ScaDS.AI DRESDEN LEIPZIG
CENTER FOR SCALABLE DATA ANALYTICS
AND ARTIFICIAL INTELLIGENCE

Surface reconstruction

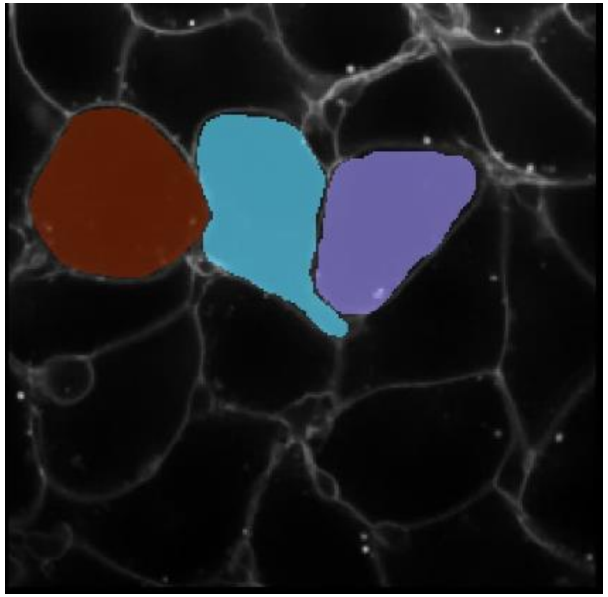
Robert Haase

Using materials from Alba Villaronga Luque and Jesse Veenvliet (MPI CBG Dresden), Marcelo Leomil Zoccoler, Johannes Soltwedel and Mara Lampert, PoL, TU Dresden

GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung

Diese Maßnahme wird gefördert durch die Bundesregierung aufgrund eines Beschlusses des Deutschen Bundestages. Diese Maßnahme wird mitfinanziert durch Steuermittel auf der Grundlage des von den Abgeordneten des Sächsischen Landtags beschlossenen Haushaltes.

Sparse Jaccard Index



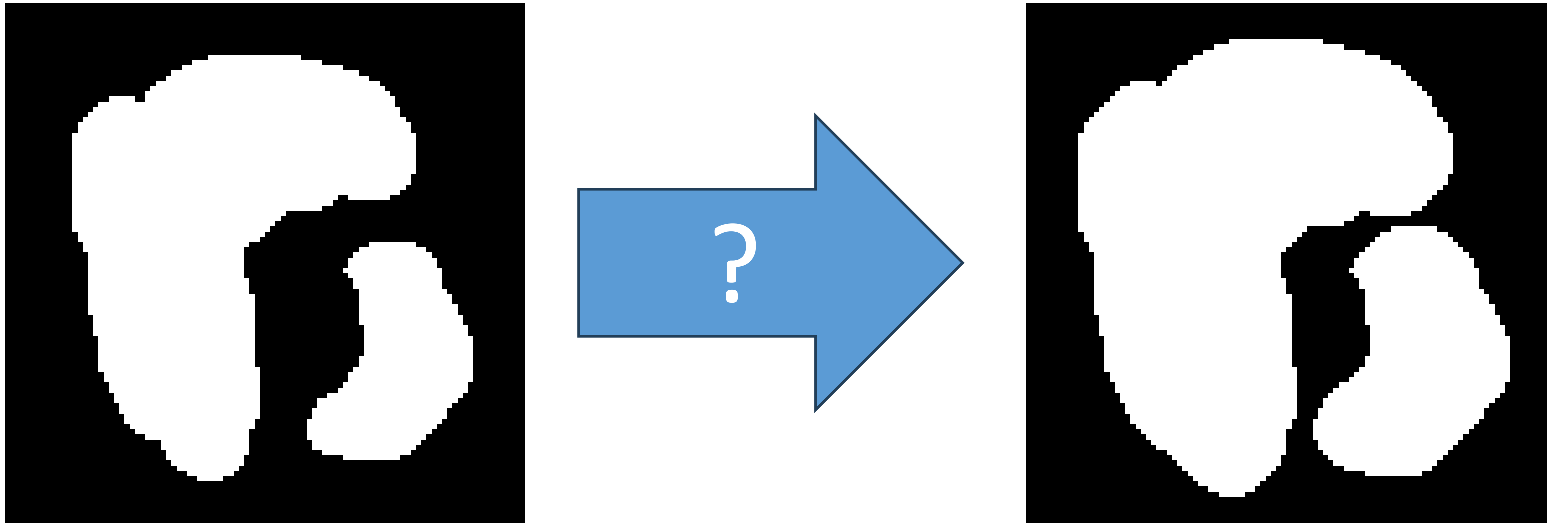
This is a …

Sparse instance segmentation

Sparse semantic segmentation

Quiz: Recap

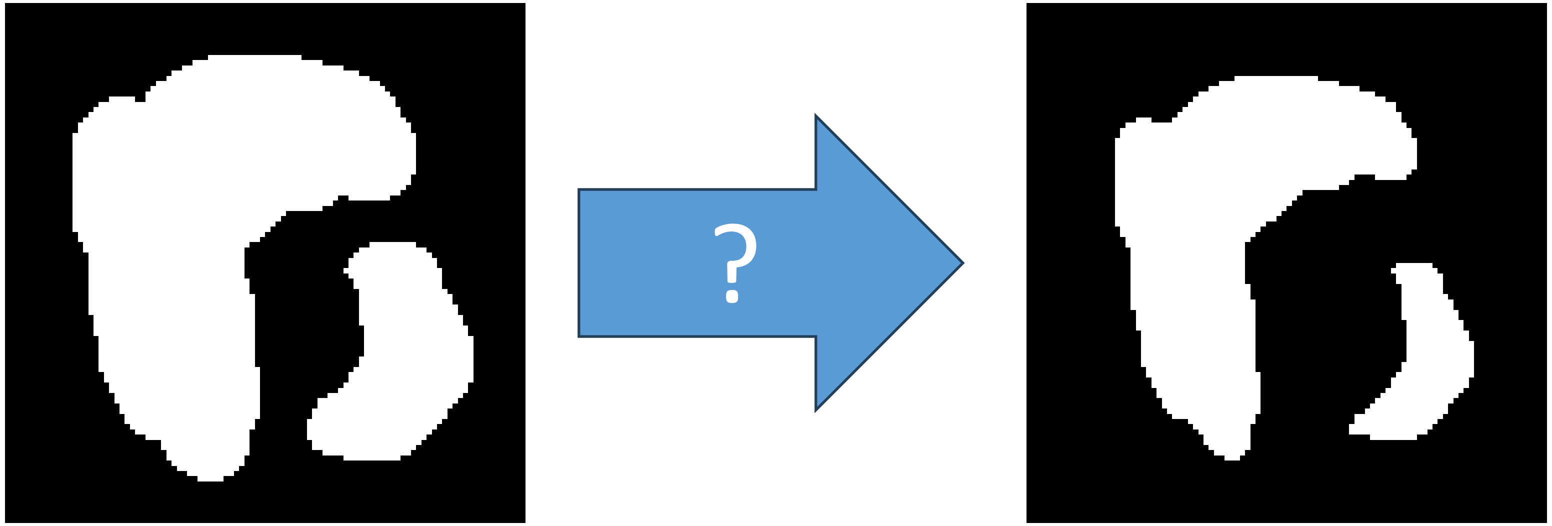
- How is this operation called?



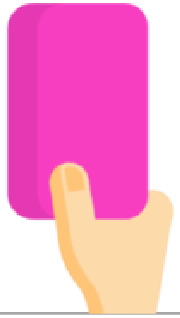
Dilation | Erosion | Opening | Closing

Quiz: Recap

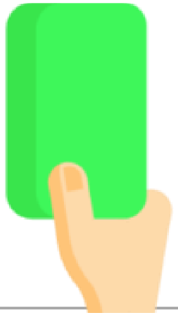
- How is this operation called?



