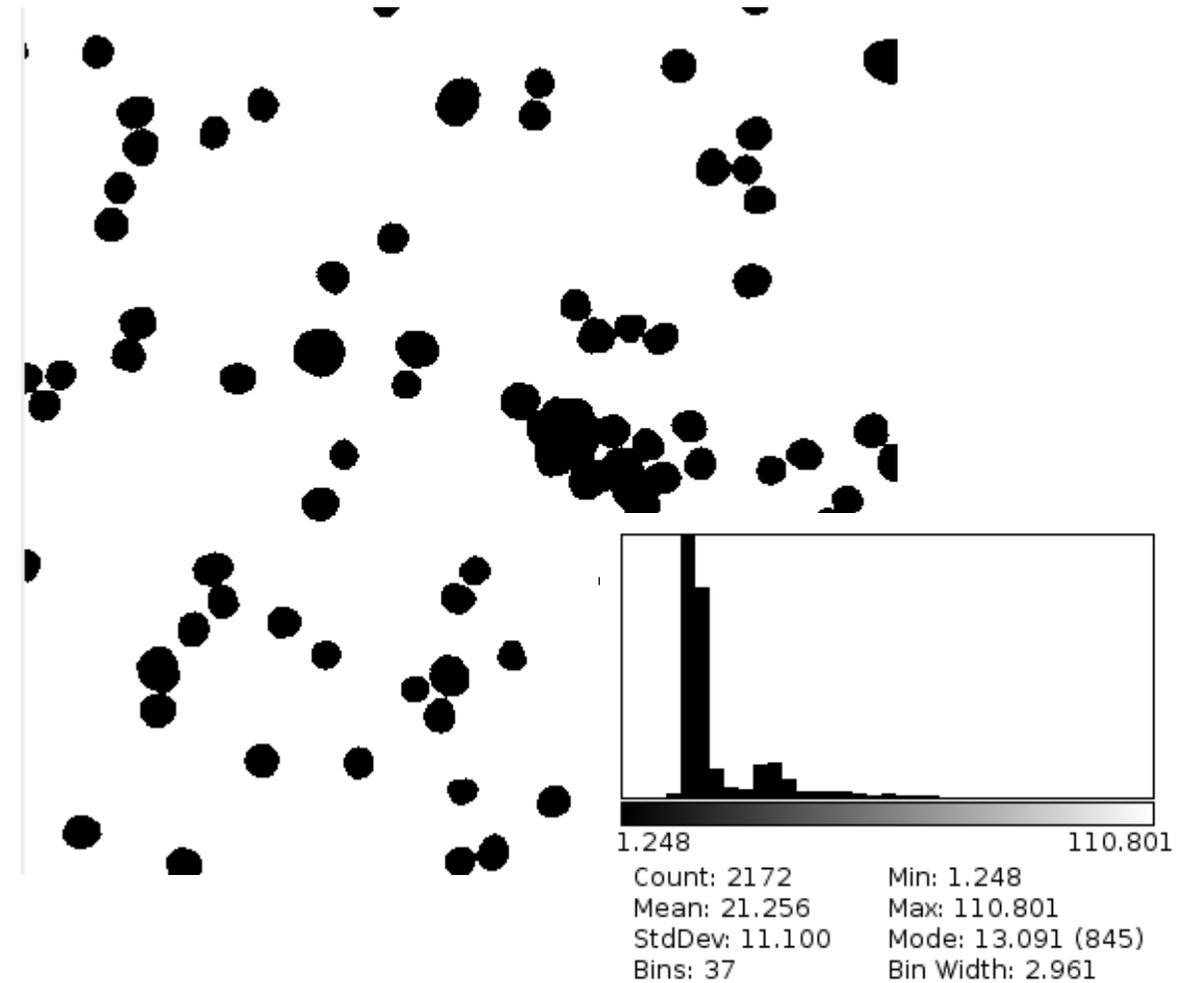
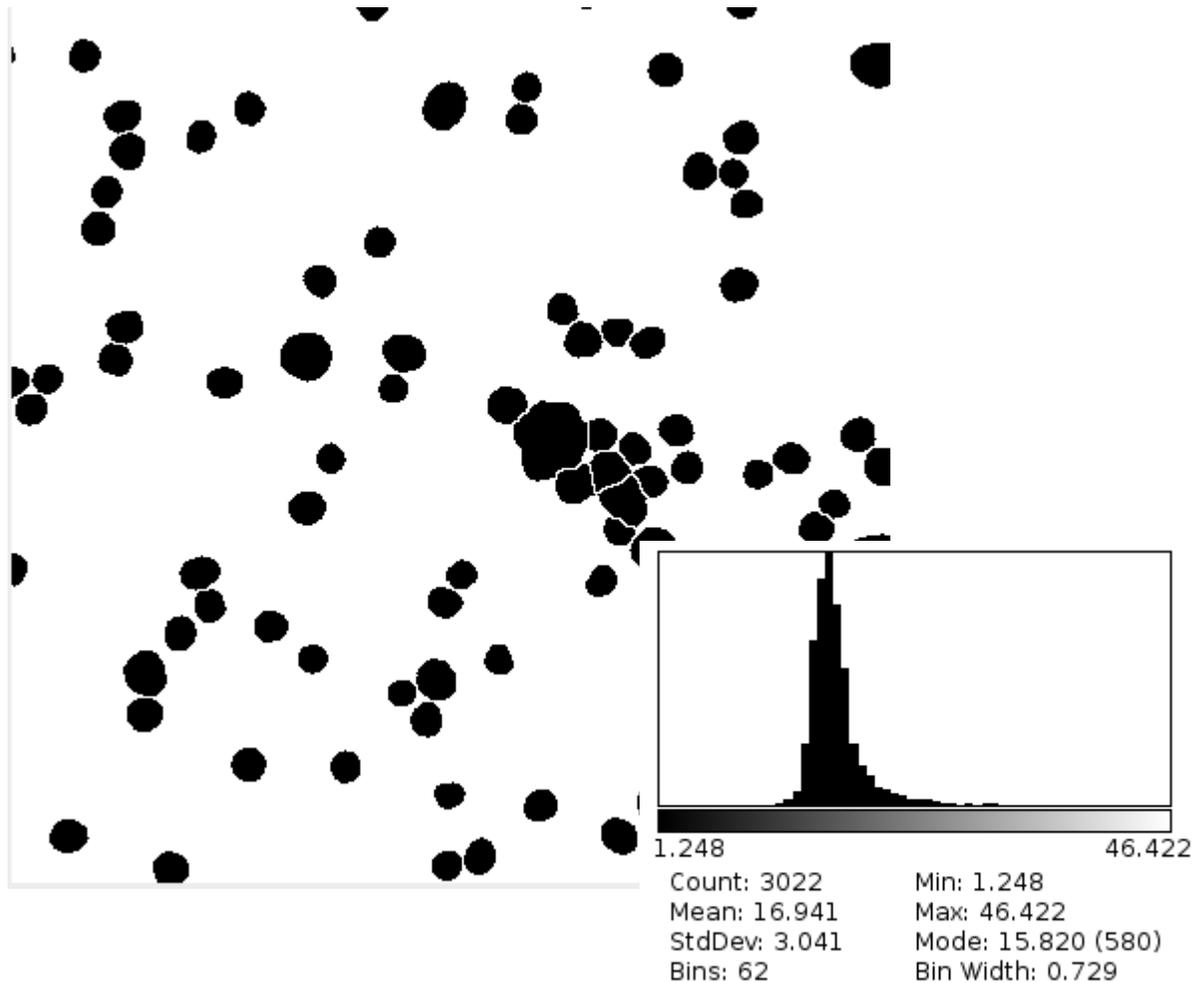
Calculate the mean radius of the AuNP in Example 2 – AuNP. Try with and without performing a watershed before. Show a distribution of the feret.

