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

Dimitri Vanhecke

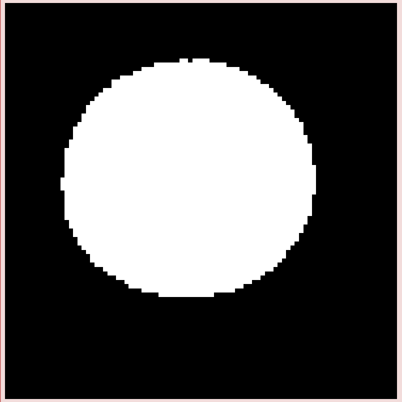
How does software measure images?

How does a software measure images?

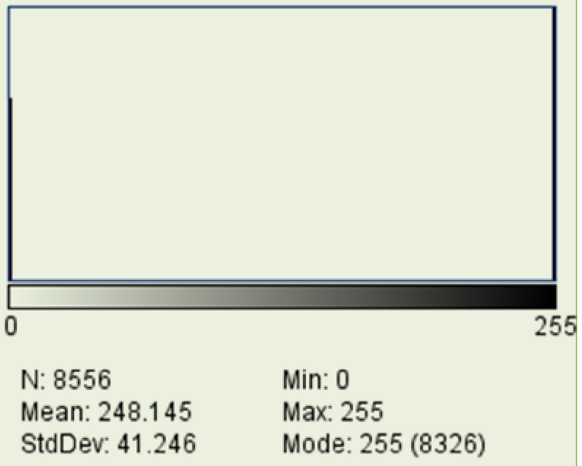
Primary units: Area of an object

Original grayscale

Thresholded



Histogram



# of pixels = area

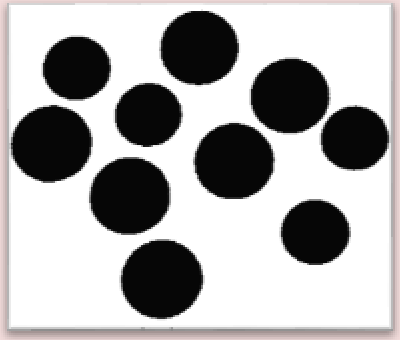
(by Analyse Particles)

How does a software measure images?

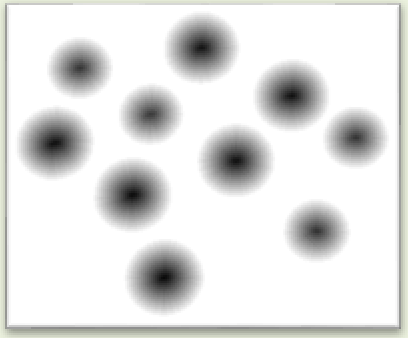
Primary units: Count objects

Grayscale

Thresholded

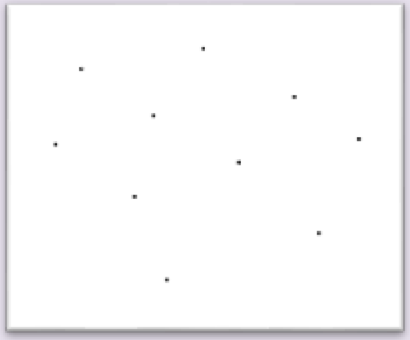


Distance transform

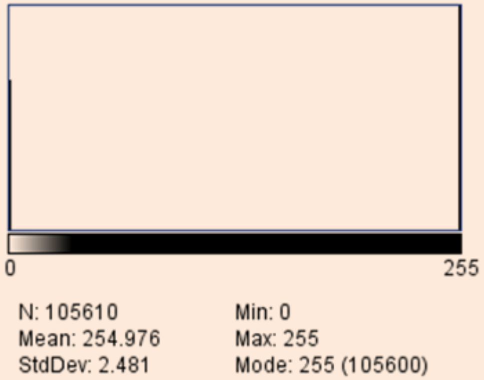


= a derived representation where every foreground pixel takes a value in function of the distance to the nearest background pixel

Max eroded points



Histogram



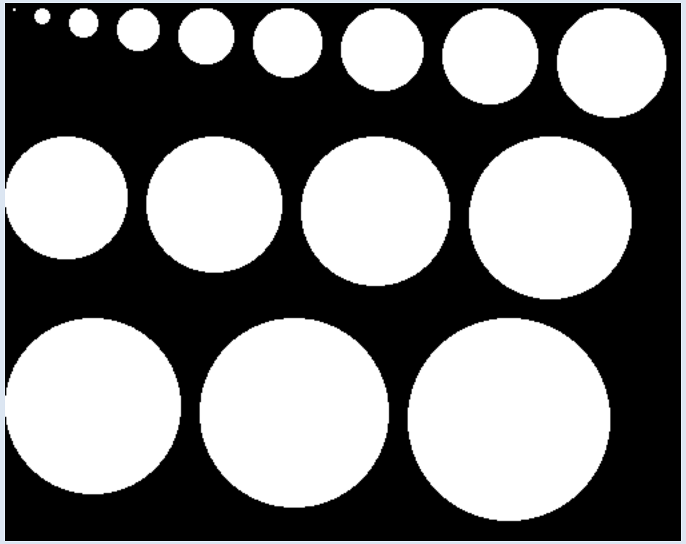
Get # of pixels = # Objects

value	count
0	10

	Area	Perim.
1	3701	226.794
2	2743	194.409
3	3754	228.208
4	2680	192.409
5	3900	234.451
6	2766	196.894
7	3831	230.208
8	3944	234.208
9	2822	198.894
10	4074	239.522

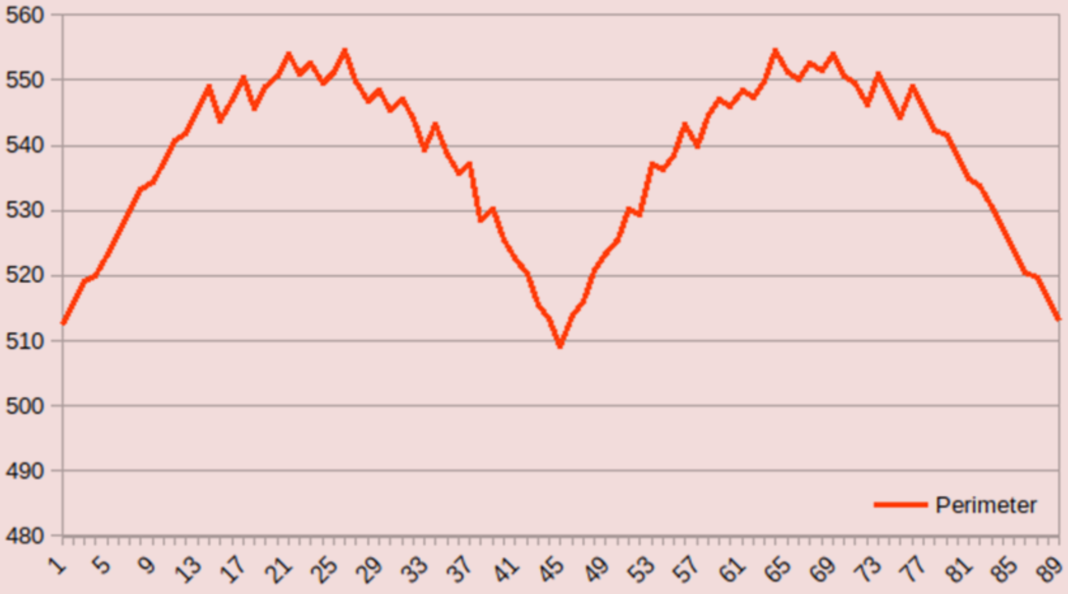
How does a software measure images?

Primary units: 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

"Perfect" circles do not have a circularity of 1

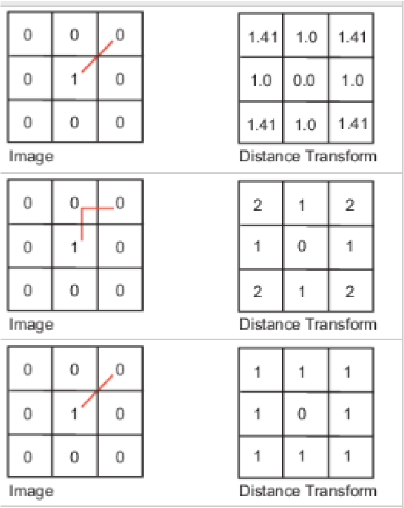


The perimeter of an object (here: 128x128 square) depends on its angular position.

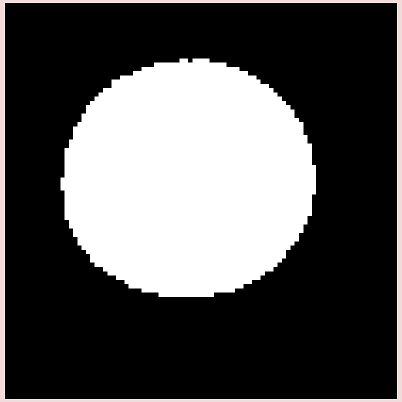
How does a software measure images?

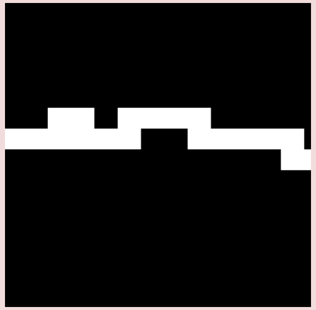
Primary units: perimeter of an object --> tricky (estimates)

Grayscale



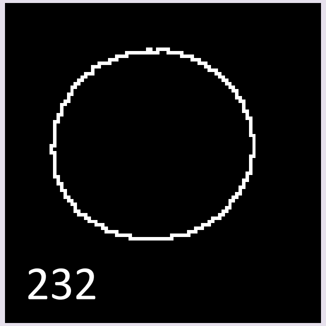
Boundary pixels



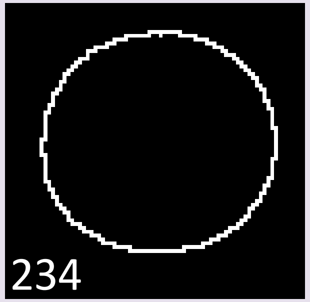


Checkerboard / Manhattan / Euclidean geometry

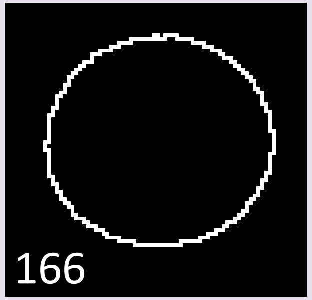
Threshold - Erosion



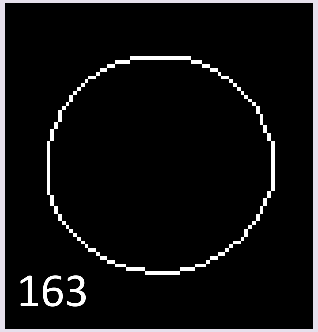
LoG $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$



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



Skeletonization



# of edge pixels

≅ perimeter

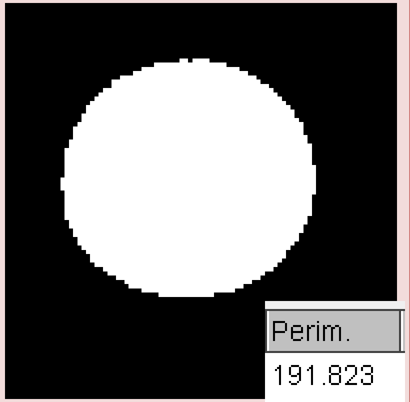
Perim.
191.823

How does a software measure images?

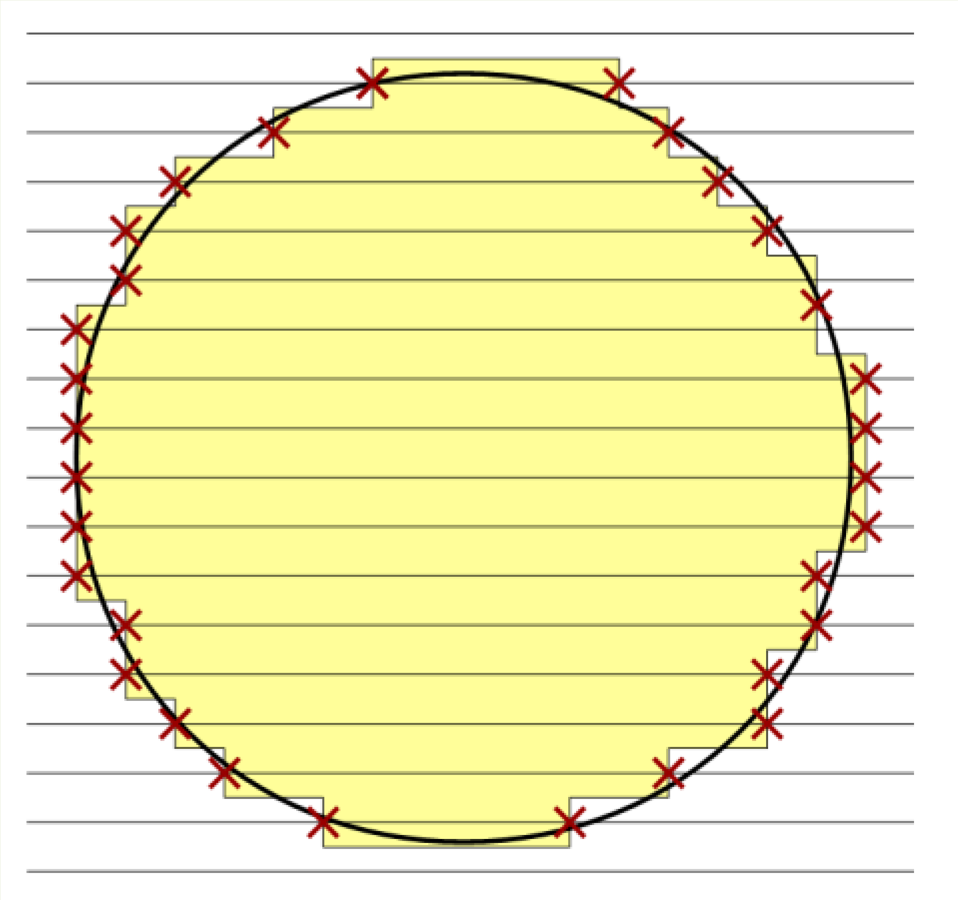
Primary units: perimeter of an object: Crofton estimator

Grayscale

Thresholded



Crofton (based on Buffon's needles)



2 Way Crofton
(horizontal and vertical)
P = 188.5

4 Way Crofton
(2-way + 2 diagonals)
P = 187.5

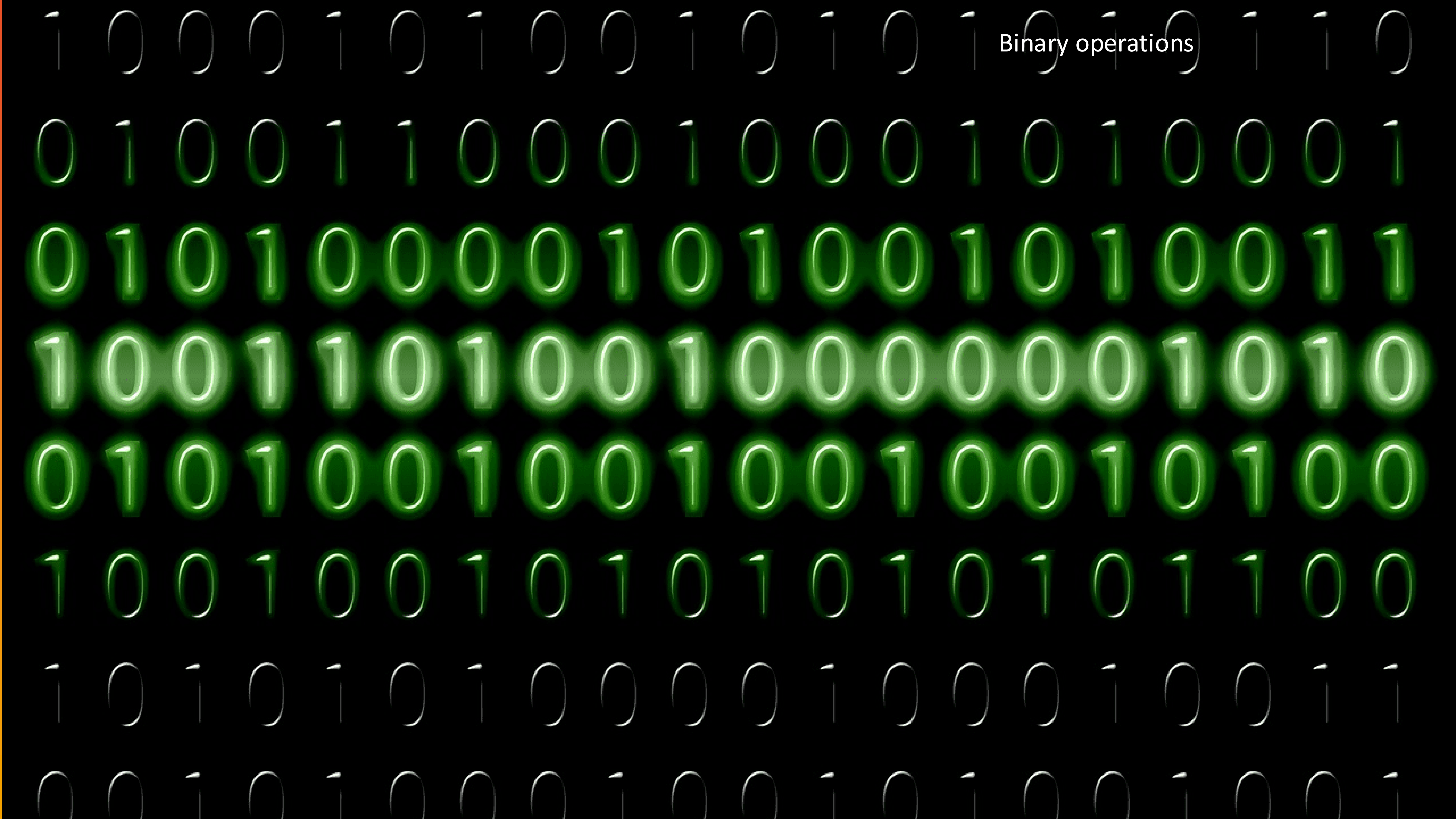
Published in "Nanoscale 9(15): 4918–4927, 2017"
which should be cited to refer to this work.

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

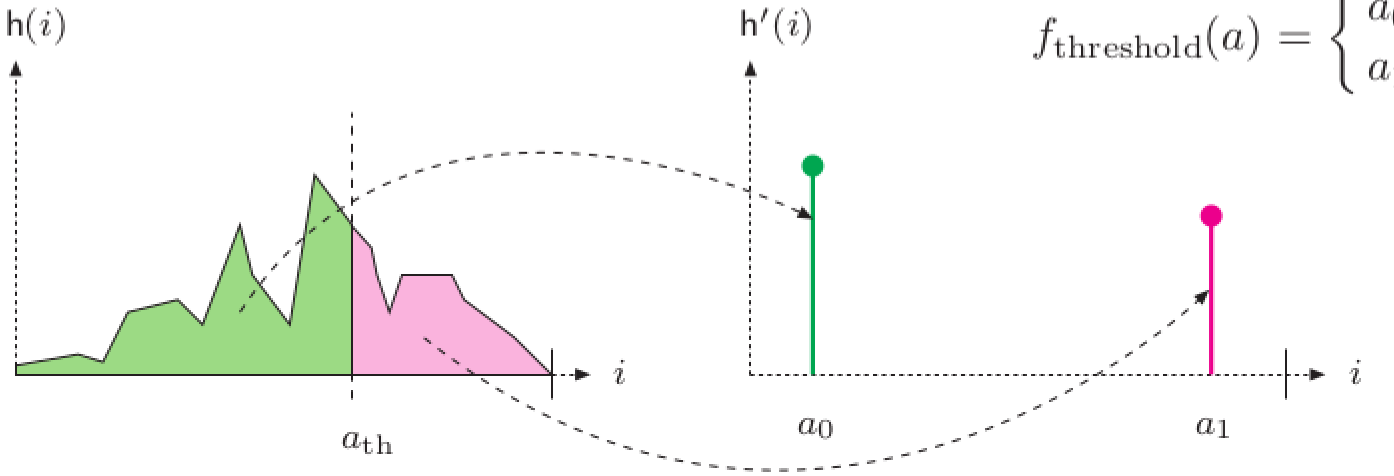
Dimitri Vanhecke, *[a] Laura Rodríguez-Lorenzo,[a] Calum Kinnear,[b] Estelle Durantie,[a] Barbara Rothen-Rutishauser[a] and Alke Petri-Fink[a,c]

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.

Binary operations



Thresholding / binarization / segmentation

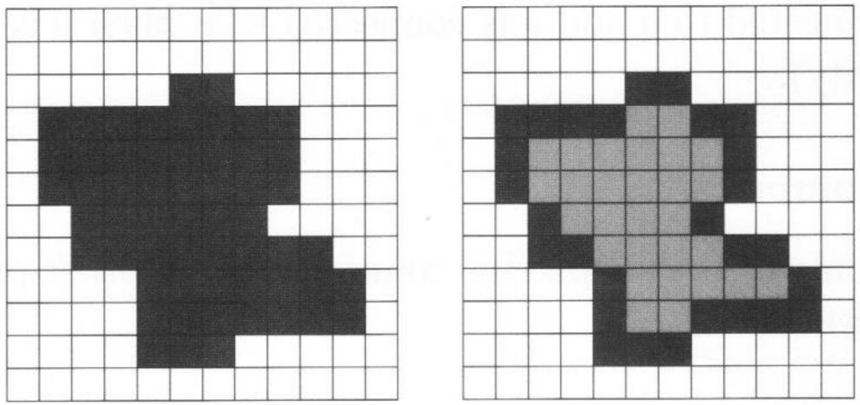


$$f_{\text{threshold}}(a) = \begin{cases} a_0 & \text{for } a < a_{\text{th}} \\ a_1 & \text{for } a \geq a_{\text{th}} \end{cases}$$

Morphological binary operations

Prerequiste: 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

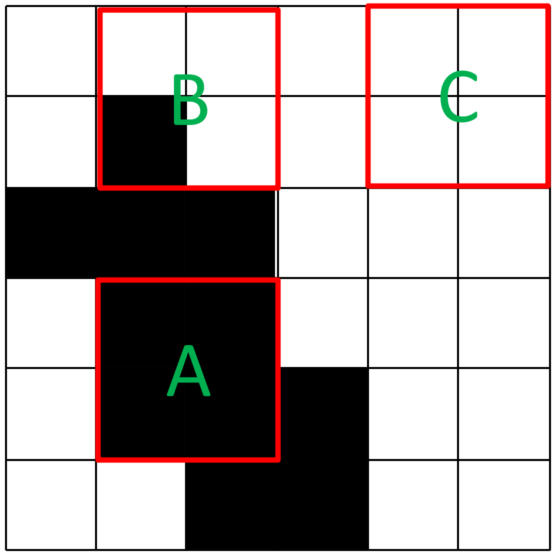
Morphological operations rely only on the relative ordering of pixel values, not on their numerical values (hence: binary data)

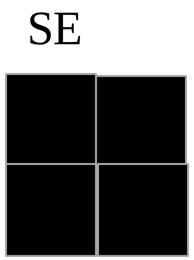
Morphological binary operations – structuring element

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.

Fits	A	SE fits within the neighbourhood
Hits	B	SE hits a boundary
None	C	Neither hits not fits

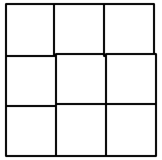
Background = 0, black
Foreground = 1, white

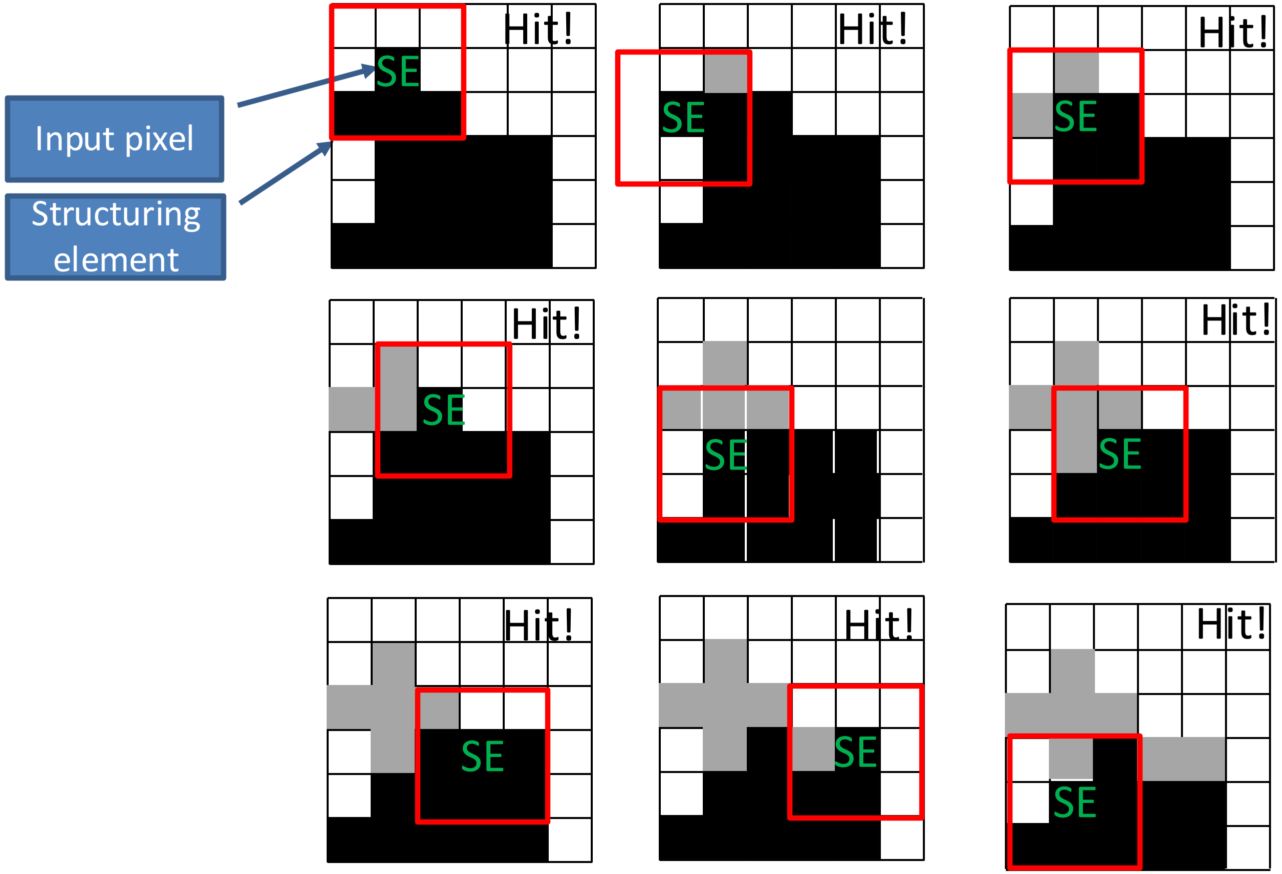


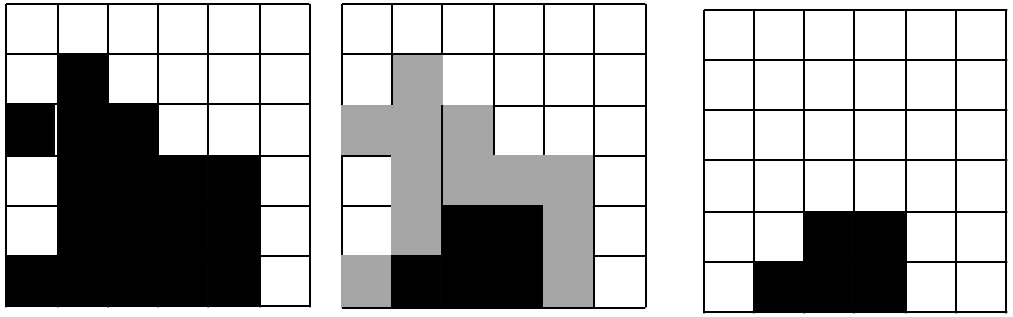


Basic (primary) binary operations: dilation

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

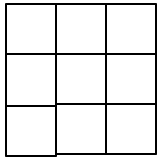


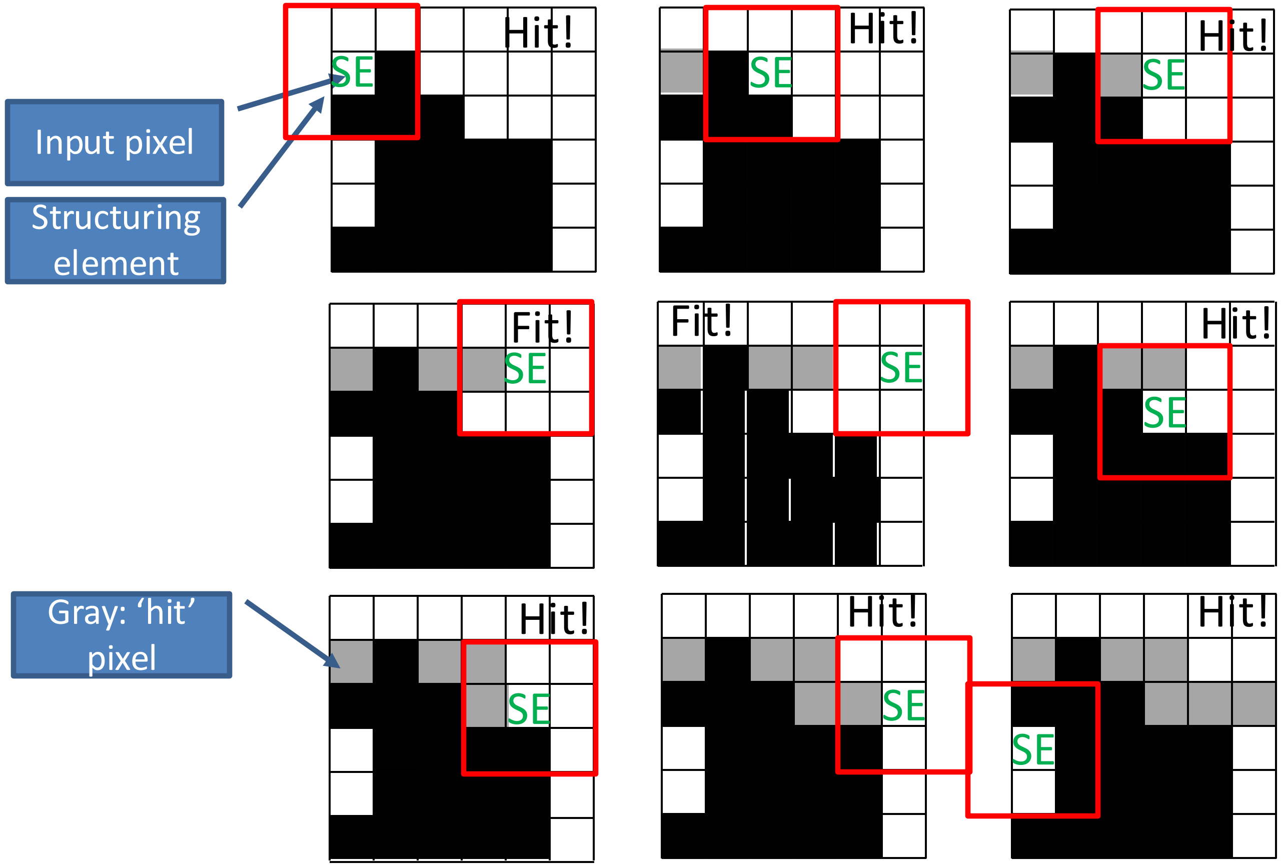


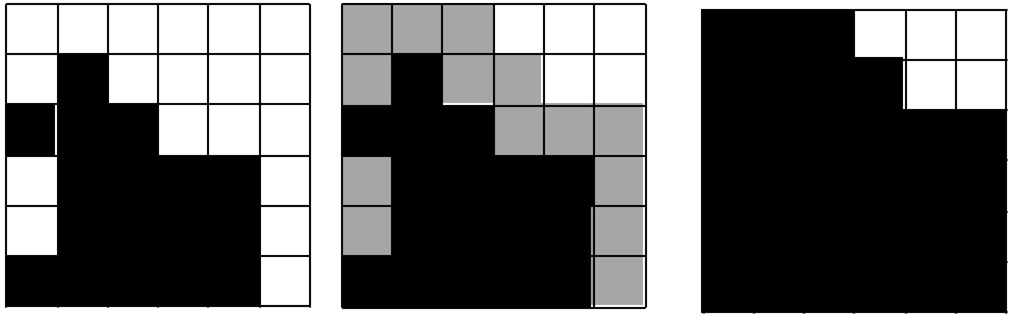


Basic (primary) binary operations: erosion

1. Consider each of the ***foreground*** pixels
2. For each foreground pixel (= *input pixel*) the SE is superimposed. (origin of the SE coincides with the input pixel).
3. **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)
4. **When fit or none**: do nothing (If all the corresponding pixels in the image are foreground the input pixel is left at the foreground value).
5. Structuring element:







Basic (primary) binary operations: dilation and erosion

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: open and close



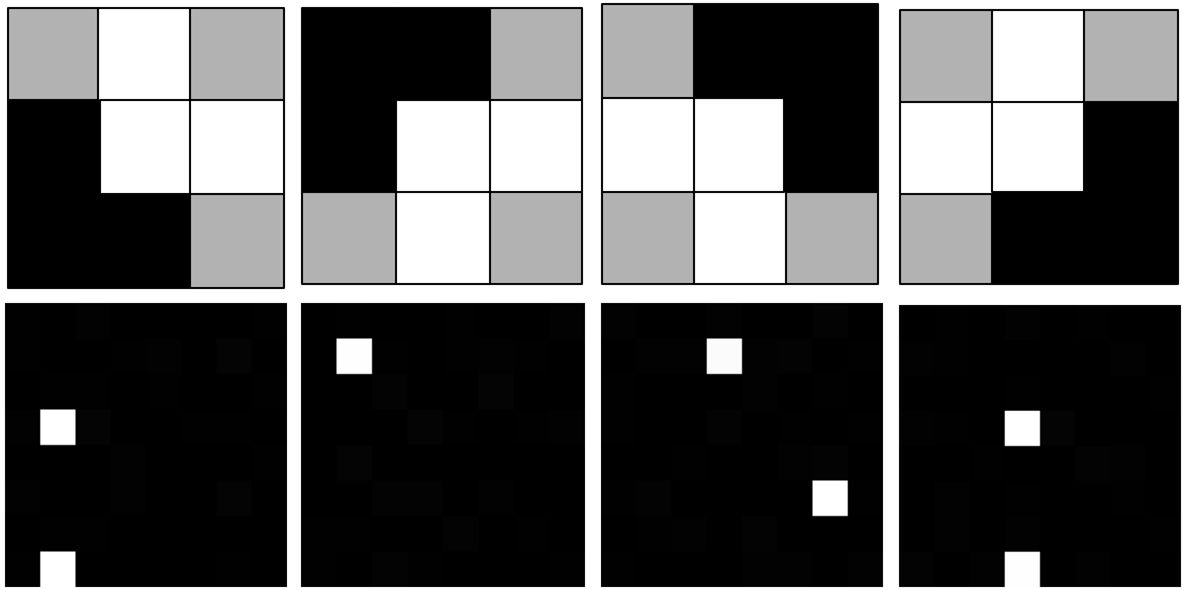

Close
First erodes, then dilates.
Gentle way to remove salt grains (=cleanup of background)

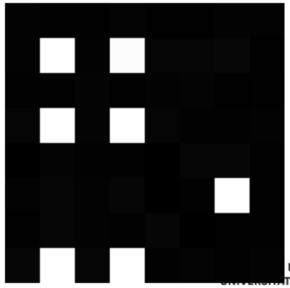
Open
First dilates, then erodes.
Gentle way to remove pepper noise (=cleanup of foreground)

Idempotence
The property of applying more than once does not produces a further change. E.g. Open and close binary operators

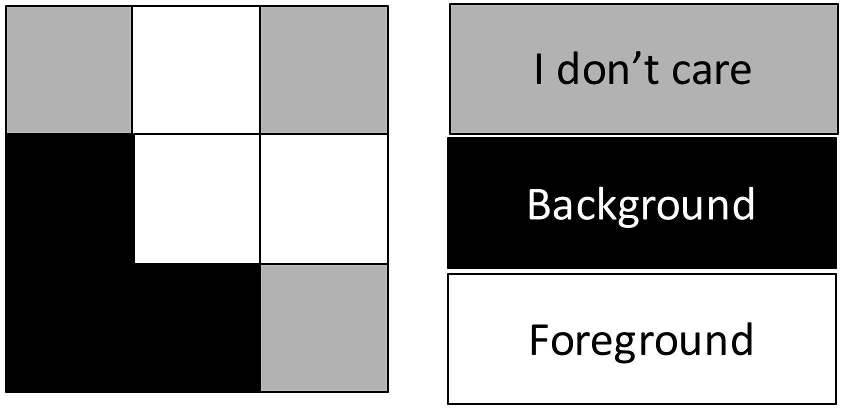
binary operations: Hit and miss

1. Foreground pixels of SE hits foreground input pixel:
   When hit: input pixel changed to background
   When fit: do nothing
2. Background pixels of SE hit background pixel:
   When hit: input pixel changed to foreground
   When fit: do nothing
3. I don't care pixels: ignore

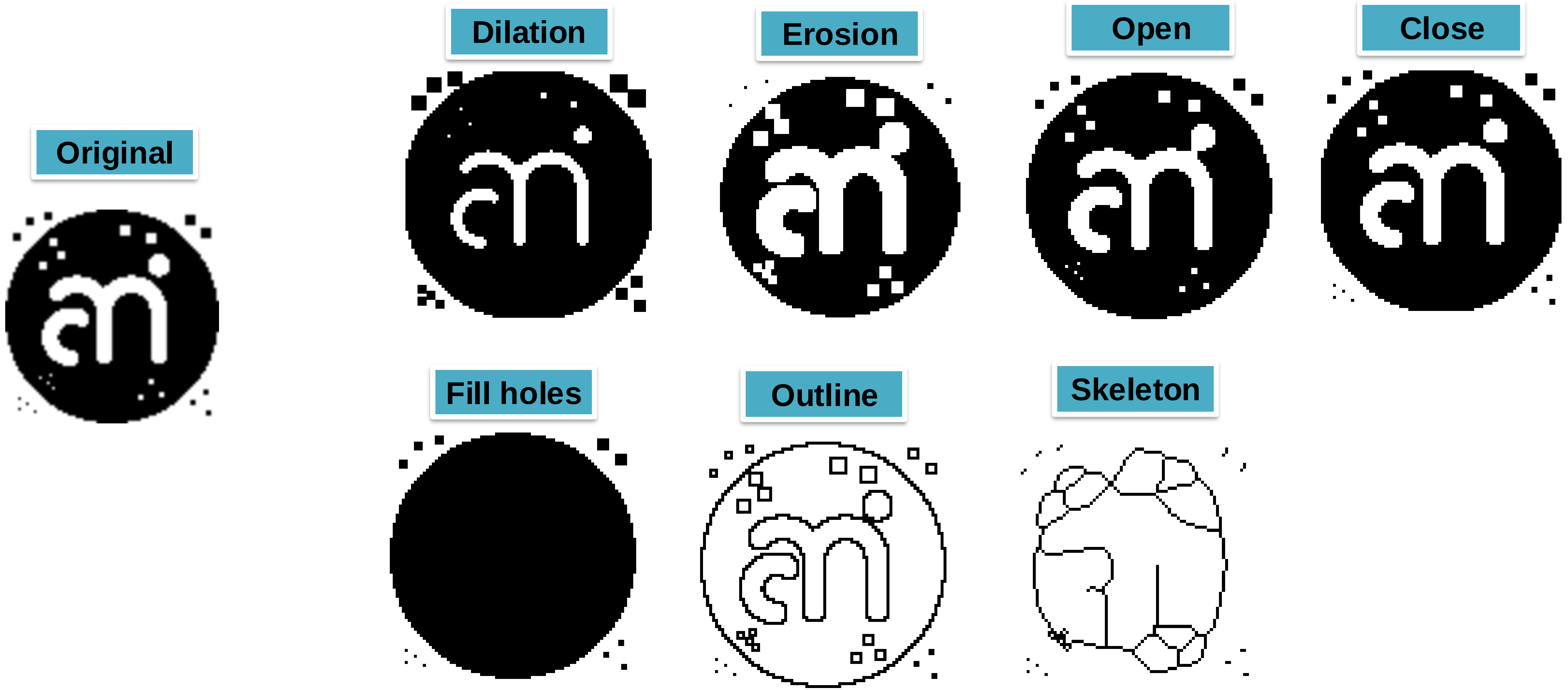




Corner detection!