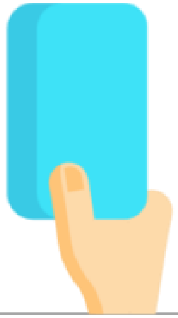
Dilation



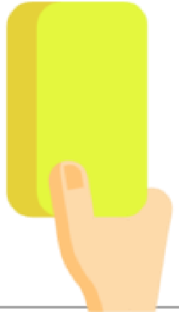
Erosion



Opening

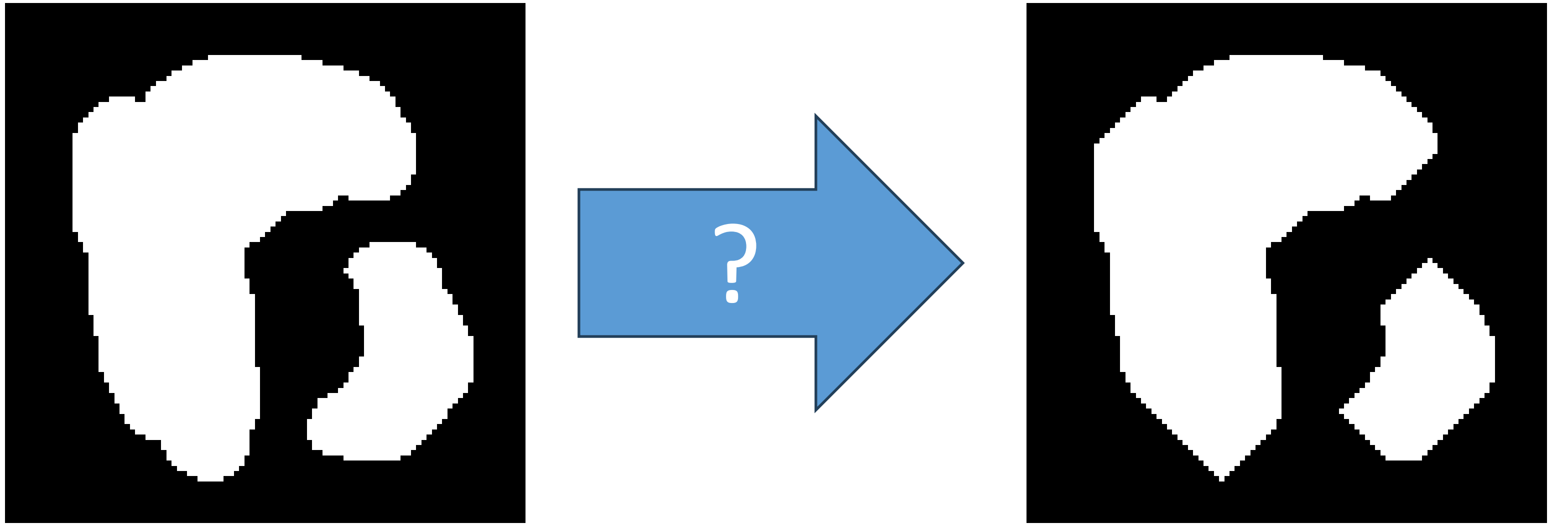


Closing



Quiz: Recap

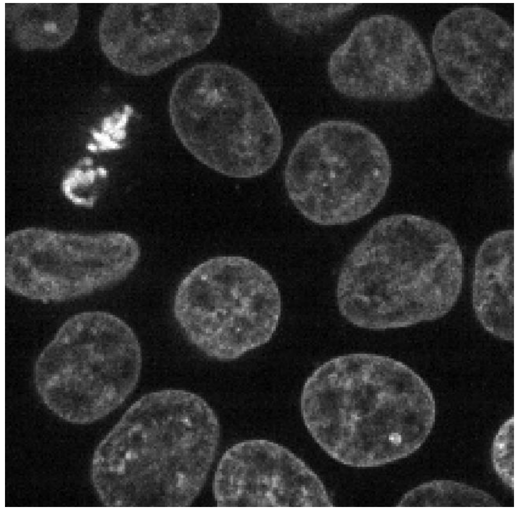
- How is this operation called?



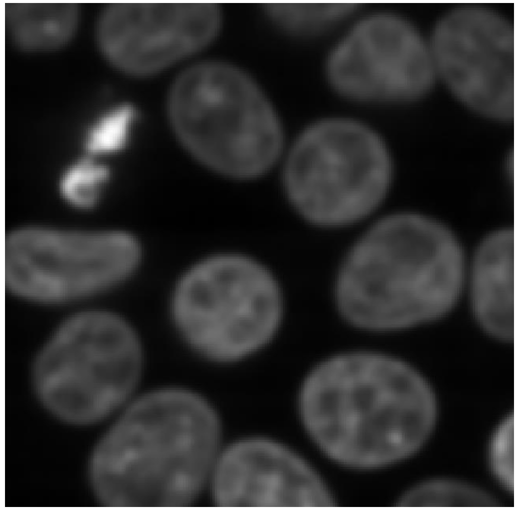
Dilation | Erosion | Opening | Closing

Motivation: Surface reconstruction

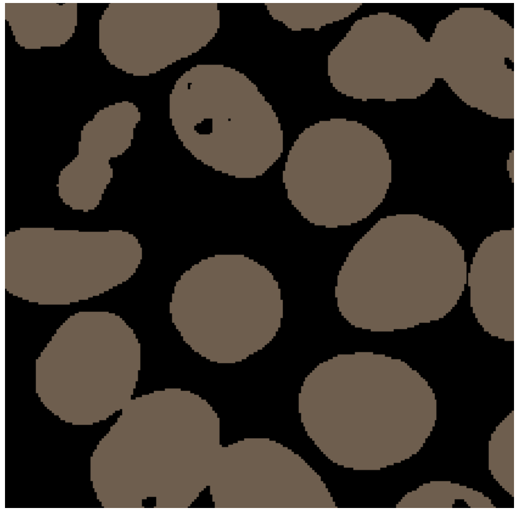
- Pixel and voxel arrays can be huge in memory
- Processing 3D arrays is time-consuming