1. Image > adjust > threshold (use Default)
2. Analyze > Measure particles
3. Analyze > Distribution

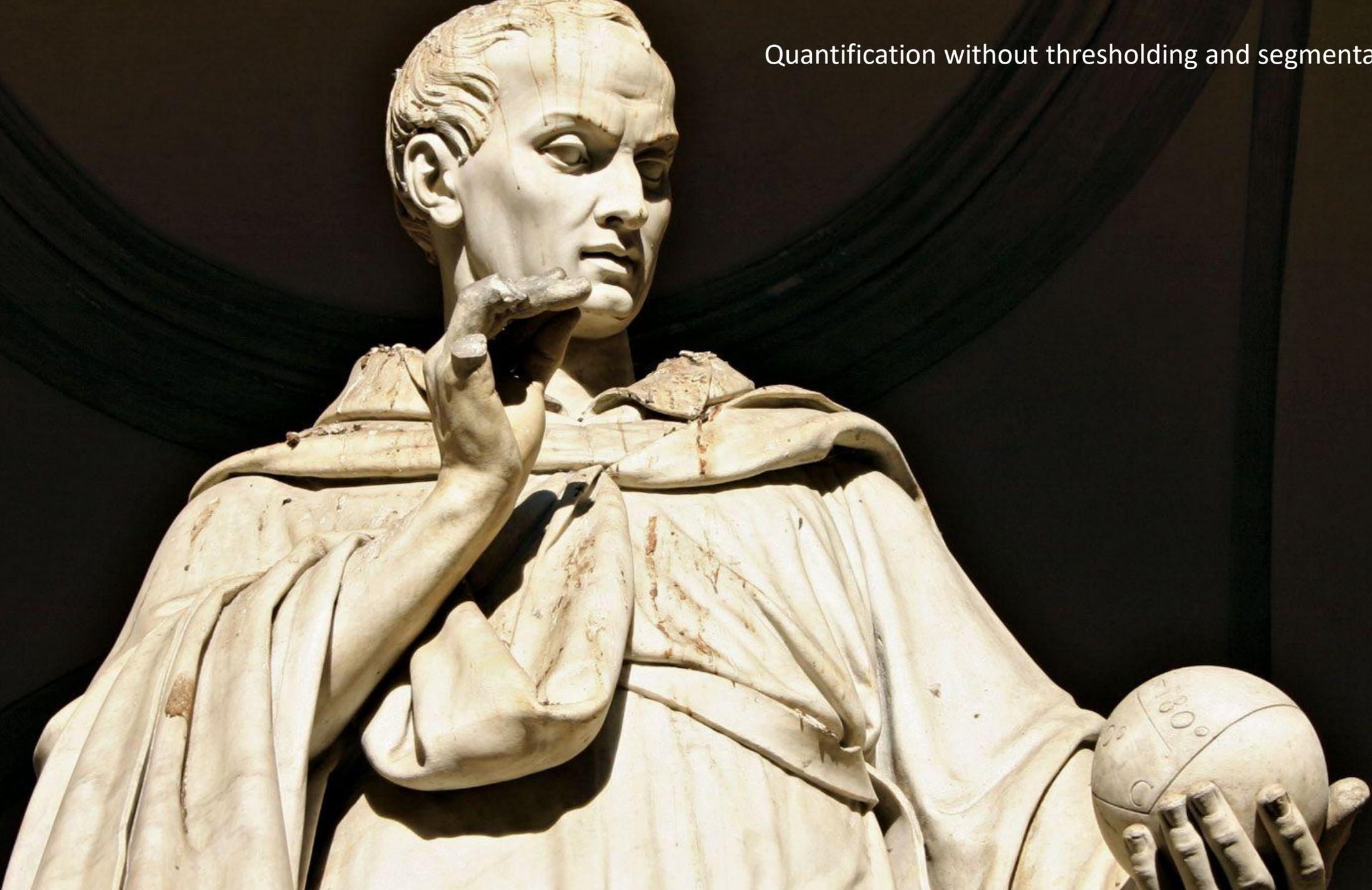
Size measurements: Results

EXERCISE

Calculate the mean radius of the AuNP in Example 2 – AuNP. Try with and without performing a watershed before. Show a distribution of the feret.