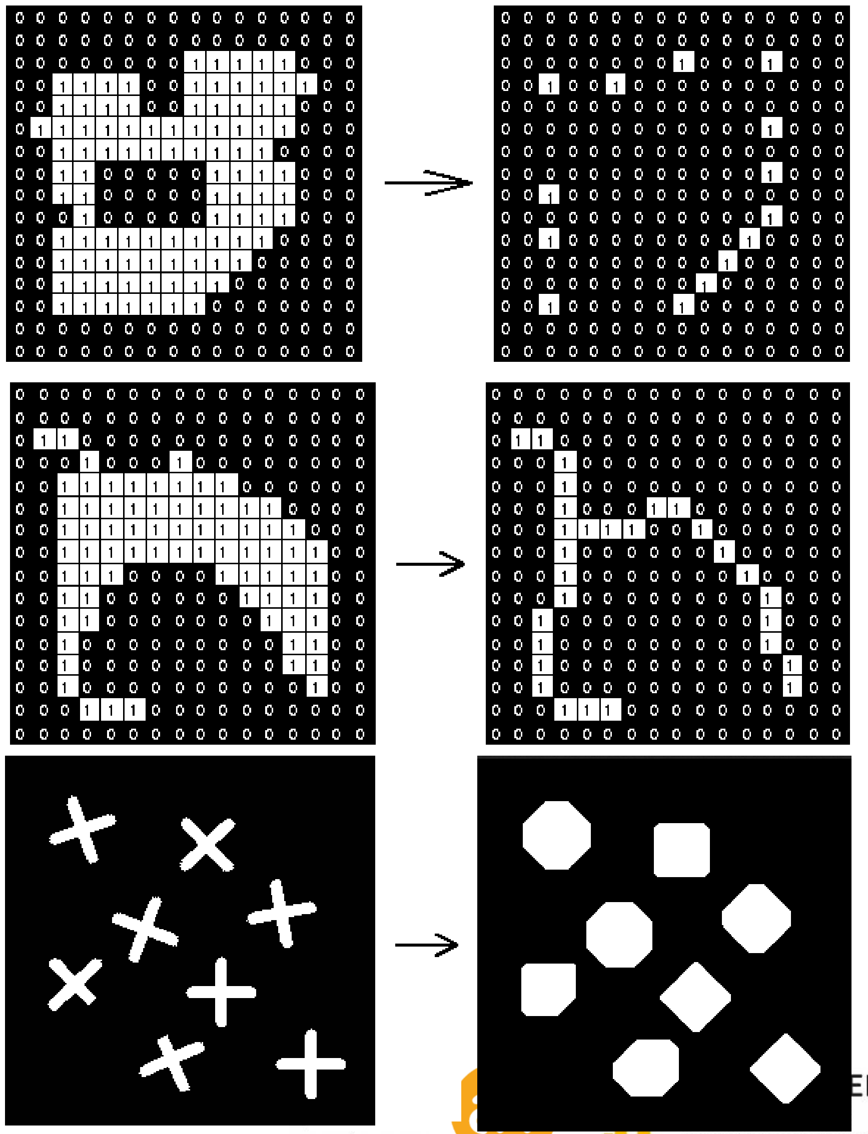


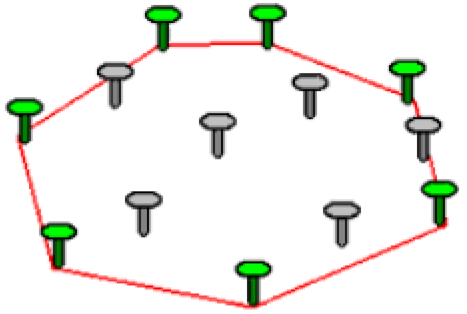
Binary operations



Binary operations

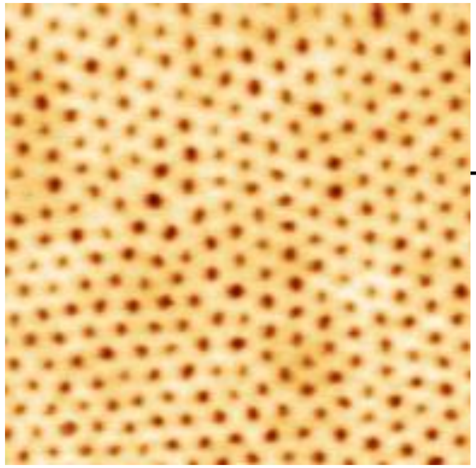
Hit or miss	Finding ends and corners
Thinning	Reduces the object to a single pixel line (skeletonization)
Thickening	Calculate convex hull of object
White top-hat	First opens (removing bright structures smaller than structuring elements), then removes the result from the original image. When applied with a large structuring element, the result is an homogenization of the background, making bright structures easier to segment.
Dark top-hat	can be used to enhance dark structures observed on an nonhomogeneous background.

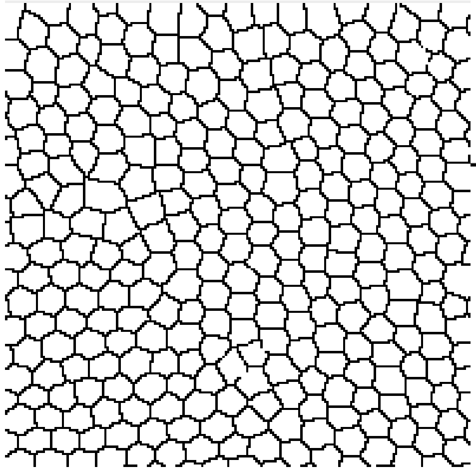




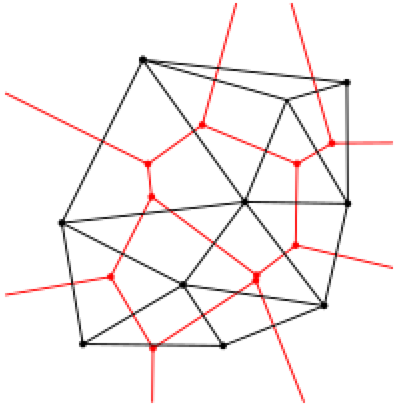
Binary operations: further applications

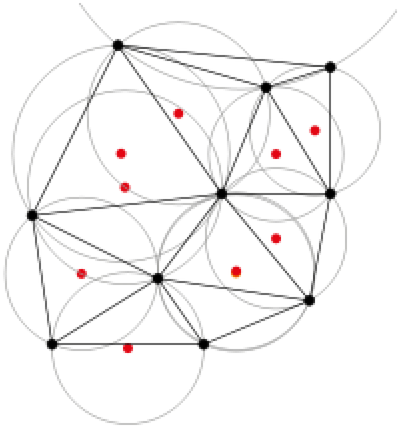
Voronoi diagrams

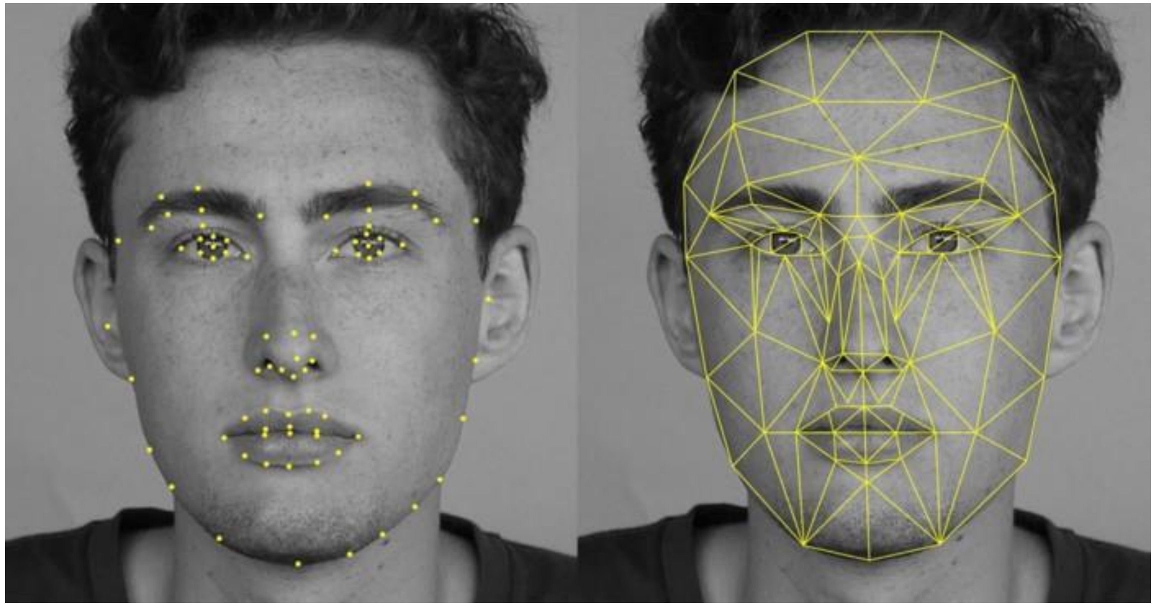


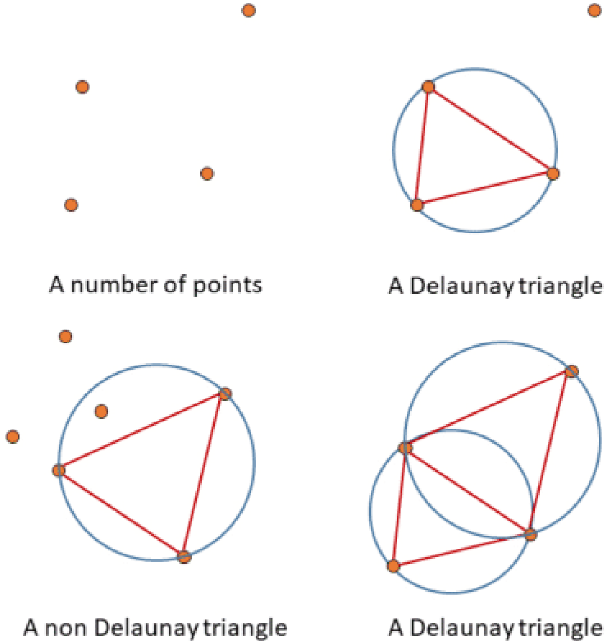


Delauney tesselation (red: Voronoi, black: Delauney)

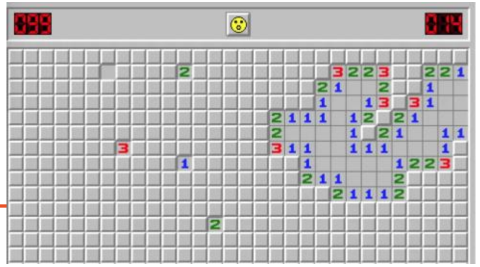








Example:
- Fingerprint analysis
- Face recognition
- …



Binary operations: Eucledian Distance transform

A **distance transform**, is a derived representation of a **binary** digital image

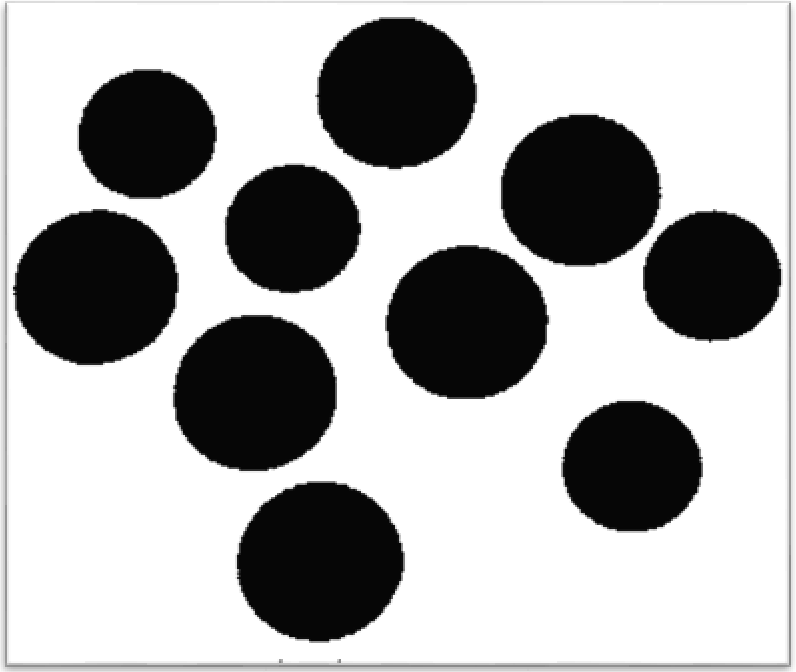
The result: the **Euclidian distance map**. Each foreground pixel in the binary image is replaced with a gray value equal to that pixel's distance from the nearest background pixel (for background pixels the EDM is 0)

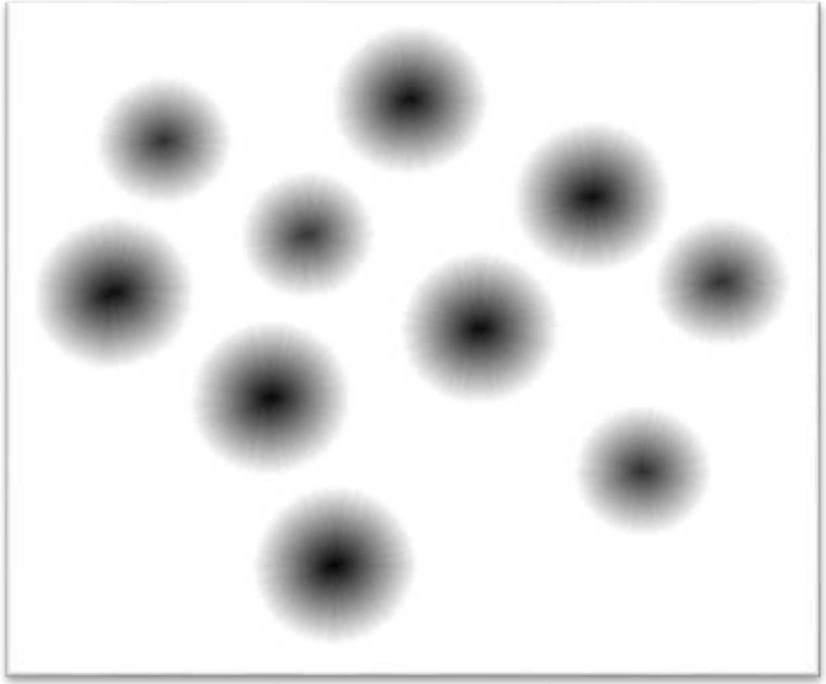
0	0	0	0	0	0	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	0	0	0	0	0	0

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

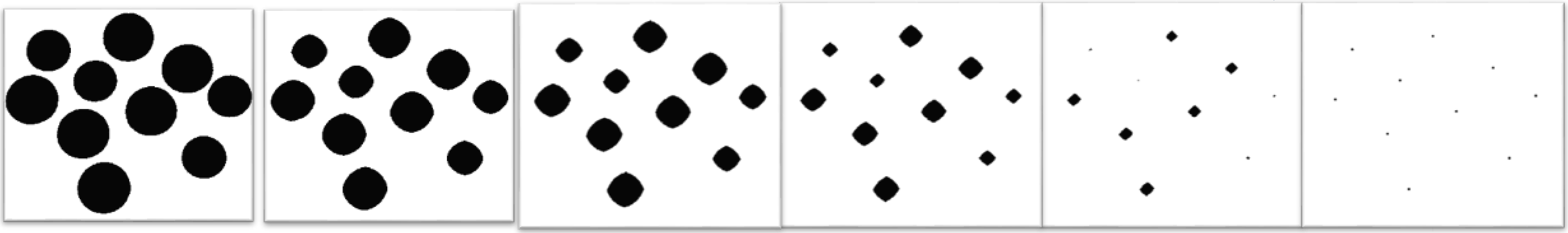




Binary operations: Ultimate eroded points

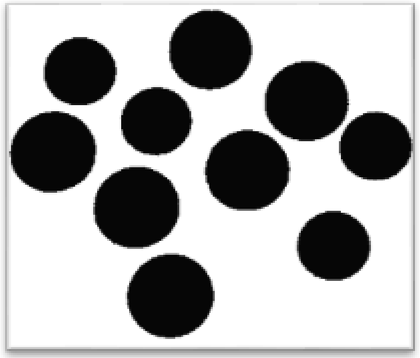
The **Ultimate Points** extracts the last point that would be removed if the object were eroded to completion.
They represent the seed of an object (=number of objects).

Iterations of eroding steps

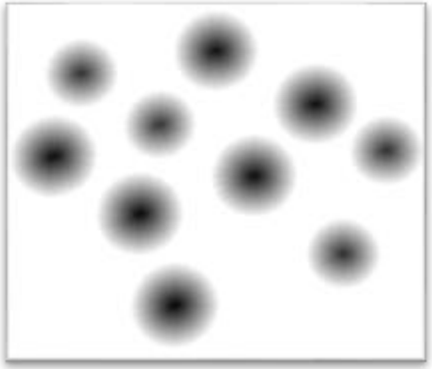


Origin

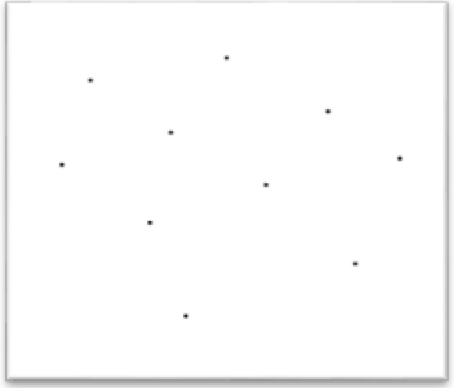
binary



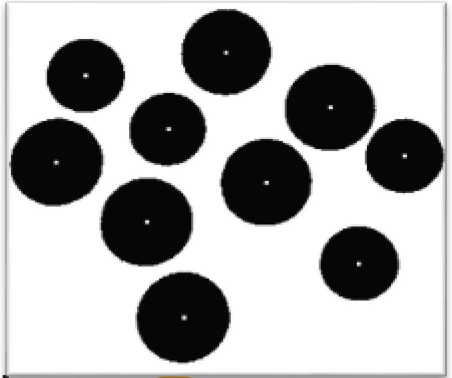
Eucledian distance map



Ultimate eroded points



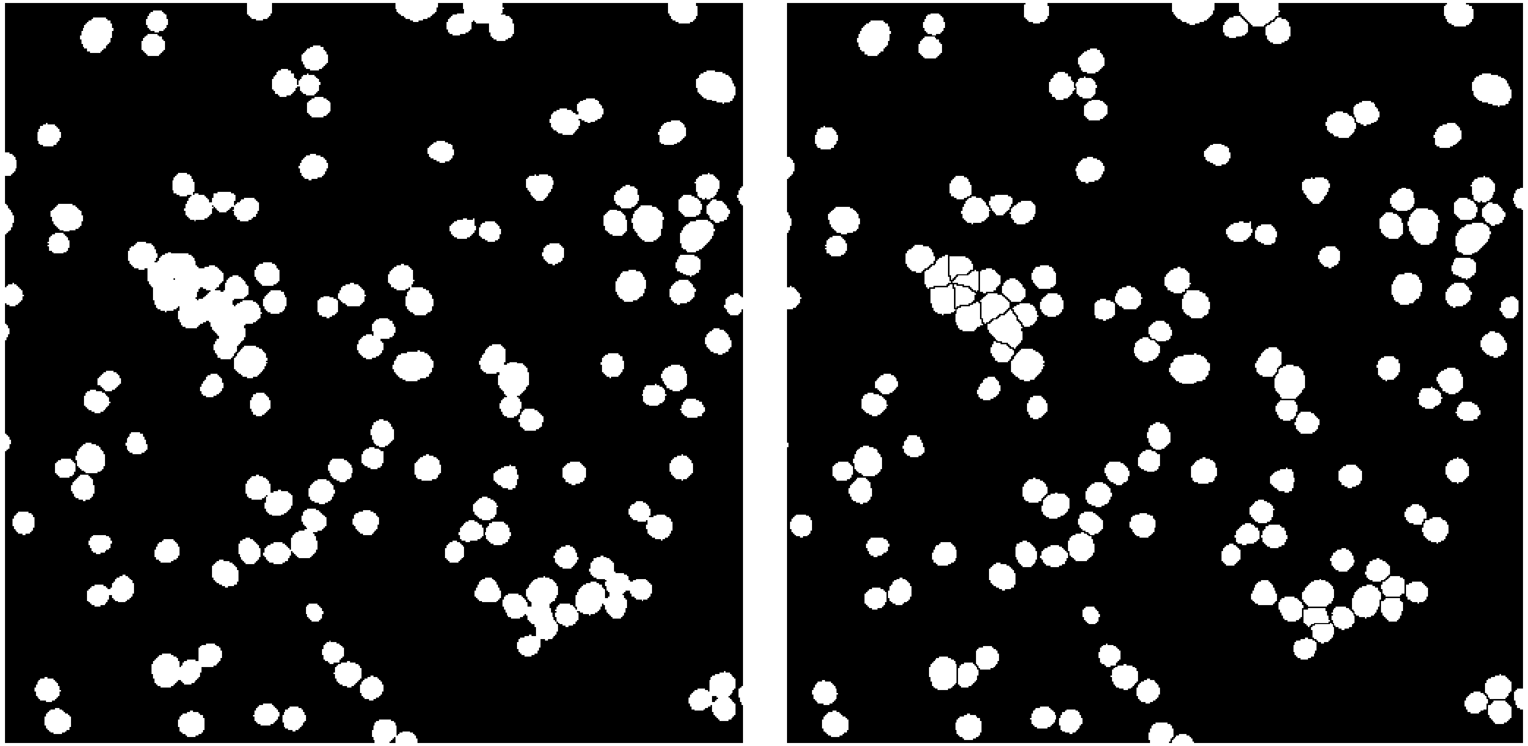
Overlay UEP with binary

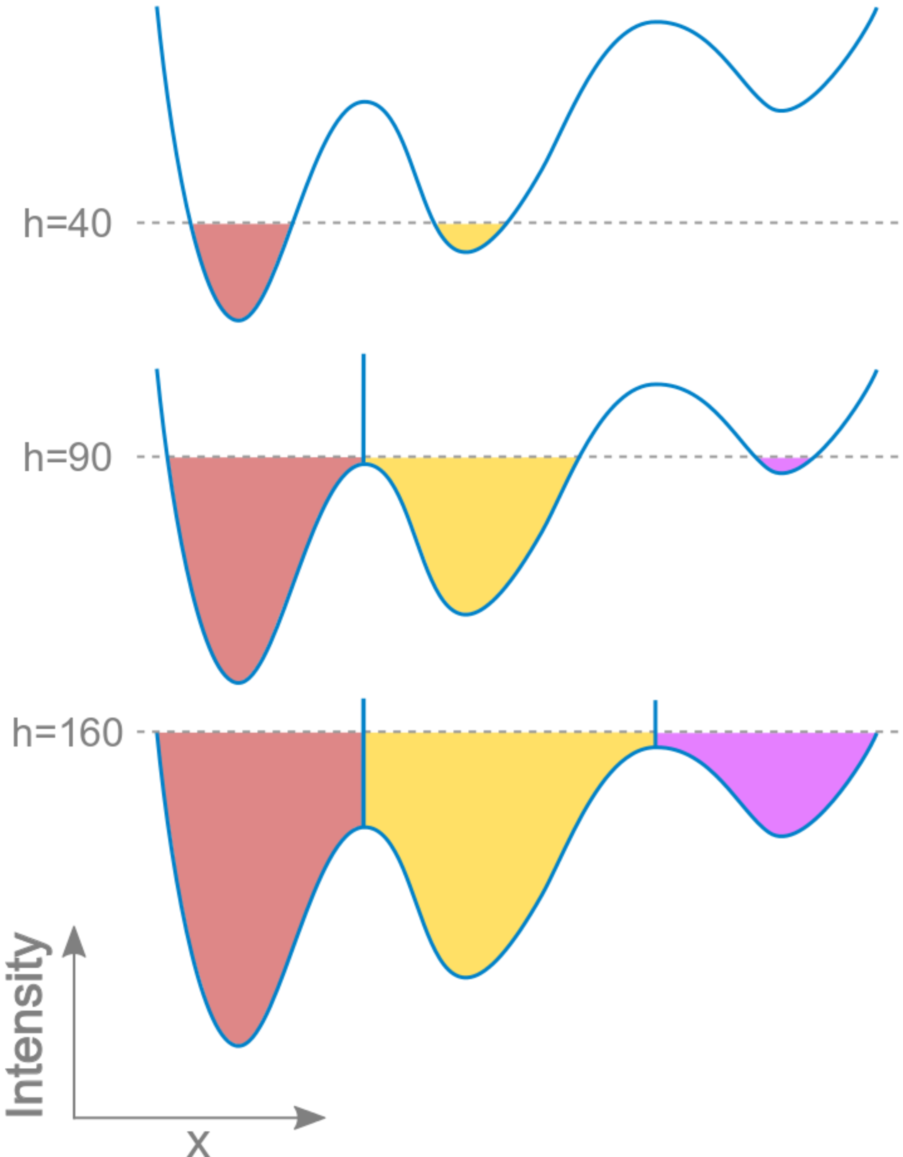


Binary operations: Watershed

Watershed segmentation is a way of automatically separating 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 as far as possible:
    1. until the edge of the original particle is reached
    2. Or the edge touches a region of another (growing) UEP.

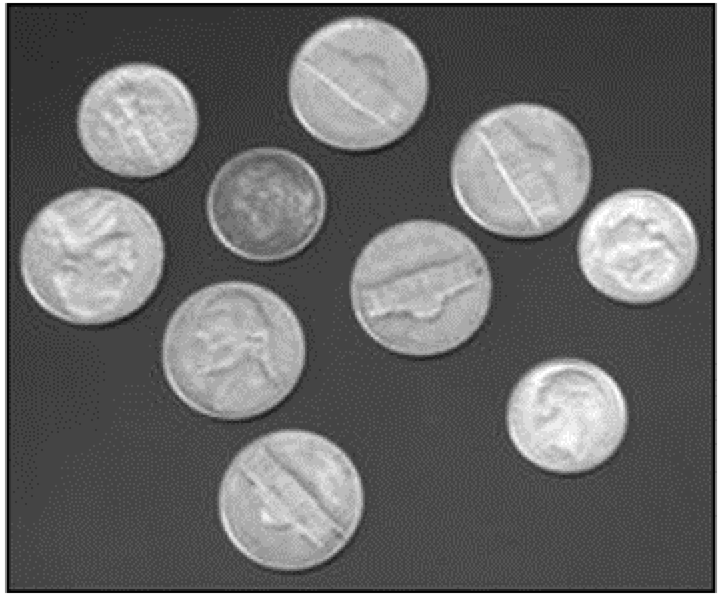


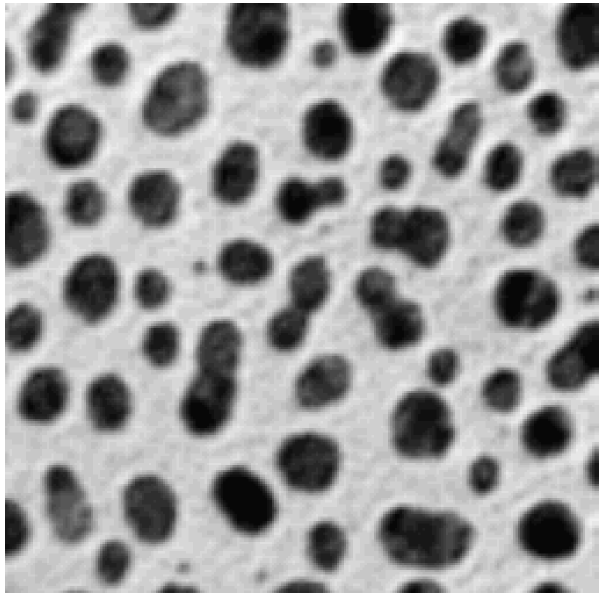


How does a software measure images?

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

Process > Binary > Ultimate points

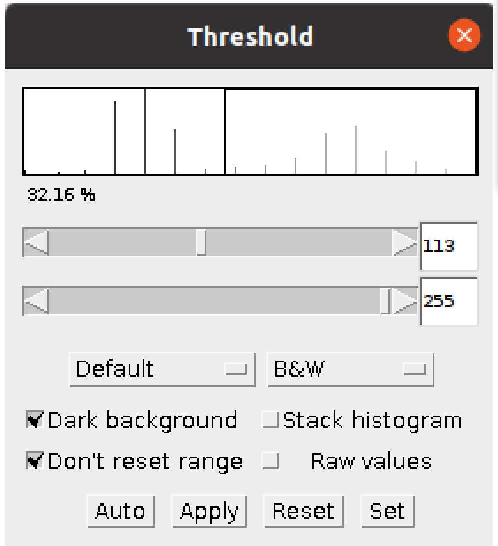


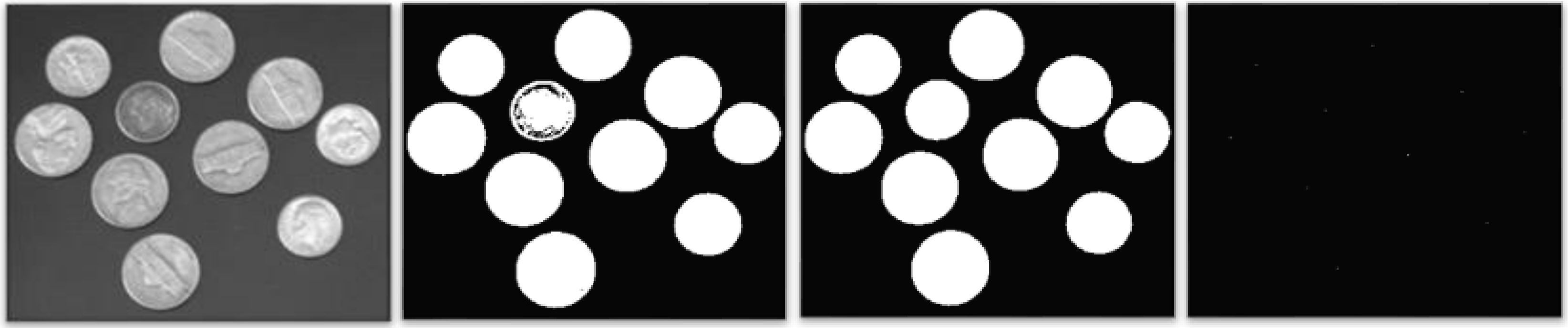


How does a software measure images?

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



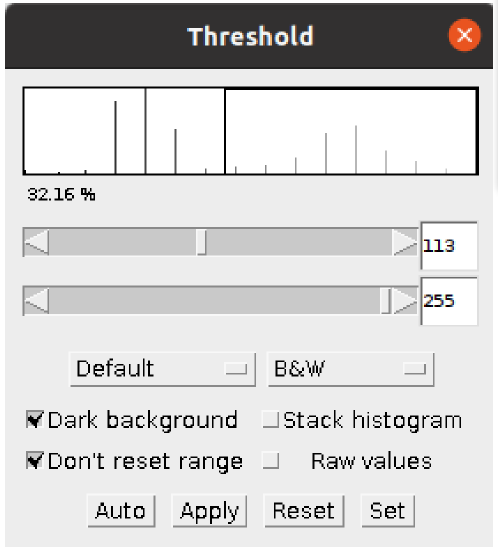


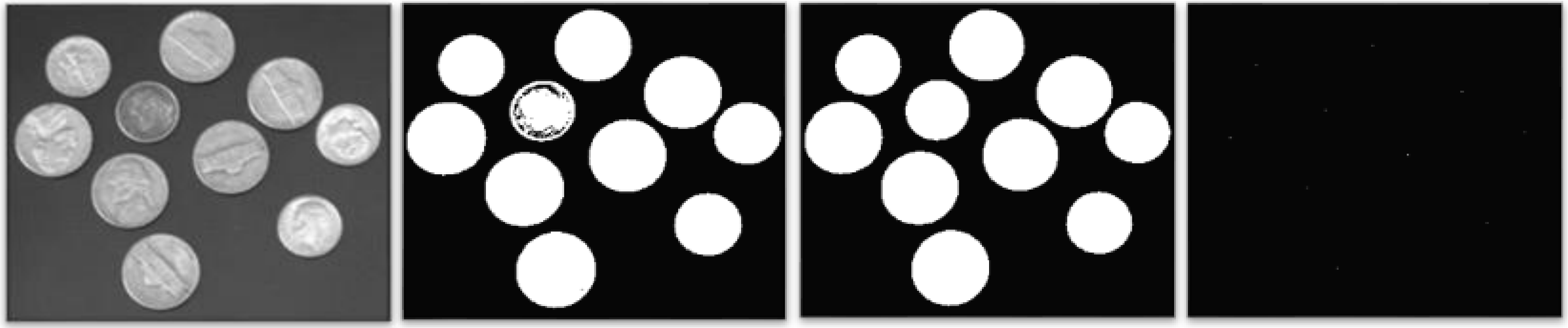
Process > Binary > Fill holes
Process > Binary > Ultimate points
(Calculates the EDT and then the UEP)
Process > Histogram
(In Histogram window) > List

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





Process > Binary > Fill holes
Process > Binary > Ultimate points
(Calculates the EDT and then the UEP)
Process > Histogram
(In Historgam window) > List

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

How does a software measure images?

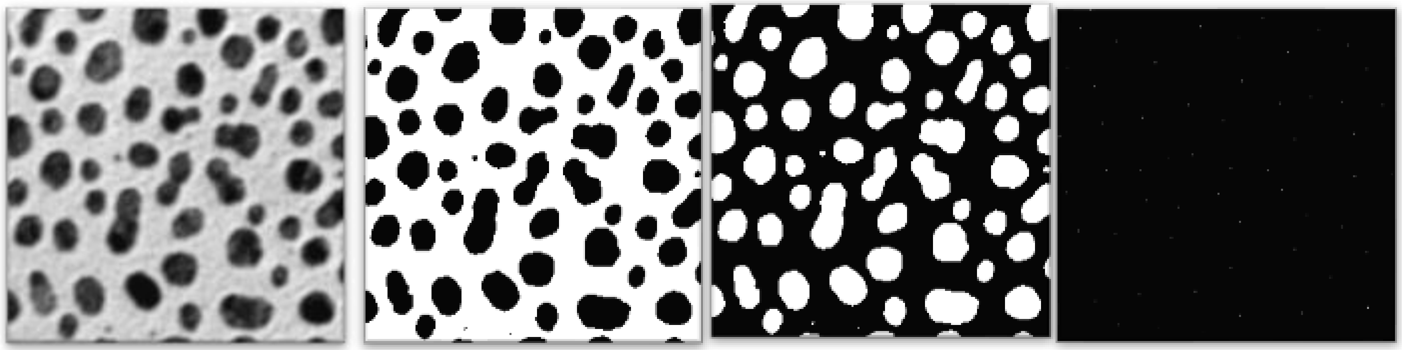
EXERCISE
Open example 1B and count the number of blobs using maximum eroded points.

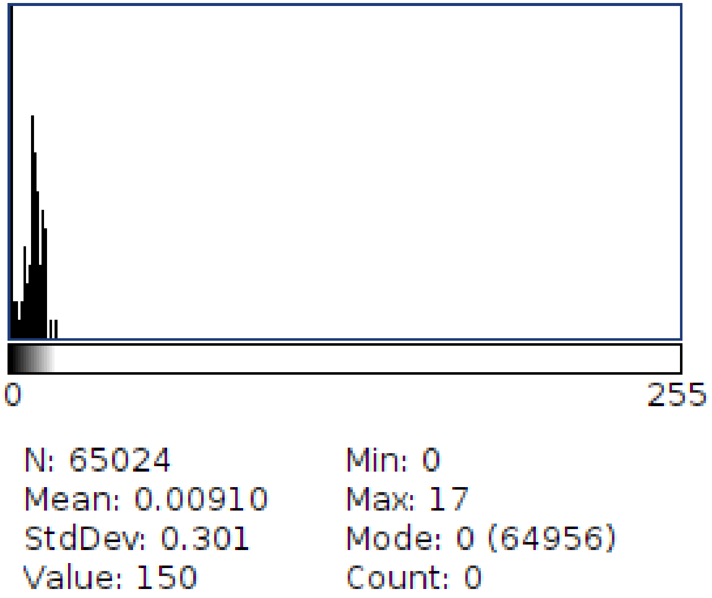
File > Open...
Image > Adjust > Threshold
Edit > Invert (make sure your objects are White)*

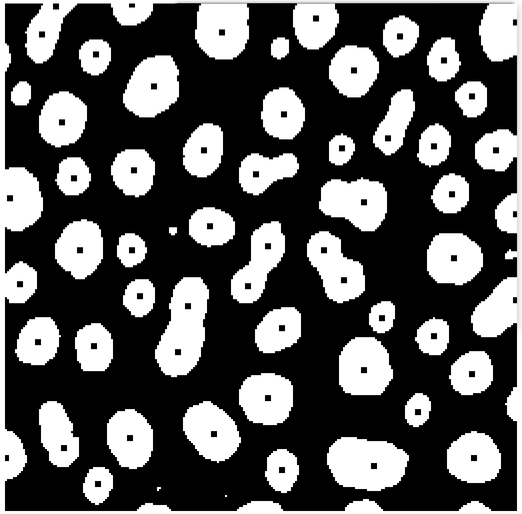
Process > Binary > Ultimate Points
Process Histogram

To count:
Process > Make binary
(In histogram) > List > check at value 255

255 56







0	64956
1	2
2	2
3	1
4	2
5	5
6	3
7	4
8	12
9	10
10	8
11	4
12	7
13	6
14	0
15	1
16	0
17	1
18	0
19	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

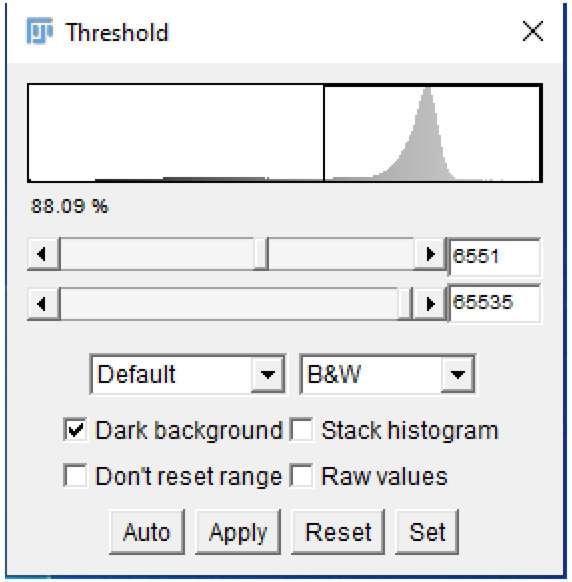
File > Open…

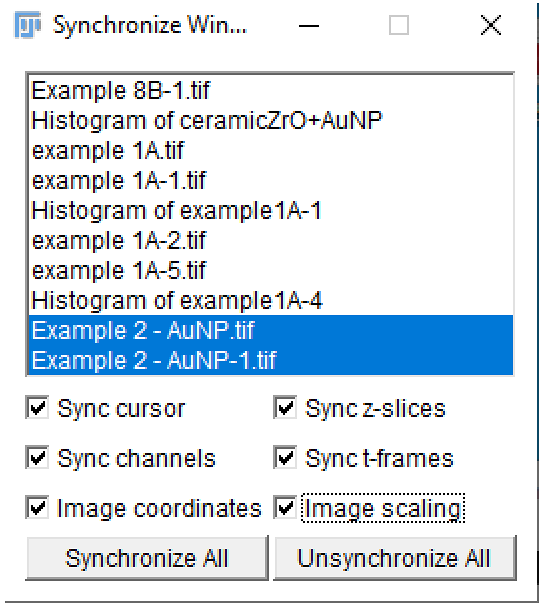
Image > Adjust > Threshold…

Click ‘Auto’

Click ‘Apply’

Duplicate the image (ctrl+shift+D)





Process > Binary > Watershed

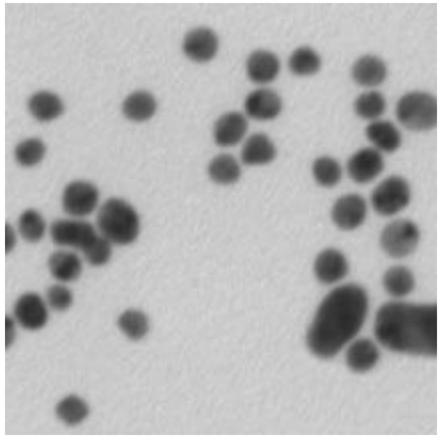
To compare the two windows: Analyze > tools > Synchronize windows

Binary operations: Watershed

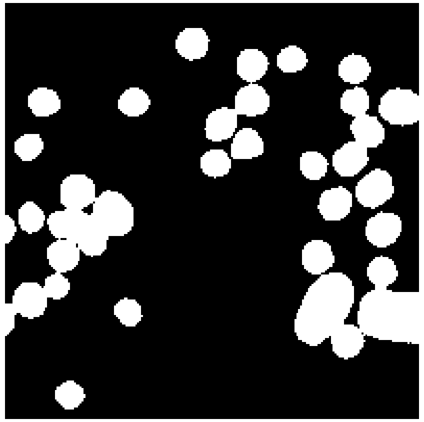
EXERCISE
Convert Example 2 – AuNP to a binary image. Compare with and without watershed

Process > Binary > Watershed

Original

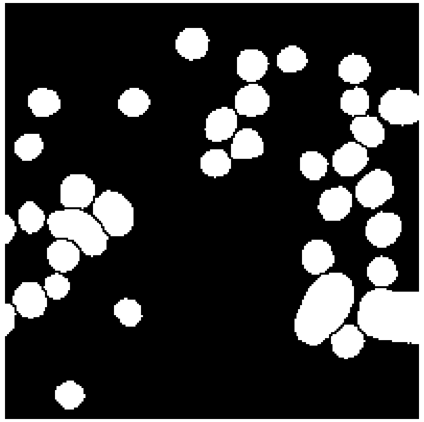


Binary

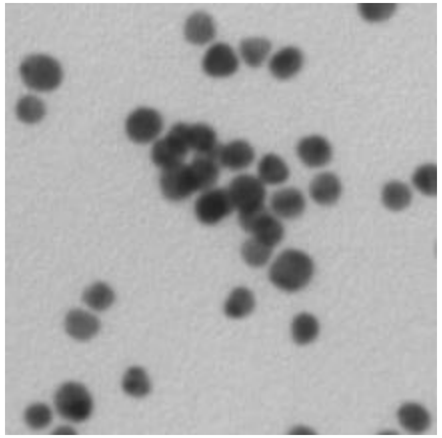


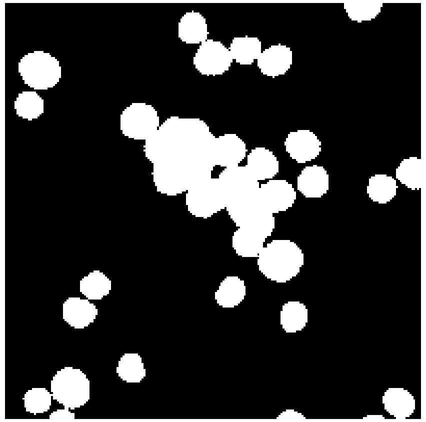
17 objects
(not touching the edge)

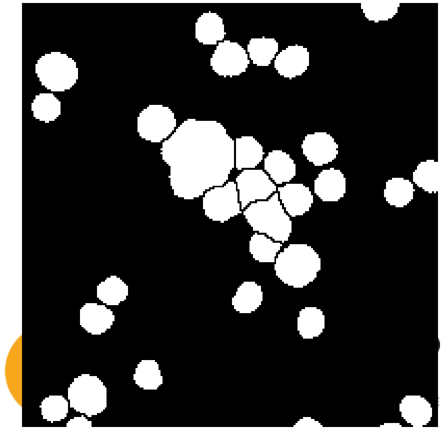
Binary + watershed



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	$\frac{4 \cdot \pi \cdot area}{perimeter^2}$, for each object
Aspect ratio	$\frac{Minor\ axis}{Mahor\ axis}$, for each object
Roundness	$\frac{4 \cdot area}{\pi \cdot major\ axis^2}$, for each object
Solidity	area/convex area.
Feret's Diameter	Longest distance between any two pixels in an object.