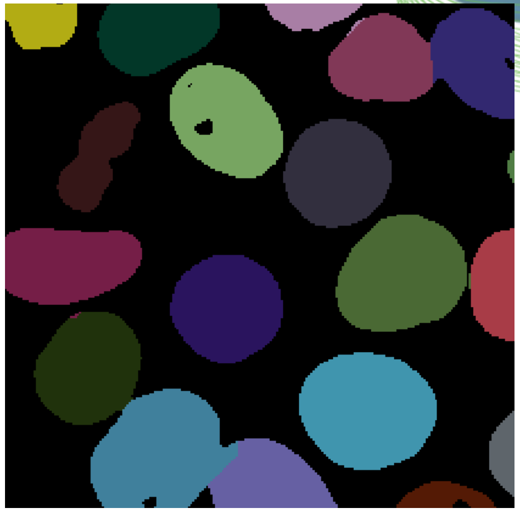
1024 x1024 x 100
16-bit image



1024 x1024 x 100
16-bit image



1024 x1024 x 100
8-bit image



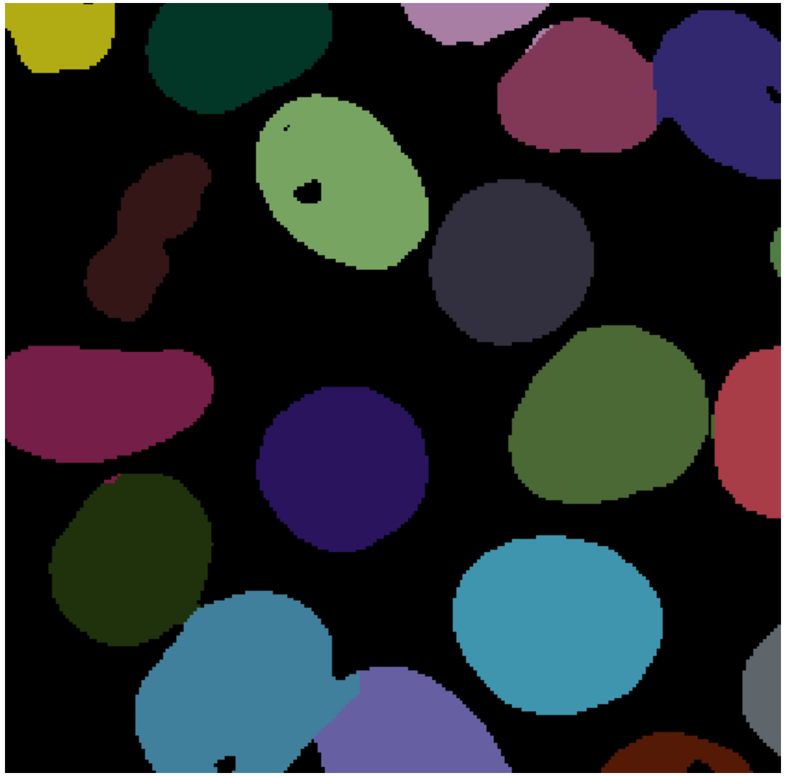
1024 x1024 x 100
16-bit image

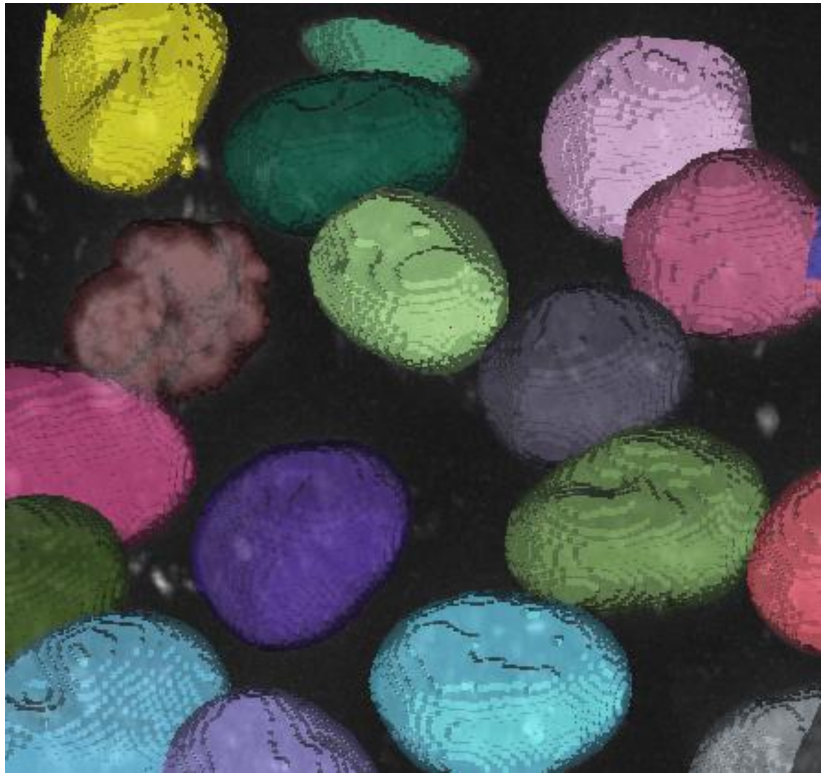
How much memory does this workflow cost?

700 MB	400 MB	4 GB	7 GB

Motivation: Surface reconstruction

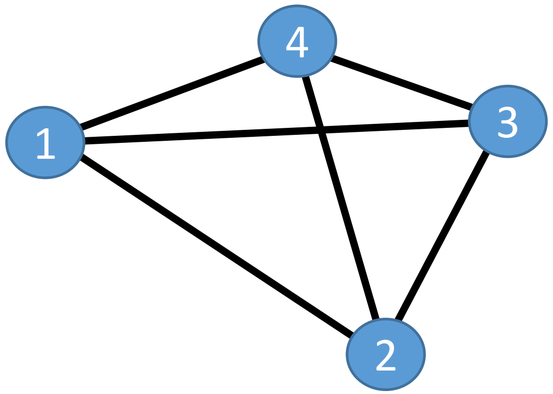
- Pixel and voxel borders introduce artifacts, potentially problematic for measurements, e.g. surface area

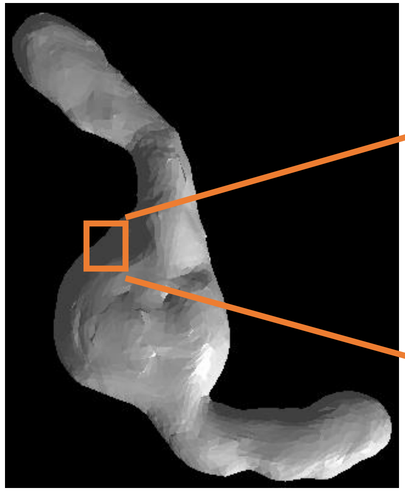




Surface meshes

- Points on a surfaces connected by triangles forma a surface mesh





"Vertices" / points

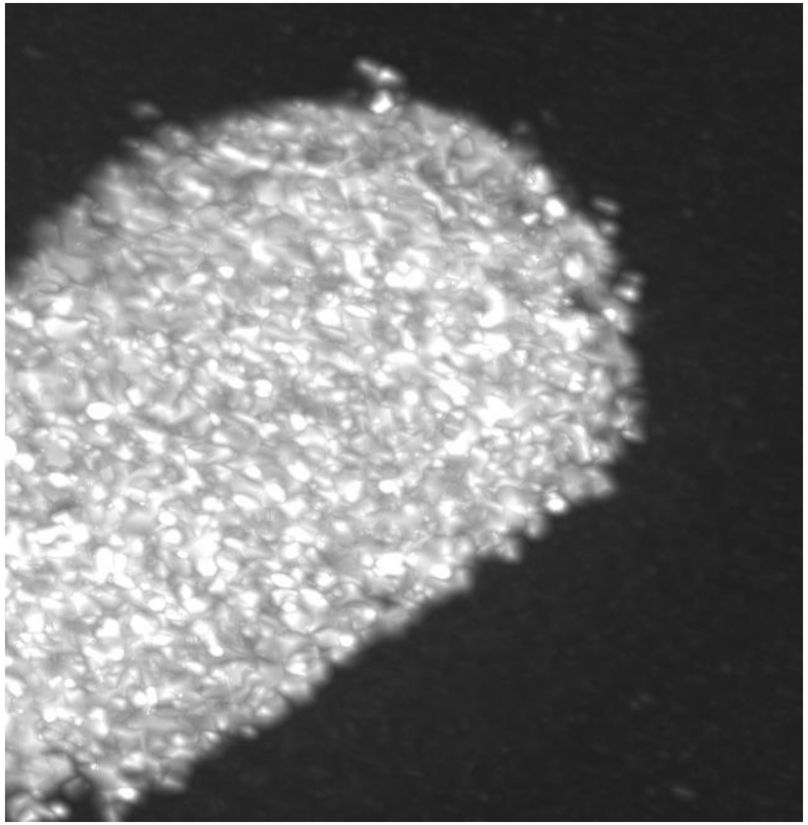
Point x	Point y	Point z
x_1	y_1	z_1
X_2	Y_2	Z_2
X_3	Y_3	Z_3
X_4	Y_4	Z_4
…	…	…

+

"Faces" / Triangles

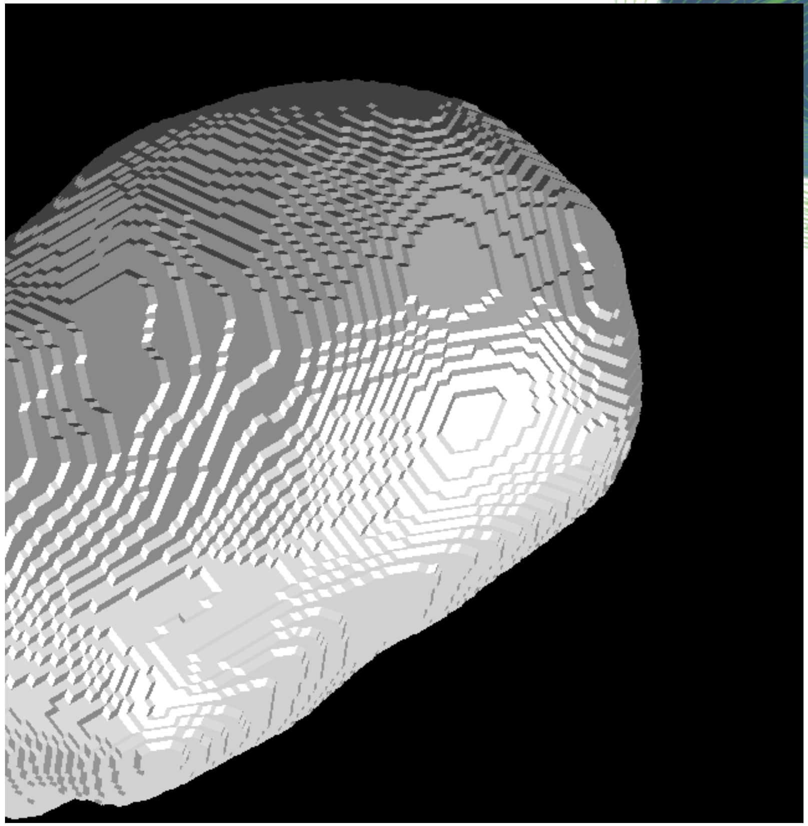
Point 1	Point 2	Point 3
1	2	3
1	2	4
2	3	4
1	3	4