Quantification without thresholding and segmentation



Volume estimation with Cavalieri

Pro

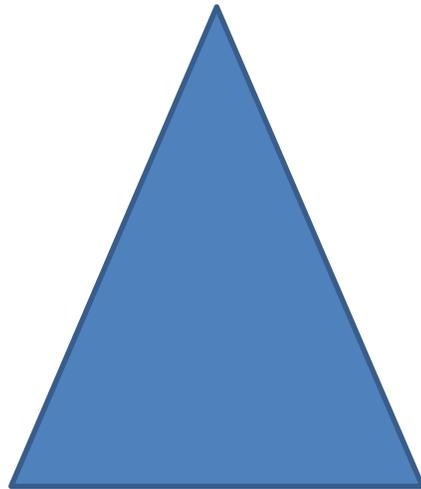
Independent of object
Optical disector
Low coefficient of error

Contra

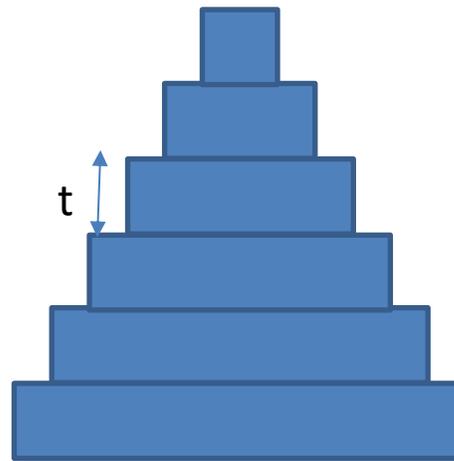
thickness must be known
Over/underprojection



3D Object

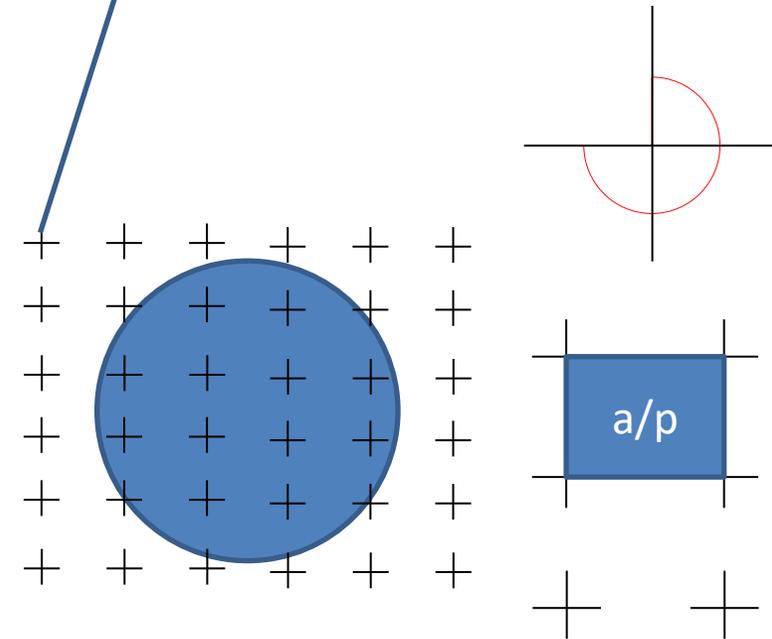


XZ side view



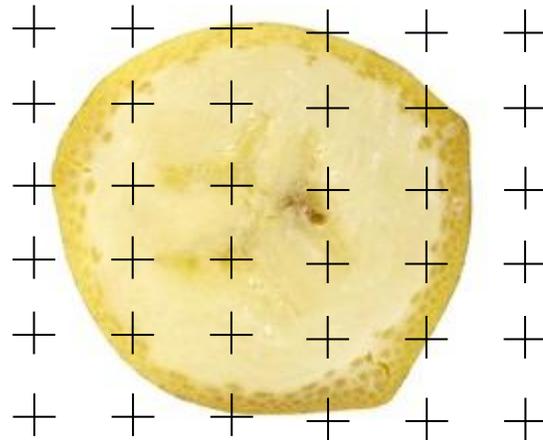
XZ view Cavalieri

Systematically uniform grid,
randomly dropped



XY view Cavalieri

Volume estimation with Cavalieri



$a/p = 1 \text{ cm}^2$

Thickness: 2 cm

Repeat i times (with i = number of banana pieces)

Volume estimation with Cavalieri



$$V = A_p \cdot t \cdot \sum P_i$$



$$V = 1 \text{ (cm}^2\text{)} \cdot 2 \text{ (cm)} \cdot 116$$
$$V = 232 \text{ cm}^3$$