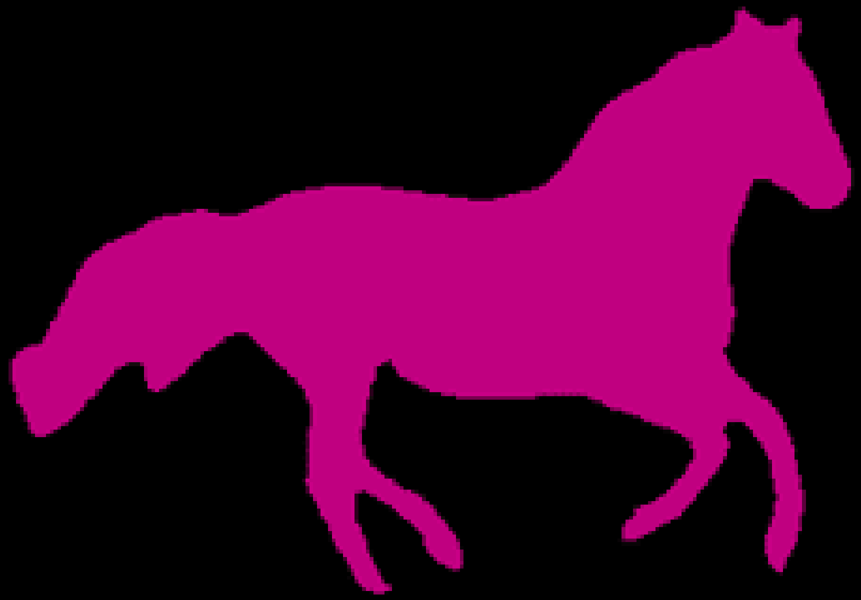
…

Everything relies on the thresholding step…

UNI FR UNIVERSITÉ DE FRIBOURG UNIVERSITÄT FREIBURG

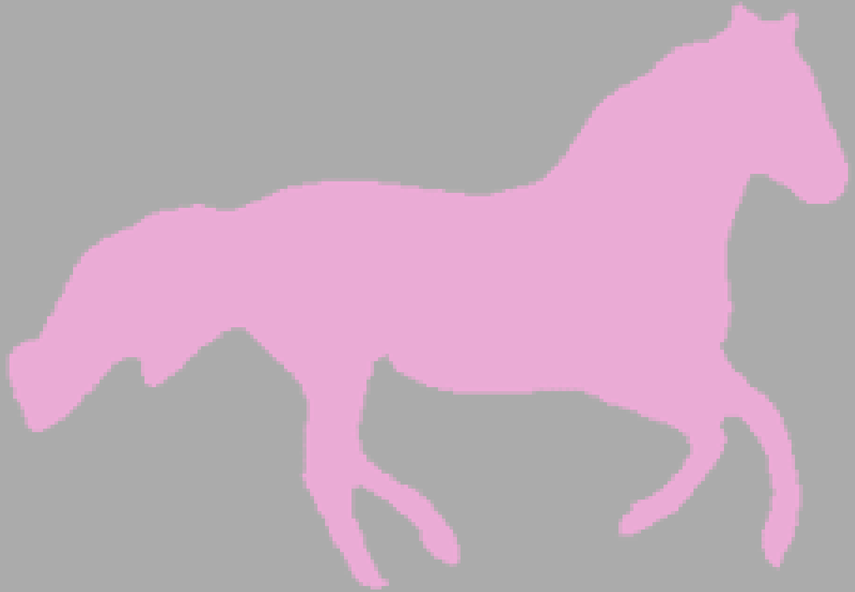
BIO-INSPIRED MATERIALS NATIONAL CENTER OF COMPETENCE IN RESEARCH

Thresholding, classification and segmentation



Thresholding, classification and segmentation

Level 1

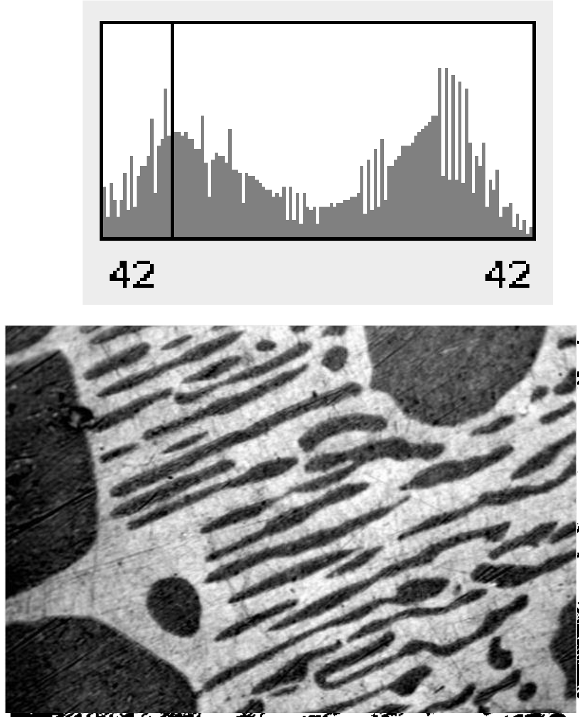


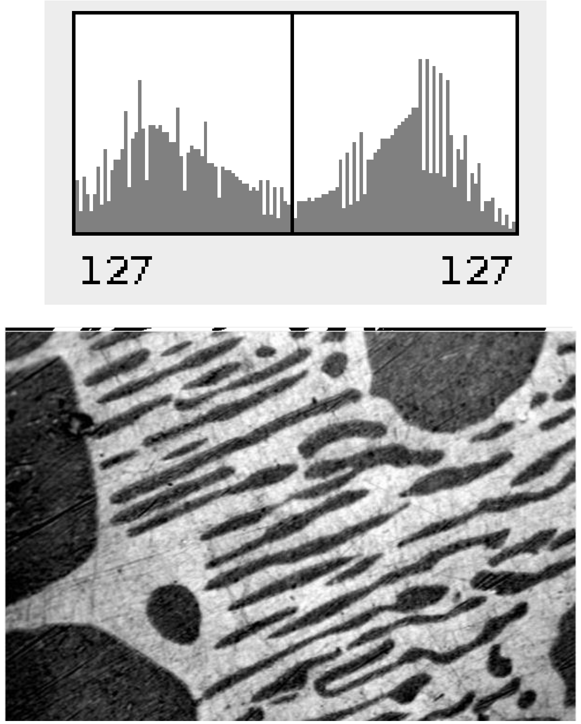
Histogram-based

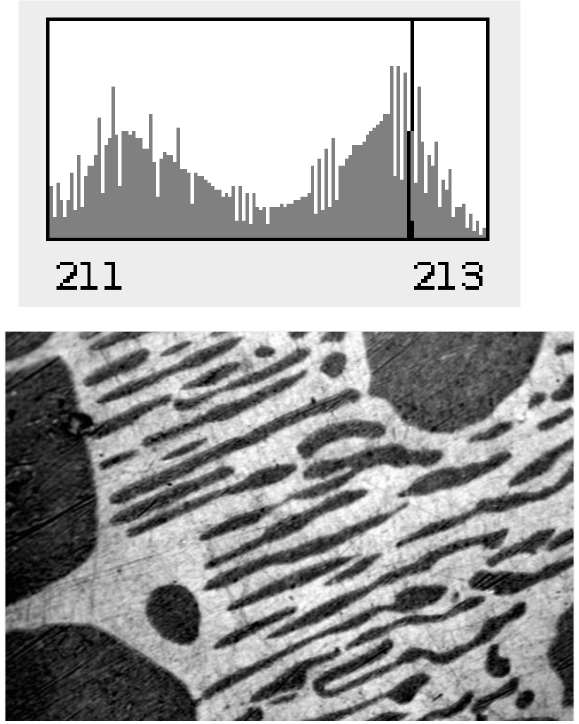
Thresholding

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

<u>Some thoughts:</u>

- 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 discribed 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.

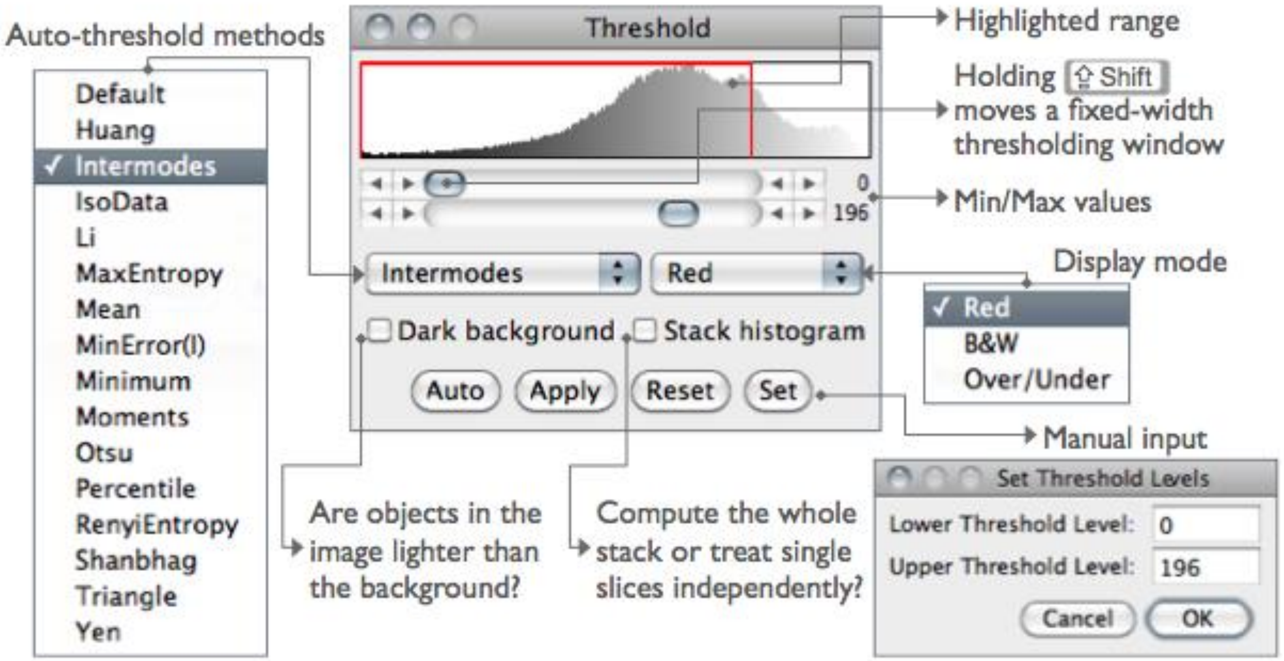
<u>Image processing to the rescue (see before):</u>

Gradient Mean filter with large kernel
Fireflies/hot pixels/dead pixels Bin your data, Median filter with a kernel as small as possible,
post thresholding: Morphological filters (open/close)
Touching objects: Watershed

GIGO

Auto – thresholding

Clustering
ISOdata
Otsu
Intermodes (assumes equal bimodal histogram)
Minimum
Mean (Mean of grayscale as threshold, initates ISOdata)
Percentile (assumes foreground pixels fraction of 0.5)
Yen
Entropy
Huang and Huang 2 (faster)
Shannon's entropy
Li
MaxEntropy
RenyiEntropy
Shanbhag
Metric
Triangle
Moments
Tsai



Unsupervised thresholding: clustering

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

The **standard deviation** within each cluster, and the distance between cluster centers is calculated

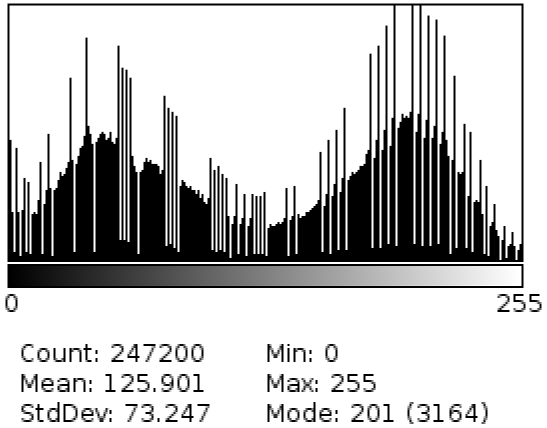
Clusters are re-arranged to minimize large standard deviations (outliers are swapped).

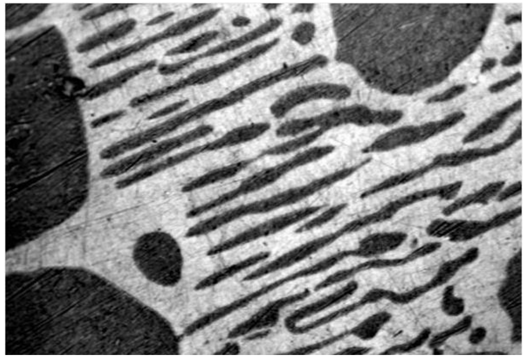
Iterate!

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

Ideal for bimodal histograms!
You do not have a bimodal histogram?
Use entropic thresholding





Better for non-bimodal histograms!

Unsupervised thresholding: entropy

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

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

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

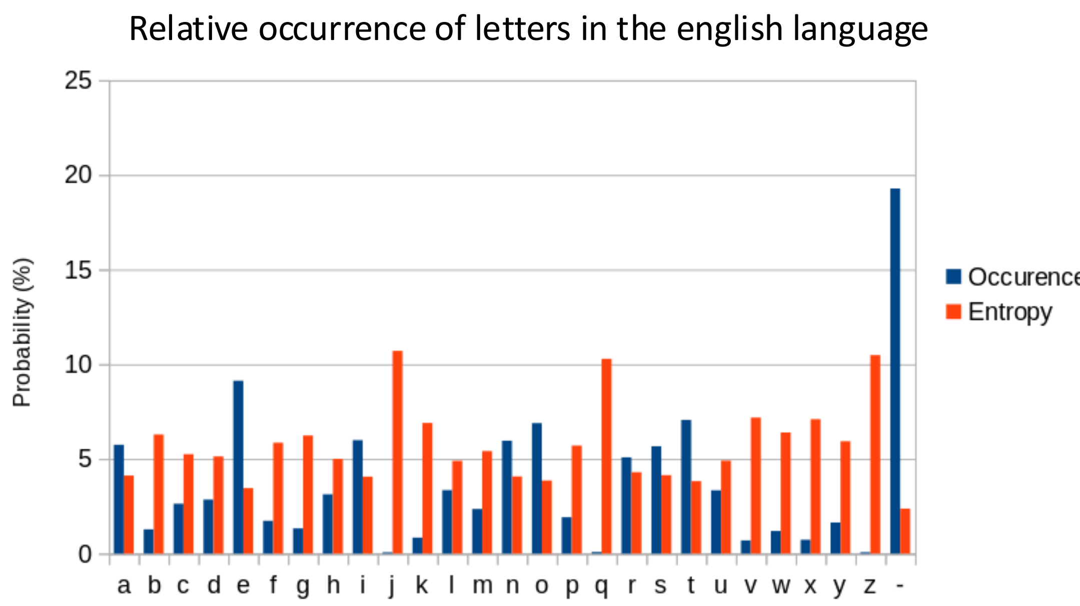
Iterate!

Clusters are re-arranged to minimize entropy (outliers are swapped).

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



Probabilities
Entropy
MaxEntropy
Huang (fuzzy)
Li
RenyiEntropy
Shanbhag

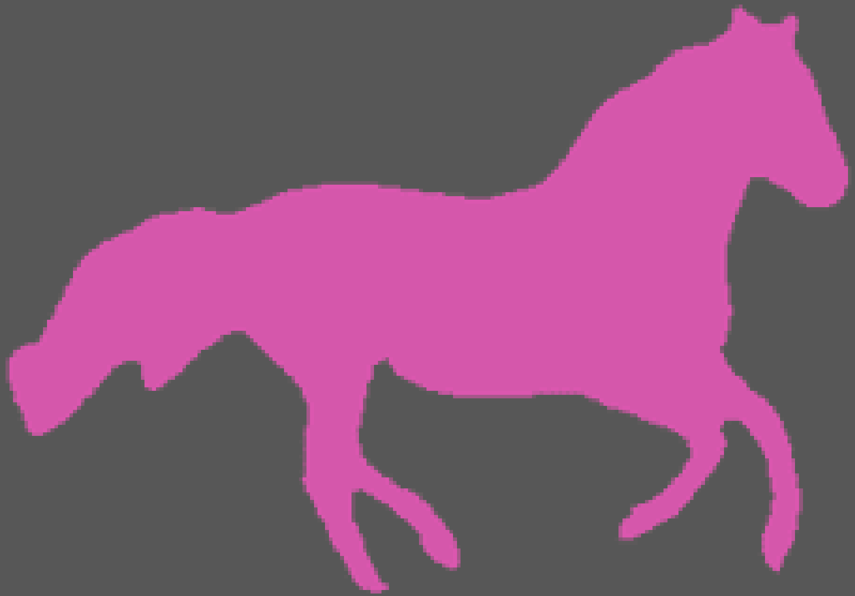
Thresholding algorithms

EXCERCISE
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

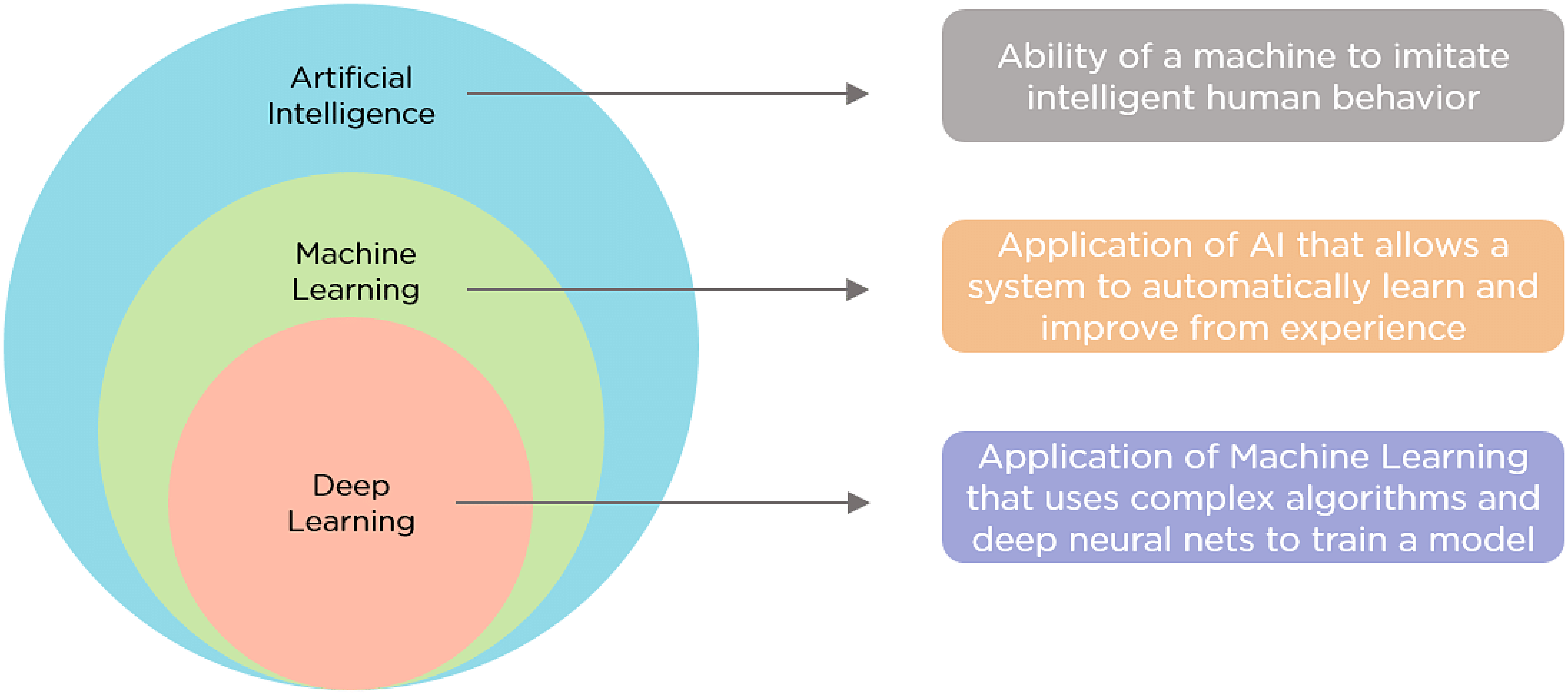
Thresholding, classification and segmentation

Level 2

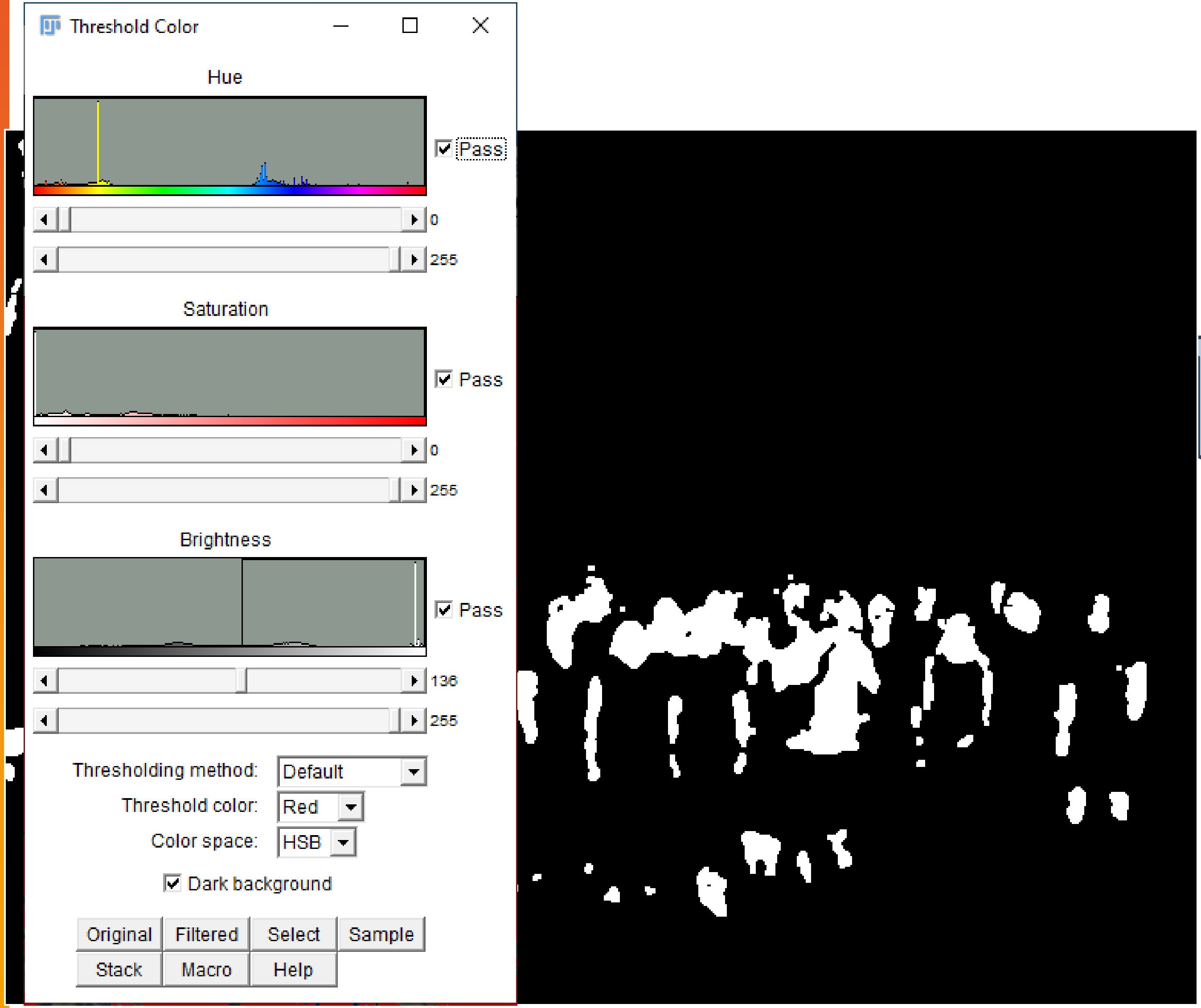


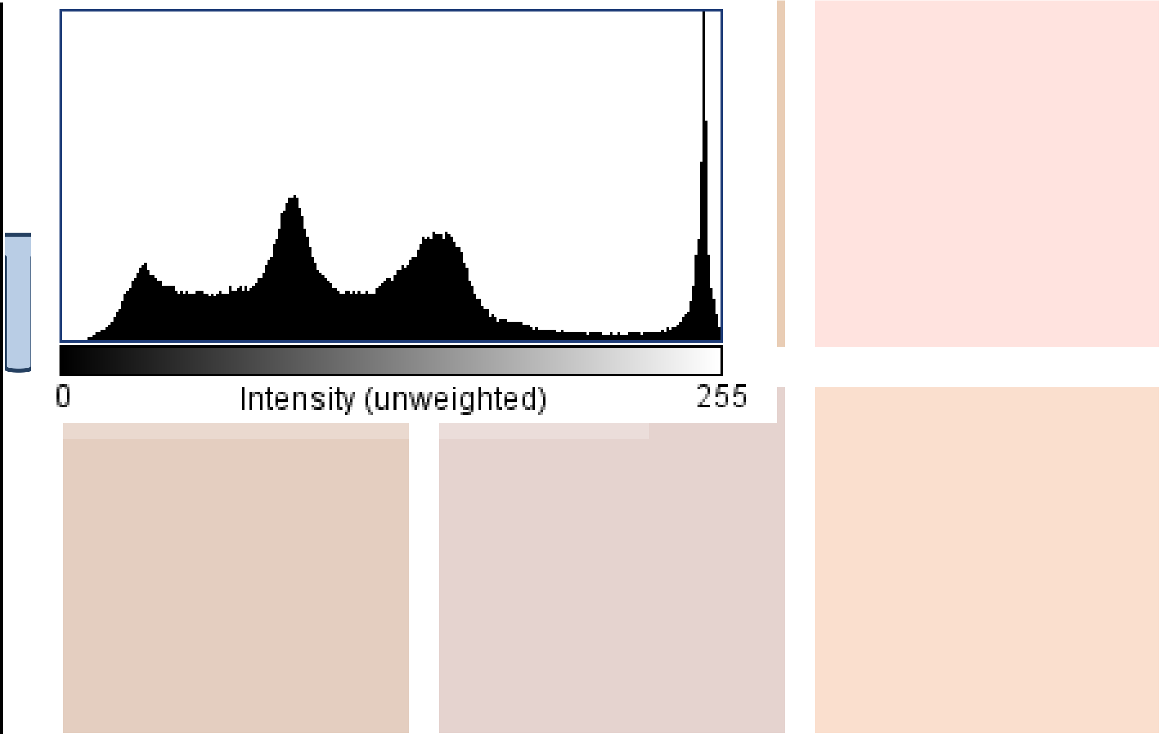
Machine learning

Thresholding: human vs machine

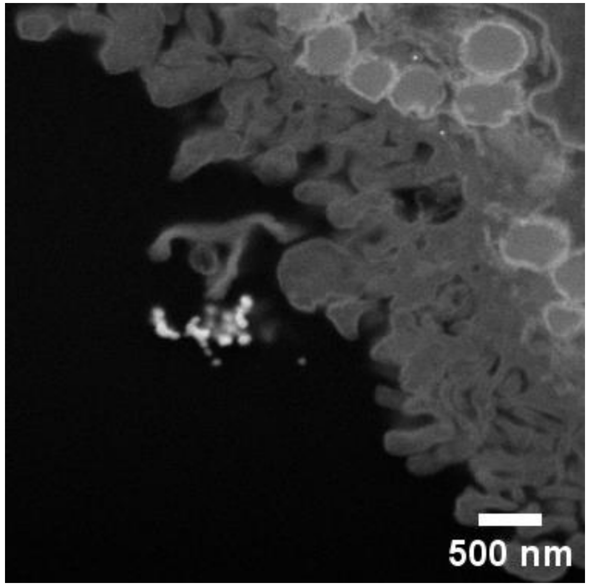


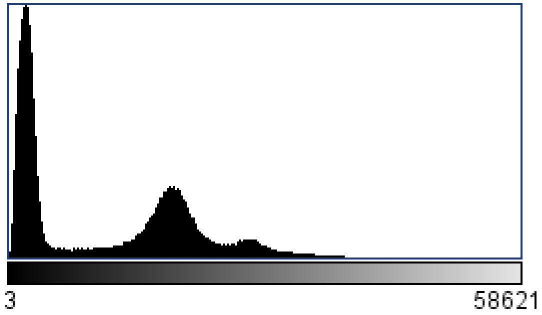
Thresholding: human vs machine

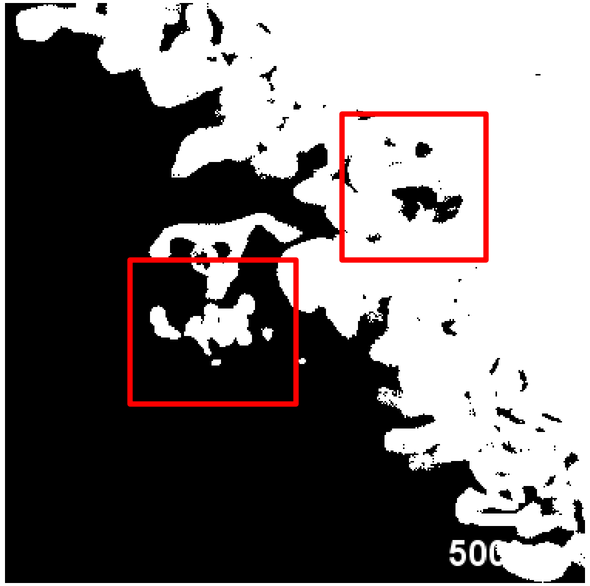


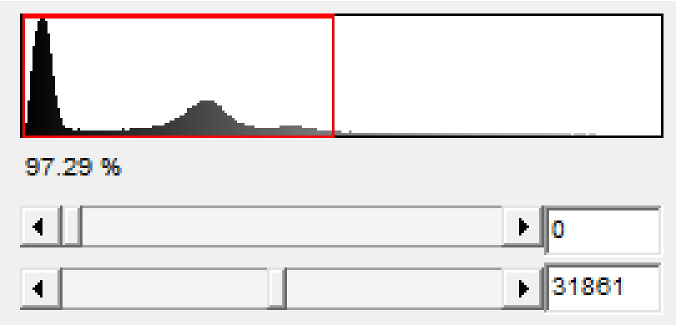


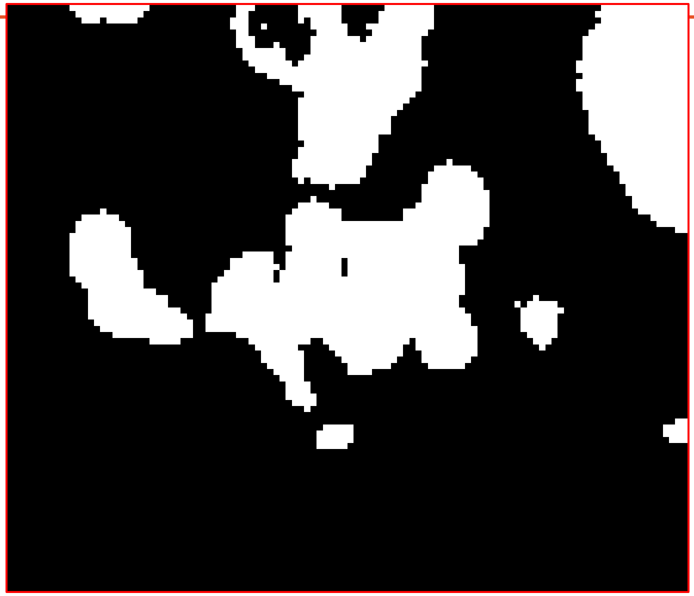
From thresholding to classification









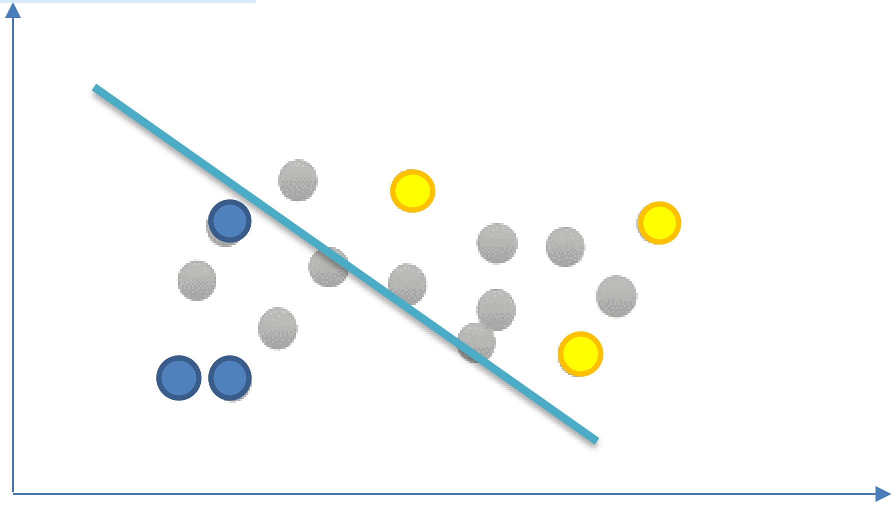


FIB Data by Henry Lee

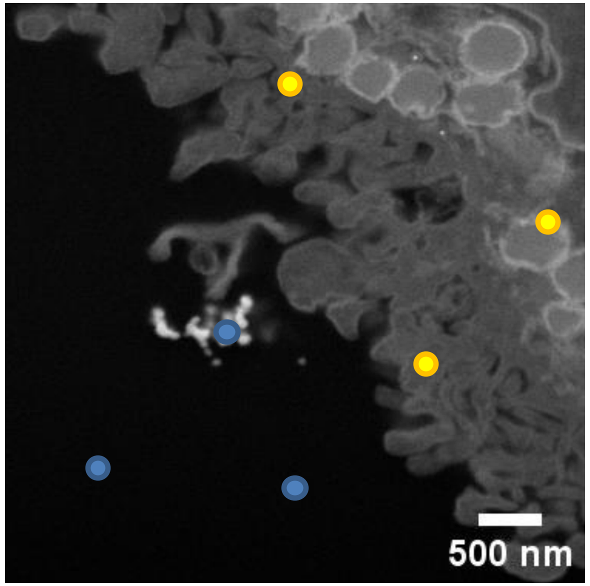
From thresholding to classification

	σ_0	σ_1	σ_2	σ_3	σ_4	σ_5	σ_6
Sigma	0.30	0.70	1.00	1.60	3.50	5.00	10.00
▼ Color/Intensity	✓	✓	✓	✓	✓	✓	✓

Intensity



Intensity in the neighbourhood (s = 1.00)
(a.k.a. Context)



Random forest classification
(theoretical example)

Is the pixel white?

Particle!

Is the neighbour pixel white?

Particle!

Is the pixel far from a strong edge?

Probably background

Is the texture smooth?