Surface reconstruction



3D image of nuclei

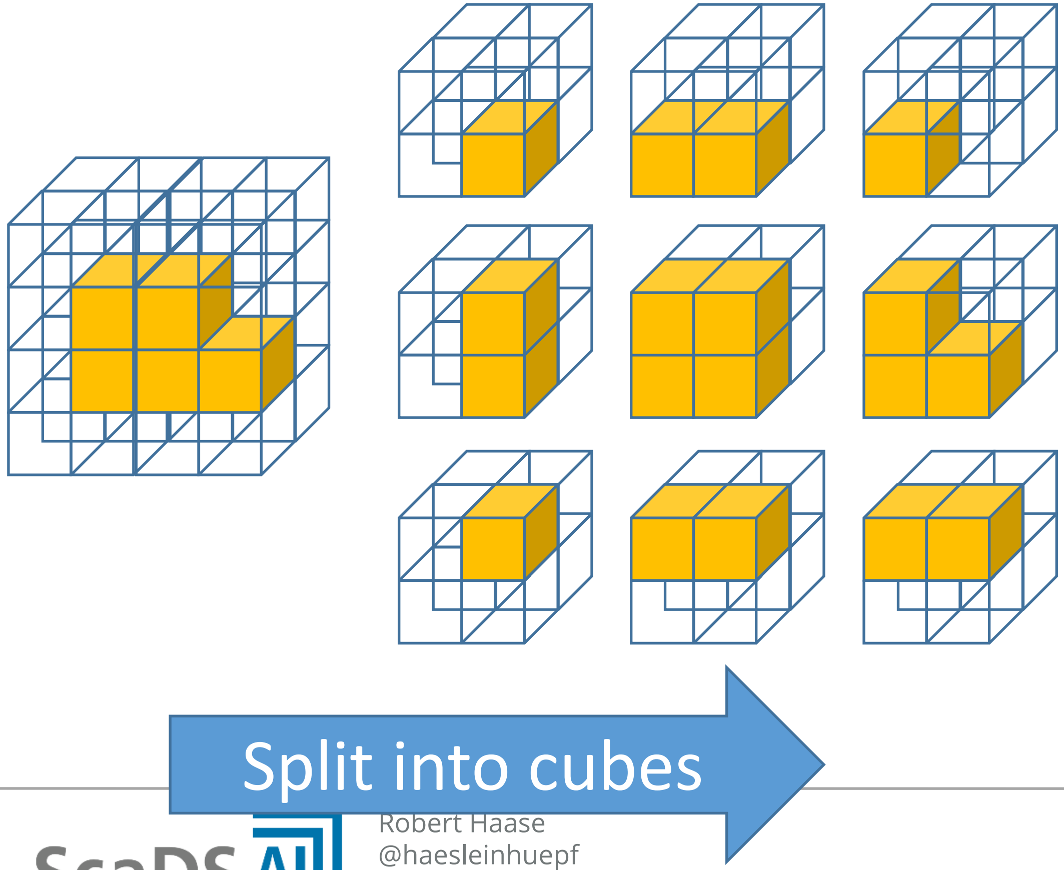
Gaussian filtered



Binary 3D image
(visualized as surface mesh)

Marching cubes algorithm

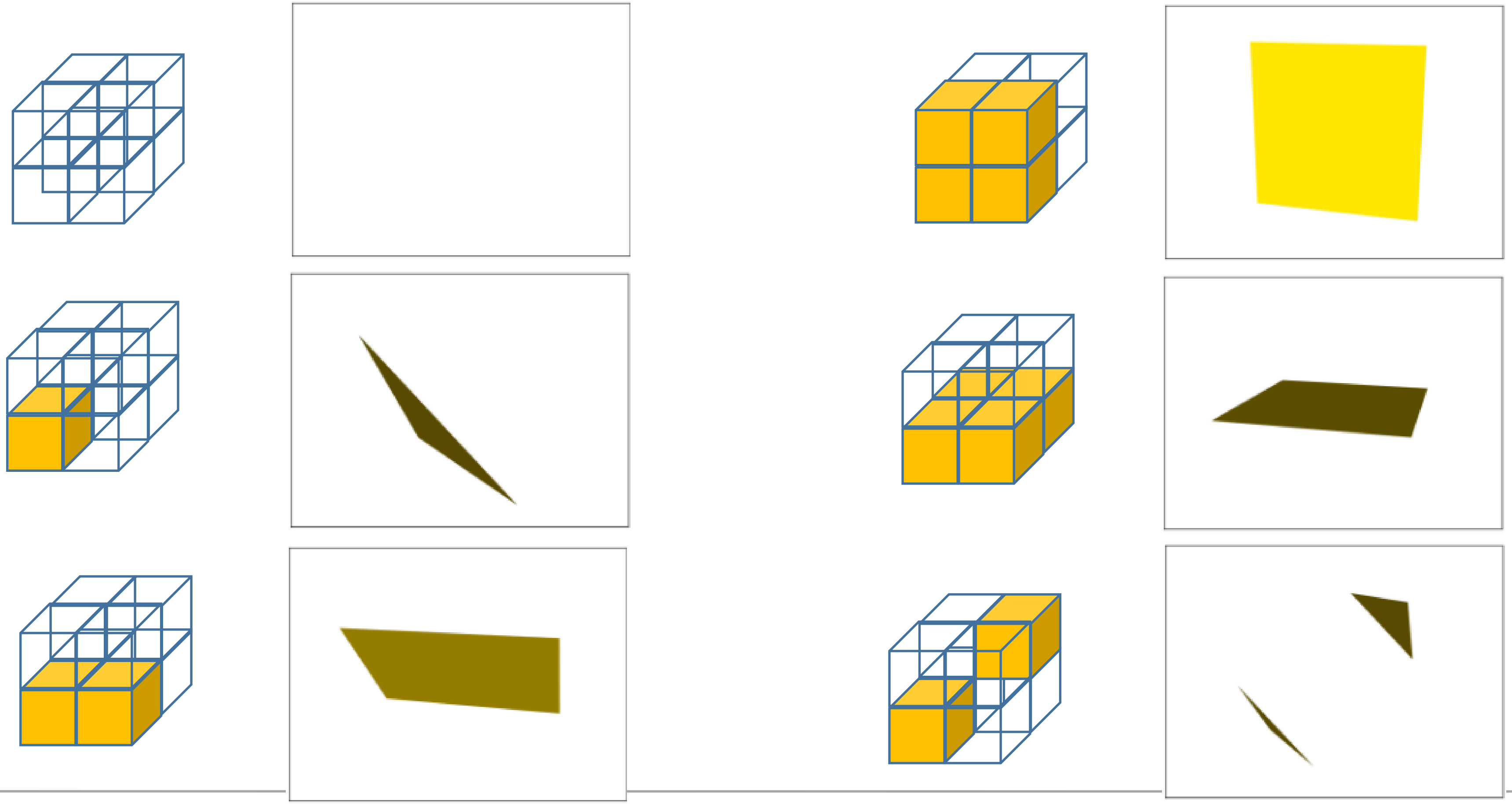
- Starting point: 3D binary image
- Cuts the image in small cubes and iterates over them



Lorensen, William E.; Cline, Harvey E. (1 August 1987). "Marching cubes: A high resolution 3D surface construction algorithm". *ACM SIGGRAPH Computer Graphics*. **21** (4): 163-169. CiteSeerX 10.1.1.545.613. doi:10.1145/37402.37422.