$$V \text{ by submersion: } 230 \text{ ml}$$
$$V = 230 \text{ cm}^3$$

$$CE_{noise} = \frac{\sqrt{Noise}}{\sum P} = 2.9 \%$$

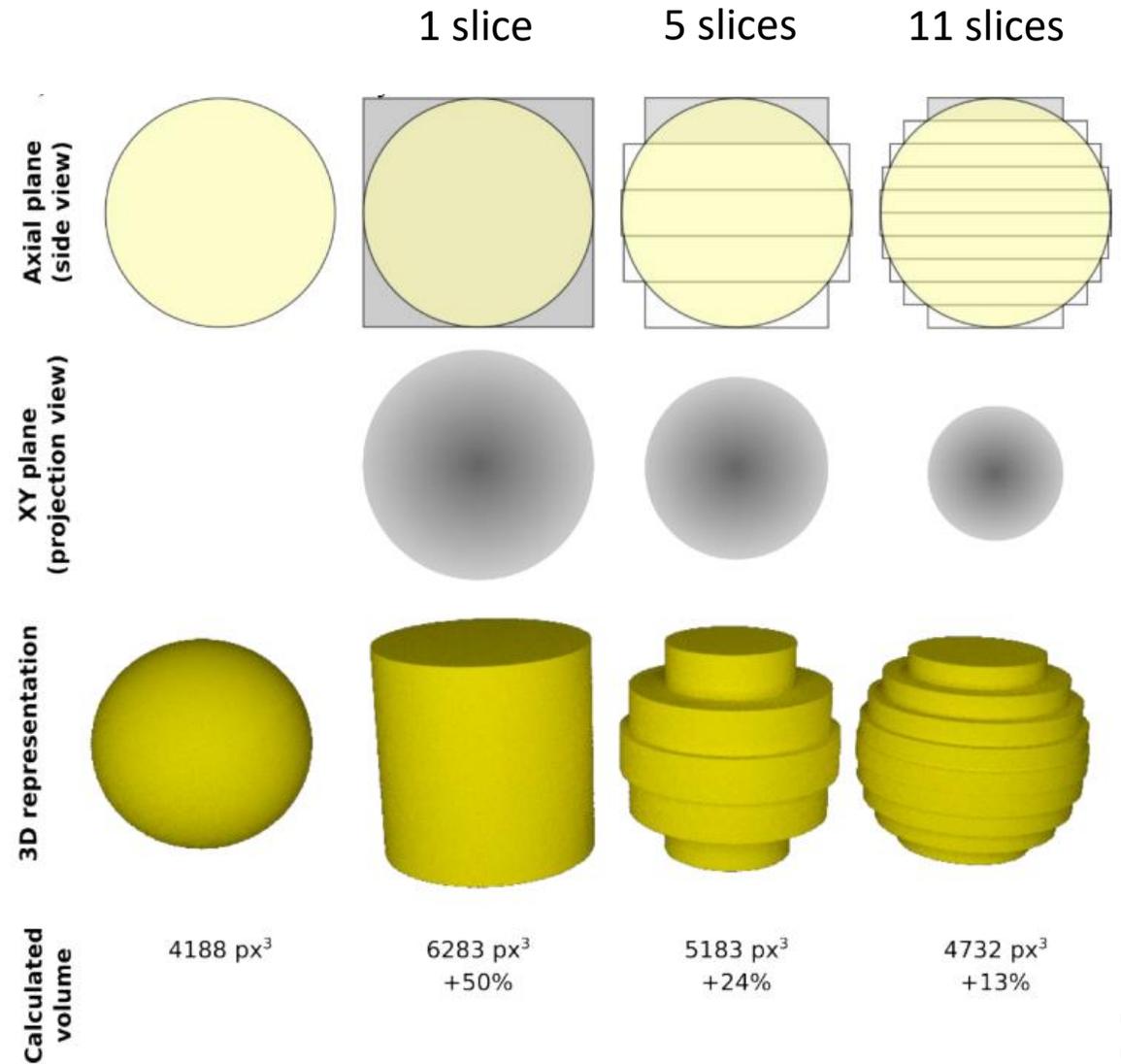
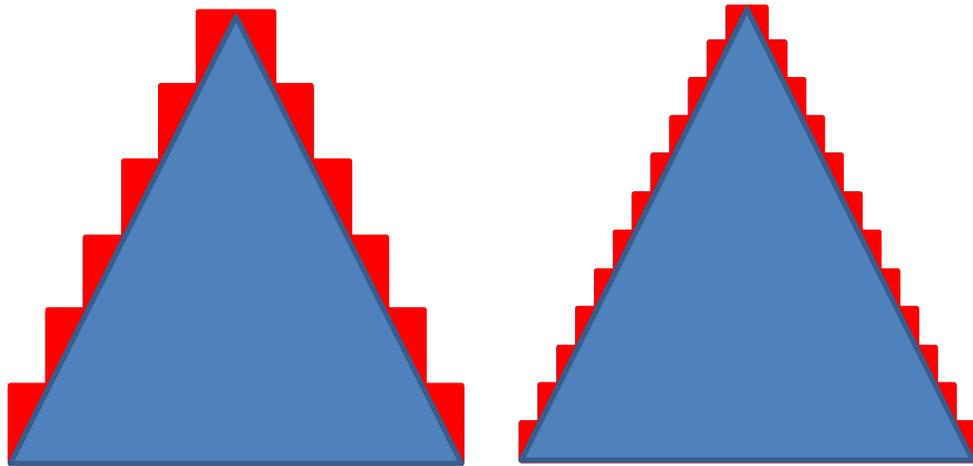
$$CE_{SURS} = \frac{\sqrt{Sampling}}{\sum P} = 0.8 \%$$

$$CE_{total} = \frac{\sqrt{Noise + Sampling}}{\sum P} = 3.7 \%$$

Volume estimation with Cavalieri: Holmes effect

Contra

thickness must be known → optical sections!
 Over/underprojection → Holmes effect



Size measurements: Results

EXERCISE

Open Example 8. Reduce the number of slices by factor 10. Then throw a random grid over the stack and do a Cavalieri estimation of the volume

1. Open Example 8A
2. Reduce the Z stack by factor 5
3. For fun (and to make it no longer a binary image): add noise (e.g. with an SD of 50)
3. Throw a random grid over the Image, A/p of roughly 150 pixel²
4. Count the number of crosses that fall onto the object, on all slices

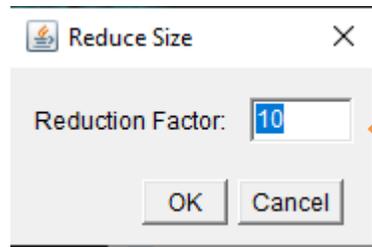
Size measurements: Results

EXERCISE

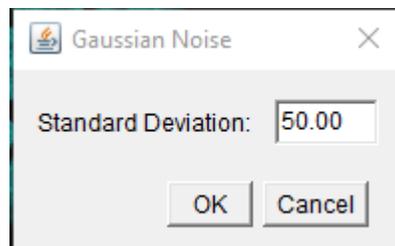
Open Example 8. Reduce the number of slices by factor 10. Then throw a random grid over the stack and do a Cavalieri estimation of the volume