Classification features

- Color/Intensity
- Texture
- Edginess
- Distance to a local edge
- Isotropy
- Curvature

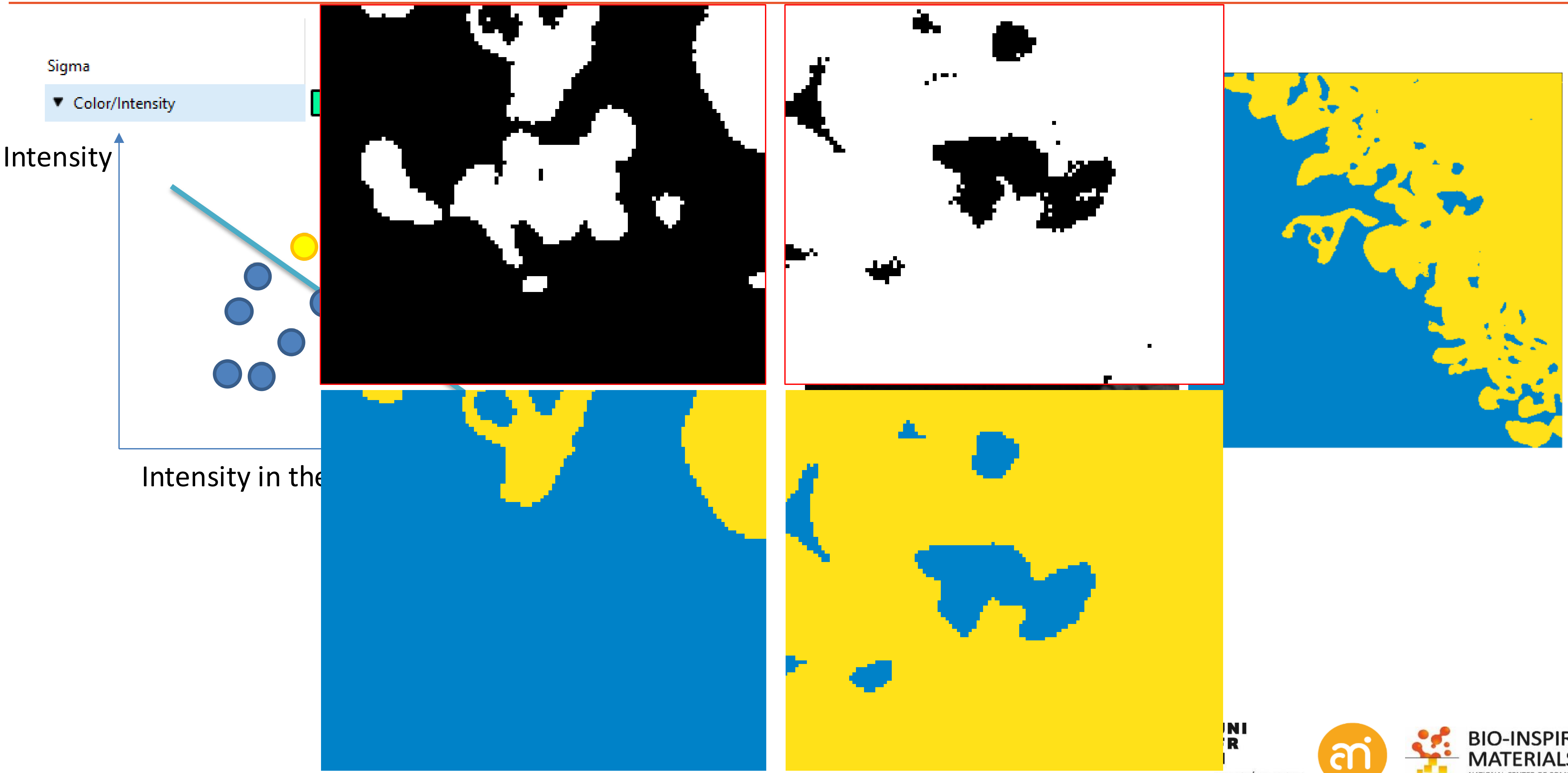
▼ Color/Intensity
Gaussian Smoothing
▼ Edge
Laplacian of Gaussian
Gaussian Gradient Magnitude
Difference of Gaussians
▼ Texture
Structure Tensor Eigenvalues
Hessian of Gaussian Eigenvalues

Statistical classification methods

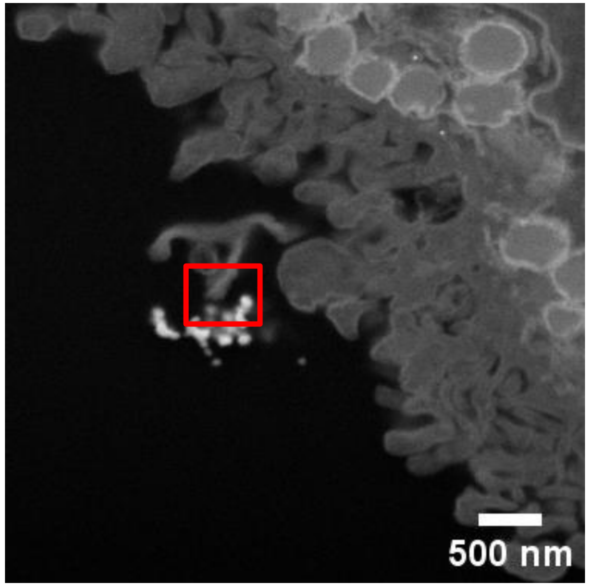
- Artificial neural networks
- Decision tree learning e.g. Random forest
- Kernel estimation e.g. k-nearest neighbour
- Linear classifier e.g. Bayes classifier
- Least squares support vector machine

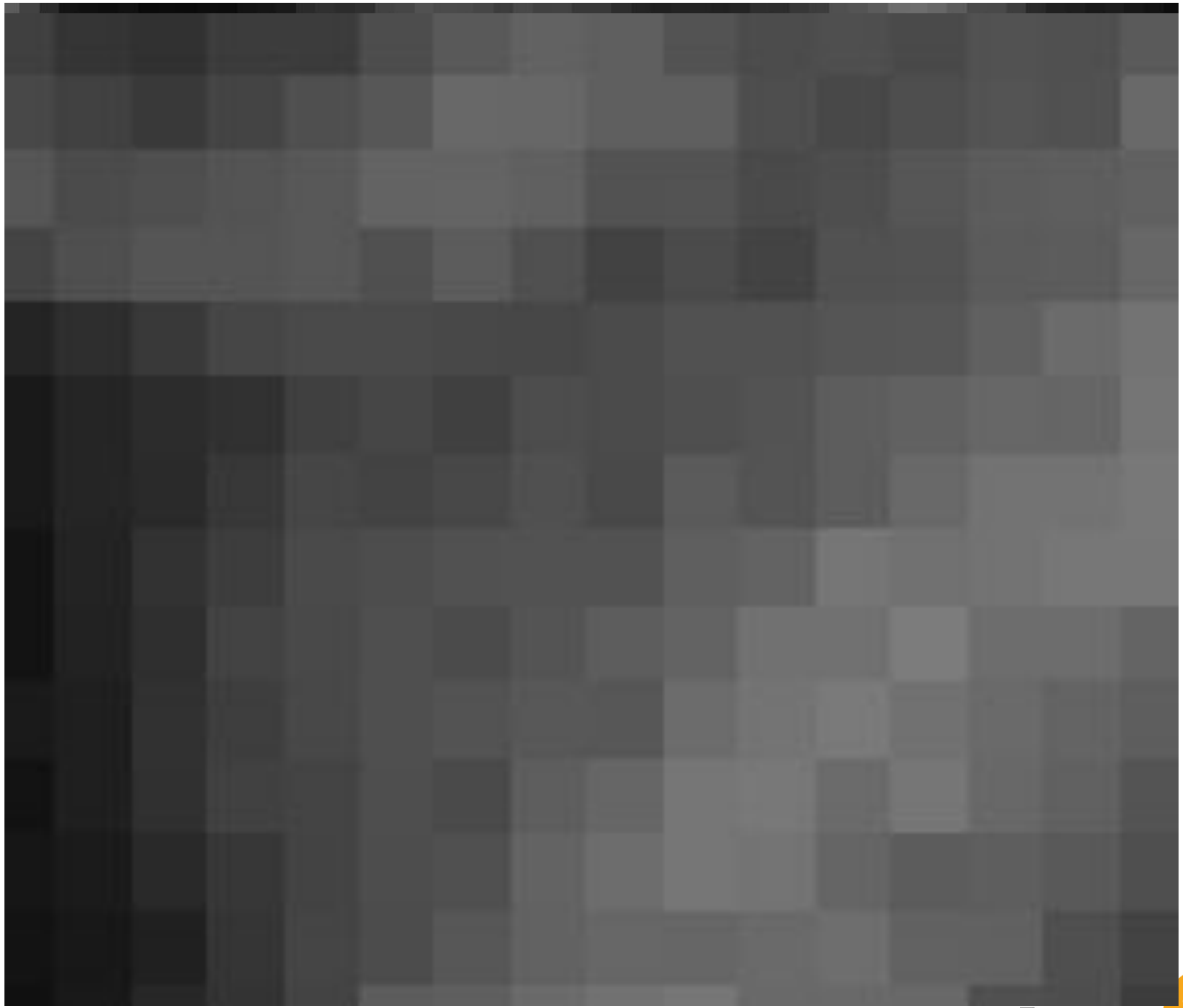
… And many many more

From thresholding to classification



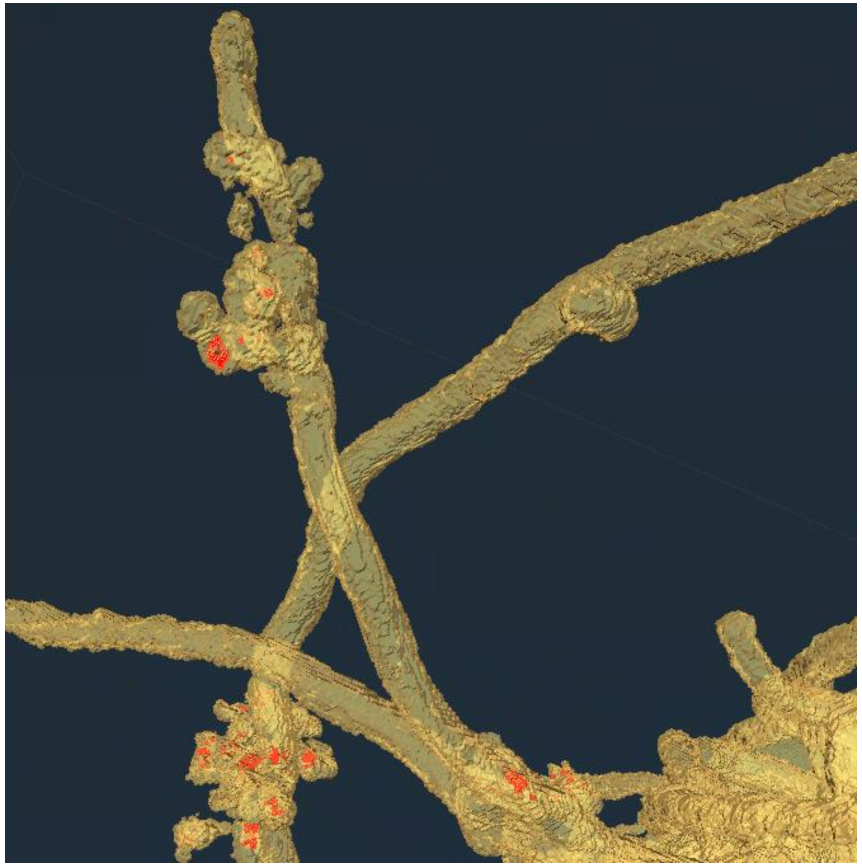
From thresholding to classification





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



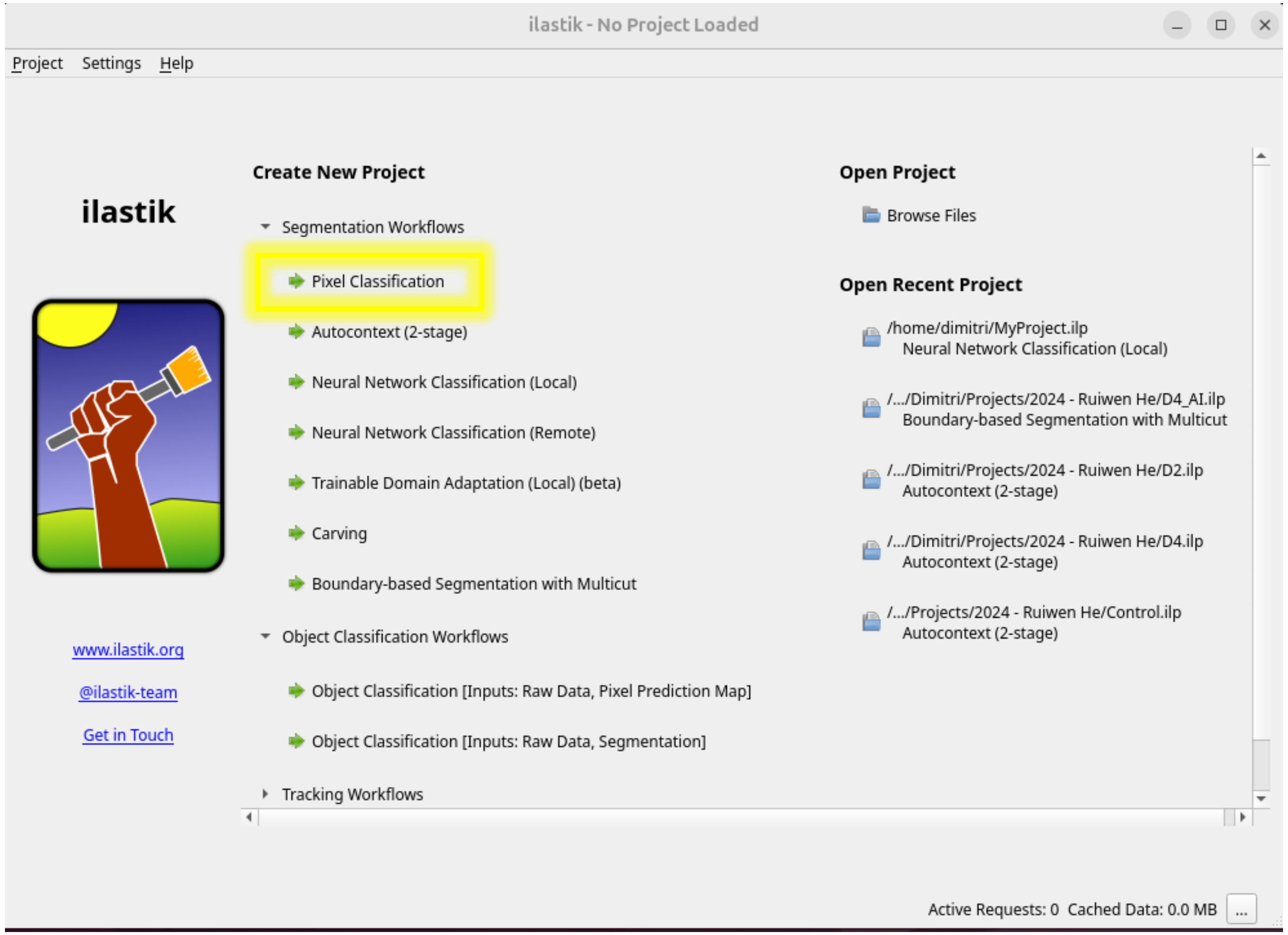
Cell volume:	1871 um^3
NP inside volume:	25.82 um^3
NP outside volume:	0.7842 um^3

(assuming spheres with a diameter of 50 nm)

Number of NP inside the cell	387815
NP per volume cell:	207 NP / um^3 cell

iLastik

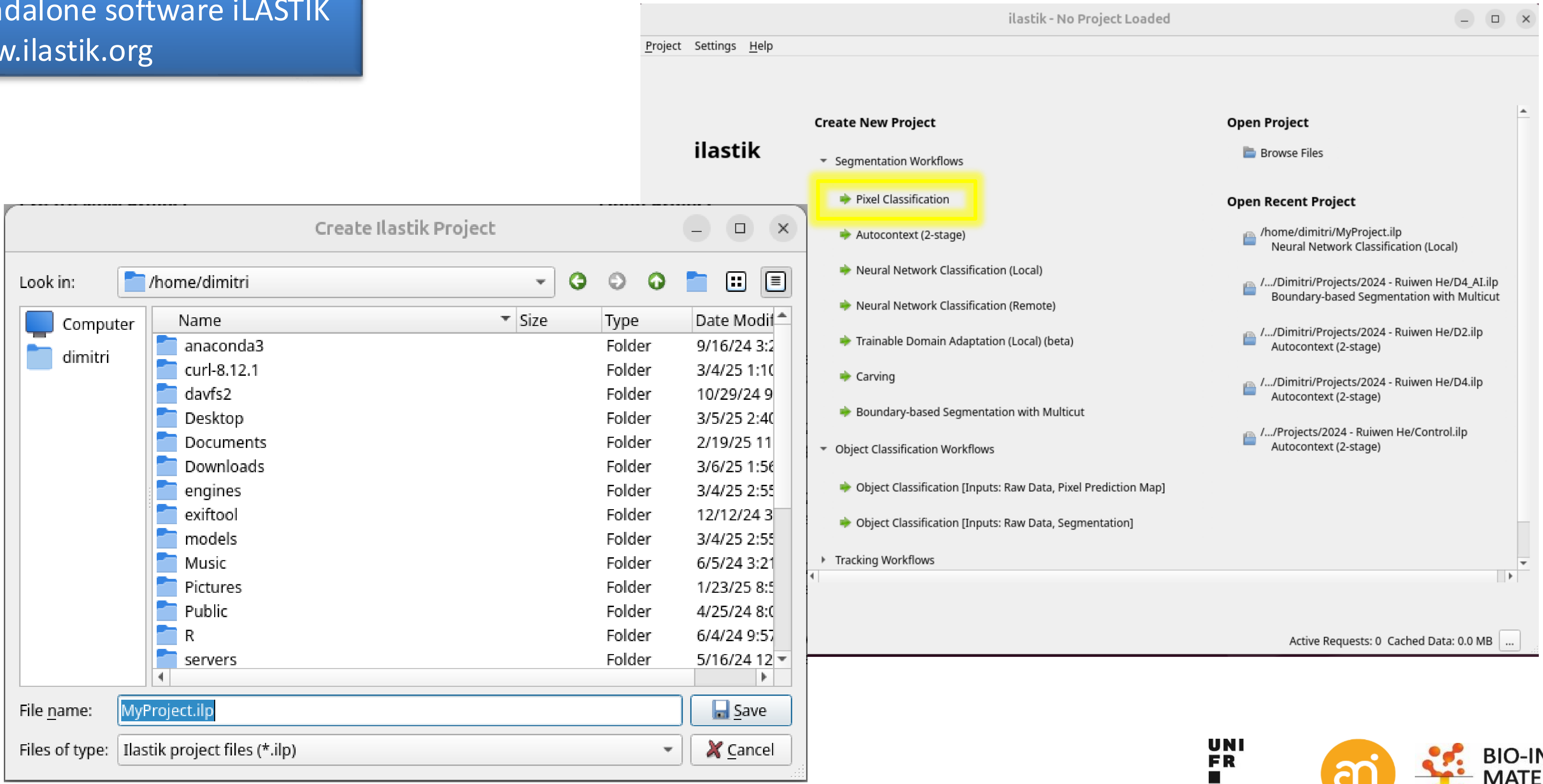
Standalone software iLASTIK
www.ilastik.org



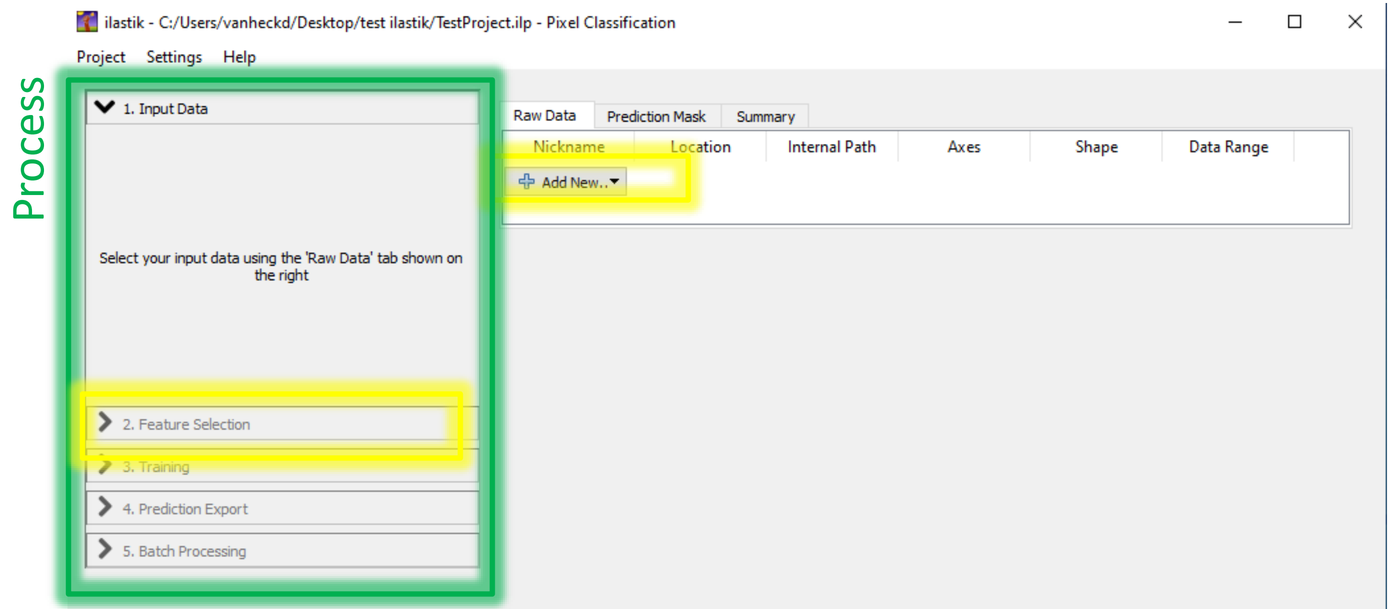
UNI FR
UNIVERSITÉ DE FRIBOURG
UNIVERSITÄT FREIBURG
BIO-INSPIRED MATERIALS
NATIONAL CENTER OF COMPETENCE IN RESEARCH

iLastik

Standalone software iLASTIK
www.ilastik.org



iLastik: 1. Input data



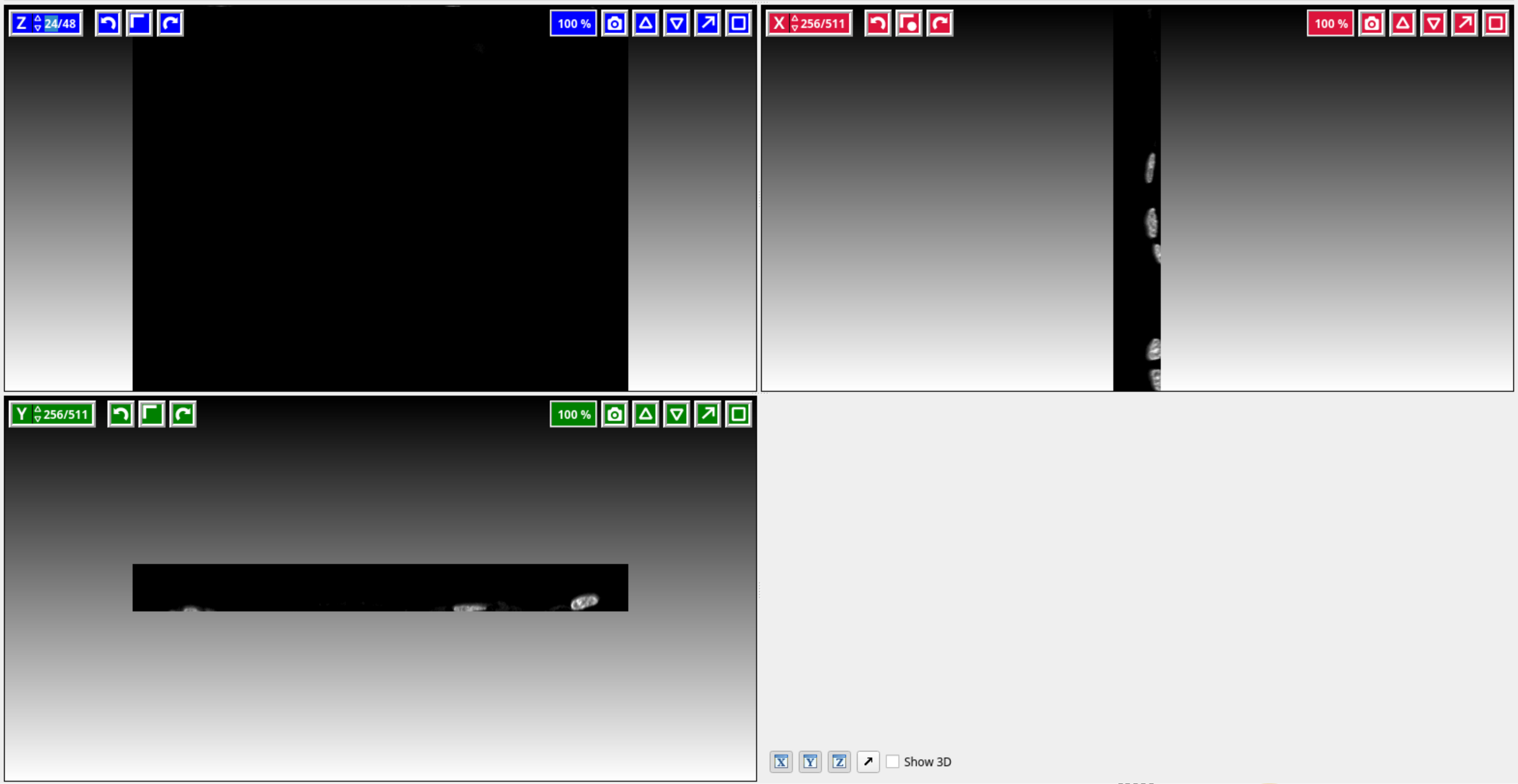
1. Add new > Add separate images… >
    - Example4-stack.h5 (for a fast PC)
    - Example4-single.h5 (for a normal PC)

2. Click in the left process menu Feature selection

make sure your training images have

- Grayscale LUT
- No scale

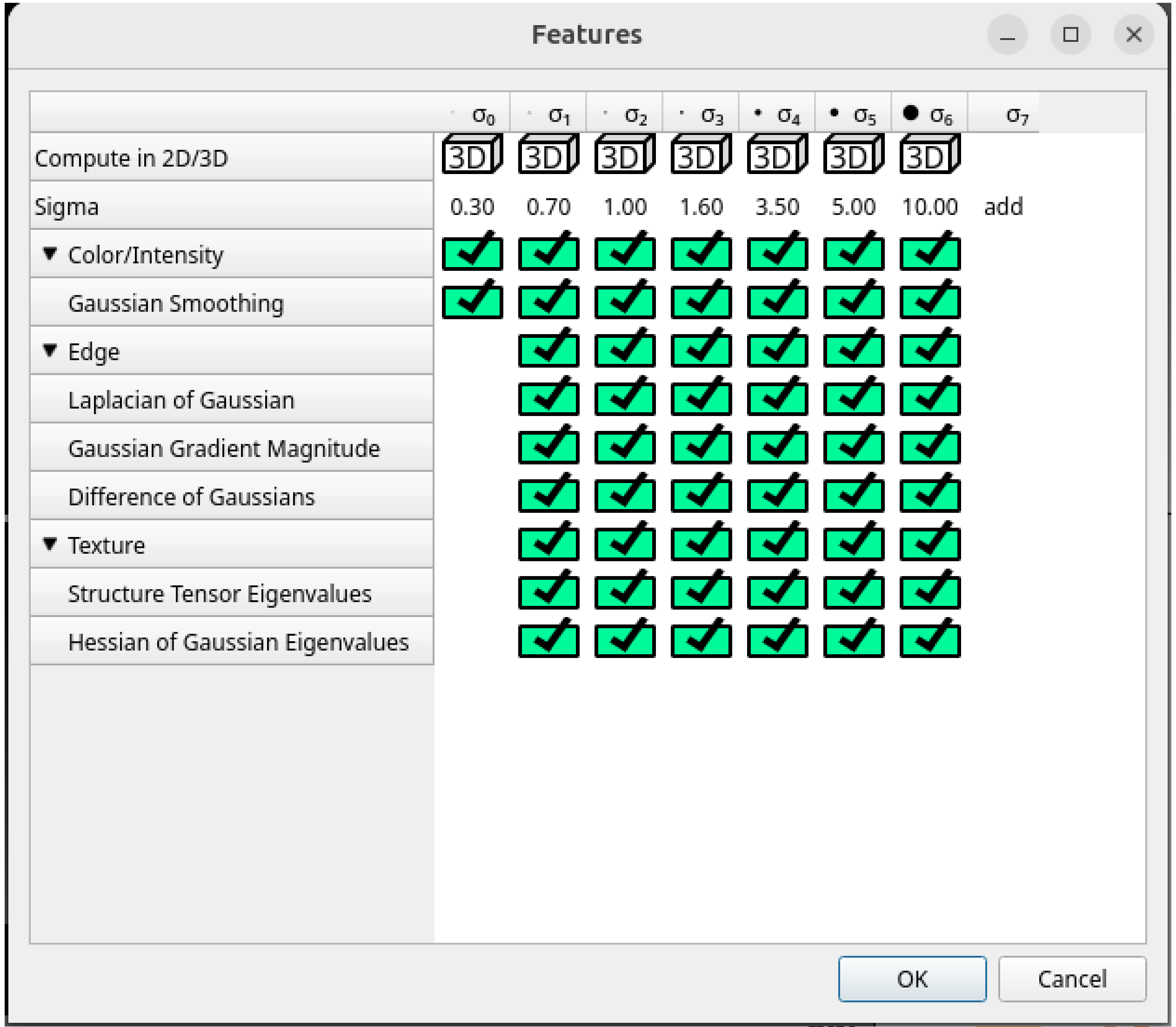
iLastik: 1. Input data



iLastik: 2 Feature selection

Process

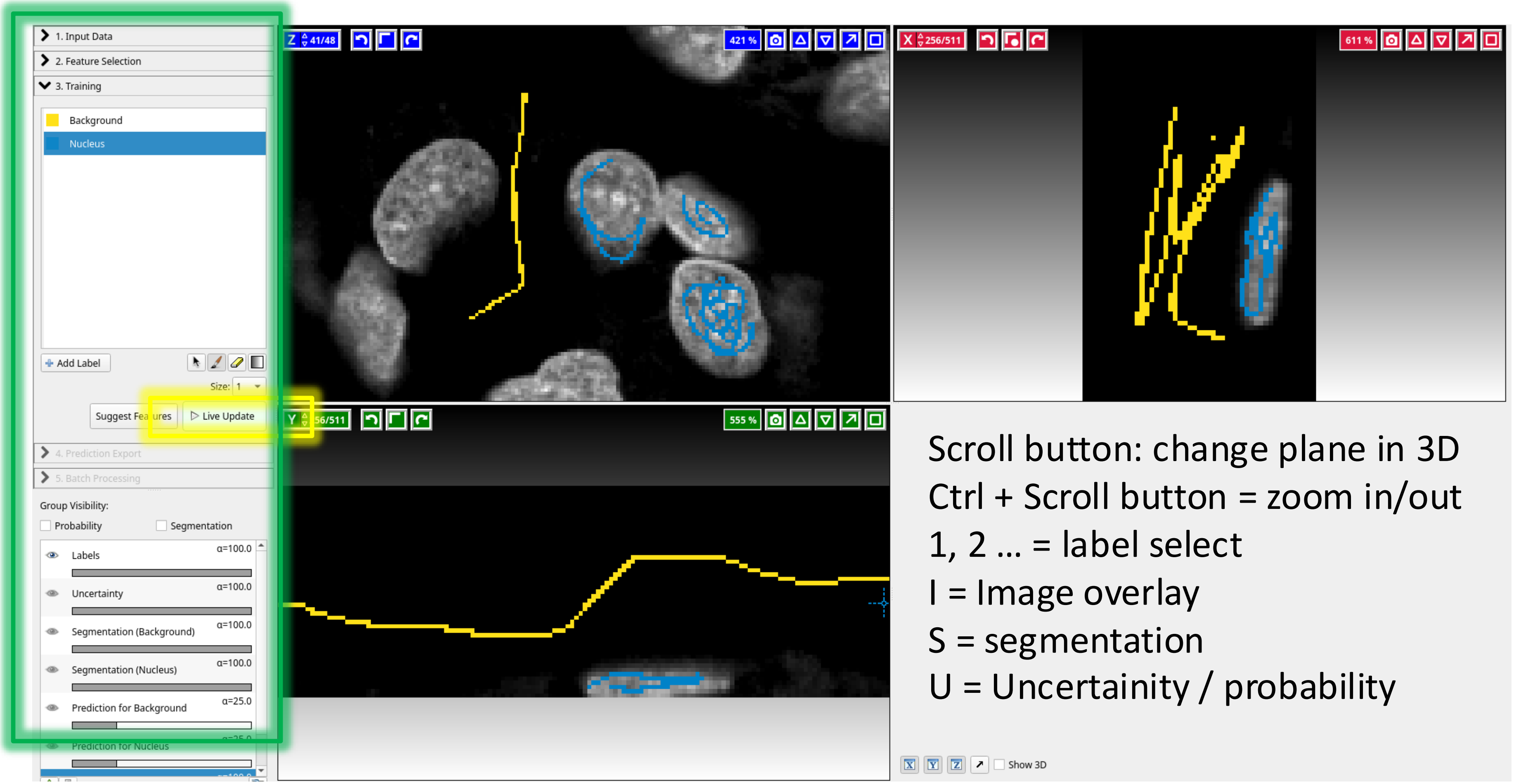
Select features…
(select all) > click OK



Click 3. Training

iLastik: 3. Training the machine

Process



Scroll button: change plane in 3D

Ctrl + Scroll button = zoom in/out

1, 2 … = label select

I = Image overlay

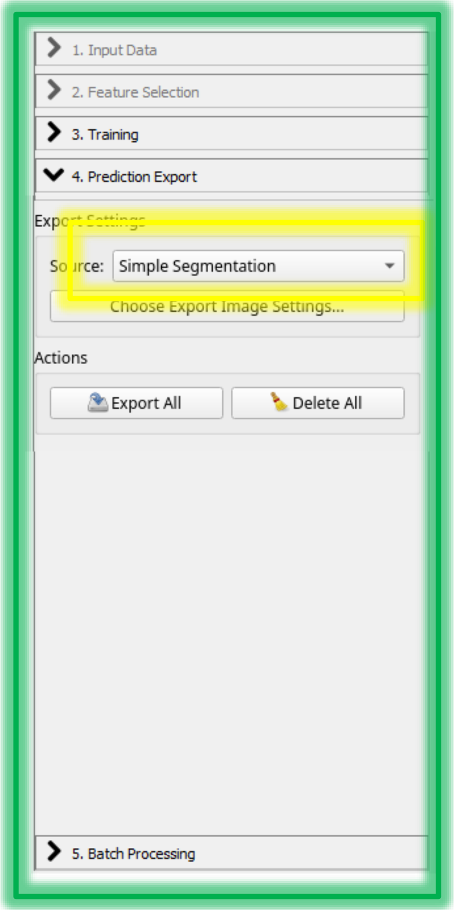
S = segmentation

U = Uncertainity / probability

Iterate and improve your result

iLastik: 4. Save the data

Process



Single Image:

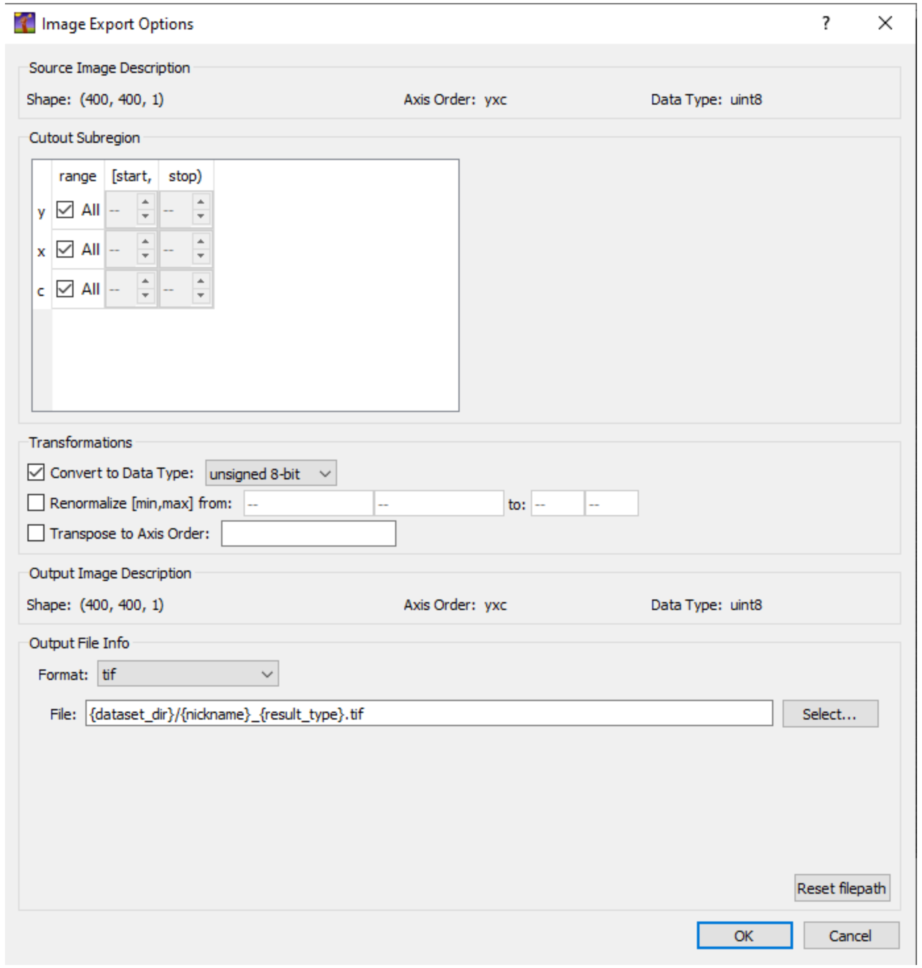
Tick: convert to unsigned 8-bit
Format: tif
Click OK

Stack:

Tick: convert to unsigned 8-bit
Format: HDF
Click OK

----- OR -----

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

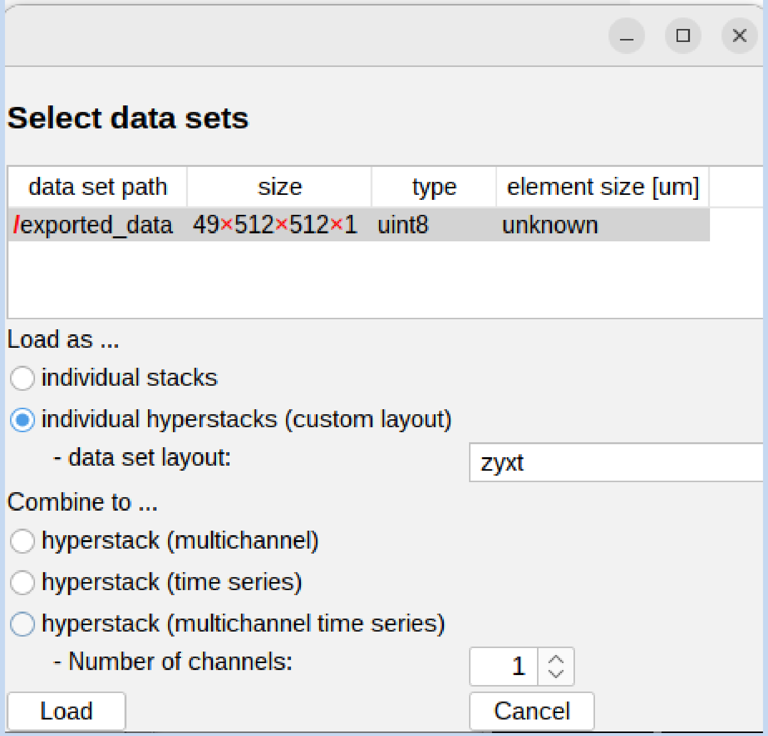


iLastik: 4. Save the data

Stack saved as HDF

File > Import > HDF5…

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

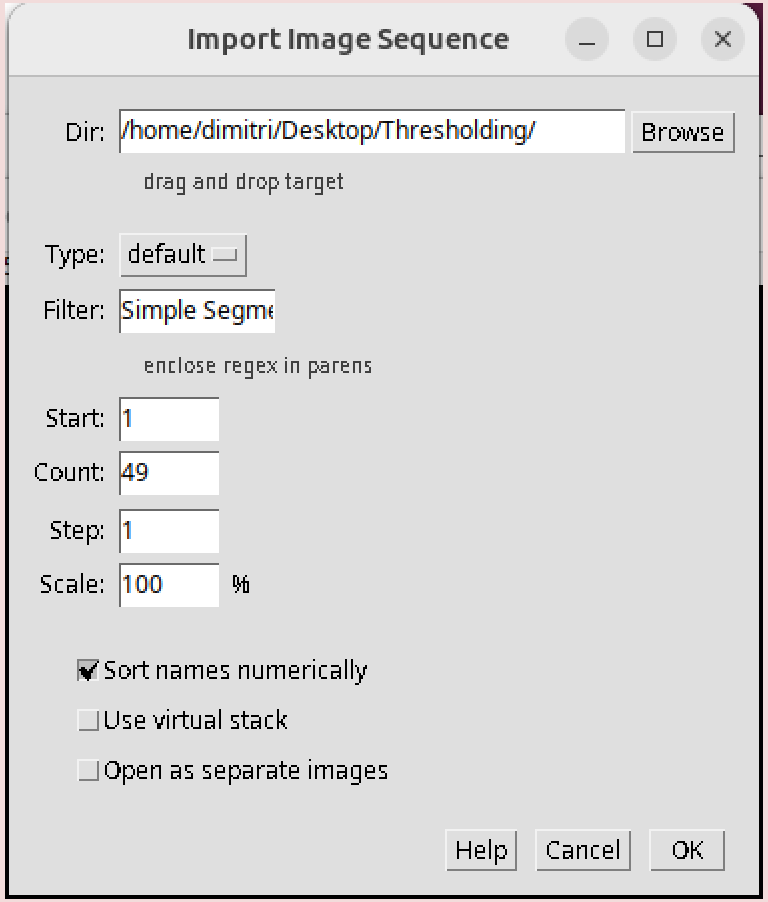


Stack saved as tif sequence

File > Import > Image sequence

Filter: “Simple Segmentation”