Marching cubes algorithm

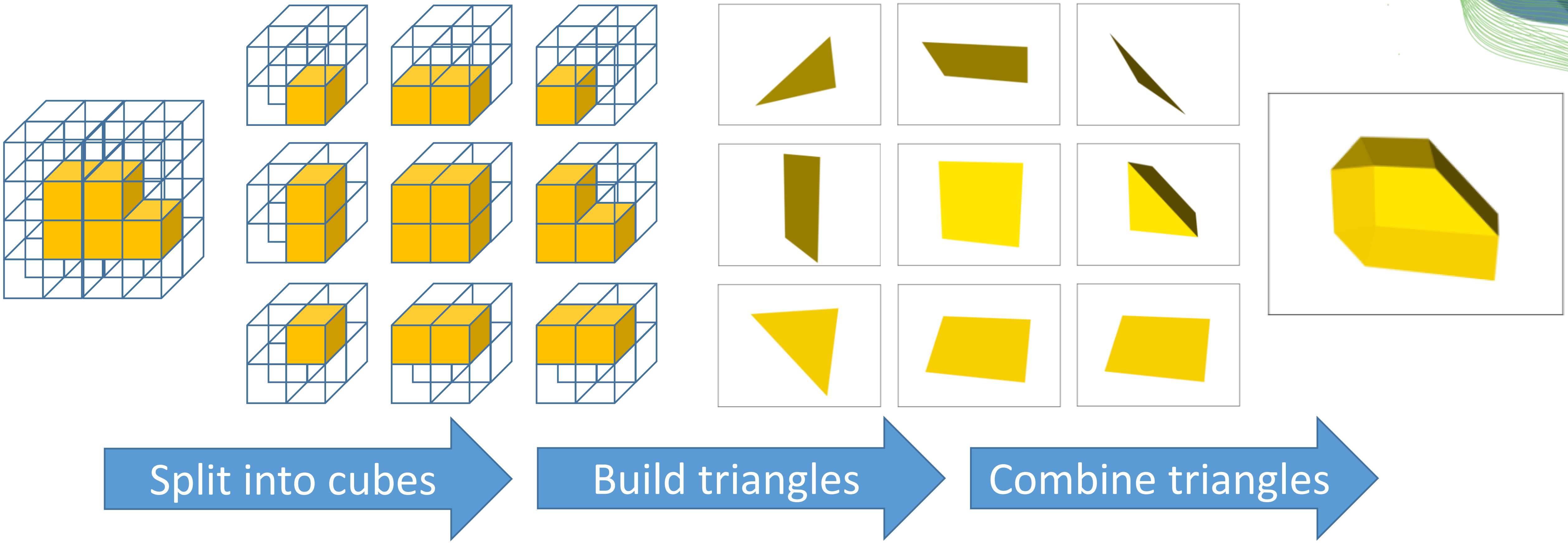
- Starting point: 3D binary image
- Cuts the image in small cubes and iterates over them



Lorensen, William E.; Cline, Harvey E. (1 August 1987). "Marching cubes: A high resolution 3D surface construction algorithm". *ACM SIGGRAPH Computer Graphics*. **21** (4): 163-169. CiteSeerX 10.1.1.545.613. doi:10.1145/37402.37422.

Marching cubes algorithm

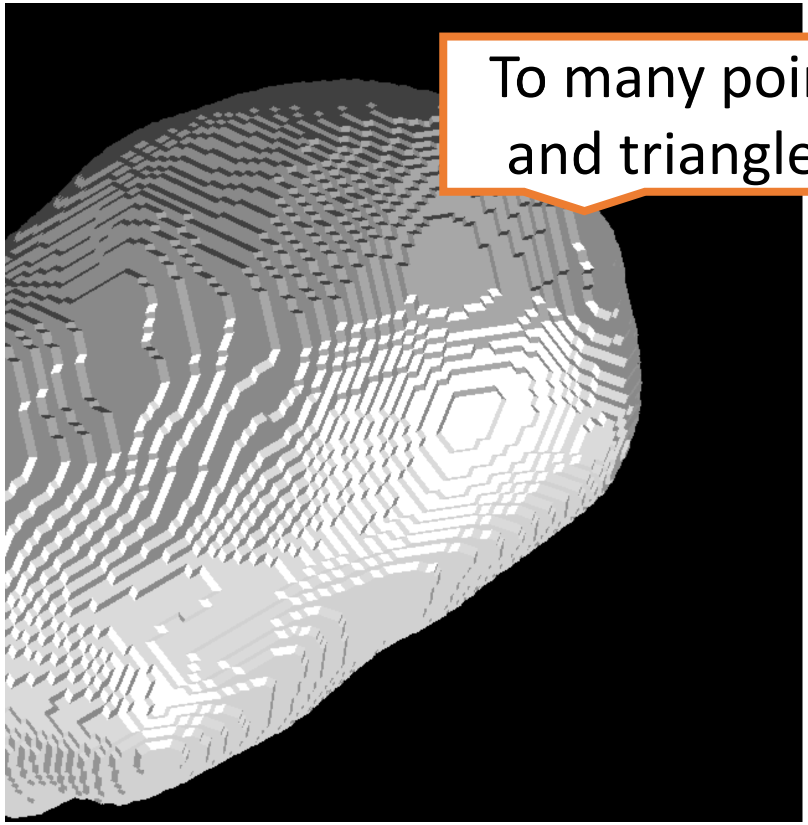
- Starting point: 3D binary image
- Cuts the image in small cubes and iterates over them



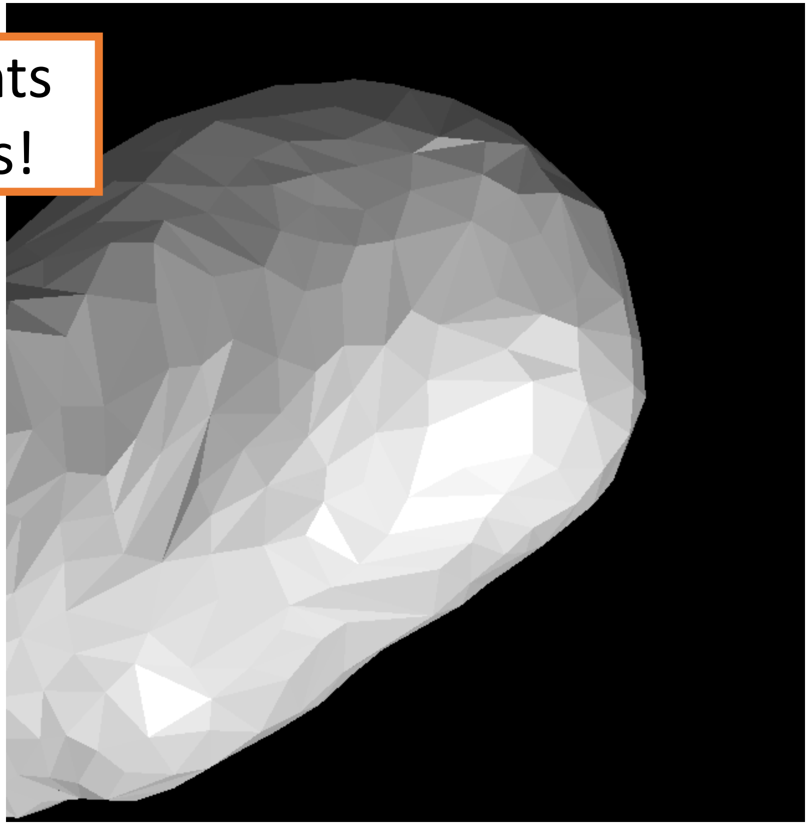
Lorensen, William E.; Cline, Harvey E. (1 August 1987). "Marching cubes: A high resolution 3D surface construction algorithm". *ACM SIGGRAPH Computer Graphics*. **21** (4): 163-169. CiteSeerX 10.1.1.545.613. doi:10.1145/37402.37422.

Surface post-processing

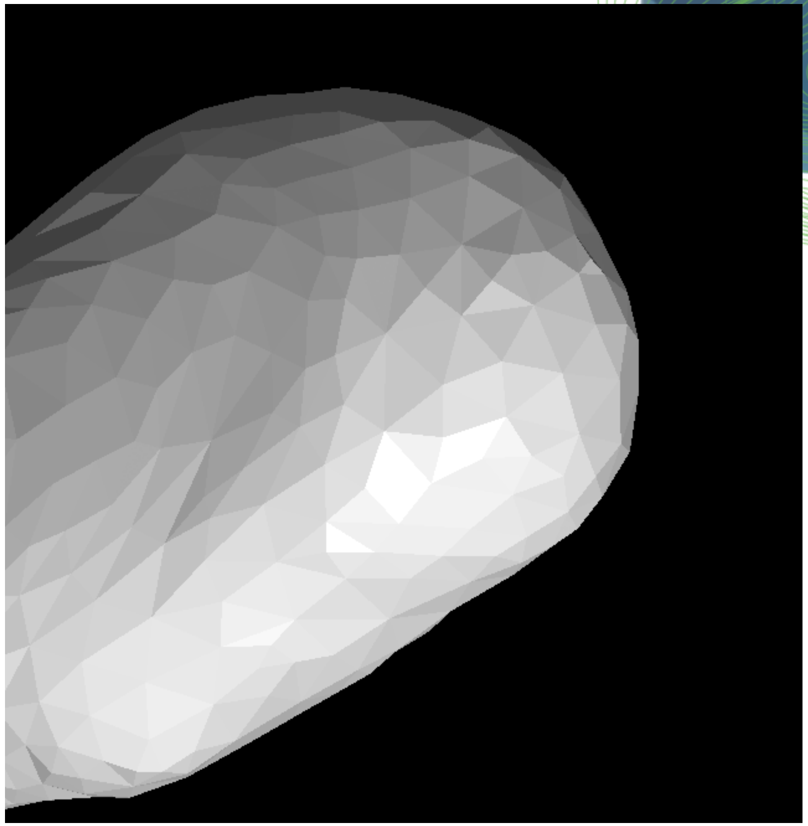
- Necessary to better match biological reality.