1. File > open

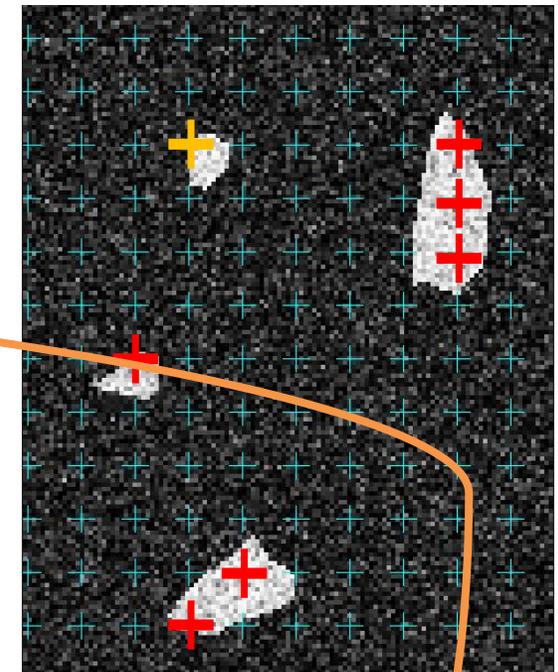
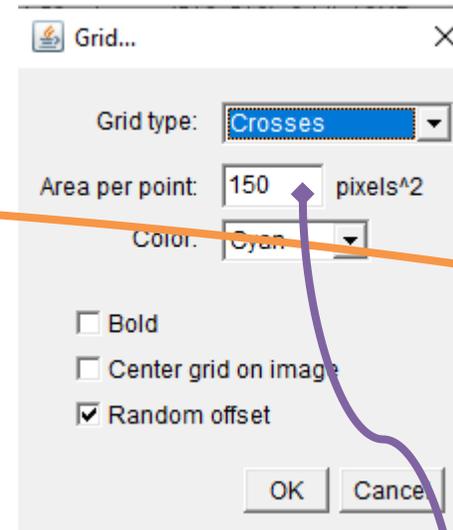
2. Image > stacks > Tools > reduce...



3. Process > Noise > Add specified noise... (yes, all slices)



4. Analyze > tools > Grid...



5. Count on all slices (in the example: 6 (7?))

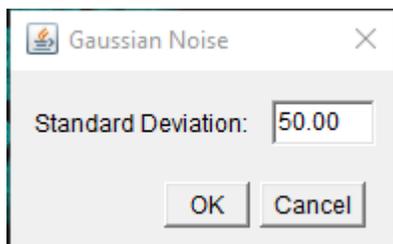
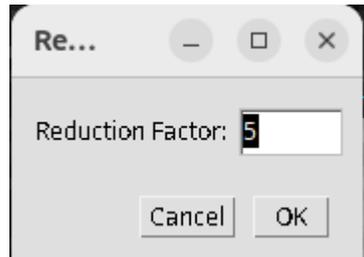
6. Volume = **count** x **Area per Point** x **Reduction factor**

Size measurements: Results

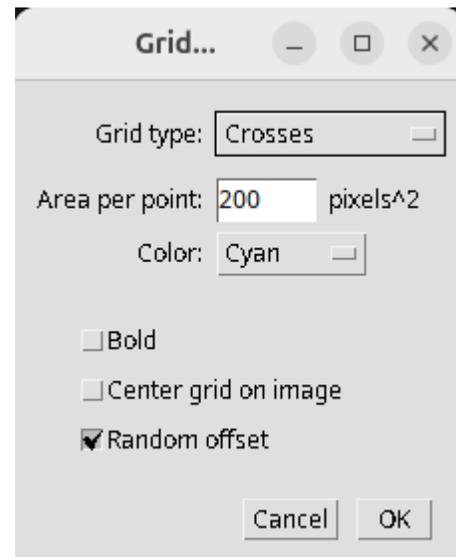
EXERCISE

Open Example 8. Reduce the number of slices by factor 10. Then throw a random grid over the stack and do a Cavalieri estimation of the volume

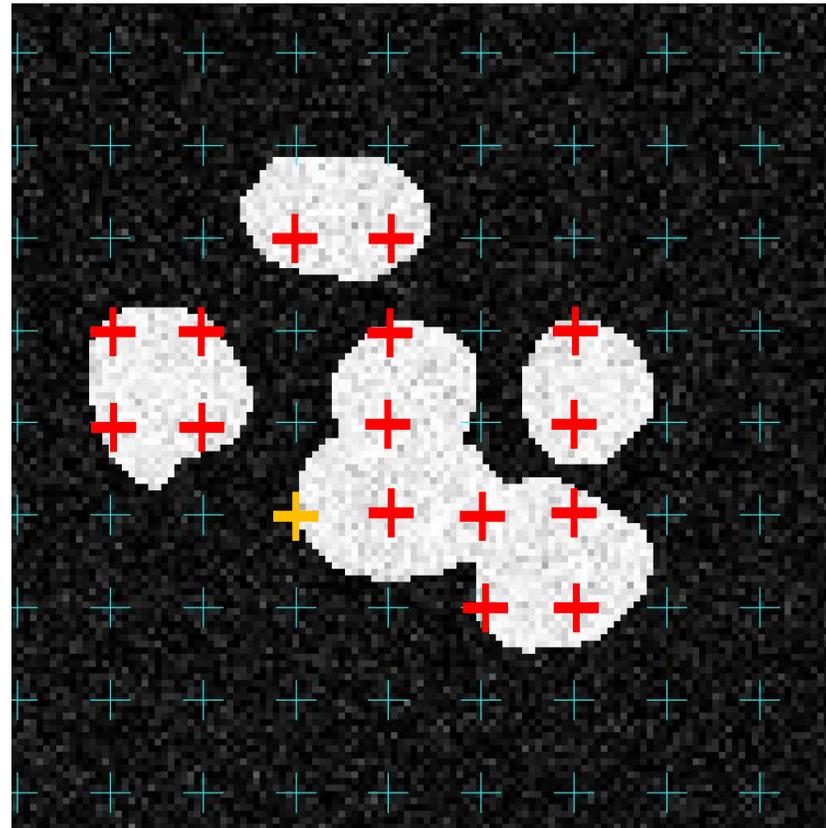
1. File > open
2. Image > stacks > Tools > reduce...
3. Process > Noise > Add specified noise... (yes, all slices)



4. Analyze > tools > Grid...

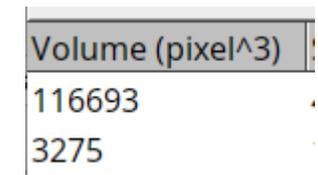


5. Count on all slices (I got 122 counts)
6. Area per point: 200 pixels²
6. Volume = **count** x **Area per Point** x **Reduction factor**