(➔ “Count” should then become 49)



Single saved as tif

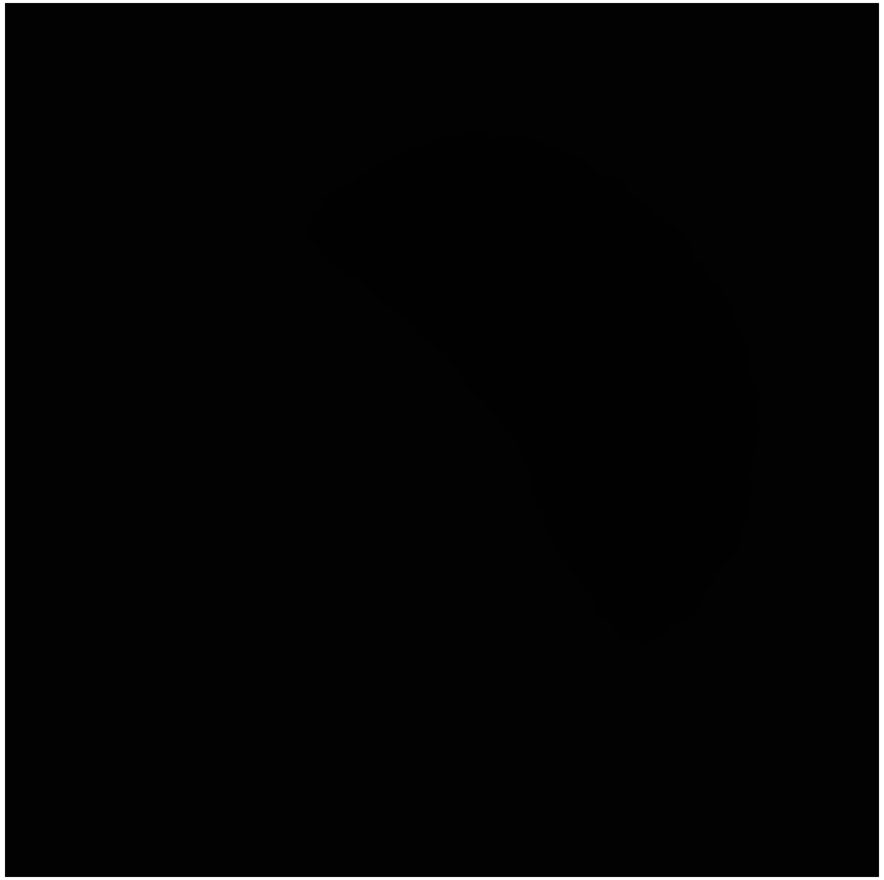
File > Open…

UNI FR UNIVERSITÉ DE FRIBOURG UNIVERSITÄT FREIBURG

BIO-INSPIRED MATERIALS NATIONAL CENTER OF COMPETENCE IN RESEARCH

Ilastik output

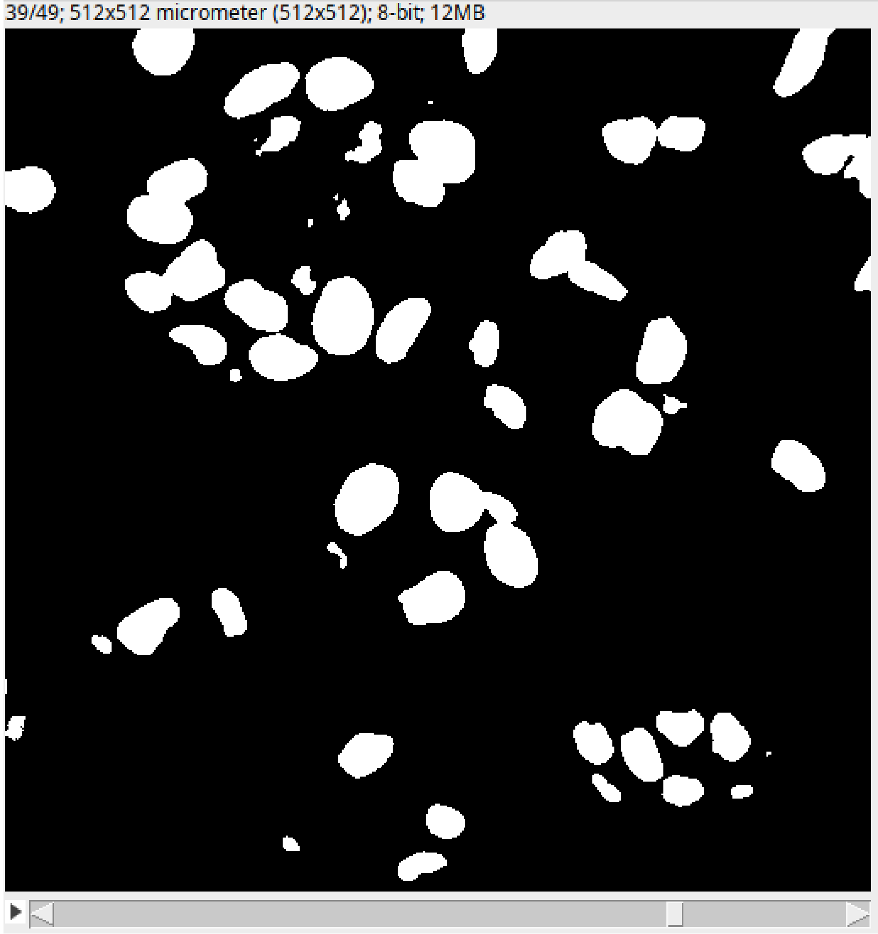
EXCERCISE
Open the segmentation result. Find out why it is black and what you can do about it.

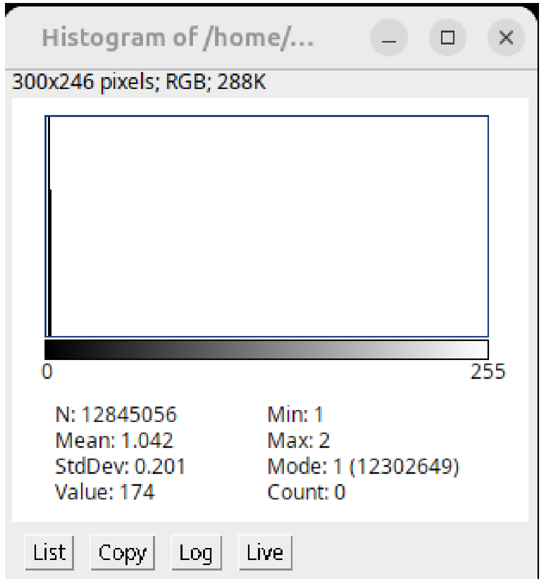


Ilastik output

EXCERCISE

Open the segmentation result. Find out why it is black and what you can do about it.

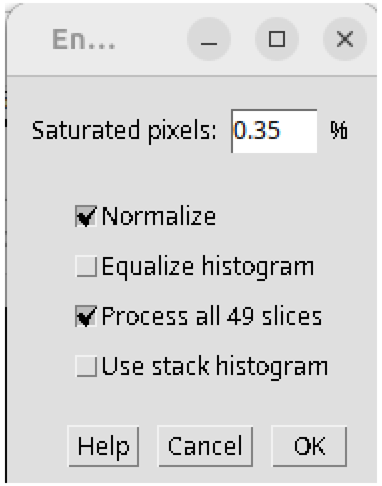




Normalize

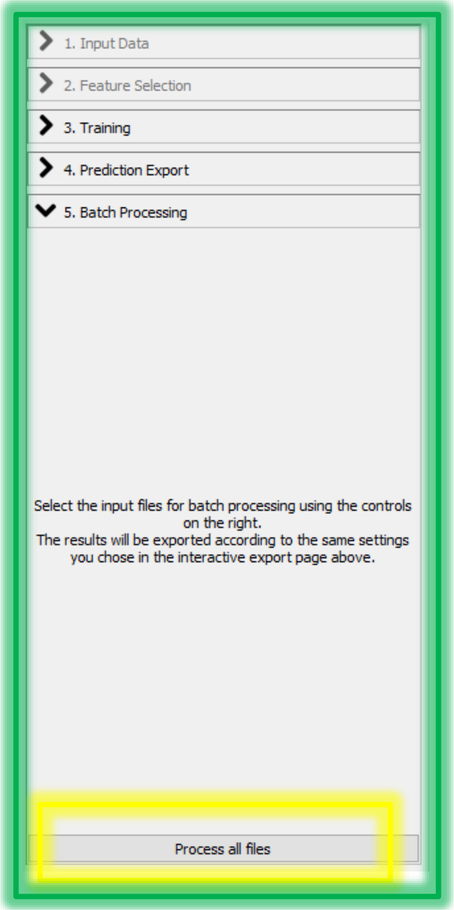
Process >

enhance contrast



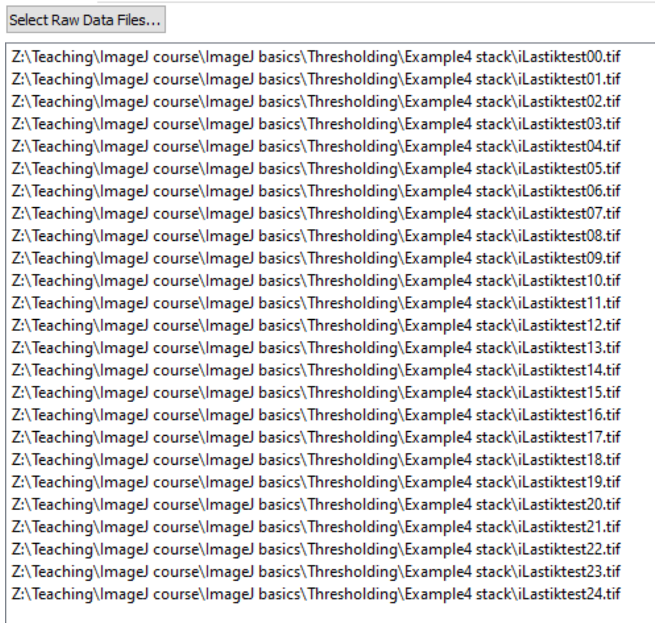
iLastik: 5. Batch processing

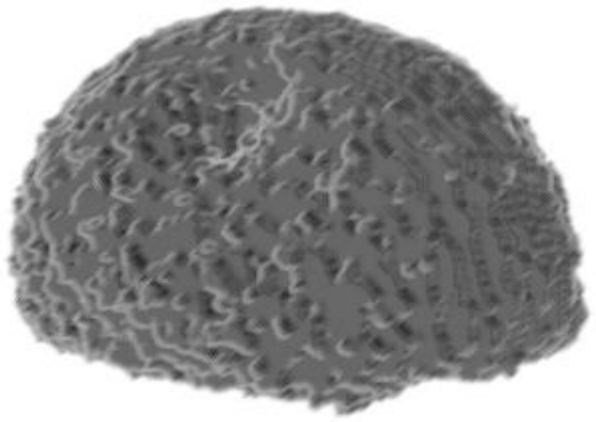
Process



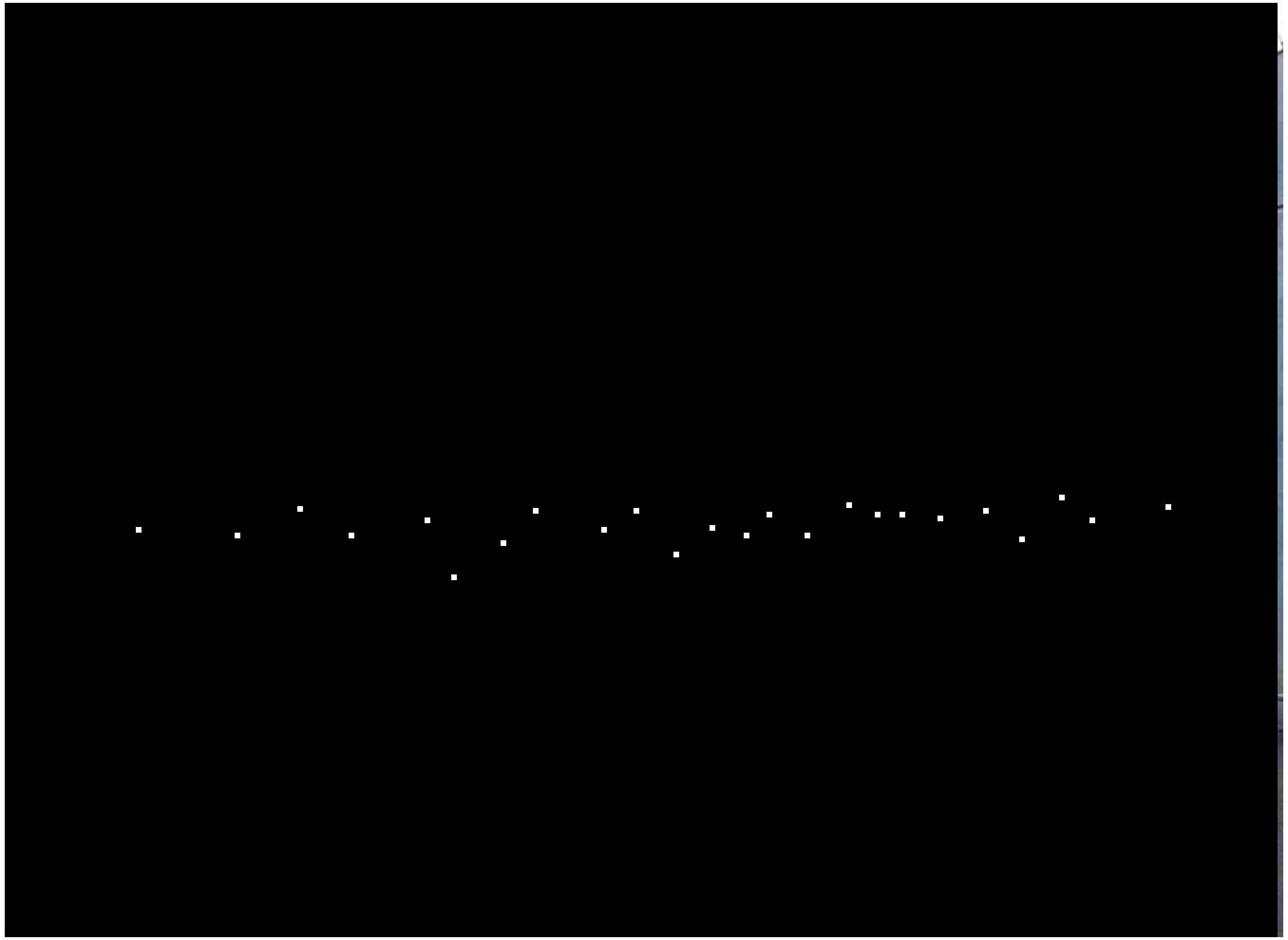
Select raw datafiles …

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





iLastik

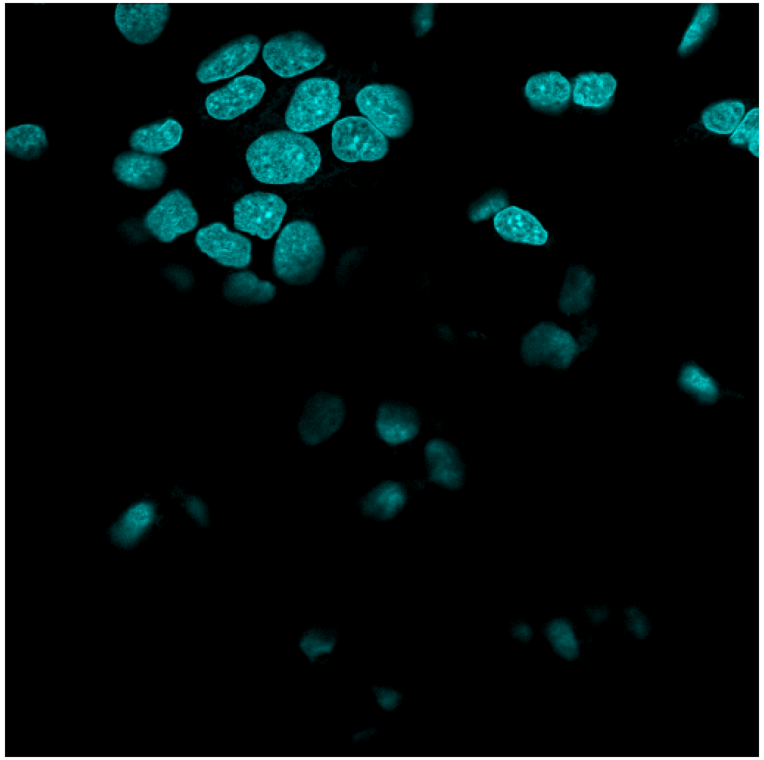


Value	# of Pixels
255	24

Labkit: example and hands-on workflow

EXCERCISE

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

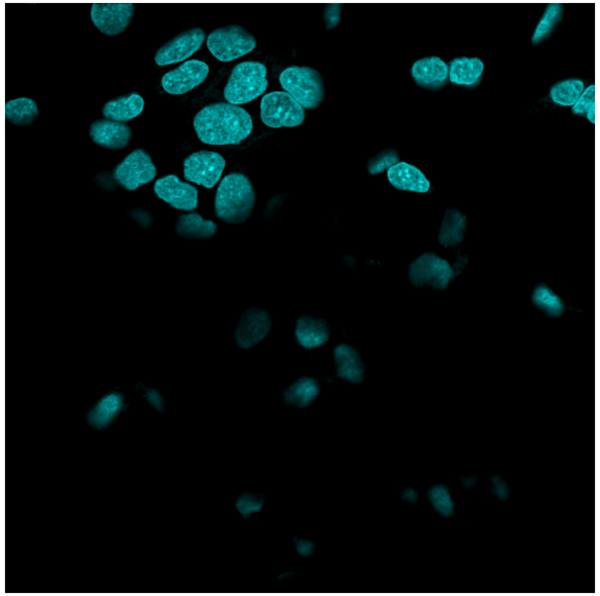


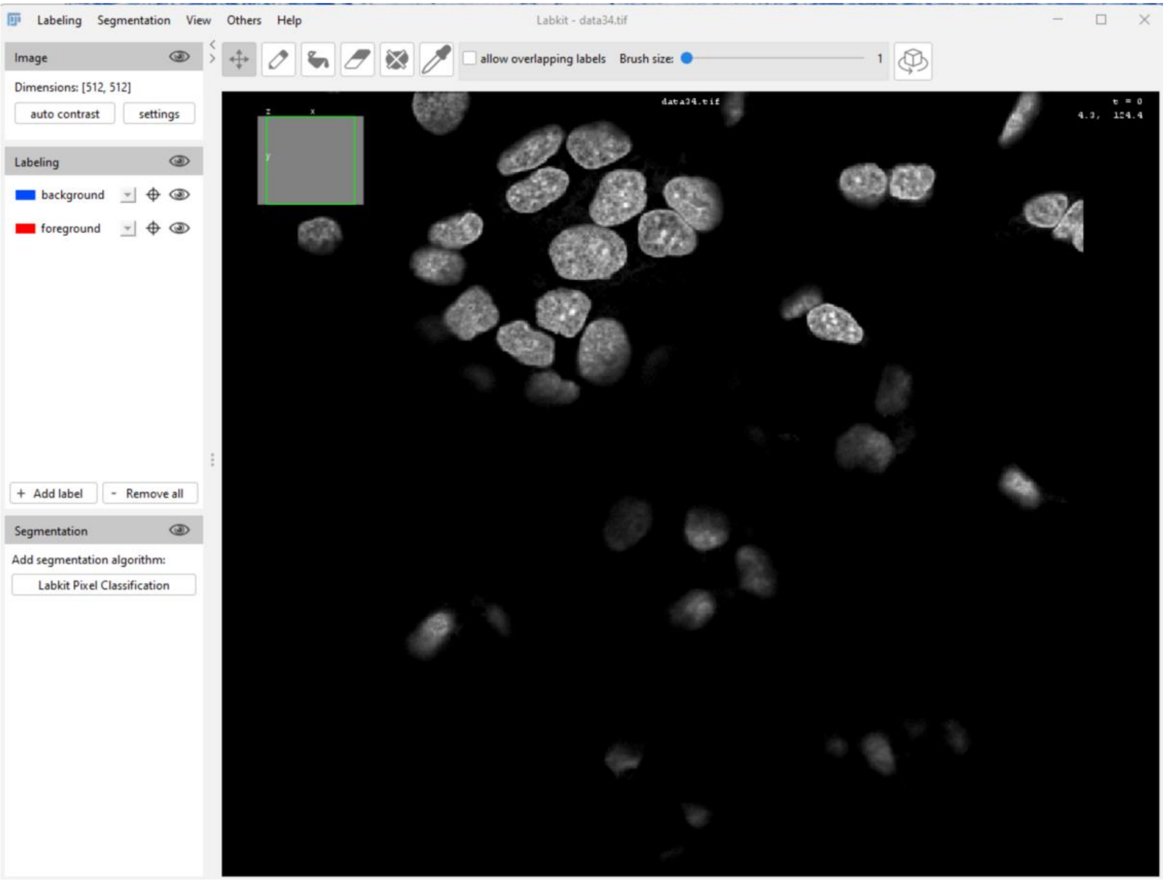
Labkit: example and hands-on workflow

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 https://sites.imagej.net/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

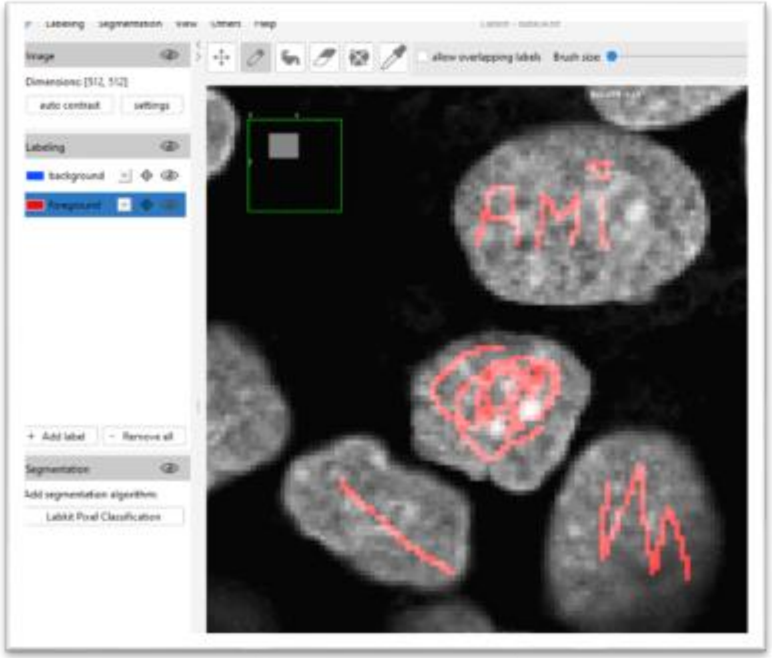
EXCERCISE

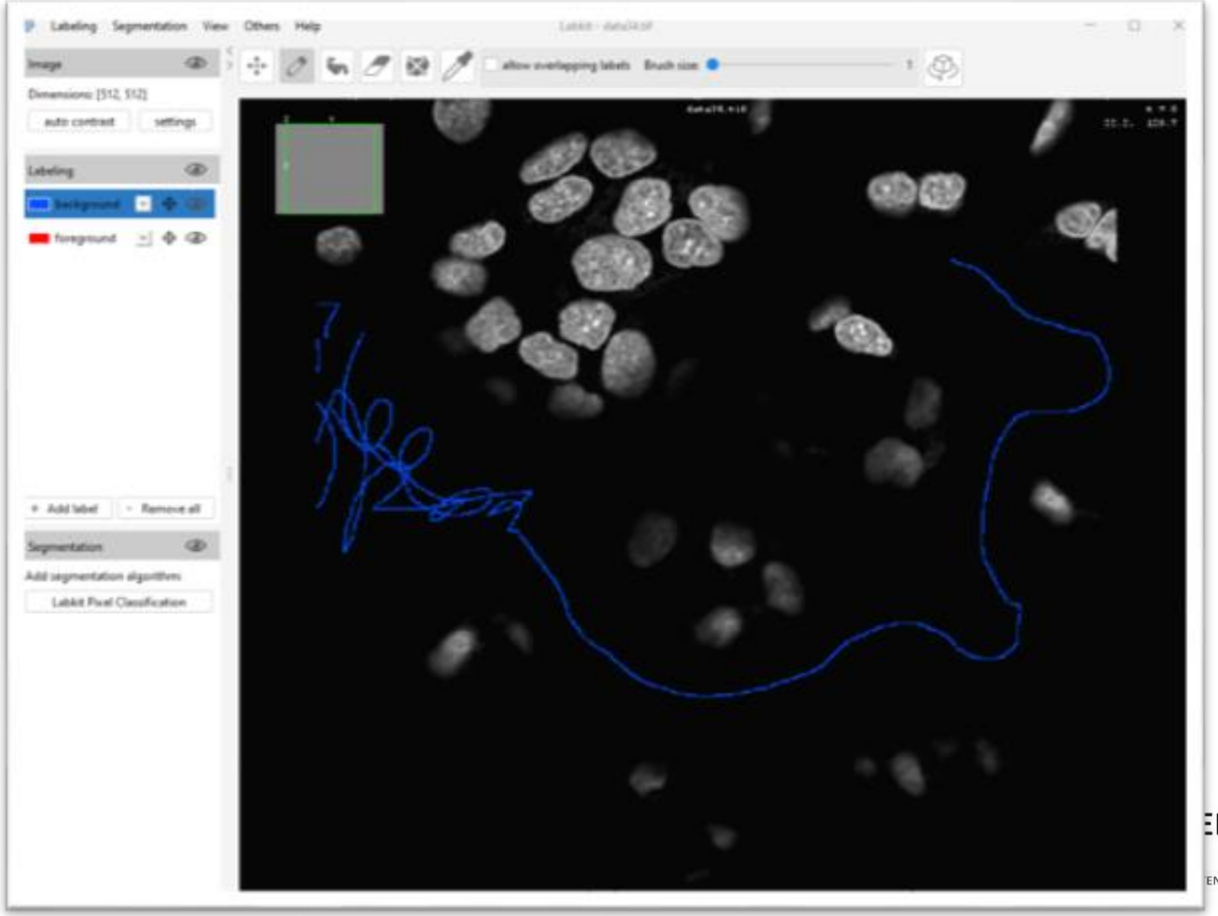
Open Example 4. Duplicate 1 image of the stack. Train a model using LabKit

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 https://sites.imagej.net/Labkit/





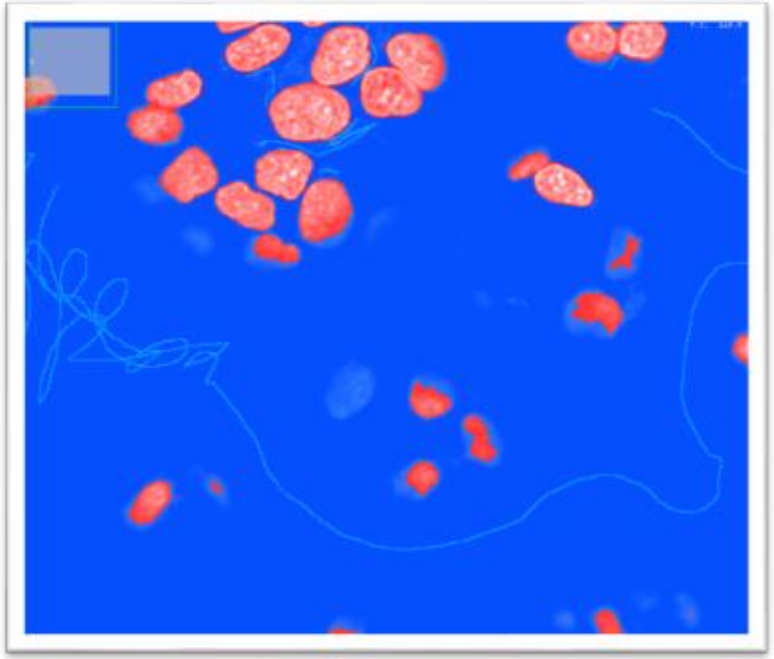
Labkit: example and hands-on workflow

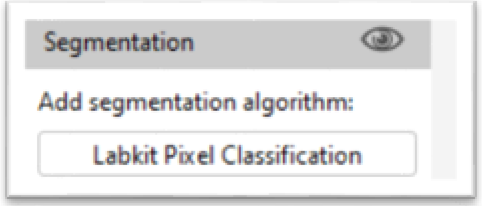
EXCERCISE

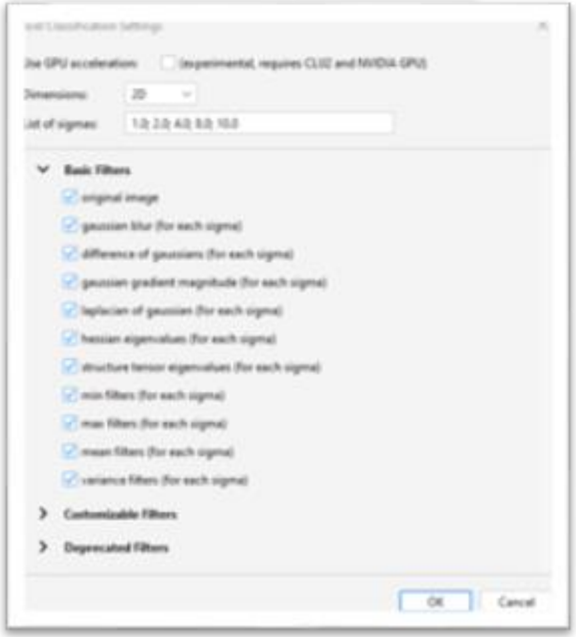
Open Example 4. Duplicate 1 image of the stack. Train a model using LabKit

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







UNI FR
UNIVERSITÉ DE FRIBOURG
UNIVERSITÄT FREIBURG

BIO-INSPIRED MATERIALS
NATIONAL CENTER OF COMPETENCE IN RESEARCH

Labkit: example and hands-on workflow

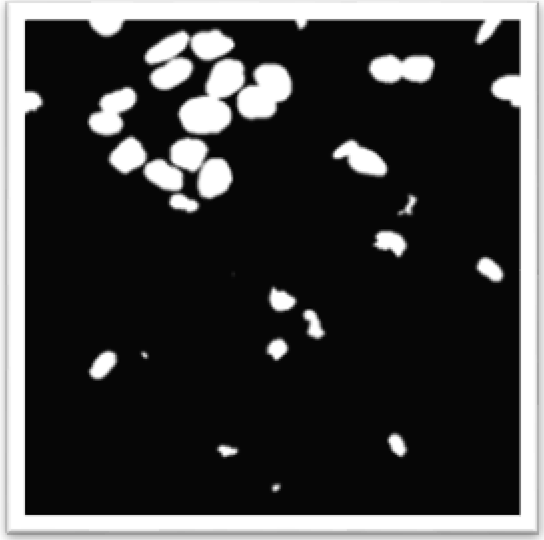
EXCERCISE

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

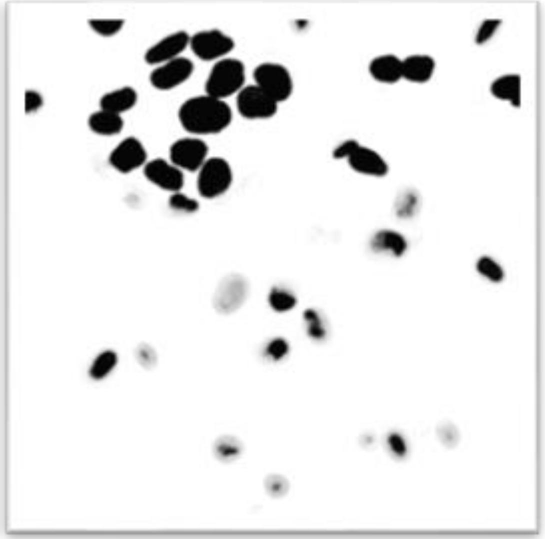
3. Check model uncertainity and segmentation

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

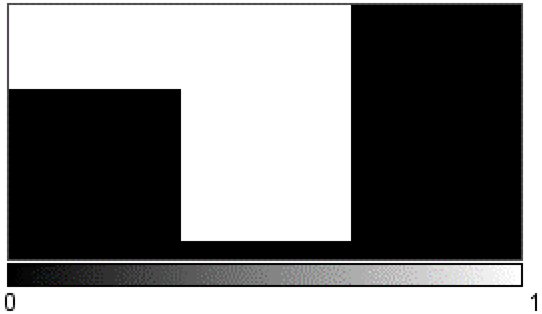
Segmentation



Probability



3 bin histogram



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

UNI FR UNIVERSITÉ DE FRIBOURG UNIVERSITÄT FREIBURG

BIO-INSPIRED MATERIALS NATIONAL CENTER OF COMPETENCE IN RESEARCH

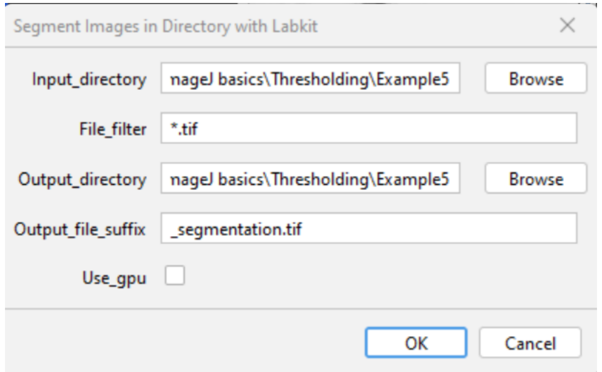
Labkit: example and hands-on workflow

EXCERCISE

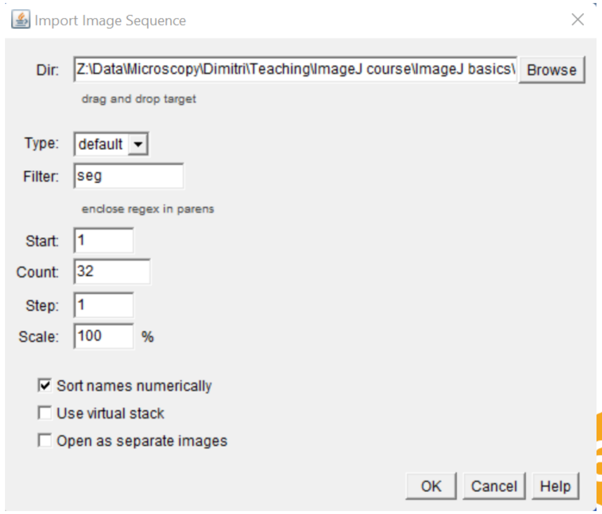
Open Example 4. Duplicate 1 image of the stack. Train a model using LabKit

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!