Marching cubes result



Simplified mesh
(less points, locally averaged)



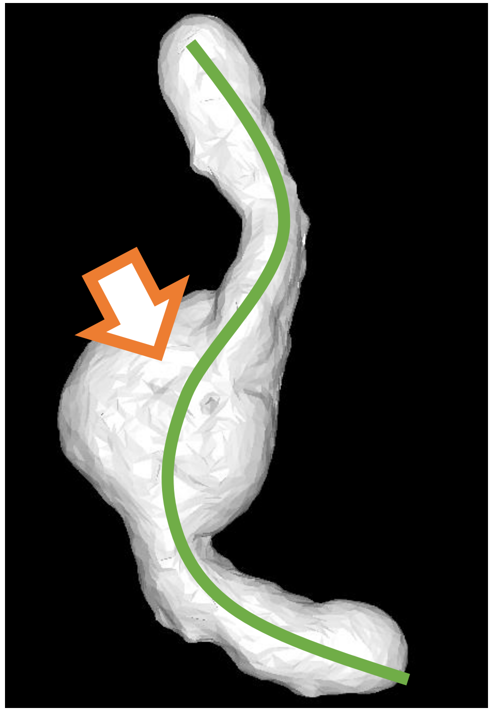
Smoothed mesh
(position locally planarized)

Surface post-processing

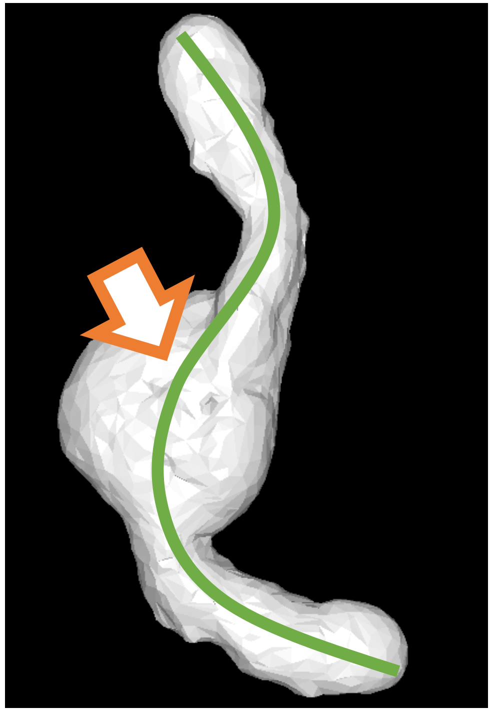
- Every processing step has consequences errors of later measurements
- Depends on desired measurement

Number of small concave regions

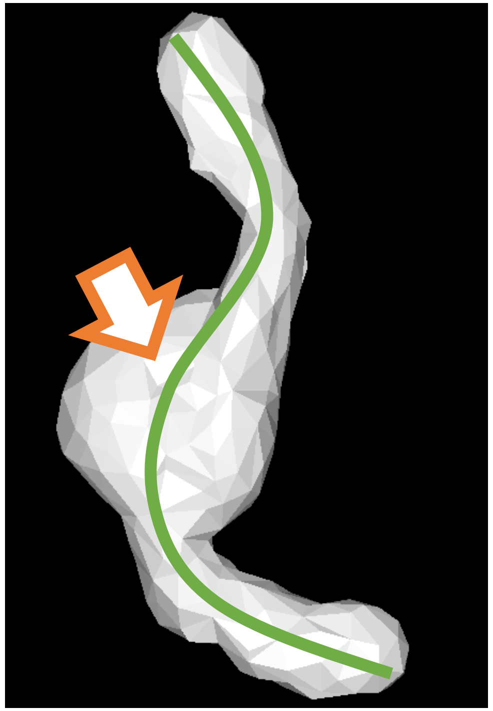
Total length



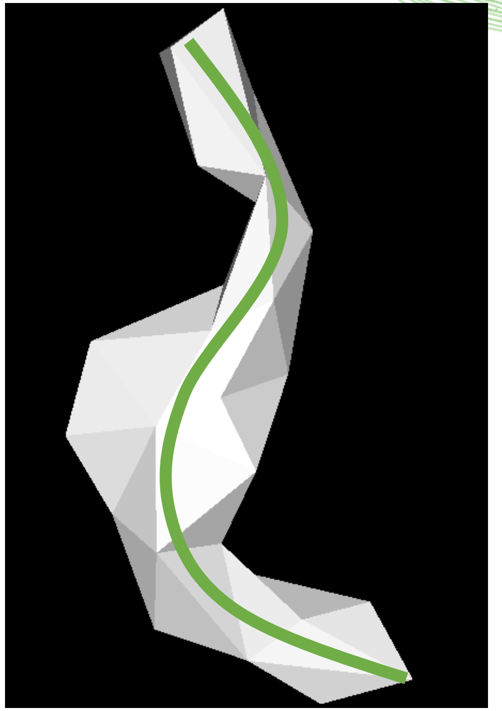
Surface mesh



Simplified by factor 0.5



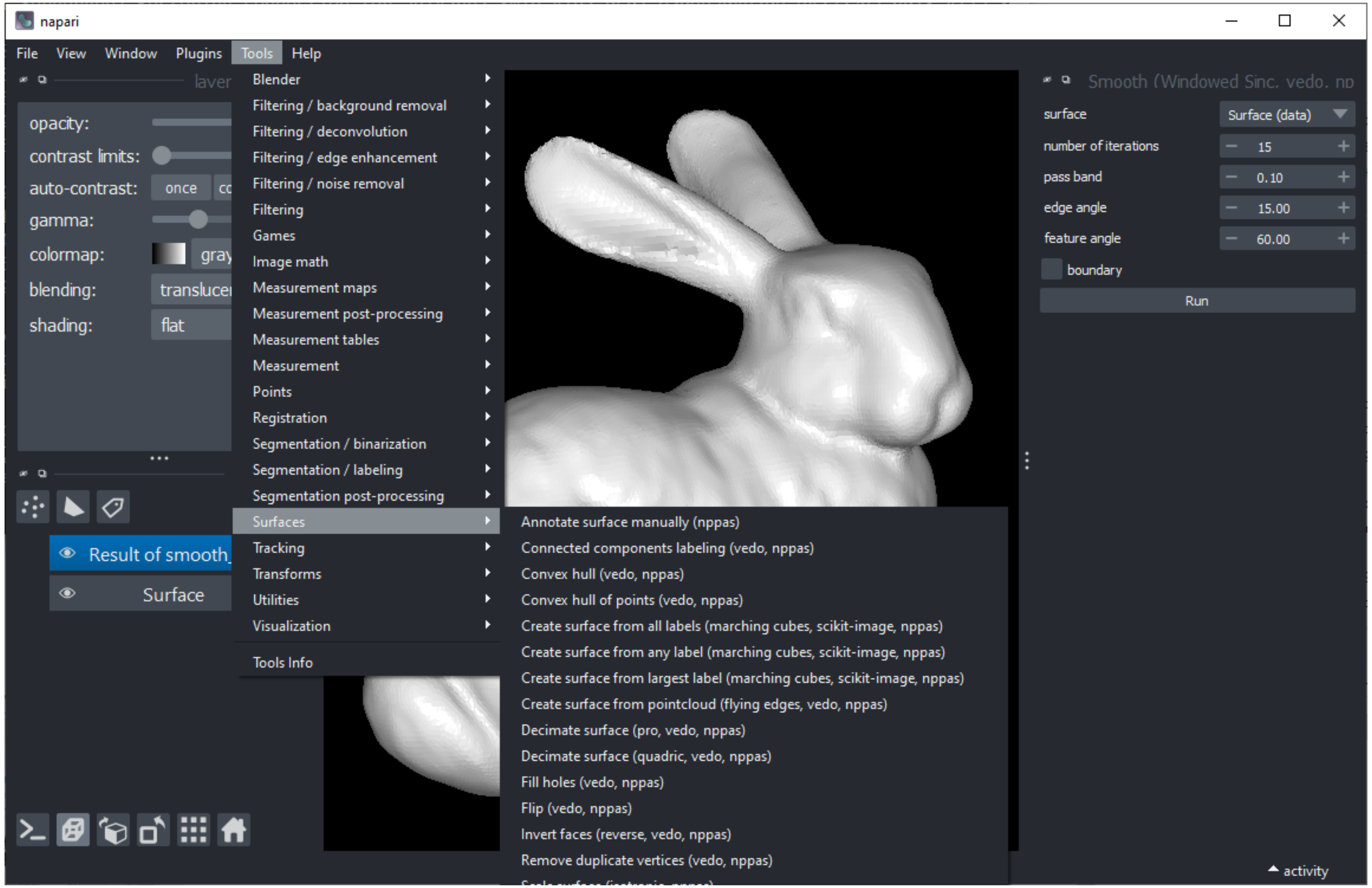
Simplified by factor 0.05



Simplified by factor 0.01

Surface reconstruction / Processing

- Tools > Surfaces > Create surface ...