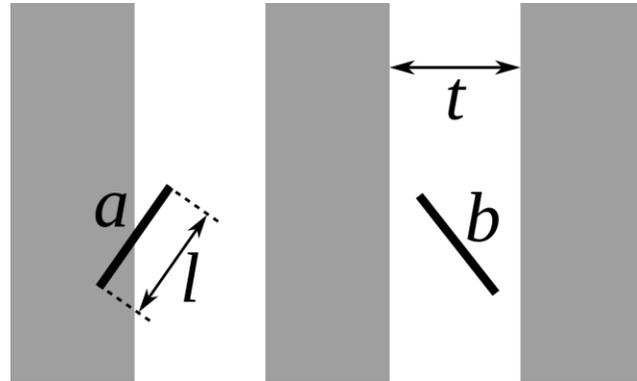
Cavalieri: 122 000 px³

ImageJ volume counter:

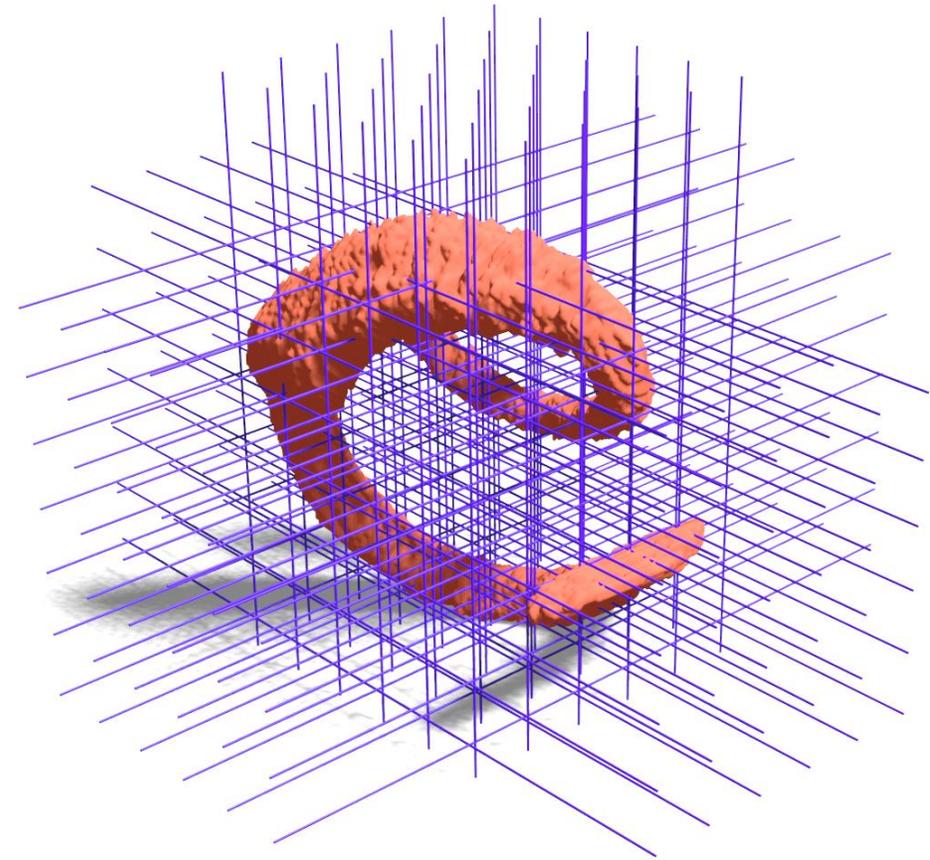


(119 968 px³)

Surface estimation with Buffon's needle



$$p = \frac{2}{\pi} \cdot \frac{l}{t}$$



Ants estimate area using Buffon's needle

Eamonn B. Mallon* and Nigel R. Franks

Centre for Mathematical Biology, and Department of Biology and Biochemistry, University of Bath, Bath BA2 7AY, UK

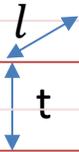
We show for the first time, to our knowledge, that ants can measure the size of potential nest sites. Nest size assessment is by individual scouts. Such scouts always make more than one visit to a potential nest before initiating an emigration of their nest mates and they deploy individual-specific trails within the potential new nest on their first visit. We test three alternative hypotheses for the way in which scouts might measure nests. Experiments indicated that individual scouts use the intersection frequency between their own paths to assess nest areas. These results are consistent with ants using a 'Buffon's needle algorithm' to assess nest areas.

Keywords: ants; colony emigration; individual-specific pheromones; *Leptothorax*; nest sites; rules of thumb