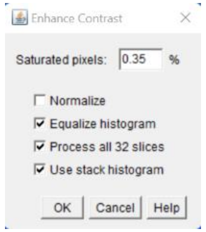


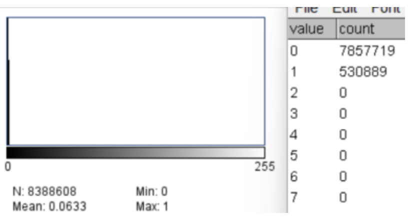
Labkit: example and hands-on workflow

EXCERCISE

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

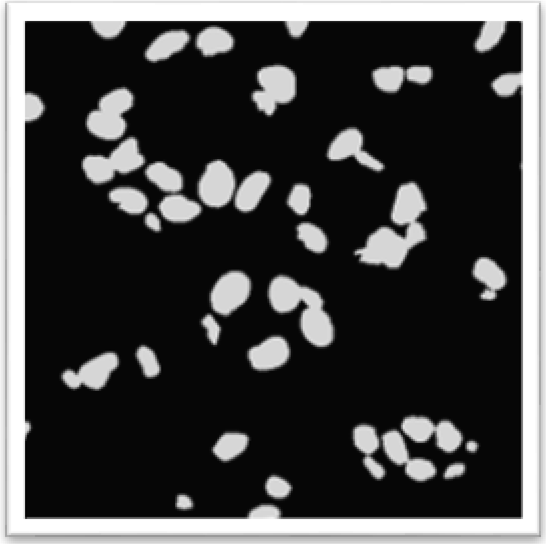




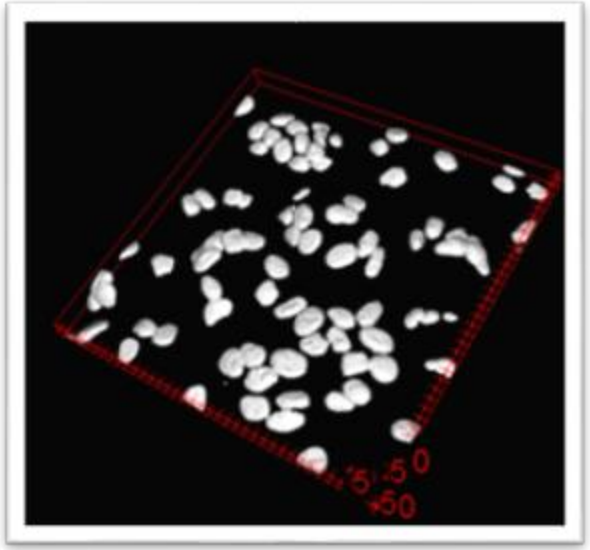
(alternative: Process > Math > Multiply: 255)

Before equalization

After equalization



3D rendering

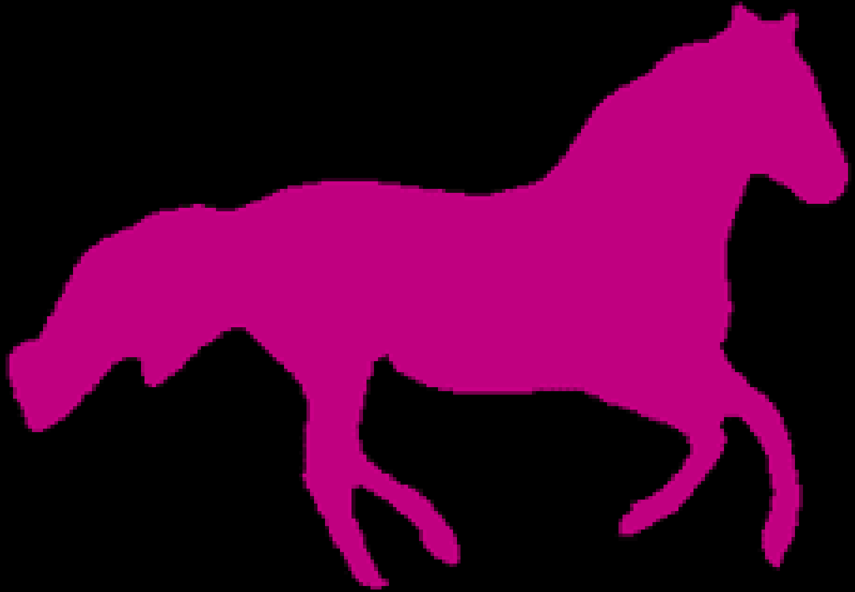


UNI FR UNIVERSITÉ DE FRIBOURG UNIVERSITÄT FREIBURG

BIO-INSPIRED MATERIALS NATIONAL CENTER OF COMPETENCE IN RESEARCH

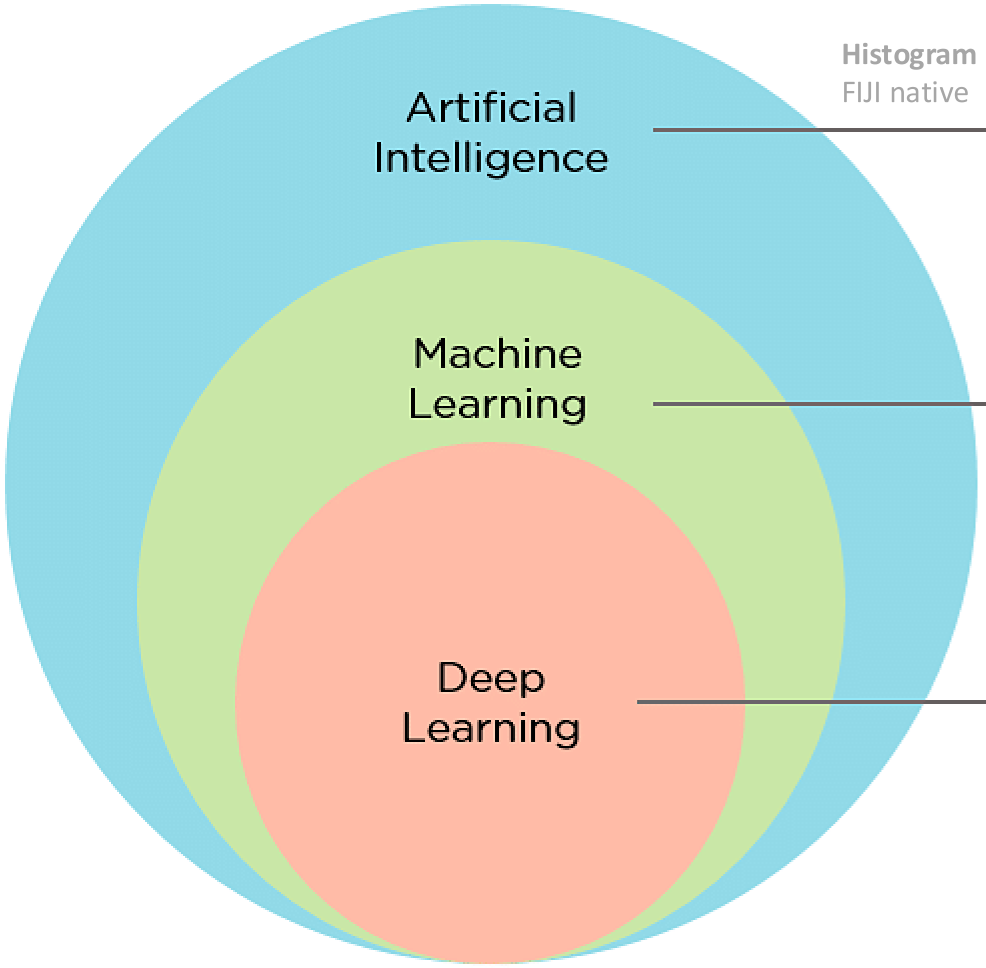
Thresholding, classification and segmentation

Level 3



Deep learning

Thresholding: human vs machine



Histogram based Thresholding
FIJI native

Ability of a machine to imitate intelligent human behavior

Classification
Labkit (FIJI plugin)
iLASTIK (standalone)

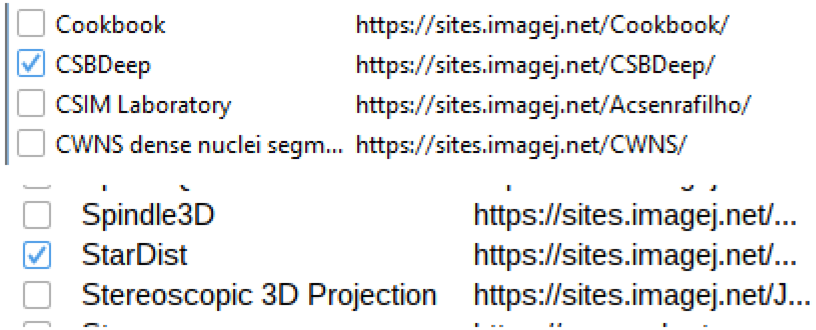
Application of AI that allows a system to automatically learn and improve from experience

Classification
DeepImageJ
CSBDeep
StarDist

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

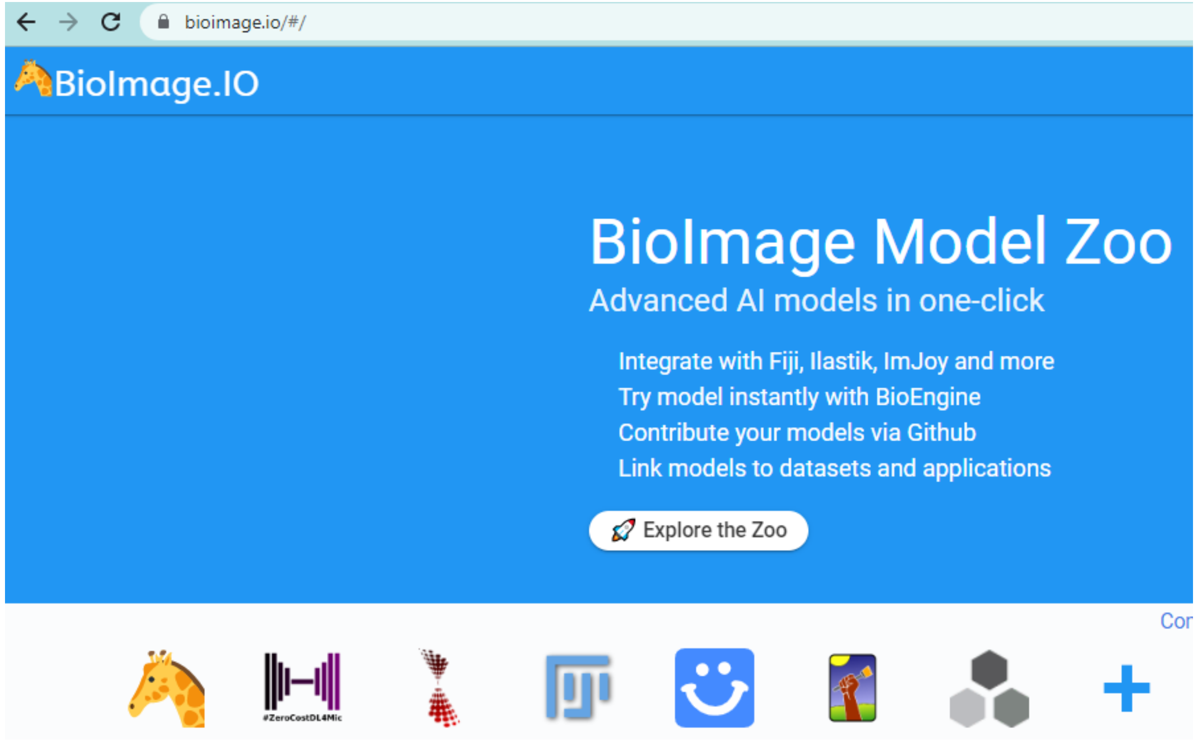
Deep Learning

From the repositories, install deepImageJ
Help > Update…
In the imageJ Updater > manage update sites. Tick **CSBDeep**

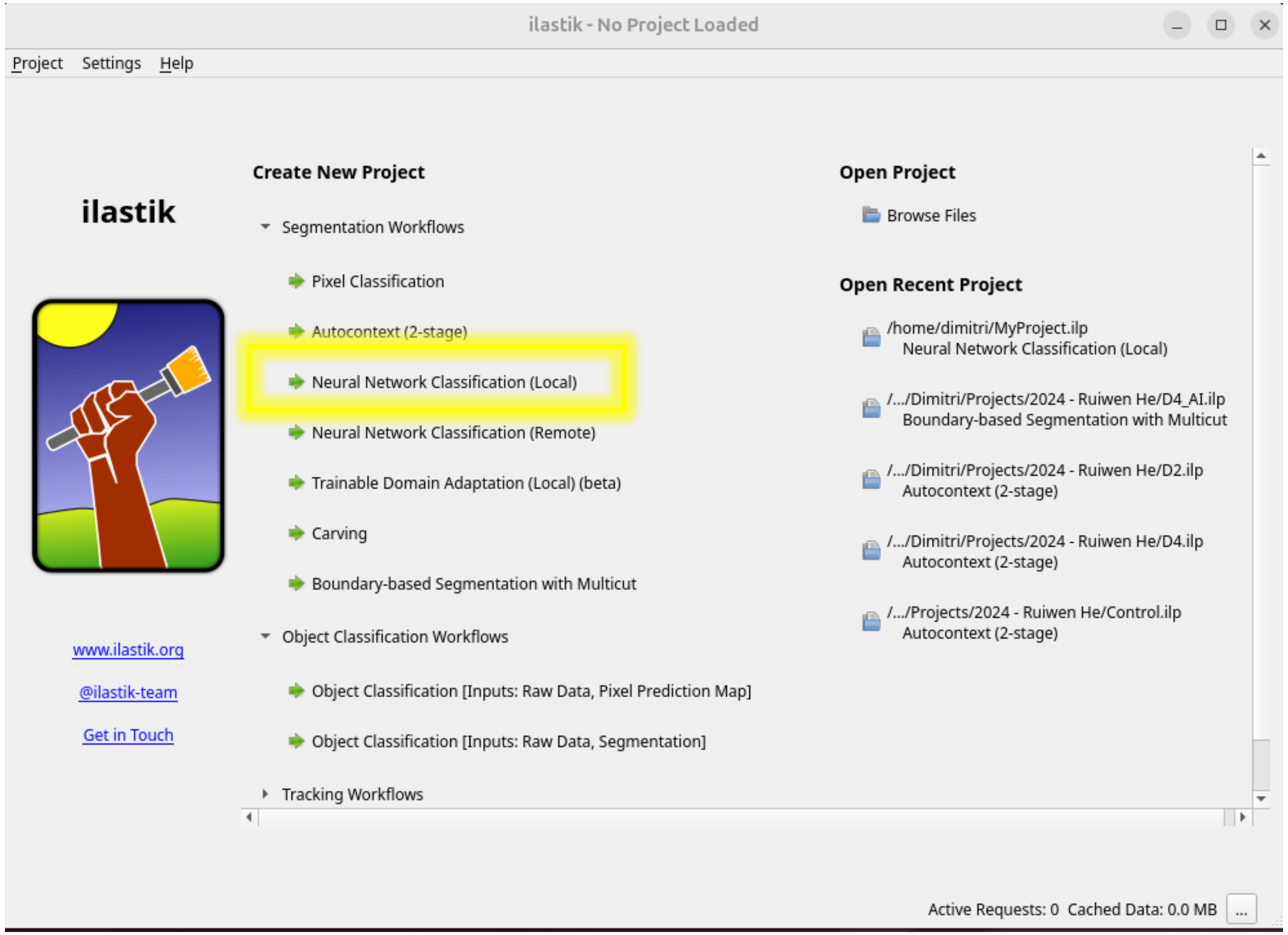


Click close
Click apply changes
Restart FIJI

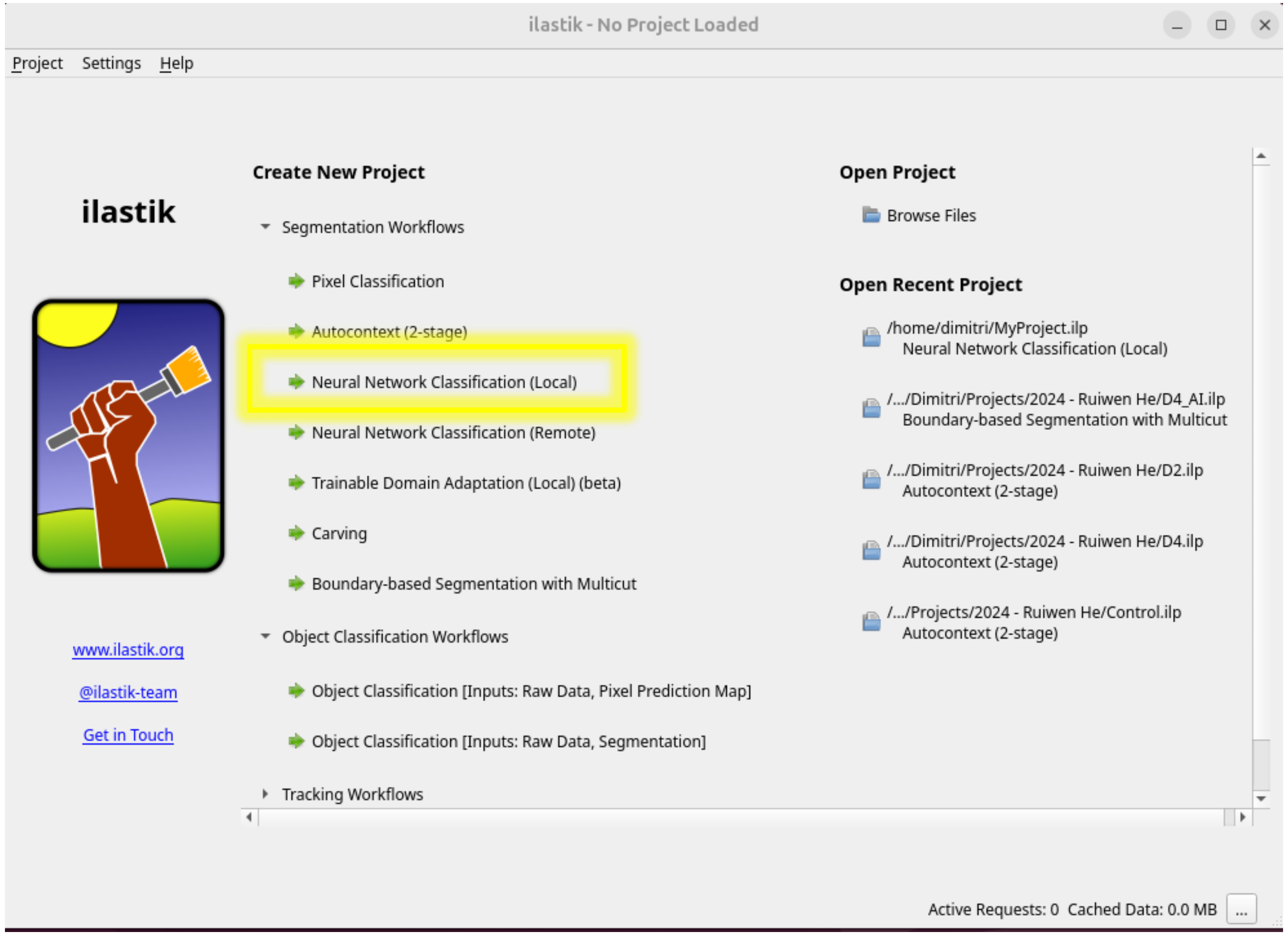
Meanwhile, have a look at **www.bioimage.io**

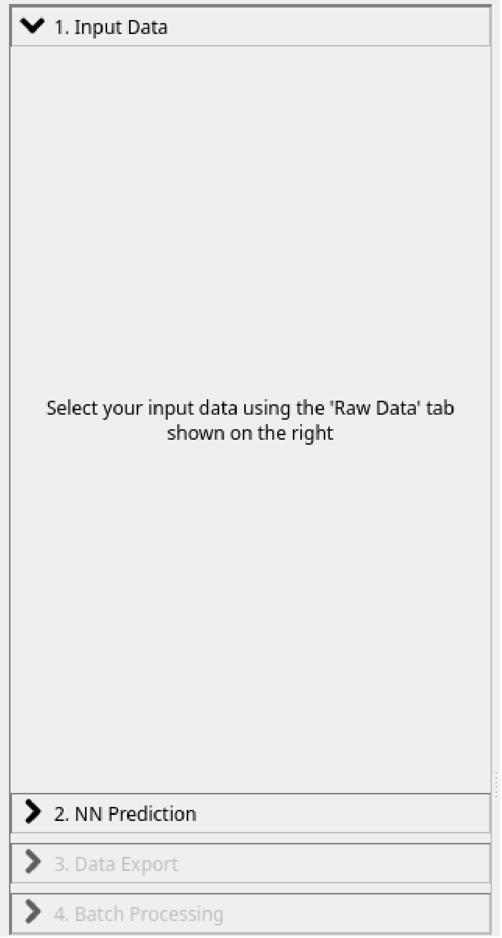


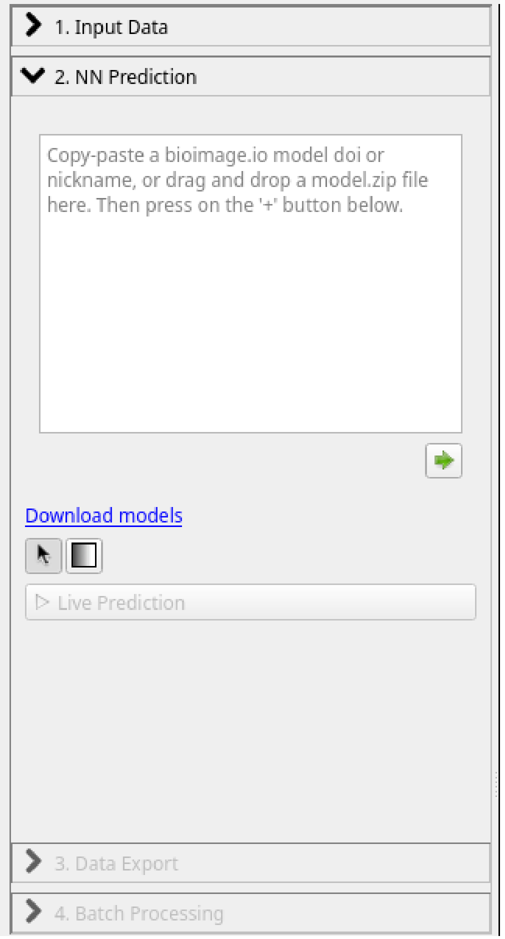
Deep Learning with Ilastik



Deep Learning with DeepImageJ

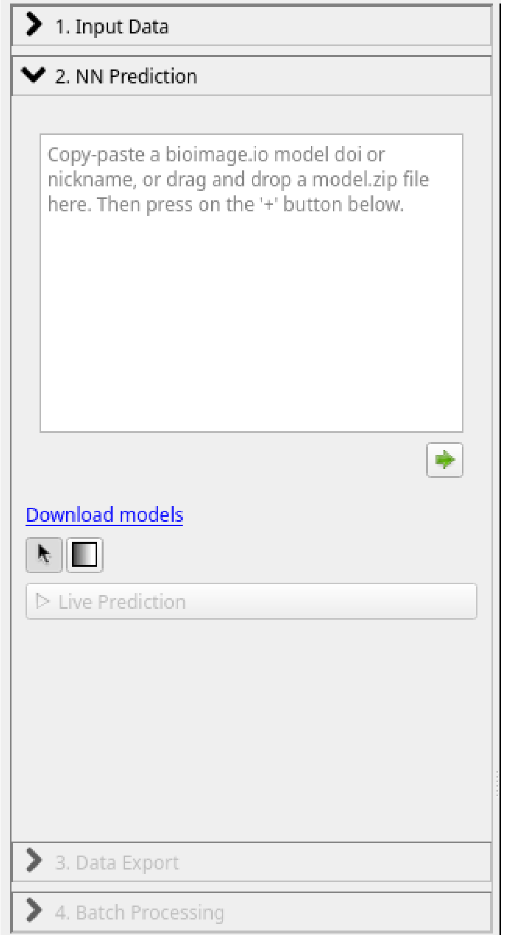


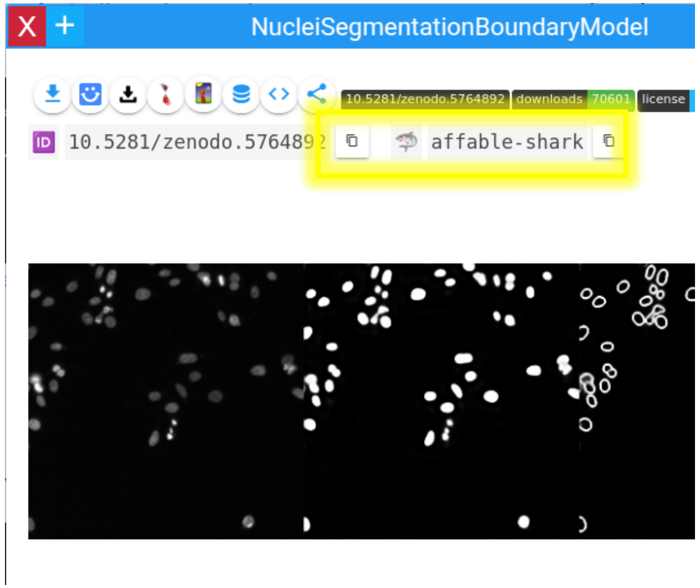


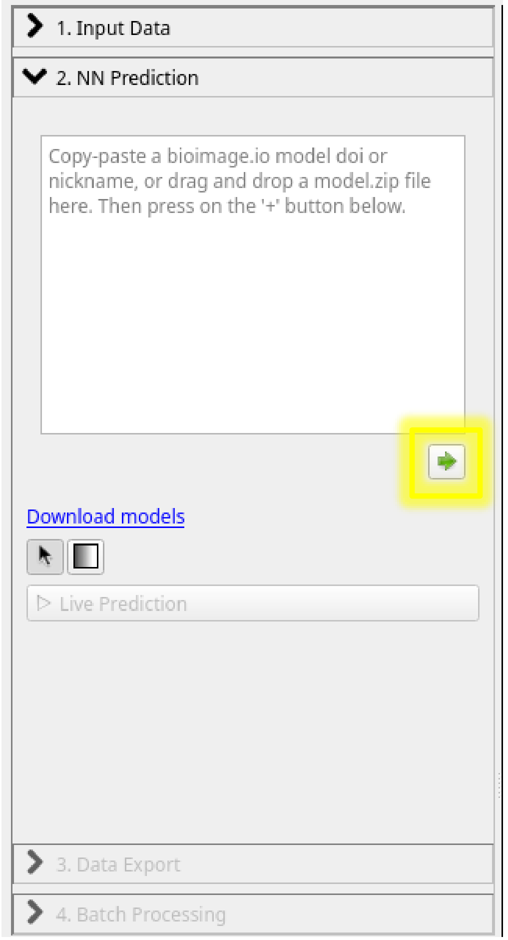


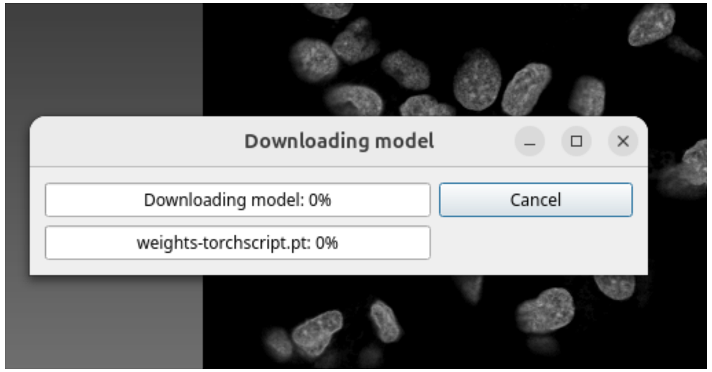
1. Add new > Add separate images… > Example4-single.h5

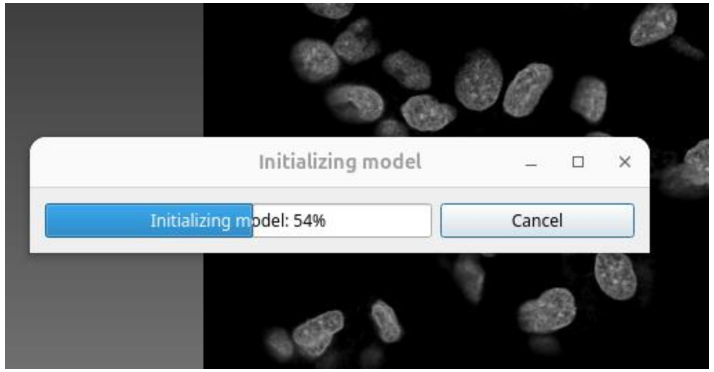
Deep Learning with DeepImageJ





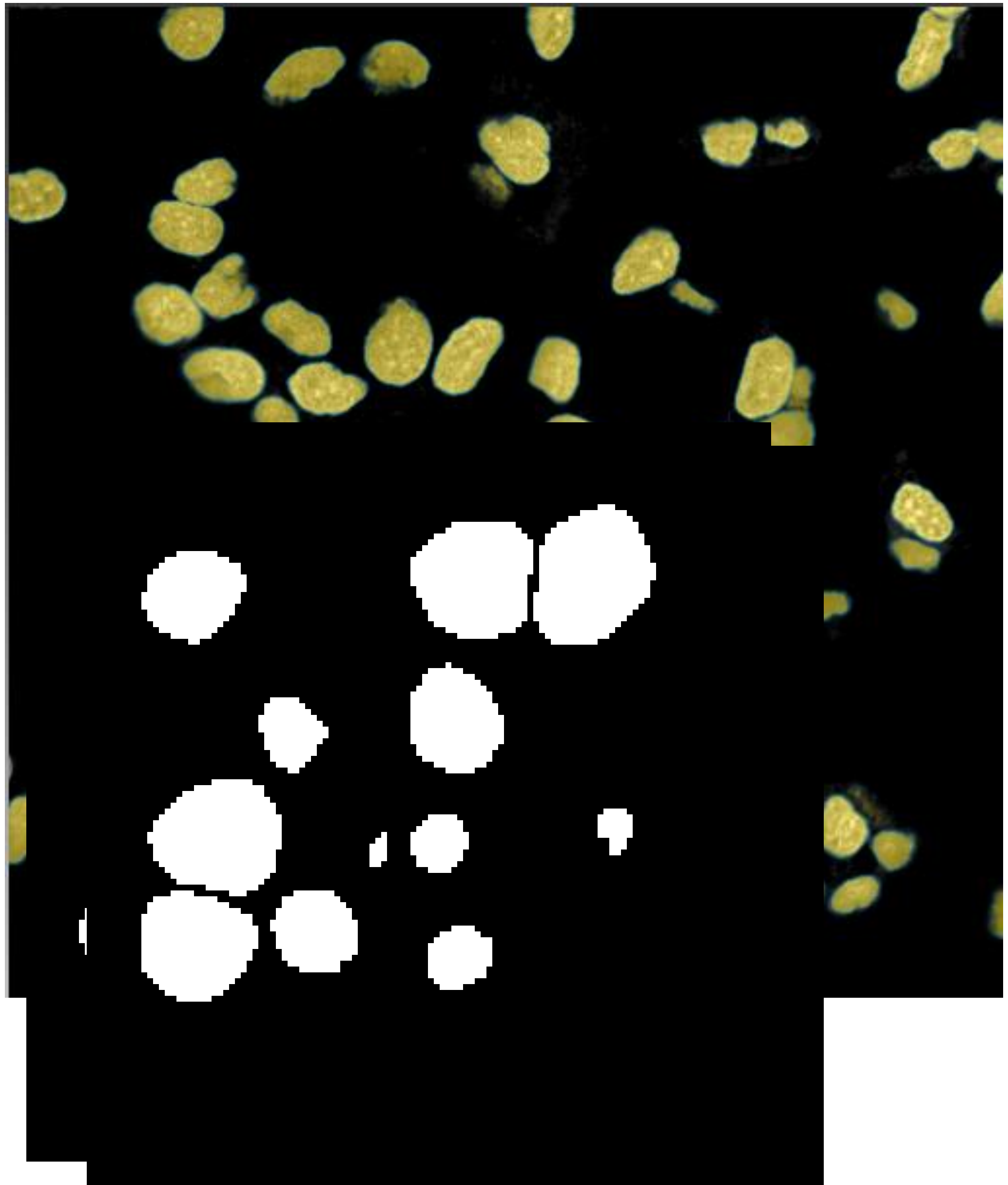


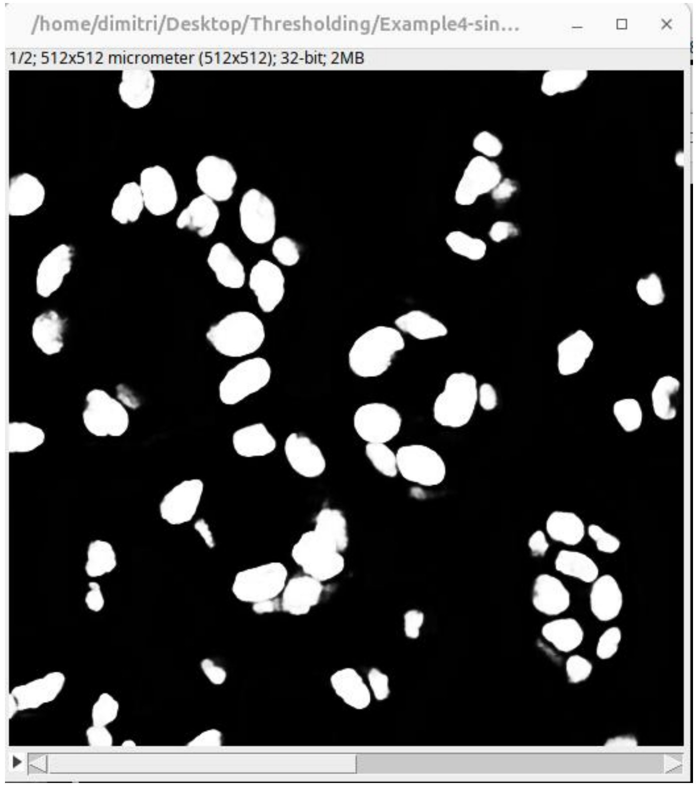


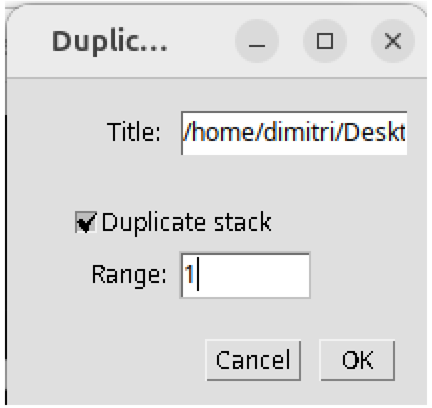


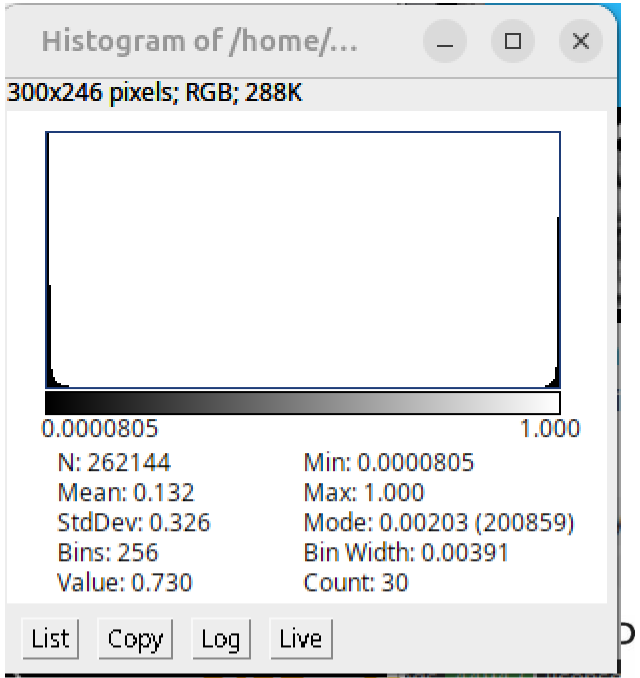
Finally: click Live prediction

Deep Learning with DeepImageJ





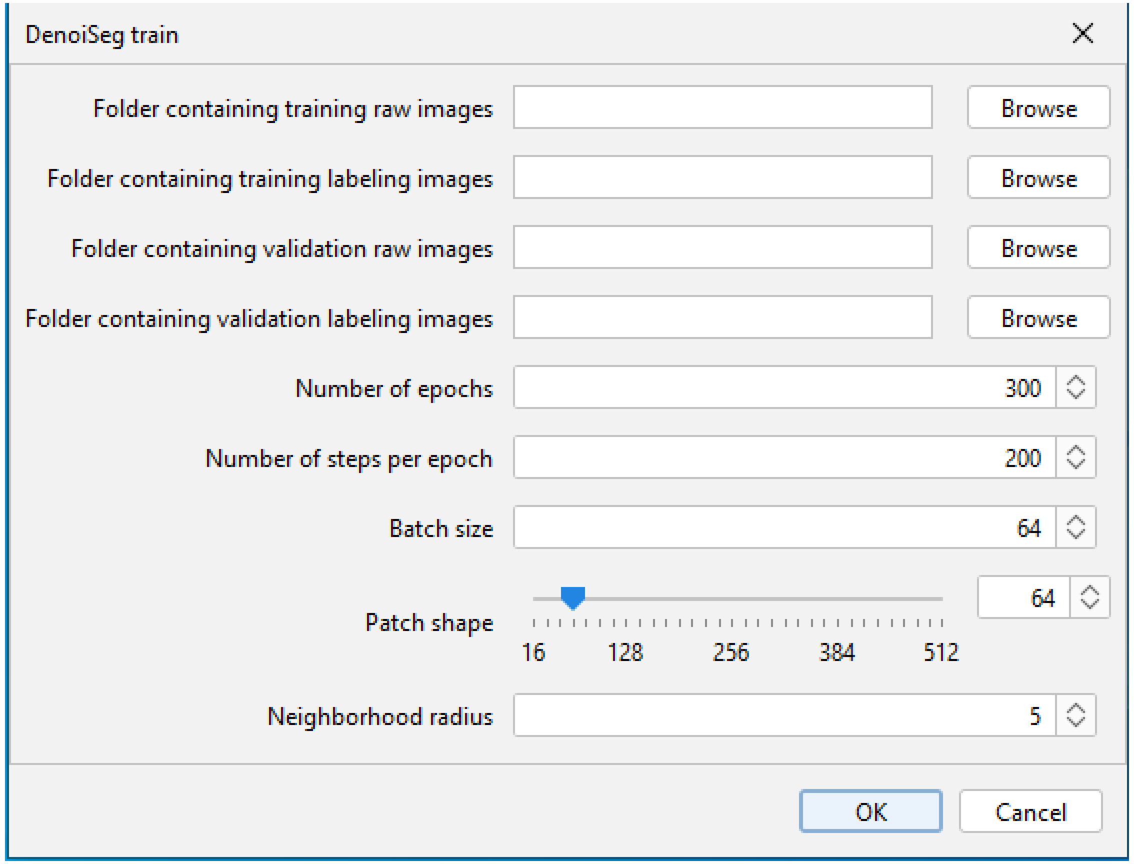




Deep Learning with CSBDeep

How to train a model yourself?
> Install CSBDeep

Plugins > CSBDeep > DenoiSeg > Train



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)

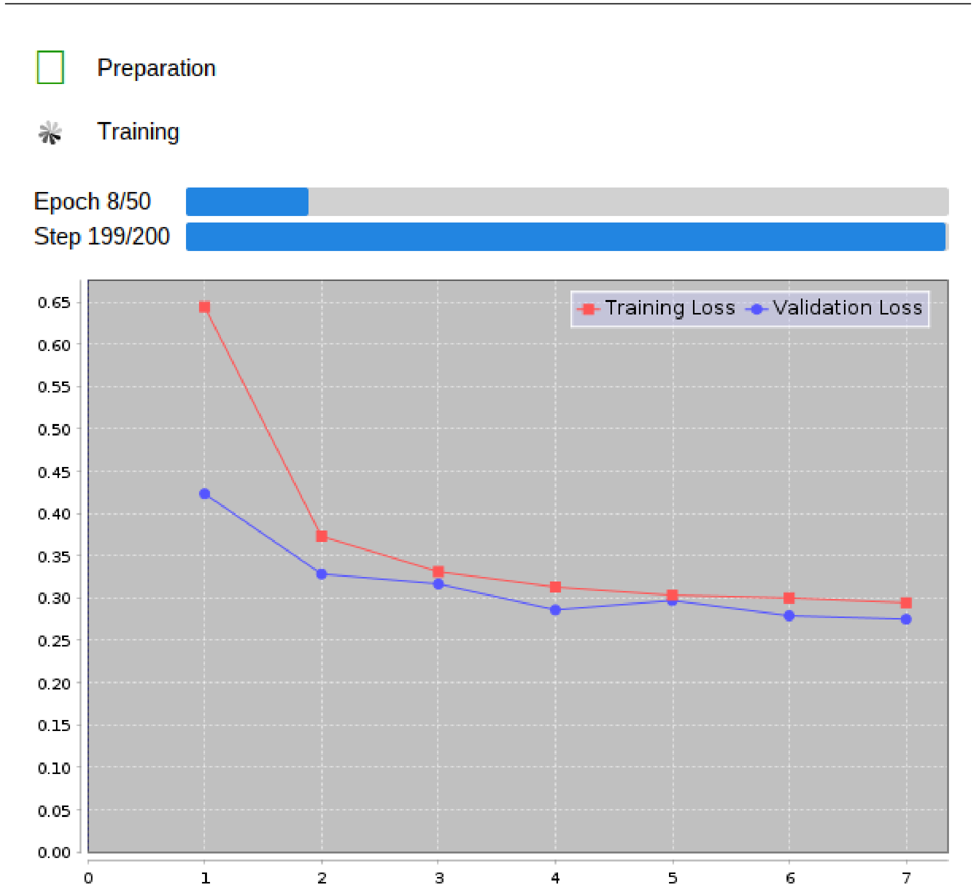
Number of Epochs: the more the better
Steps per Epoch: the more the better
Batch/Patch size: do not change

Then: wait...