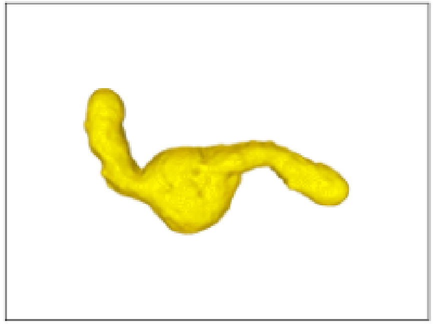
You need to install an extra napari-plugin: https://github.com/haesleinhuepf/napari-process-points-and-surfaces

Surface reconstruction

- Turn binary and/or label images into surface meshes

```
surface = nppas.all_labels_to_surface(binary_filled)
```

```
surface
```



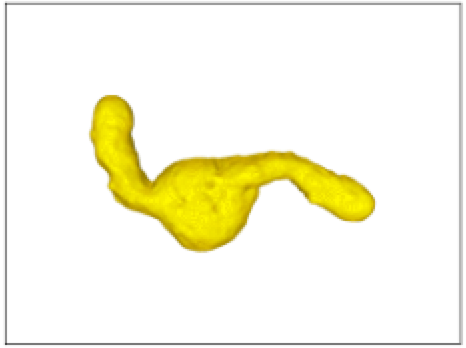
nppas.SurfaceTuple

origin (z/y/x)	[0. 0. 0.]
center of mass(z/y/x)	57.710,309.963,440.042
scale(z/y/x)	1.000,1.000,1.000
bounds (z/y/x)	12.500...113.500 111.500...461.500 169.500...807.500
average size	170.769
number of vertices	330776
number of faces	661548

Surface mesh processing

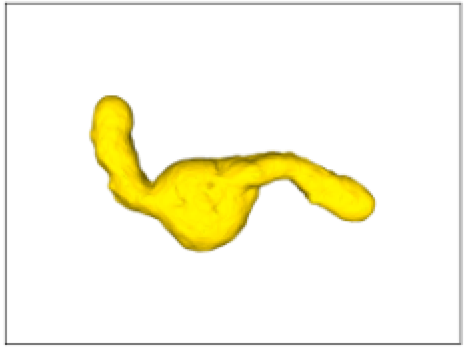
- Surface mesh simplification
- To prevent the computer freezing

```
simplified_surface = nppas.decimate_quadric(surface, fraction=0.01)
simplified_surface
```



nppas.SurfaceTuple

origin (z/y/x)	[0. 0. 0.]
center of mass(z/y/x)	57.710,309.963,440.042
scale(z/y/x)	1.000,1.000,1.000
bounds (z/y/x)	12.500...113.500 111.500...461.500 169.500...807.500
average size	170.769
number of vertices	330776
number of faces	661548



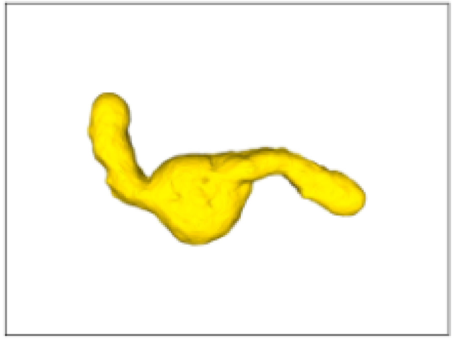
nppas.SurfaceTuple

origin (z/y/x)	[0. 0. 0.]
center of mass(z/y/x)	57.928,308.938,440.985
scale(z/y/x)	1.000,1.000,1.000
bounds (z/y/x)	13.231...113.510 111.642...461.602 170.022...806.468
average size	170.083
number of vertices	3310
number of faces	6615

Surface mesh processing

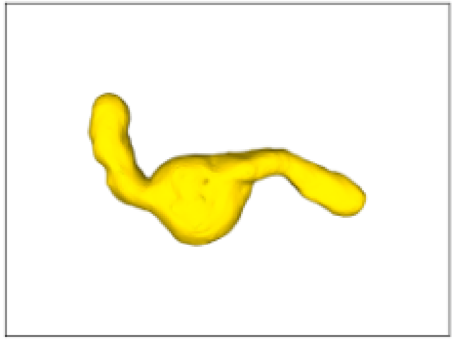
- Surface mesh smoothing

```
smoothed_surface = nppas.smooth_surface(simplified_surface)
smoothed_surface
```



nppas.SurfaceTuple

origin (z/y/x)	[0. 0. 0.]
center of mass(z/y/x)	57.928,308.938,440.985
scale(z/y/x)	1.000,1.000,1.000
bounds (z/y/x)	13.231...113.510 111.642...461.602 170.022...806.468
average size	170.083
number of vertices	3310
number of faces	6615