Surface estimation with Buffon's needle

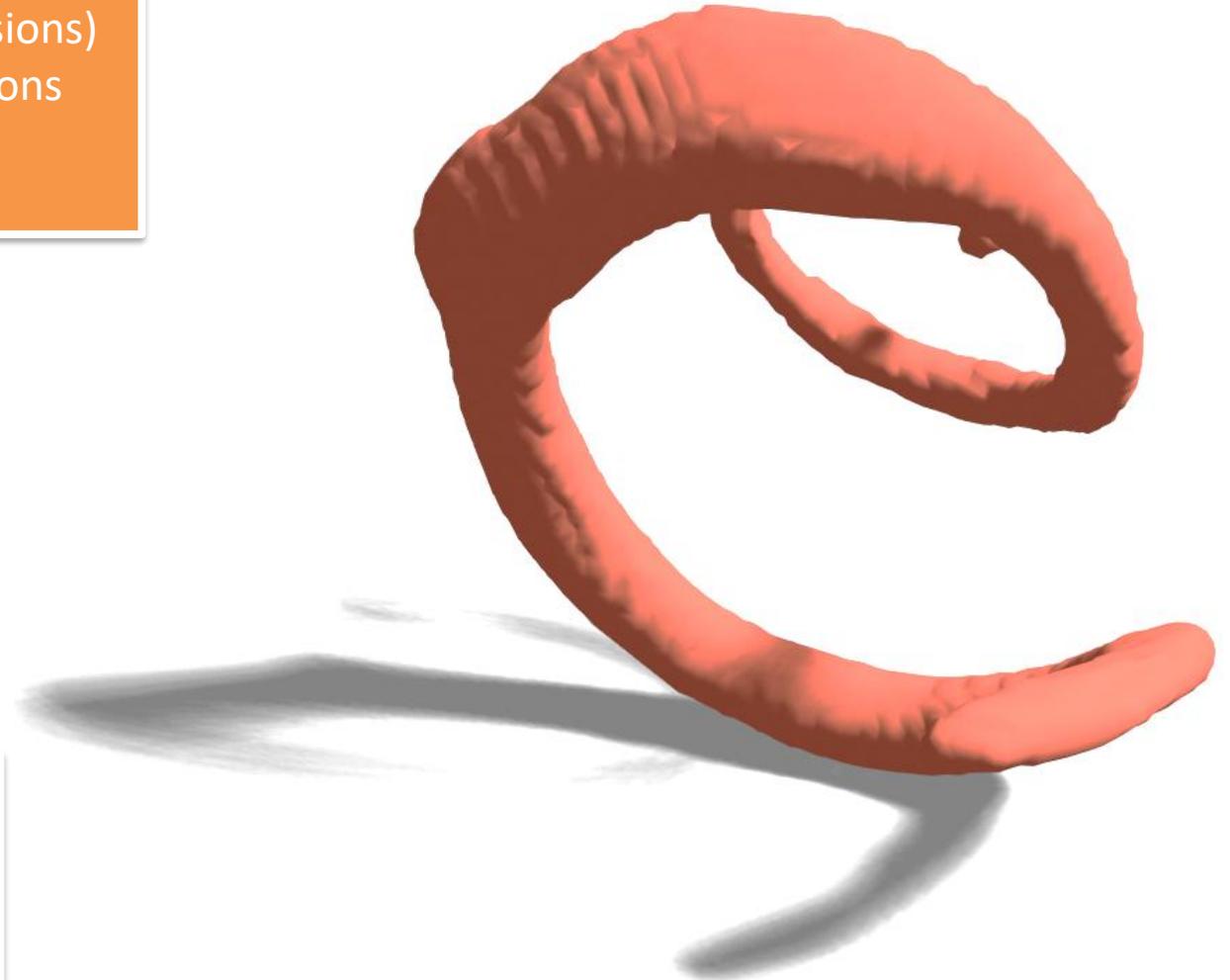
$$S = 2 \cdot \frac{1}{n} \cdot \sum_{i=1}^n \frac{v}{l_i} \cdot l$$

$n = 3$ (number of dimensions)
 l = number of intersections
 $\frac{v}{l_i}$ = Area per volume



t = distance between two slices
 l = distance between two lines

$$\frac{v}{l_i} = t \cdot l$$



Surface estimation with Buffon's needle

XY	XZ	YZ
142	164	168

$$S = 2 \cdot \frac{1}{n} \cdot \sum_{i=1}^n \frac{v}{l_i} \cdot l_i$$

$n = 1$ (number of dimensions)

$$\sum l_i = 474$$

$$\frac{v}{l_i} = 70.7$$

→ $\sqrt{200} = 14$ (distance between two grid lines)

→ 5 (reducing factor, distance between adjacent slices)