Deep Learning with CSBDeep

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

→ Data in folder: Example 7




```
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
```

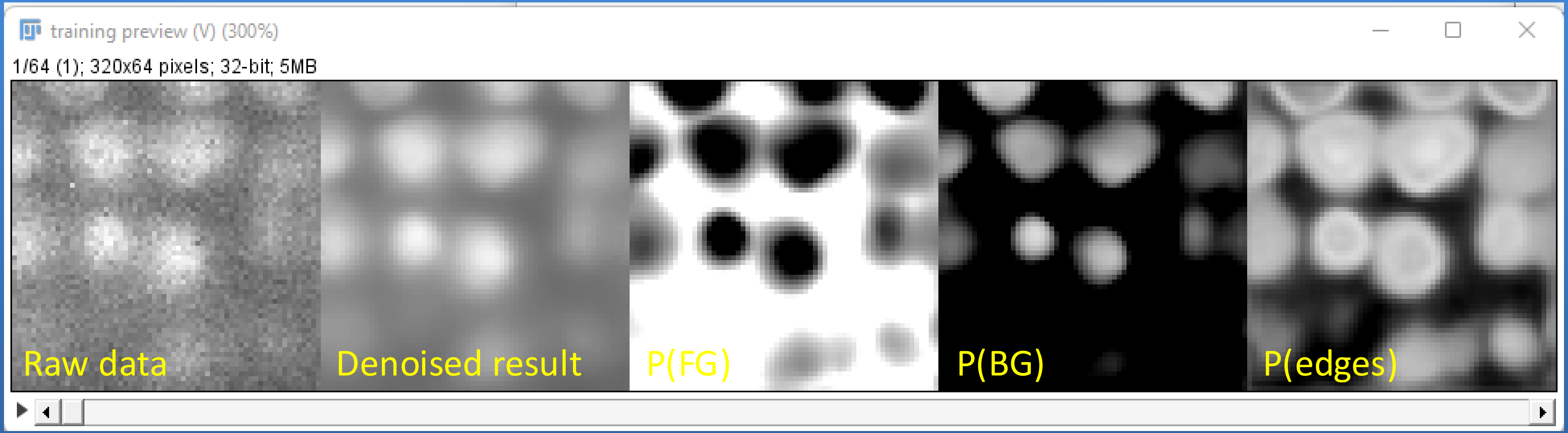
UNIVERSITÉ DE FRIBOURG
UNIVERSITÄT FREIBURG

BIO-INSPIRED MATERIALS
NATIONAL CENTER OF COMPETENCE IN RESEARCH

Deep Learning with CSBDeep

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

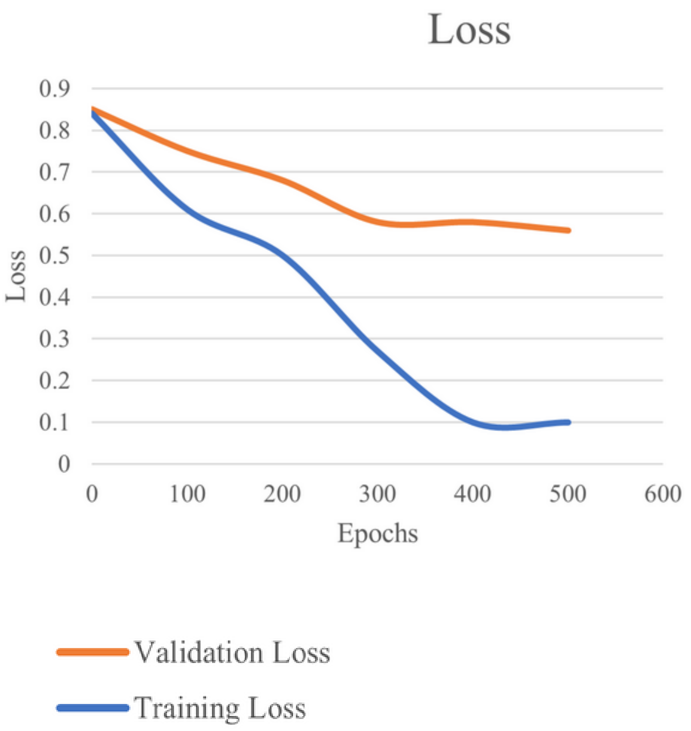
Preview window



Deep Learning with CSBDeep

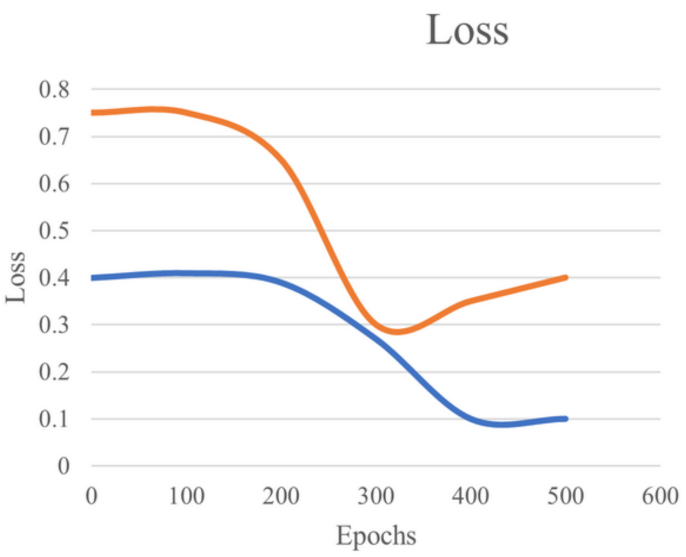
Underfitting

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

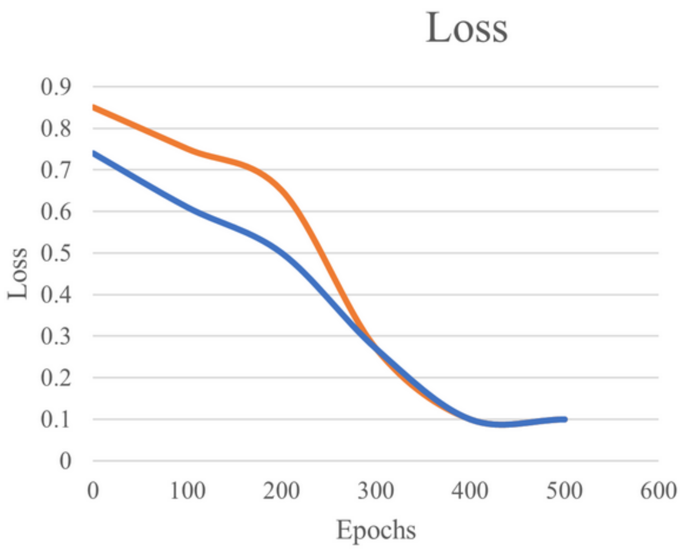


Overfitting

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



Good fit

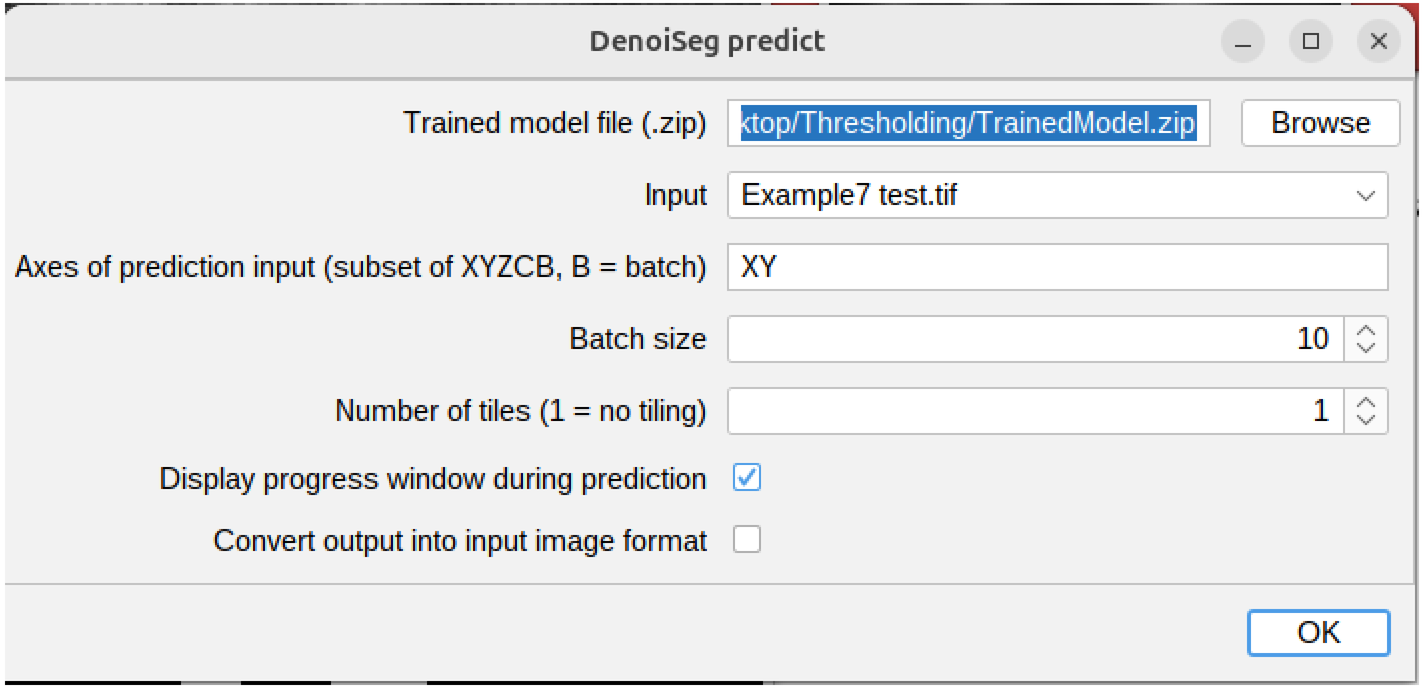


Deep Learning with CSBDeep

EXCERCISE

Use the trained model on data from Example 7

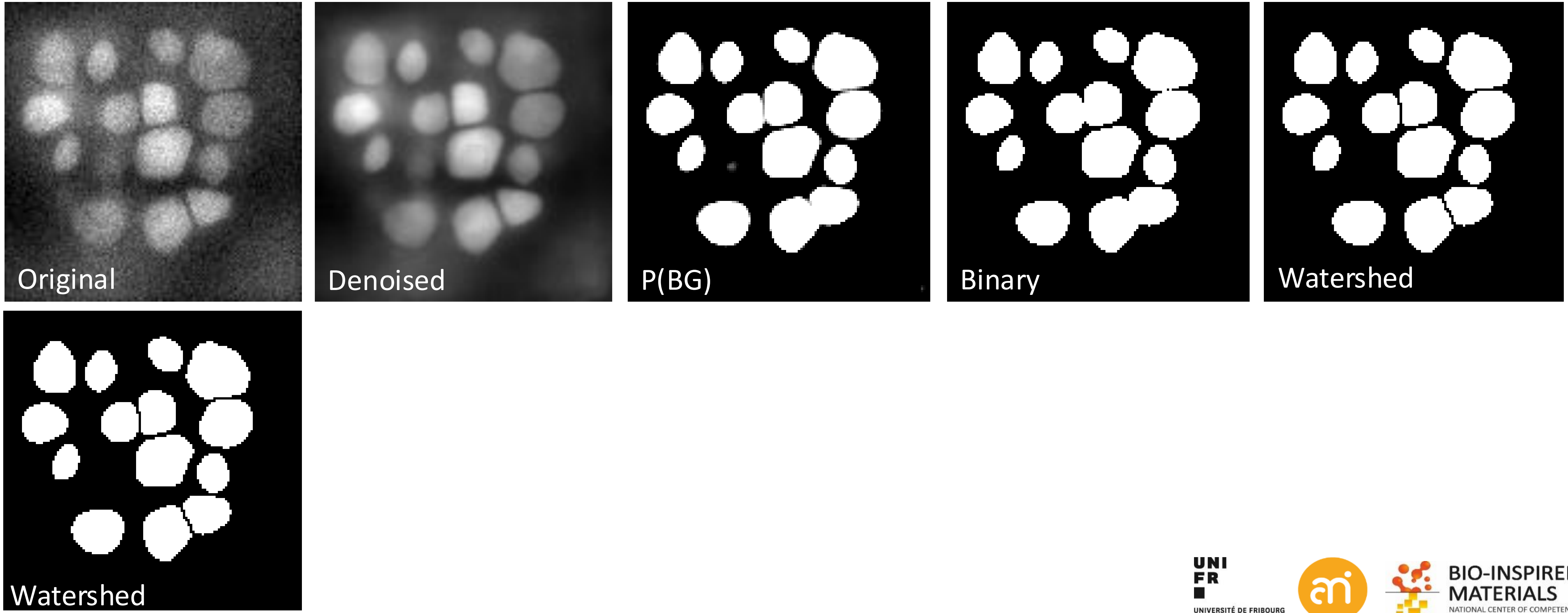
- Open a dataset from the trained images (e.g. Images > test > Example7 test.tif)
- Duplicate 1 image
- Plugins > CSBDeep > DenoiSeg > DenoiSeg predict



Deep Learning with CSBDeep

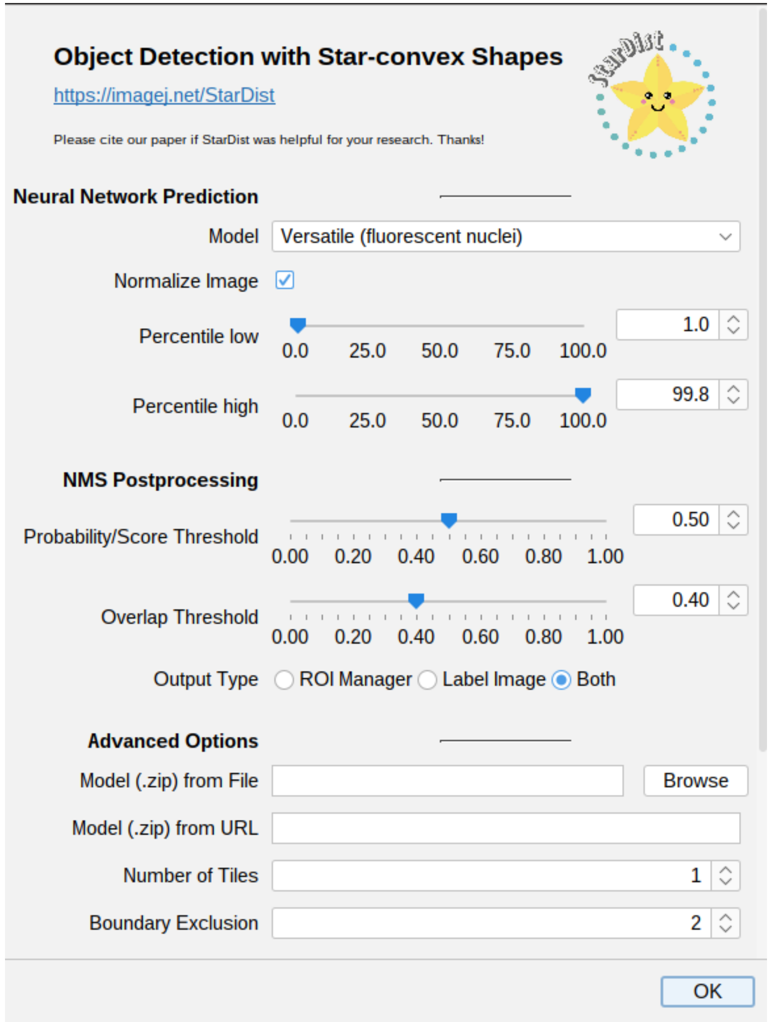
EXCERCISE

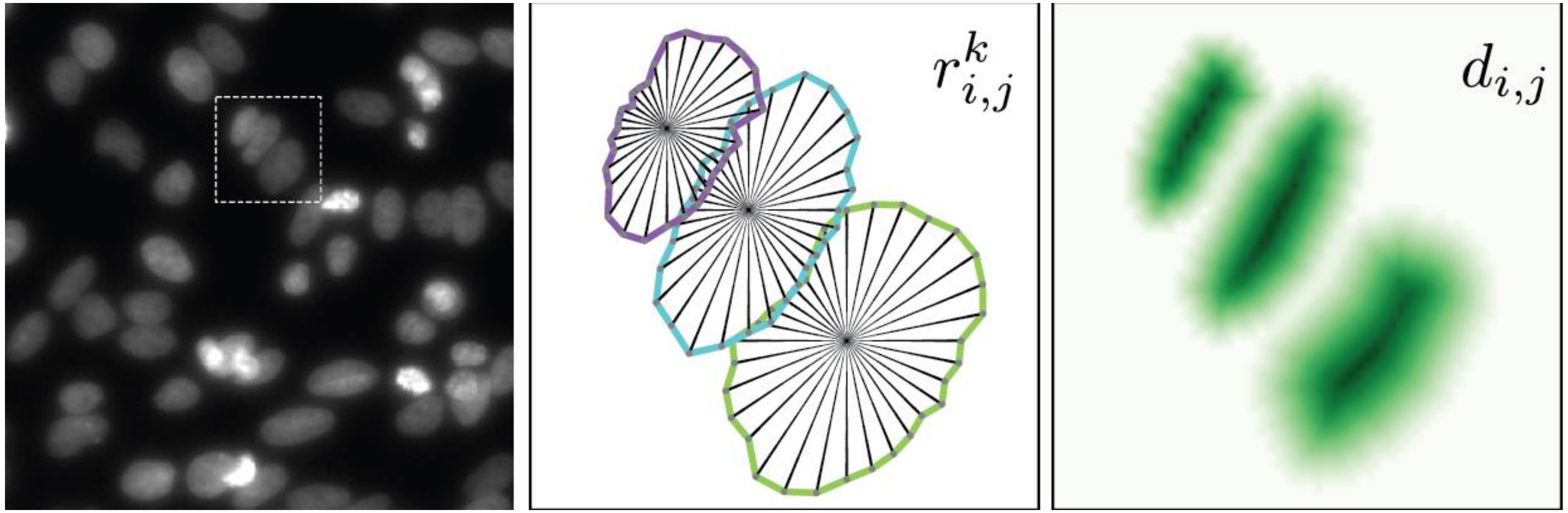
Use the trained model on data from Example 7



Deep Learning with StarDist

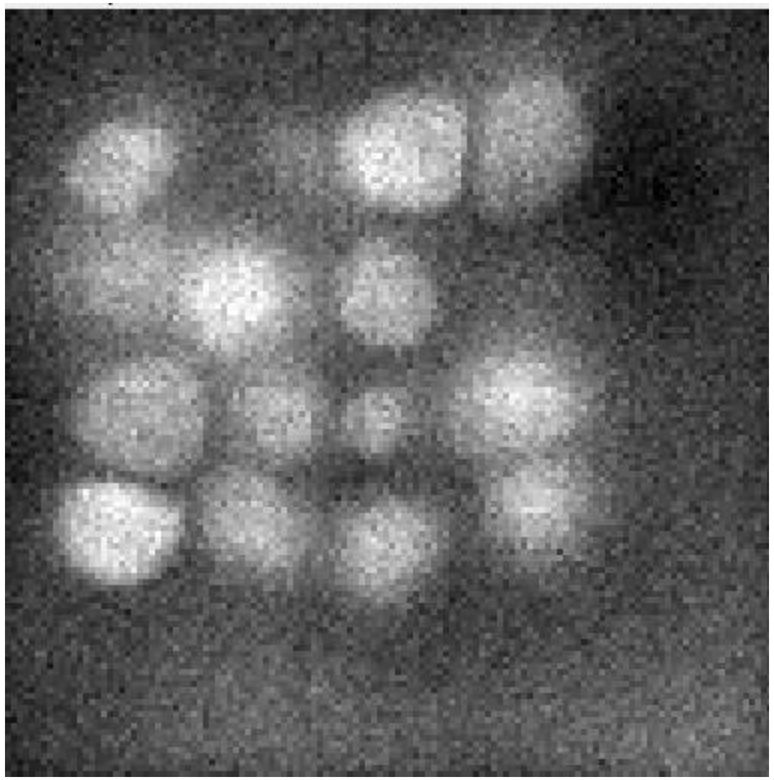
Looks to work well for segmentation of fluorescence data (e.g. nuclei), but 2D
Help > Update… > Manage update sites > Stardist
Can be scripted

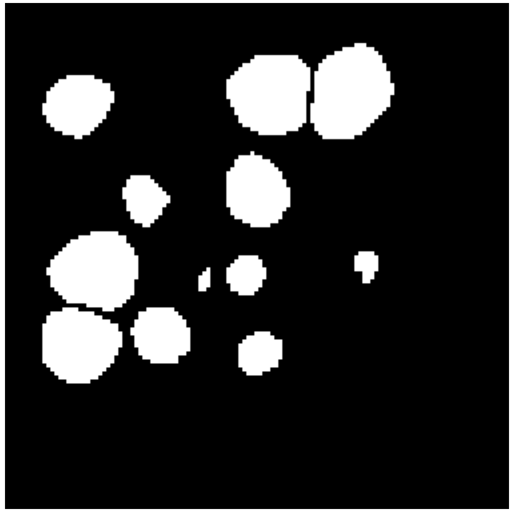




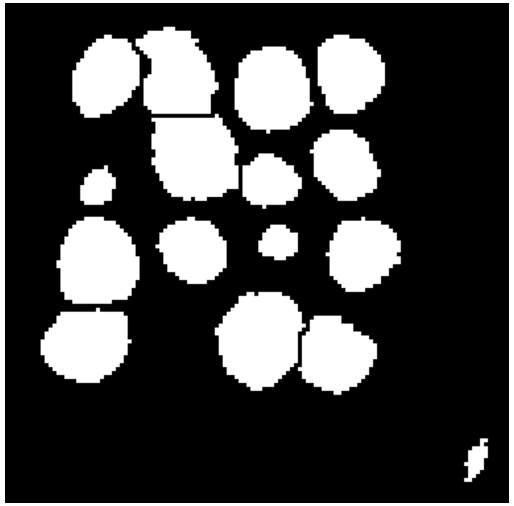
Deep Learning with StarDist

Looks to work well for segmentation of fluorescence data (e.g. nuclei), but 2D
Help > Update… > Manage update sites > Stardist
Can be scripted





Ilastik, quick manual

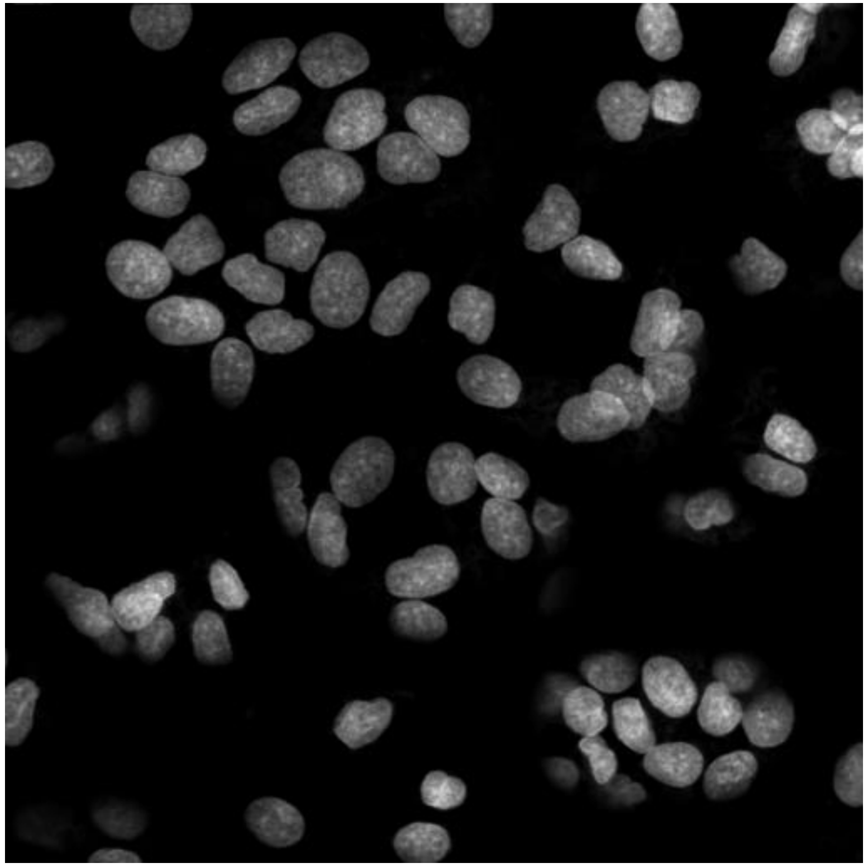


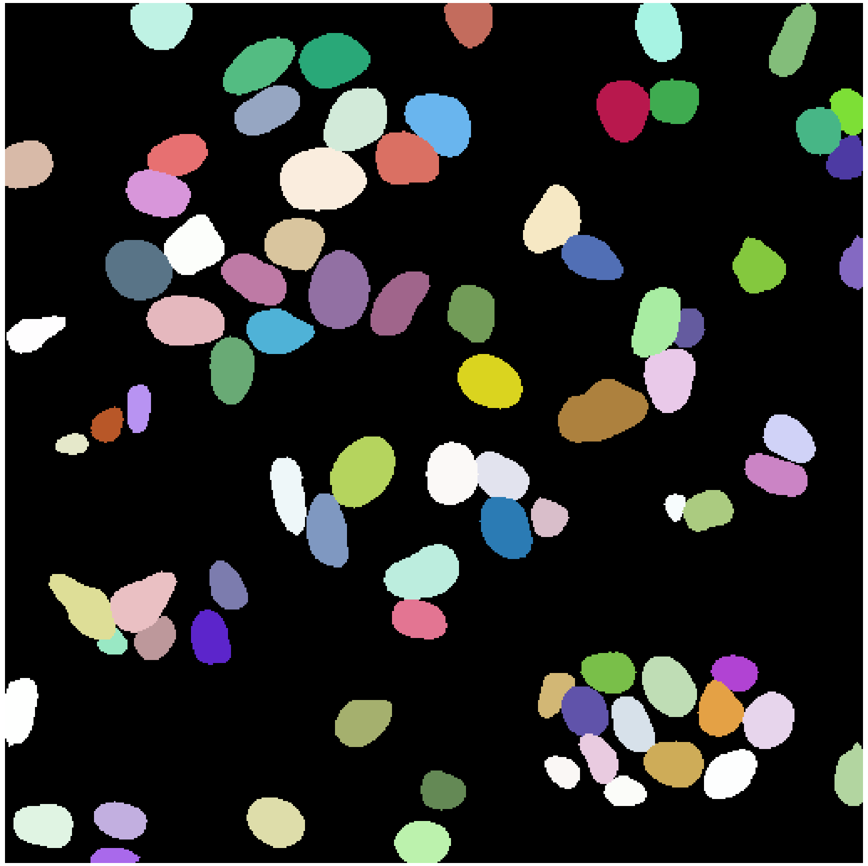
Affable-shark

Stardist

Deep Learning with StarDist

Looks to work well for segmentation of fluorescence data (e.g. nuclei), but 2D
Help > Update… > Manage update sites > Stardist
Can be scripted





Number extraction

Blob analysis aka particle counting

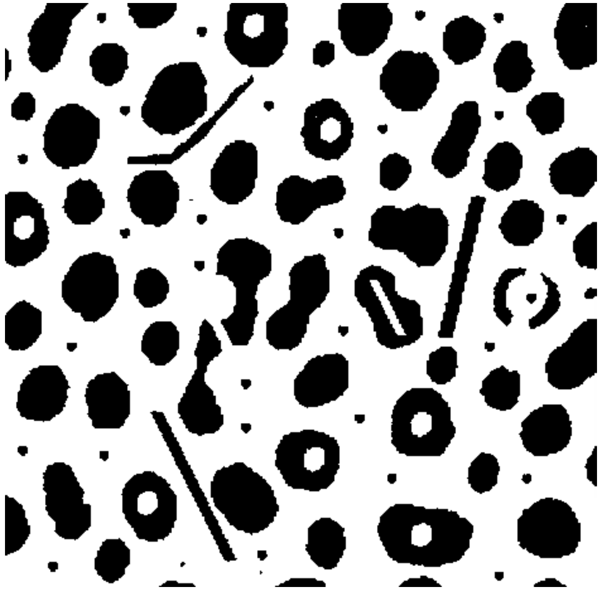
Assumption
Your data is binary (or at least segmented)

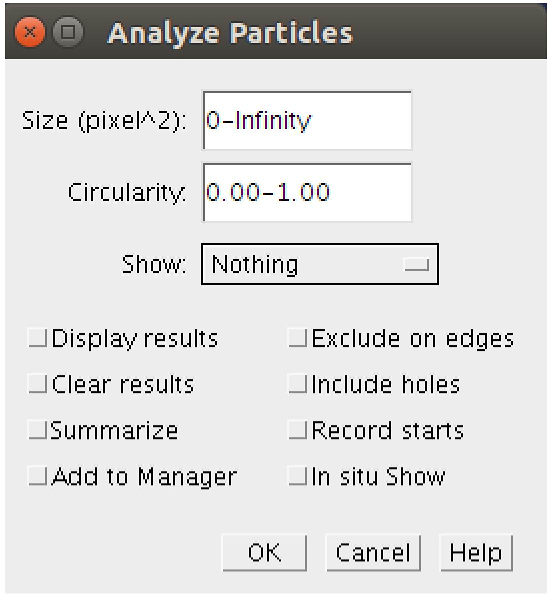
Before you start:

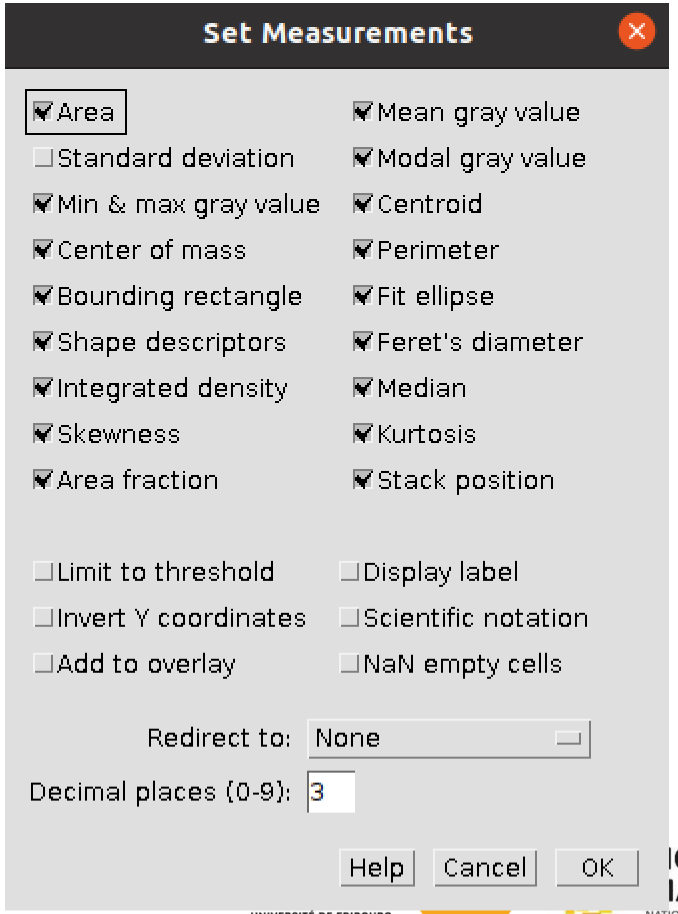
- Can you trust your binary image?
- Is the scale properly set? (Analyse > set scale)
- Is the foreground particle white (if not: invert: ctrl+i)
- What do you want to measure (Analyse > Set Measurements)

Two step procedure:

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



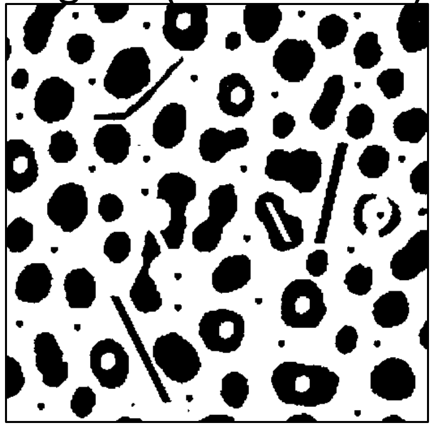




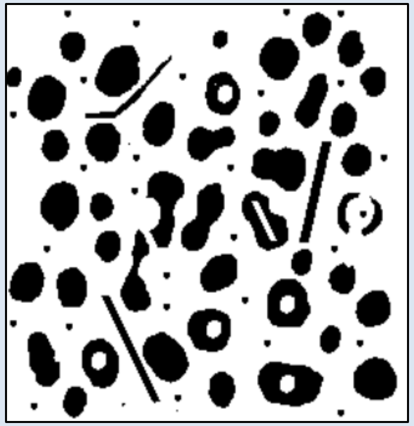
IO-INSPIRED
IATERIALS
UNIVERSITÉ DE FRIBOURG
UNIVERSITÄT FREIBURG
NATIONAL CENTER OF COMPETENCE
IN RESEARCH

Size measurements: filters

Original (thresholded)



Edge filter

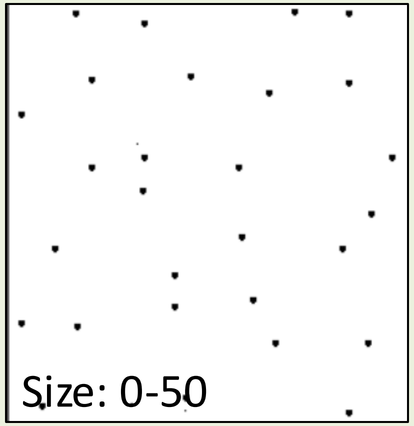


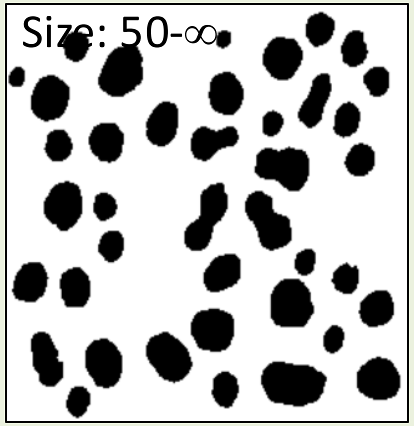
particles touching the edge will be ignored

Include holes

Interior holes will be included

Size filter

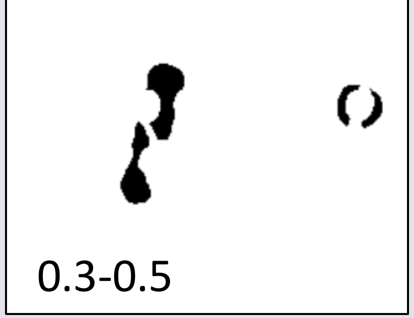


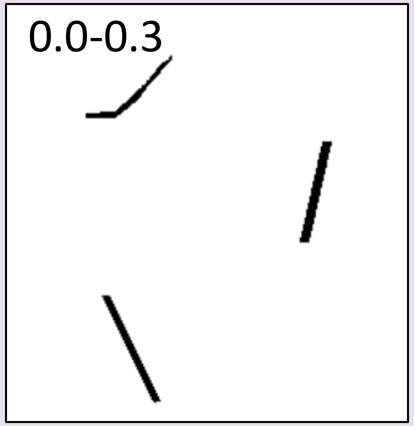


Particles with size (=area) outside the range specified in this field are ignored.

Circularity filter

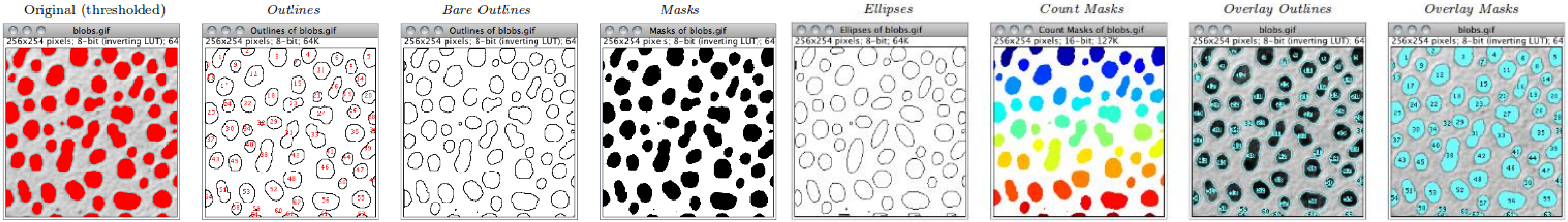
$$\text{Circ.} = 4\pi \times \frac{Area}{Perimeter^2}$$





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

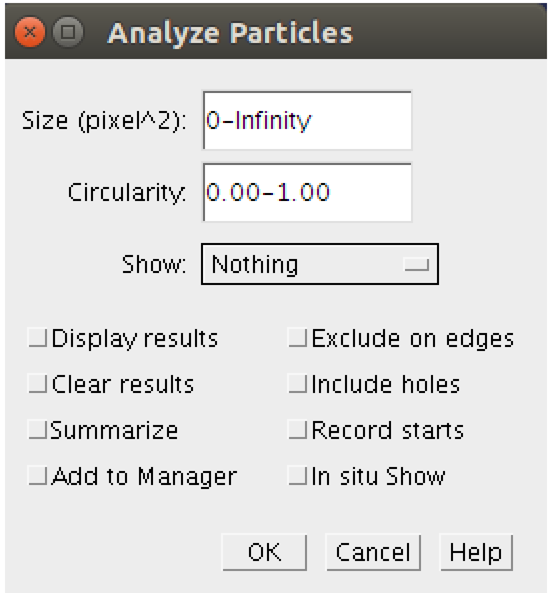
Size measurements: Outlines, masks and overlays



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. . .

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

Size measurements: Results

EXERCISE

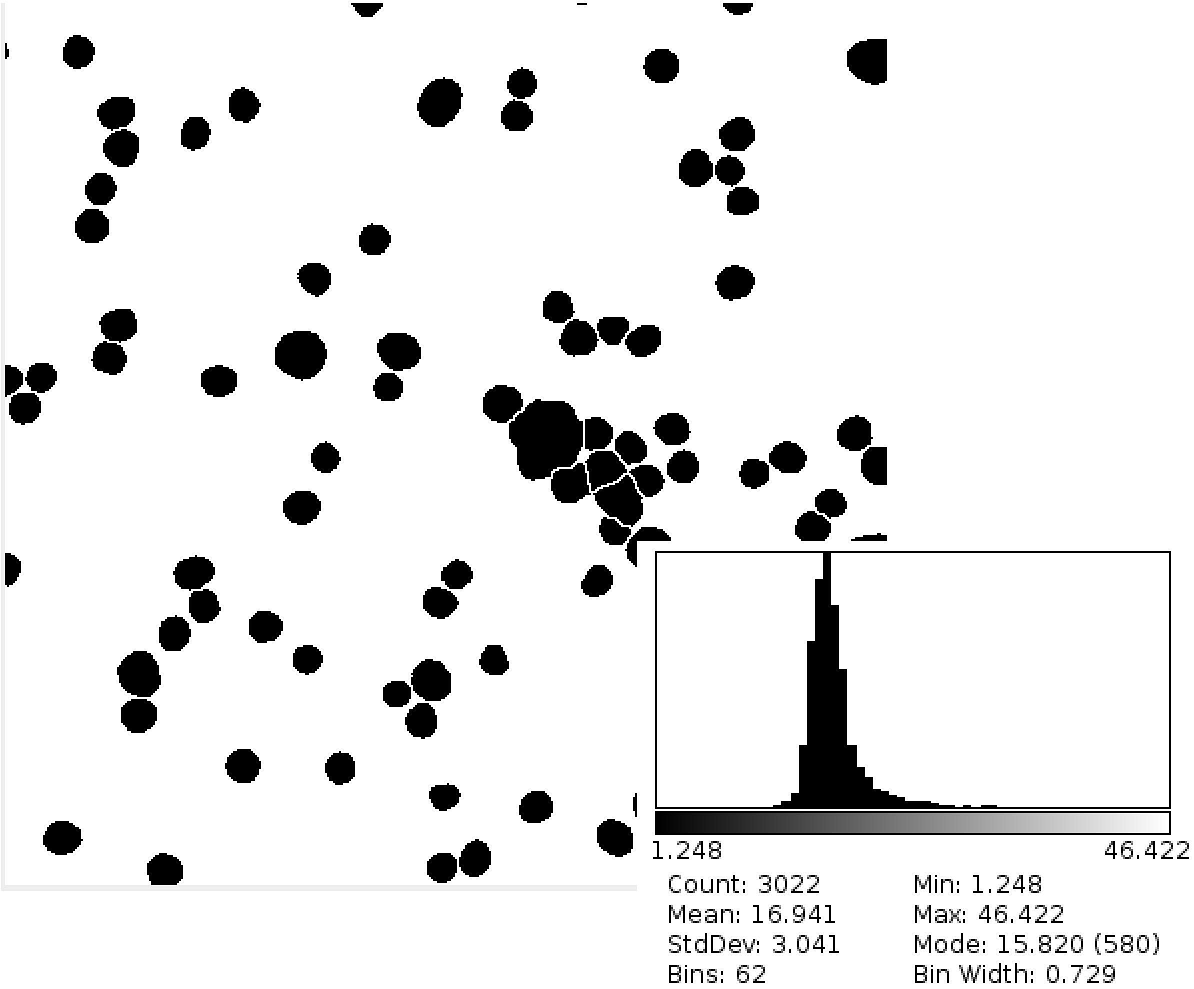
Calculate the mean radius of the AuNP in Example 2 – AuNP. Try with and without performing a watershed before. Show a distribution of the feret.