$$S = \frac{2}{3} \cdot 474 \cdot 14 \cdot 14 \cdot 5 = 22\,340 \text{ px}^2$$

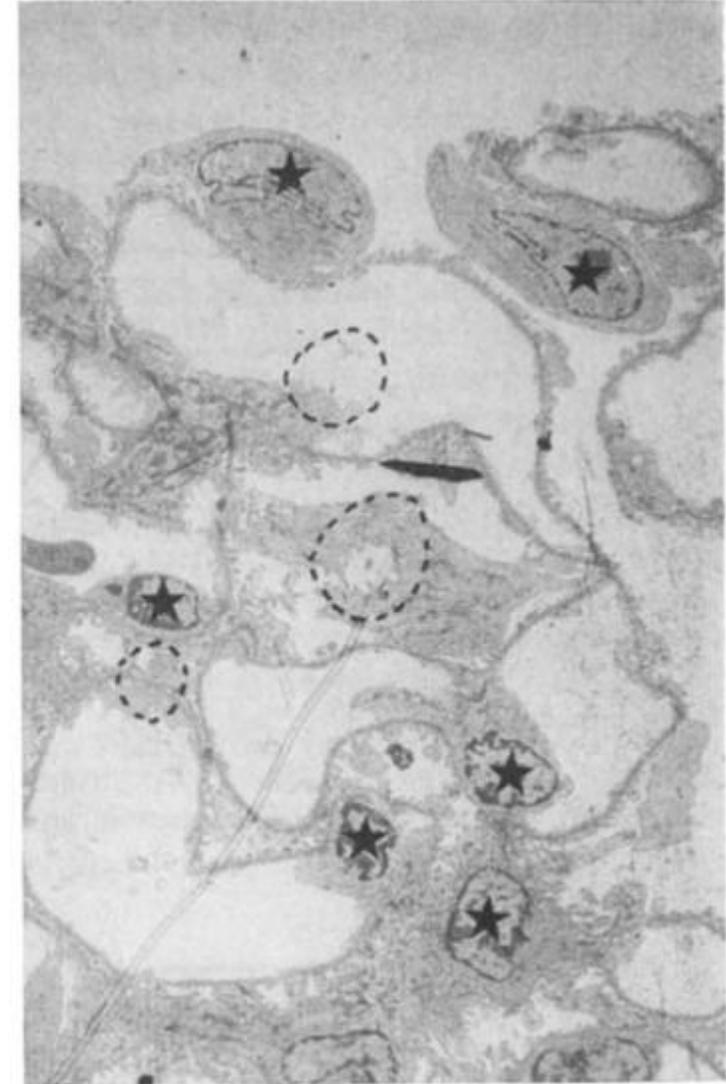
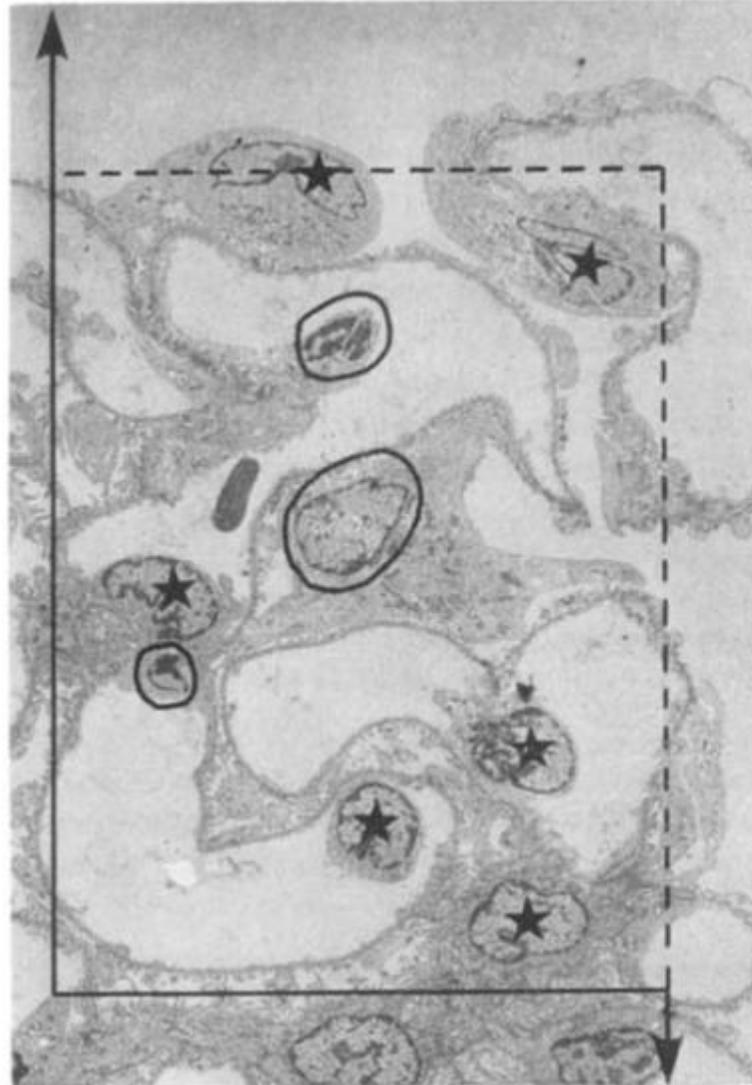
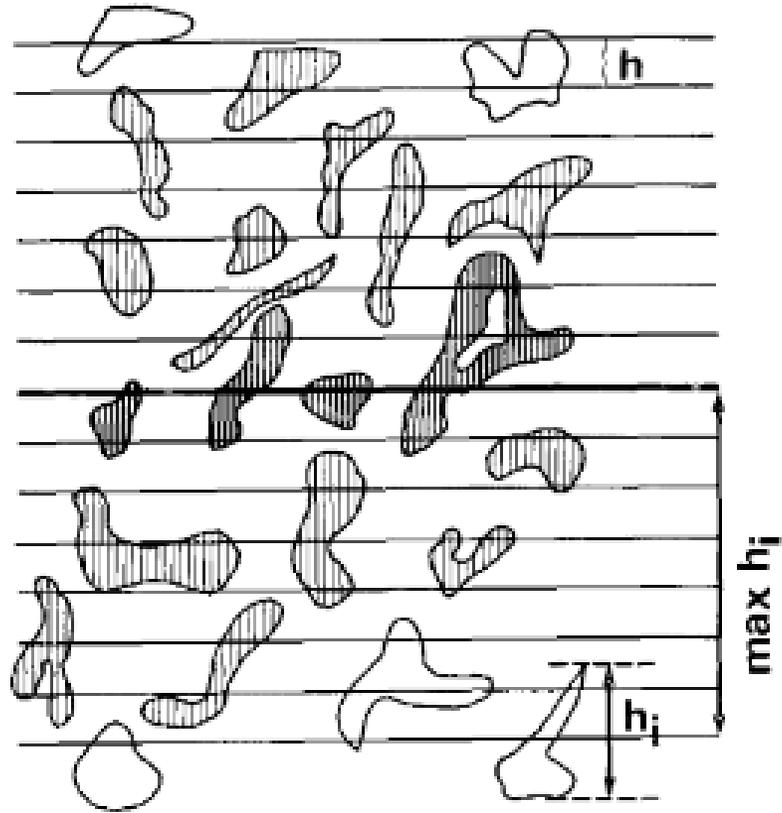
By software: $S = 45\,236 \text{ px}^2$

(The surface area is resolution-dependent! Using median filter to reduce the resolution:

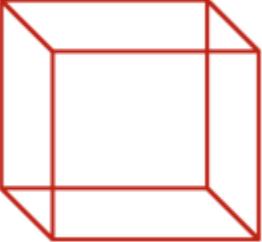
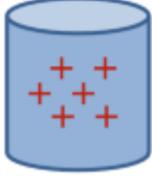
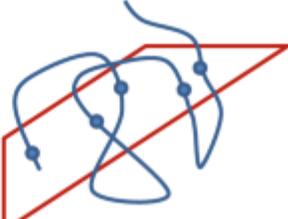
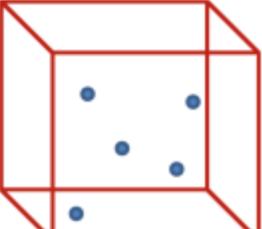
Kernel 2 radius: $S = 37\,528 \text{ px}^2$

Kernel 4 radius: $S = 33\,744 \text{ px}^2$

Number estimation with the disector



Stereology

Structure	Volume (3D) 	Surface (2D) 	Length (1D) 	Number (0D) 
Test system	Points (0D) 	Line (1D) 	Plane (2D) 	Volume (3D) 
Structure \cap Test system	$3D \cap 0D = 3D$ 	$2D \cap 1D = 3D$ 	$1D \cap 2D = 3D$ 	$0D \cap 3D = 3D$ 

✓ Congratulations,
You finished Part III, Thresholding, segmentation and
(particle) size analysis