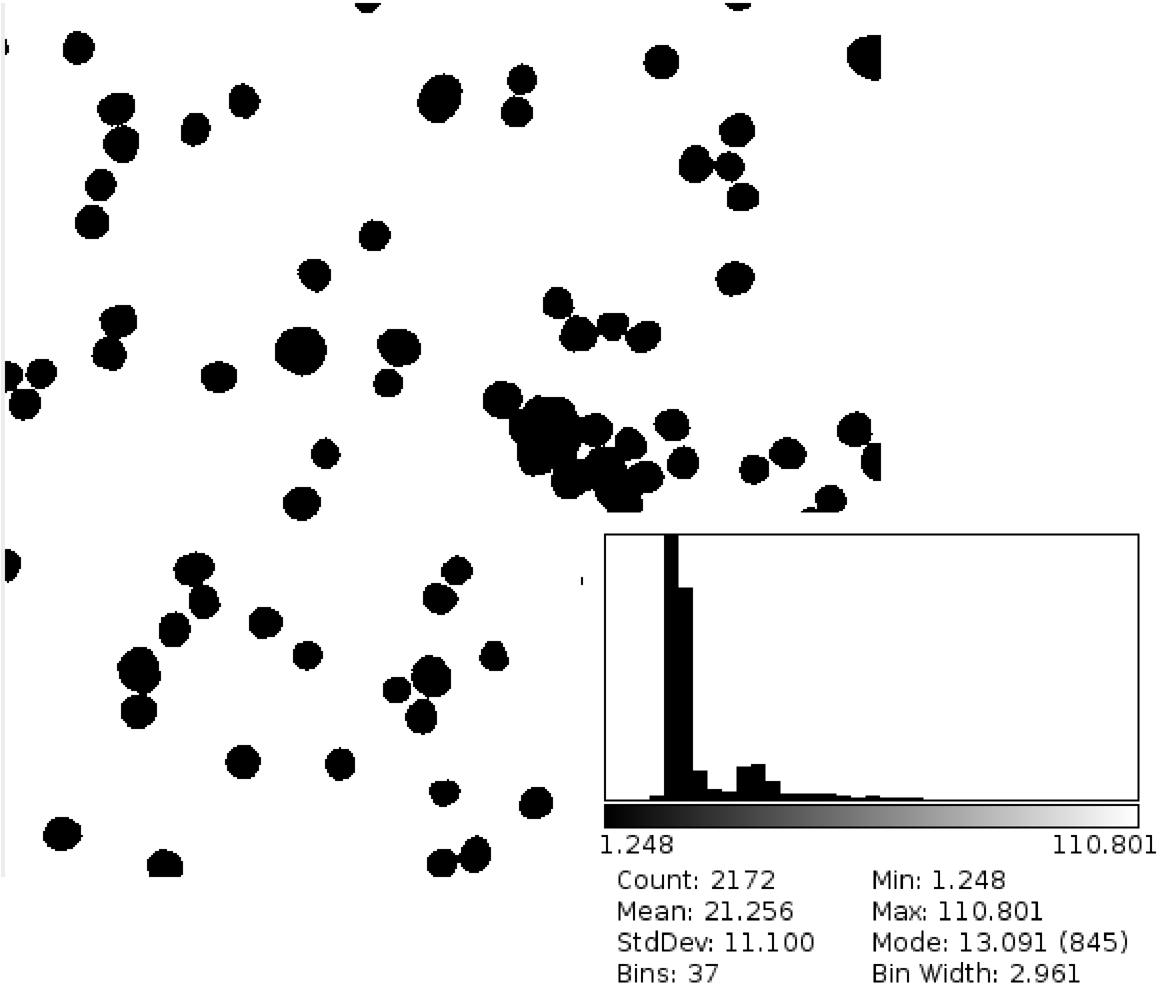
1. Image > adjust > threshold (use Default)
2. Analyze > Measure particles
3. Analyze > Distribution

Size measurements: Results

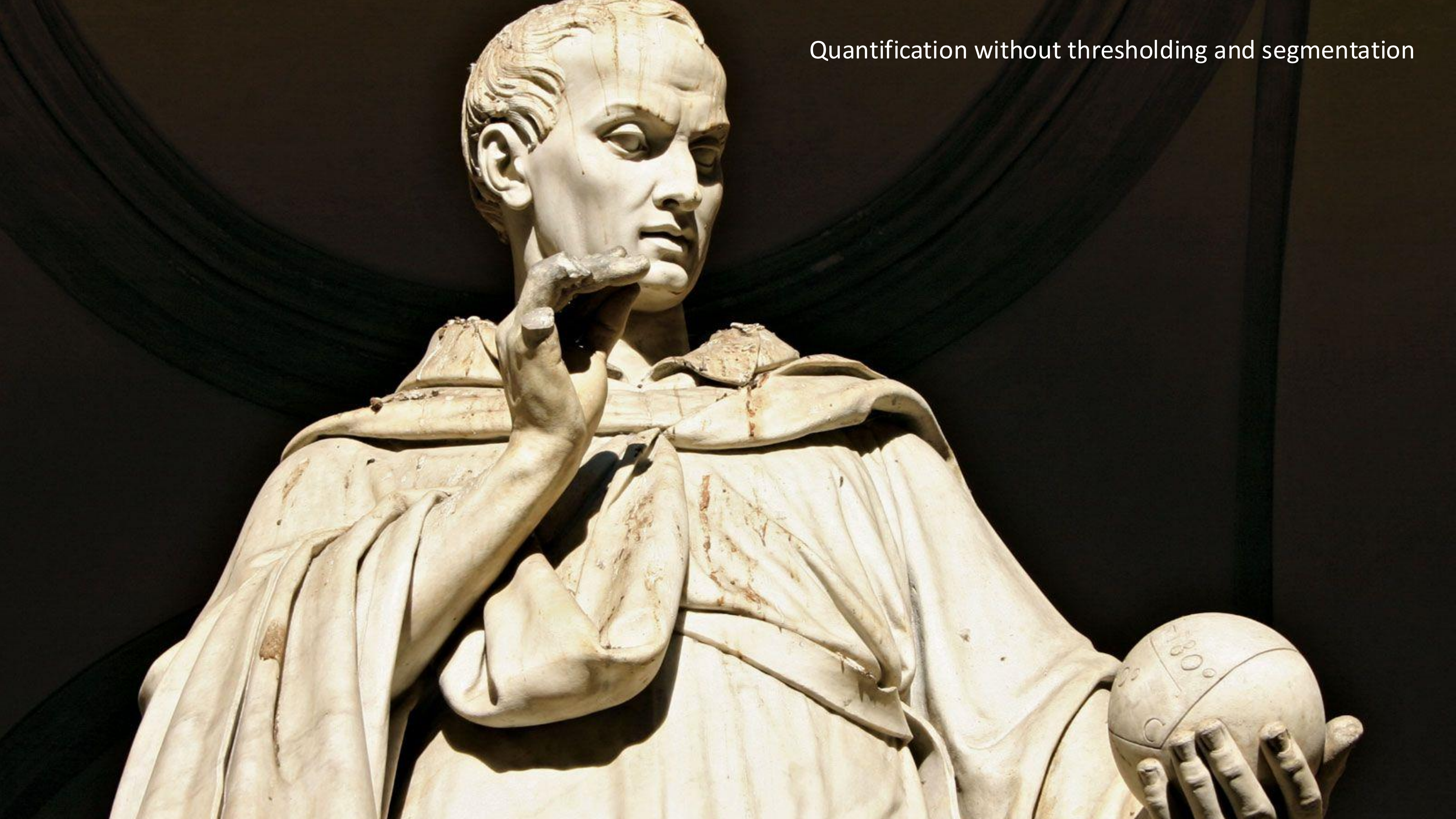
EXERCISE

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

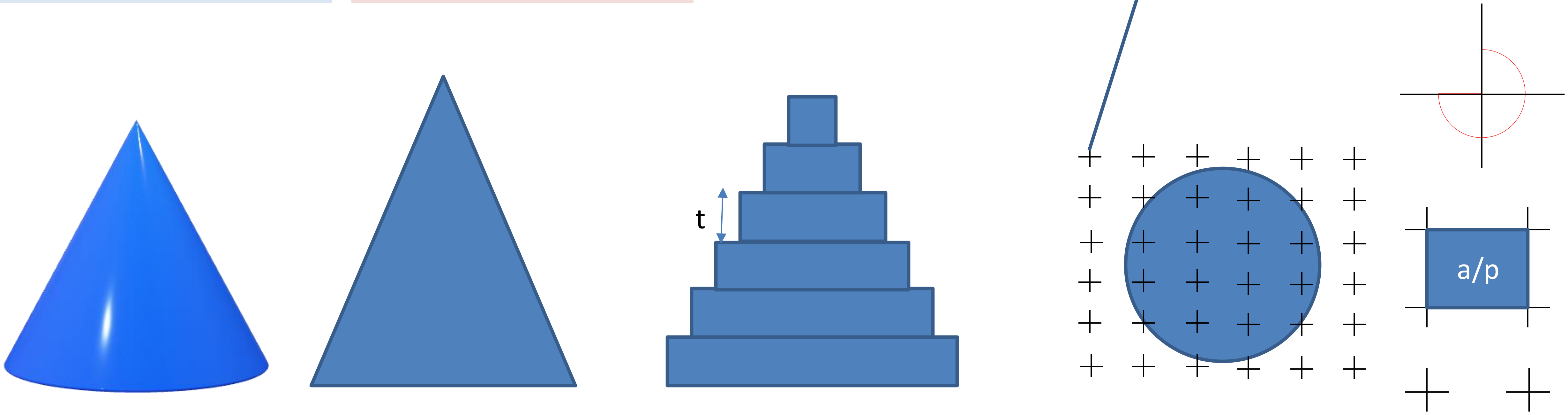


Volume estimation with Cavalieri

Pro
Independent of object
Optical disector
Low coefficient of error

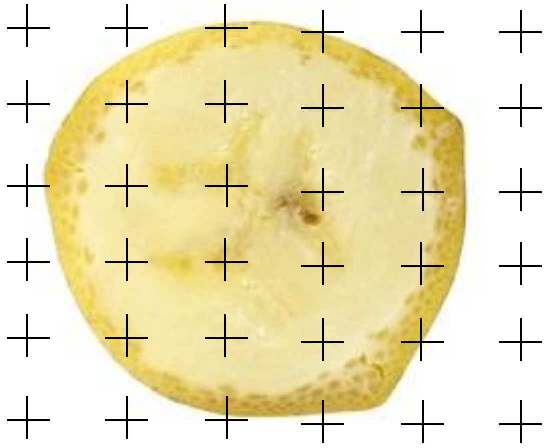
Contra
thickness must be known
Over/underprojection

Systematically uniform grid, **randomly** dropped



3D Object | XZ side view | XZ view Cavalieri | XY view Cavalieri

Volume estimation with Cavalieri



$a/p = 1\ cm^2$
Thickness: 2 cm
Repeat i times (with i = number of banana pieces)

Volume estimation with Cavalieri

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

$V = 1\,(cm^2) \cdot 2(cm) \cdot 116$
$V = 232\ cm^3$

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

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

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

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

UNI FR
UNIVERSITÉ DE FRIBOURG
UNIVERSITÄT FREIBURG

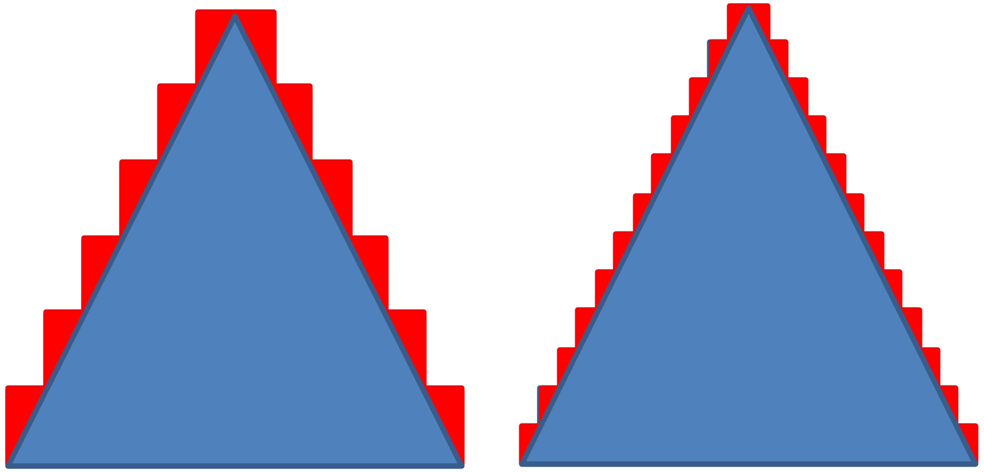
BIO-INSPIRED MATERIALS
NATIONAL CENTER OF COMPETENCE IN RESEARCH

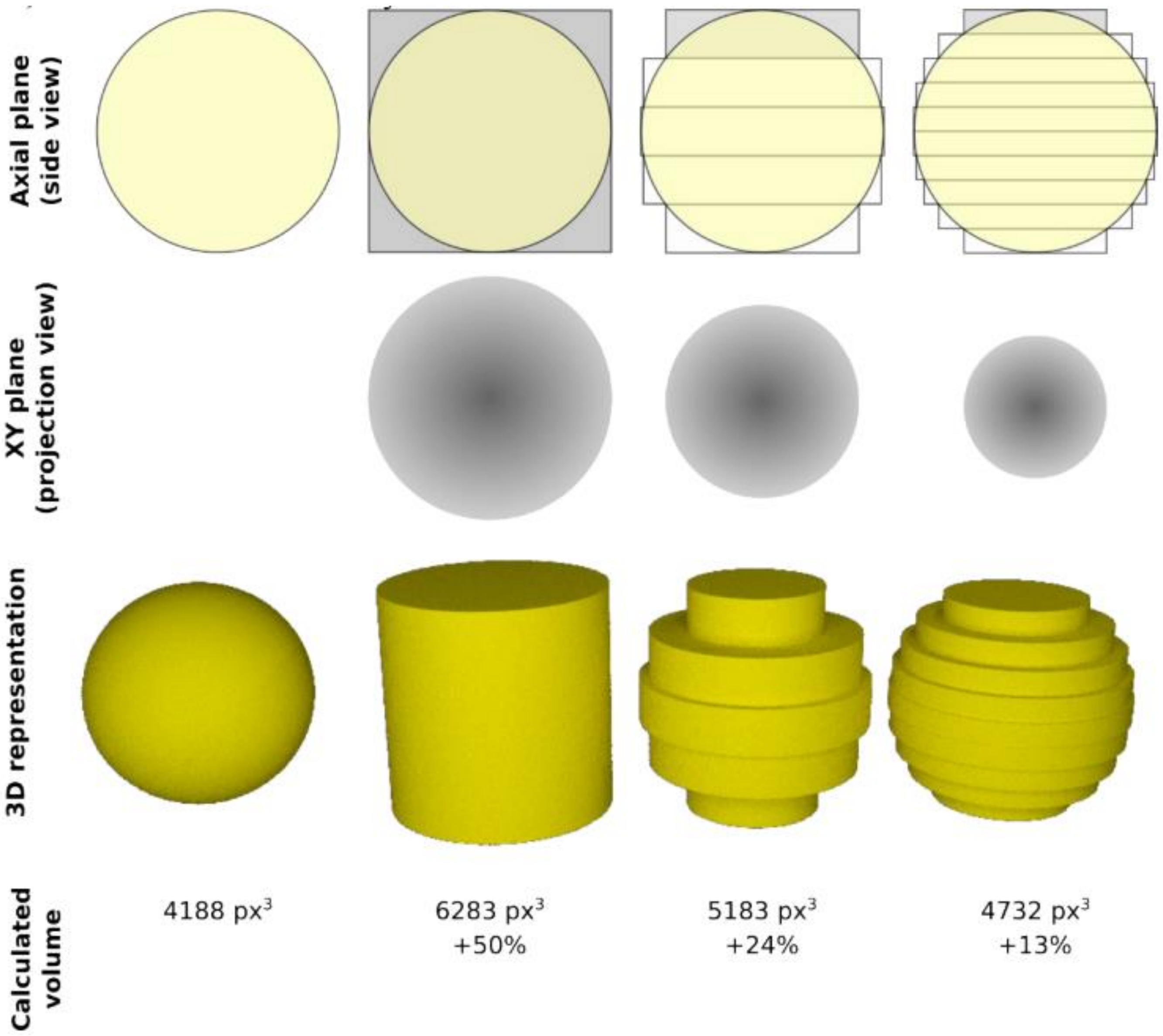
Volume estimation with Cavalieri: Holmes effect

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

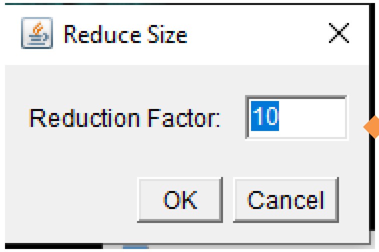
1. Open Example 8
2. Reduce the Z stack by factor 10
3. For fun (and to make it no longer a binary image): add noise (e.g. with an SD of 50)
3. Throw a random grid over the Image, A/p of roughly 150 pixel^2
4. Count the number of crosses that fall onto the object, on all slices

Size measurements: Results

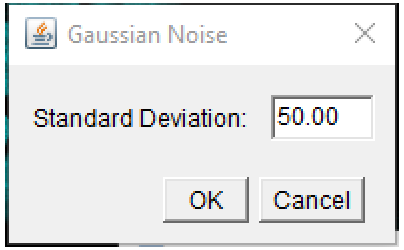
EXERCISE

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

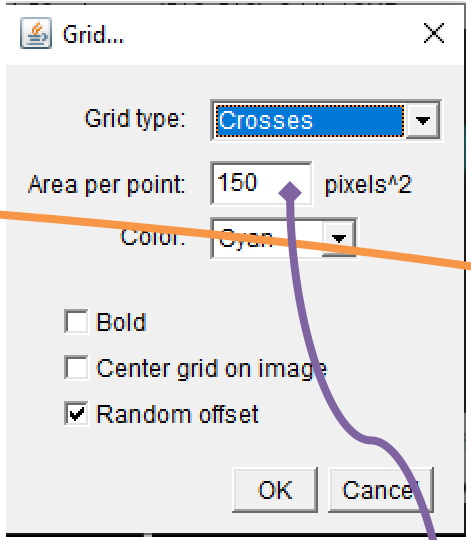
1. File > open
2. Image > stacks > Tools > reduce…

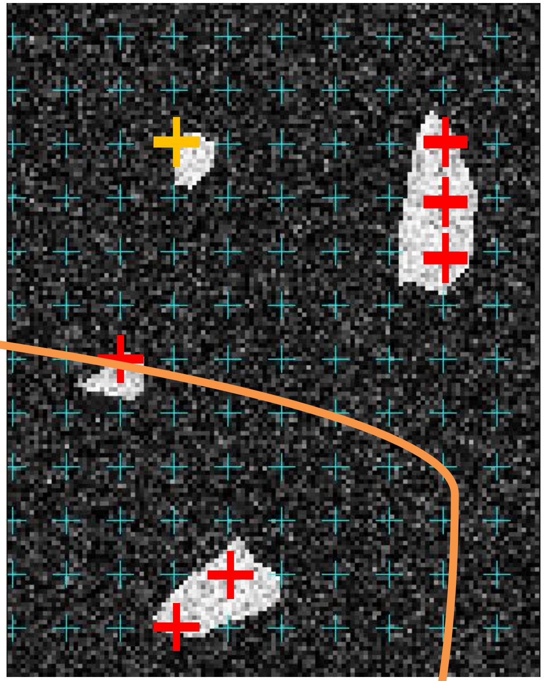


3. Process > Noise > Add specified noise… (yes, all slices)



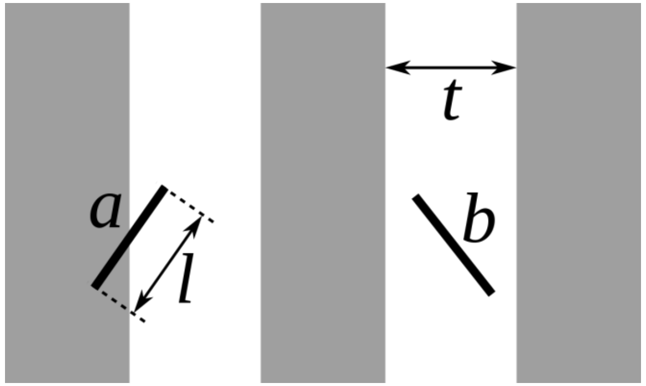
4. Analyze > tools > Grid…



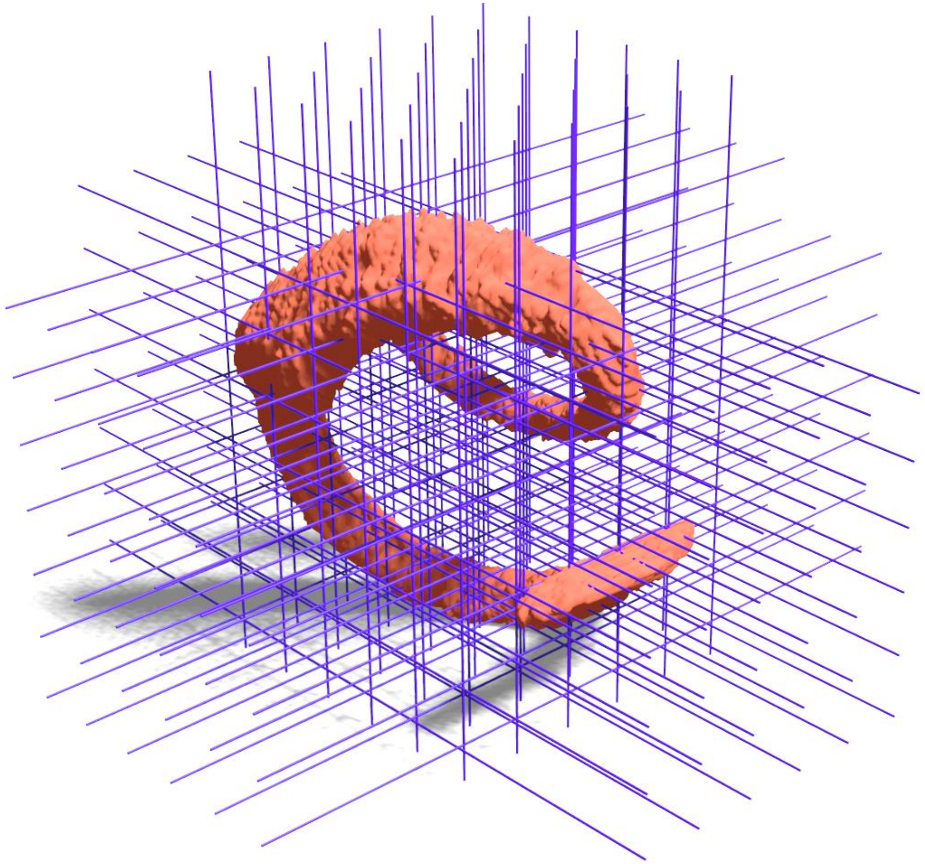


5. Count on all slices (in the example: 6 (7?))
6. Volume = count x Area per Point x Reduction factor

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

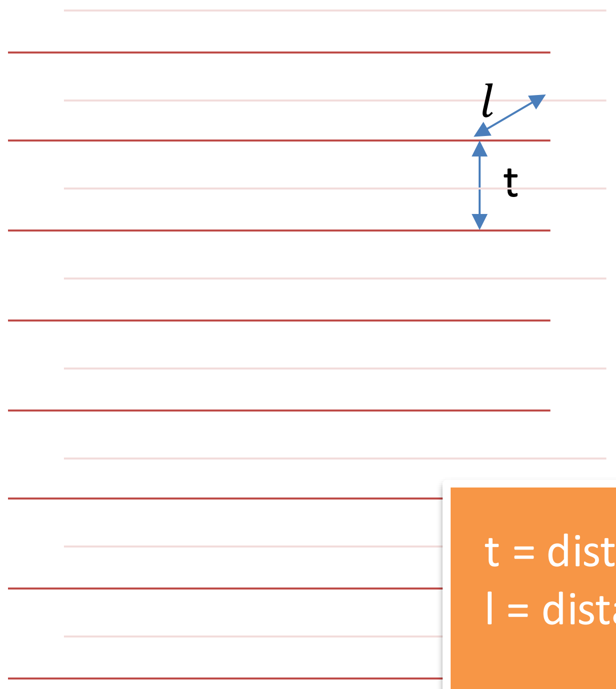
We show for the first time, to our knowledge, that ants can measure the size of potential nest sites. Nest size assessment is by individual scouts. Such scouts always make more than one visit to a potential nest before initiating an emigration of their nest mates and they deploy individual-specific trails 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

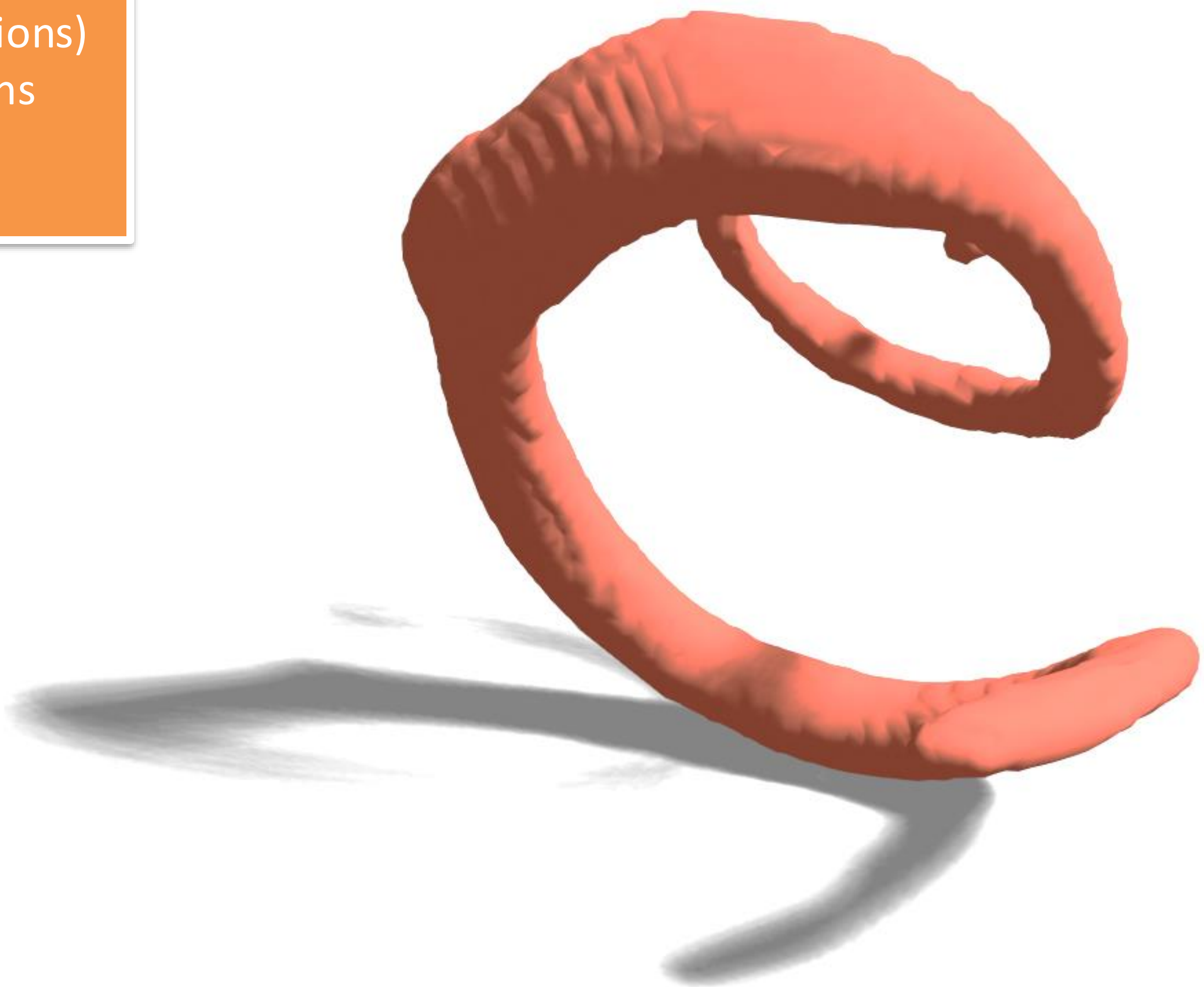
$$S = 2 \cdot \frac{1}{n} \cdot \sum_{i=1}^{n} \frac{v}{l_i} \cdot l_i$$

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



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

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



UNI FR UNIVERSITÉ DE FRIBOURG UNIVERSITÄT FREIBURG
BIO-INSPIRED MATERIALS NATIONAL CENTER OF COMPETENCE IN RESEARCH

Surface estimation with Buffon's needle

Cochlea XY	Cochlea YZ	Cochlea XZ
160	156	137

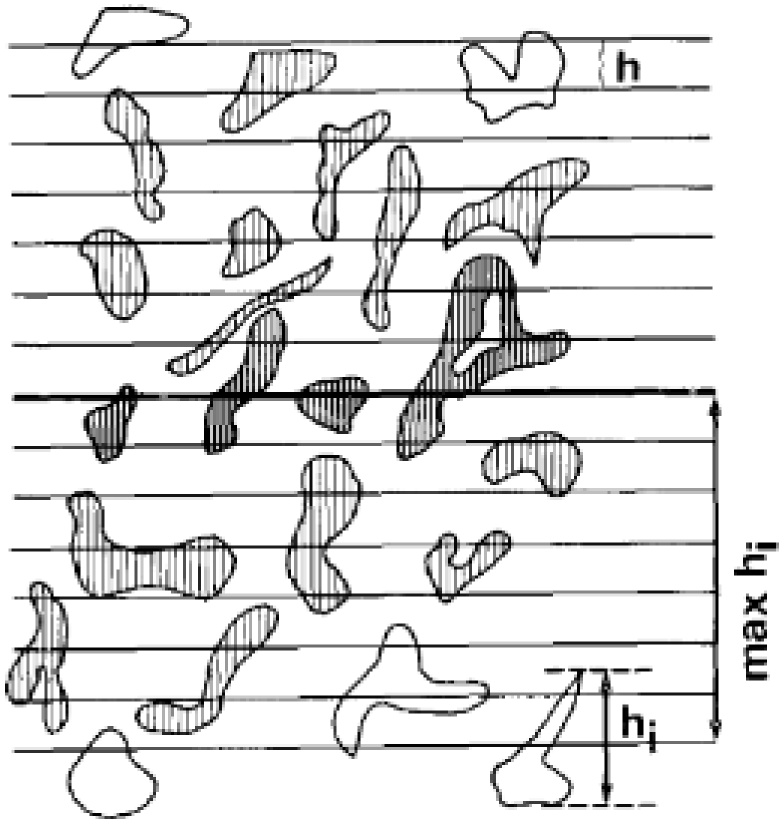
$$S = 2 \cdot \frac{1}{n} \cdot \sum_{i=1}^{n} \frac{v}{l_i} \cdot l_i$$

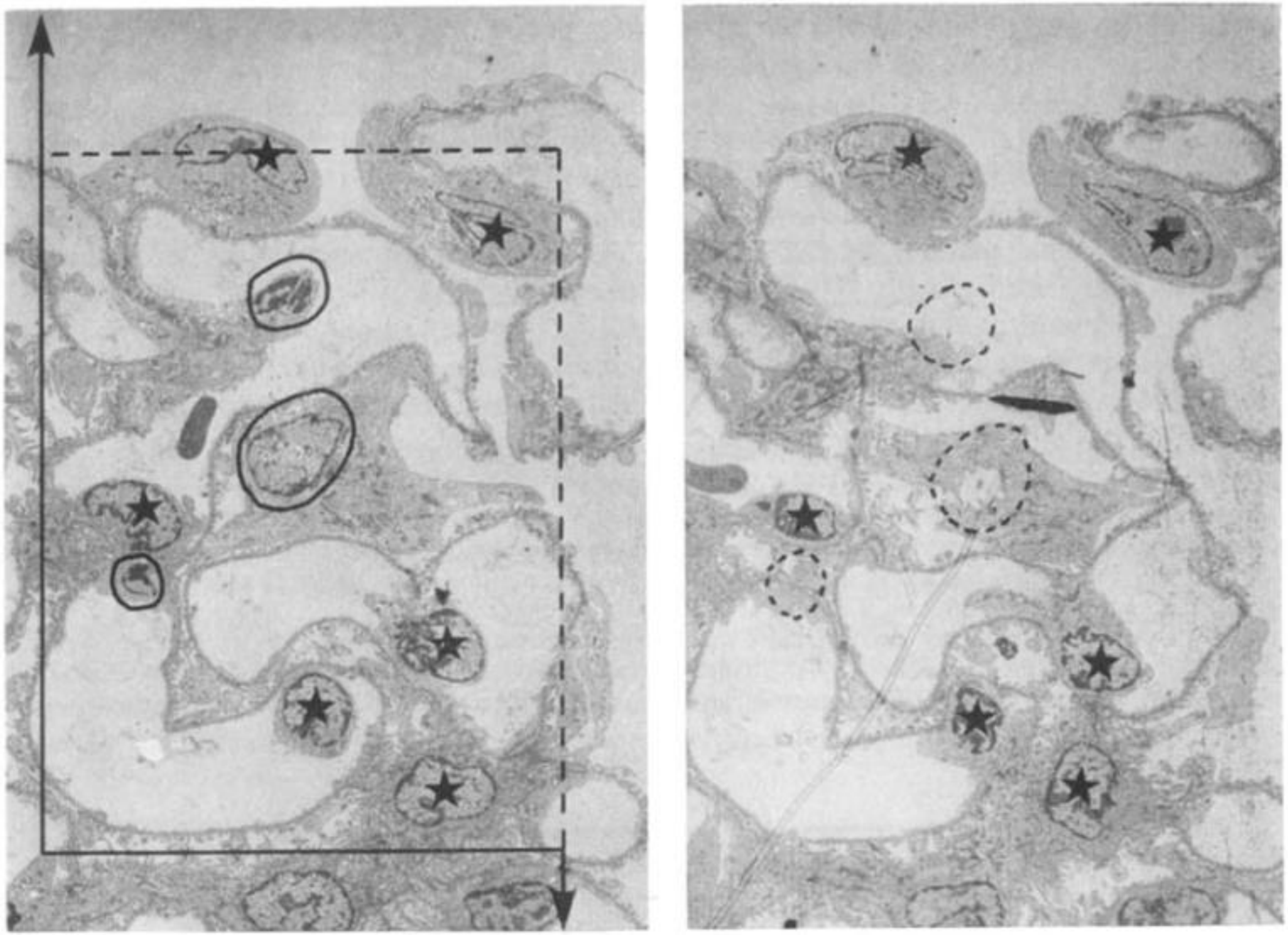
n = 3 (number of dimensions)
$\sum l_i$= 453
$\frac{v}{l_i}$= 100
→ 10 (distance between the grid lines) .
→ 10 (reducing factor, distance between adjacent slices, in pixels)

S = $\frac{2}{3} \cdot 453 \cdot 100$ = 30 200 px^2

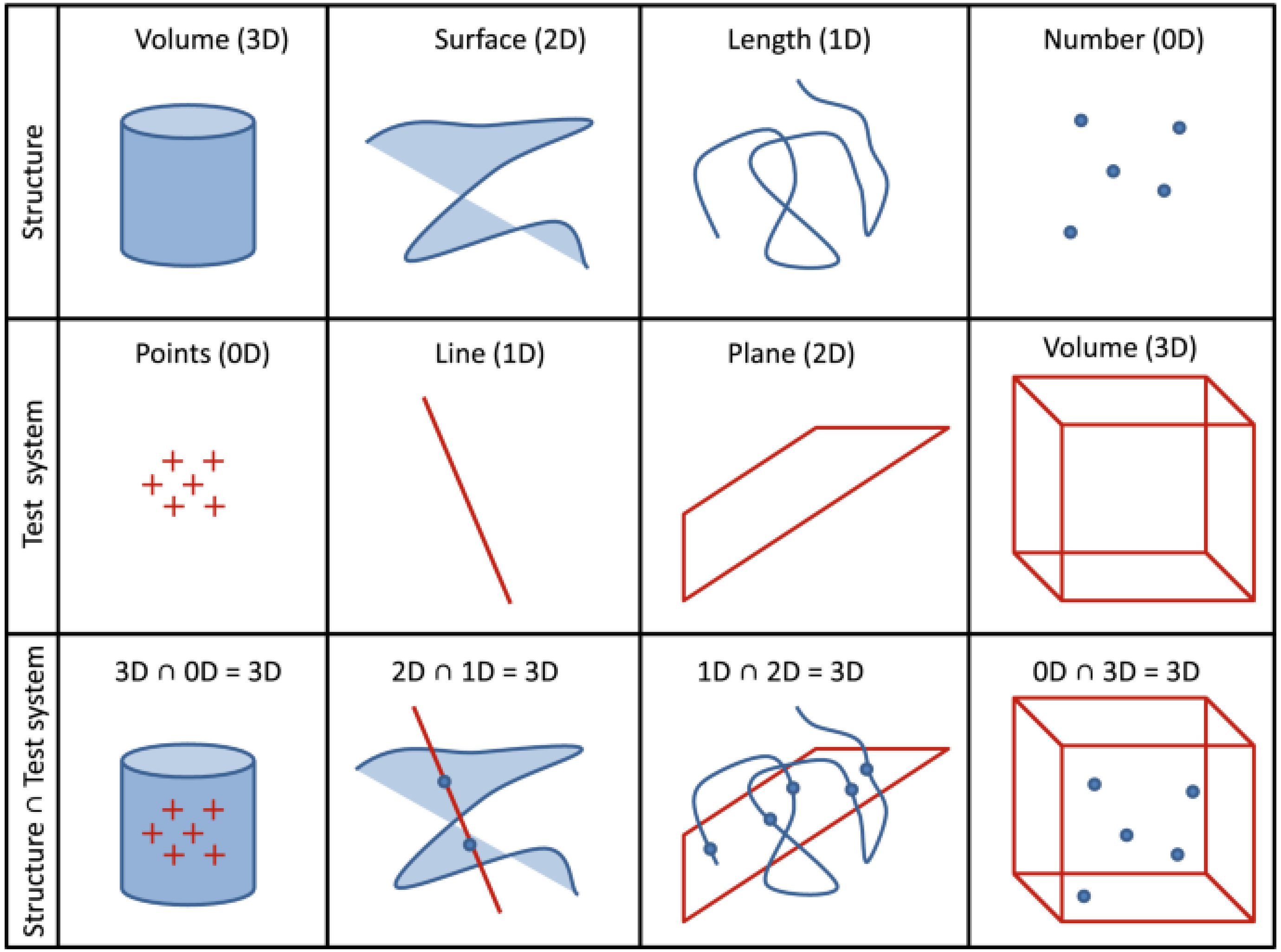
By software: S = 32 450 px^2

Number estimation with the disector



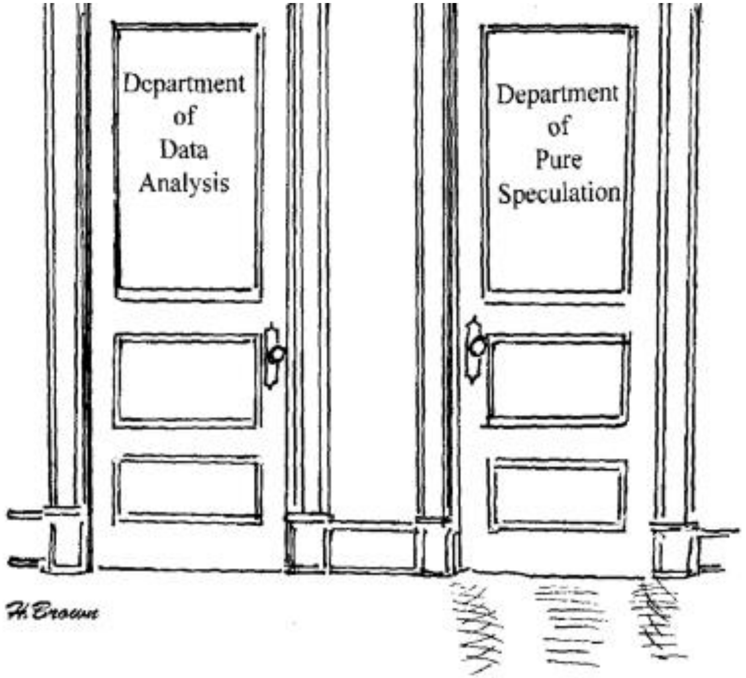


Stereology


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

✓ Congratulations,

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



UNI FR UNIVERSITÉ DE FRIBOURG UNIVERSITÄT FREIBURG

BIO-INSPIRED MATERIALS NATIONAL CENTER OF COMPETENCE IN RESEARCH