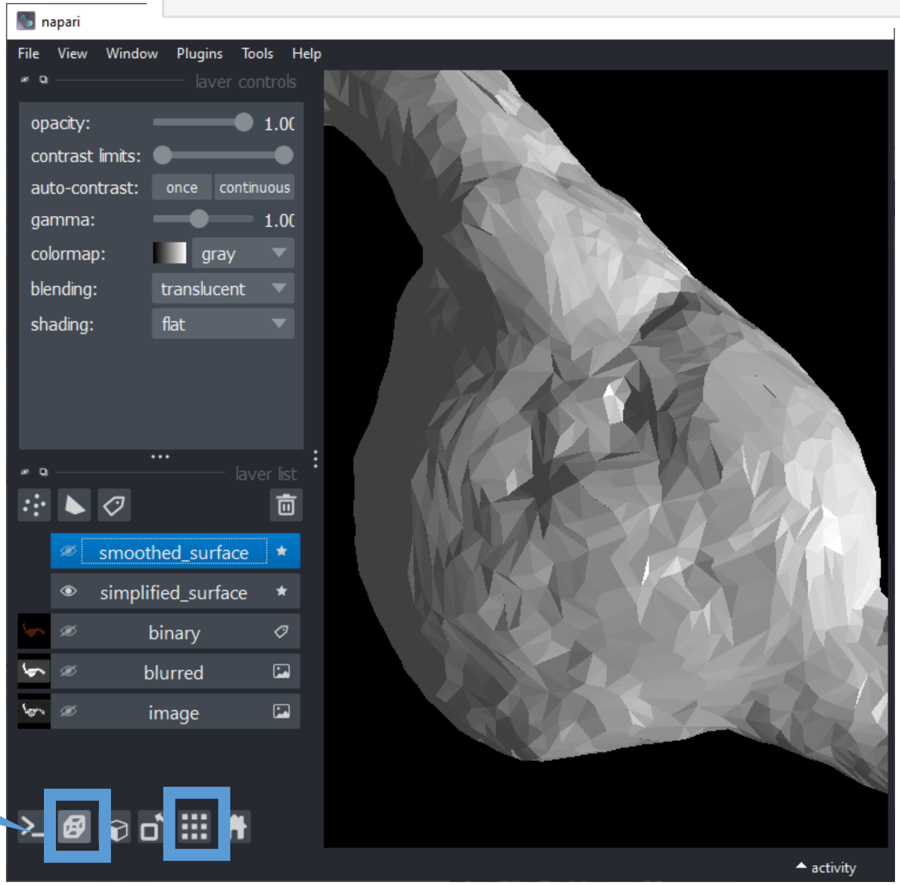
nppas.SurfaceTuple

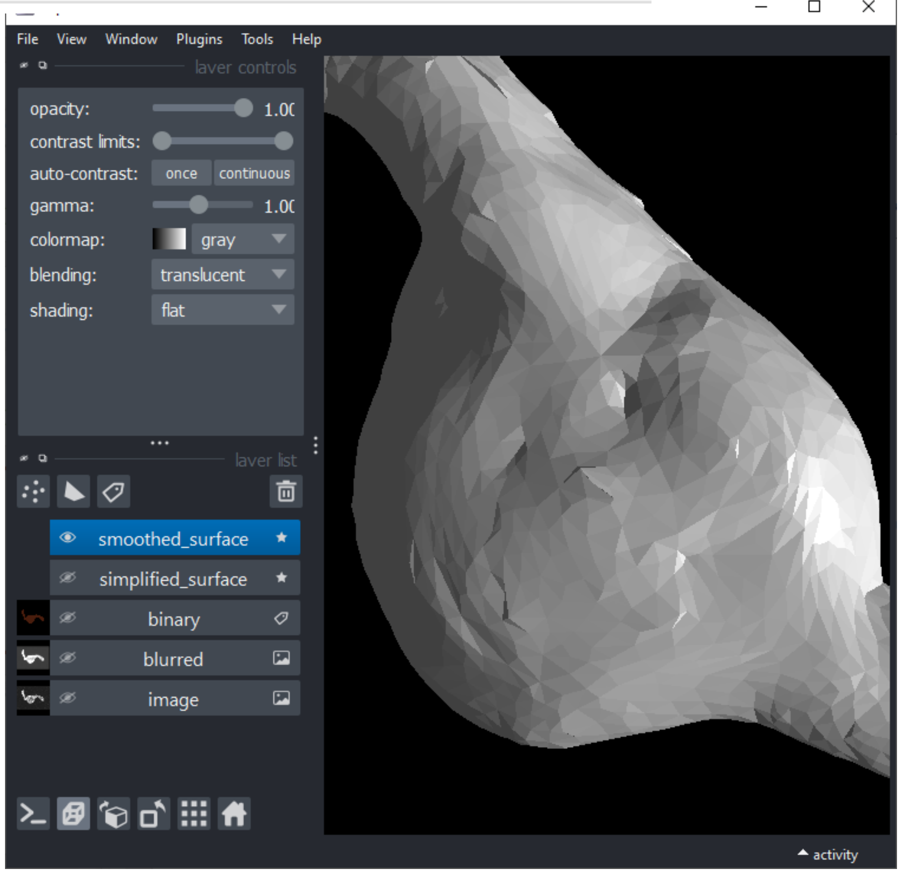
origin (z/y/x)	[0. 0. 0.]
center of mass(z/y/x)	57.913,308.988,440.878
scale(z/y/x)	1.000,1.000,1.000
bounds (z/y/x)	13.901...113.627 110.982...461.191 169.711...807.193
average size	170.378
number of vertices	3310
number of faces	6615

View surface meshes in Napari

```
viewer.add_surface(surface, scale=[zoom, zoom, zoom])
viewer.add_surface(simplified_surface, scale=[zoom, zoom, zoom])
viewer.add_surface(smoothed_surface, scale=[zoom, zoom, zoom])
```

In case your computer freezes, comment out this line





Toggle 3D view and grid mode

ScaDS.AI
DRESDEN LEIPZIG
CENTER FOR SCALABLE DATA ANALYTICS
AND ARTIFICIAL INTELLIGENCE

Segmentation quality estimation

Robert Haase

Reusing materials from Lena Maier-Hein, Annika Reinke (DKFZ) et al. and Martin Schätz (Charles Uni Prague)

GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung

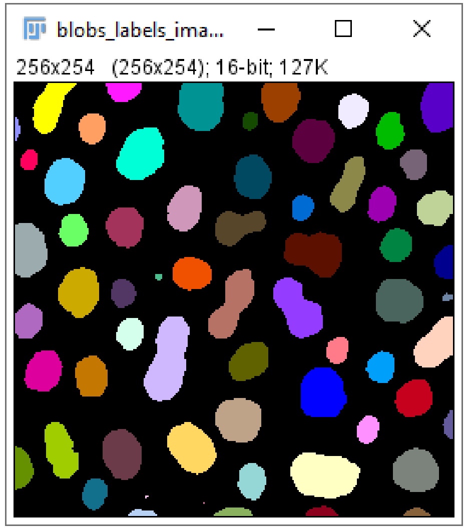
SACHSEN
Diese Maßnahme wird gefördert durch die Bundesregierung aufgrund eines Beschlusses des Deutschen Bundestages. Diese Maßnahme wird mitfinanziert durch Steuermittel auf der Grundlage des von den Abgeordneten des Sächsischen Landtags beschlossenen Haushaltes.

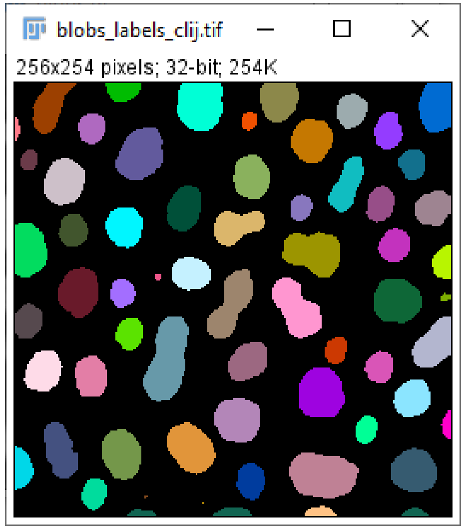
Goal

- Compare label images quantitatively, to know
  - how “good” a segmentation algorithms is and/or
  - how “variable” segmentations (from humans or computers are)

How can we know if these results are the same?

Human annotation (“ground truth”)





Algorithm result

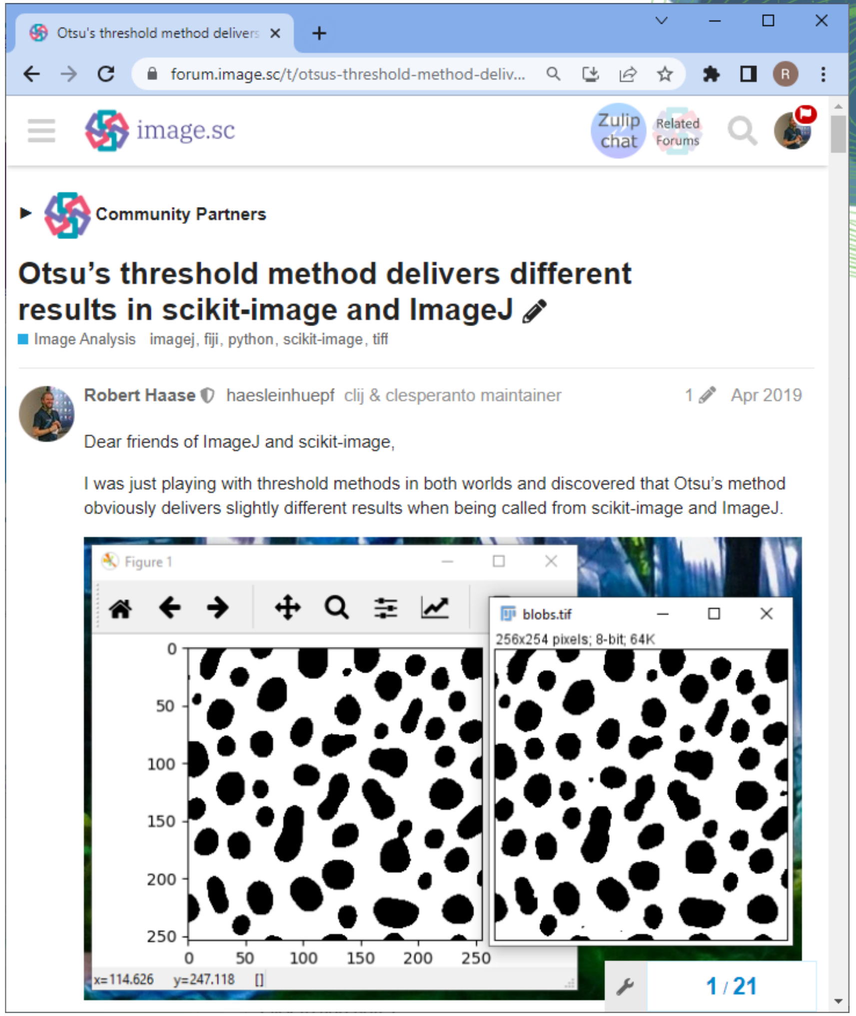
Why do results vary?

Potential reasons of same workflows delivering different results:

- Image data type (8/16/32-bit float/int)
- Workflow implementation
- How histograms are determined
- How the threshold is determined from the histogram

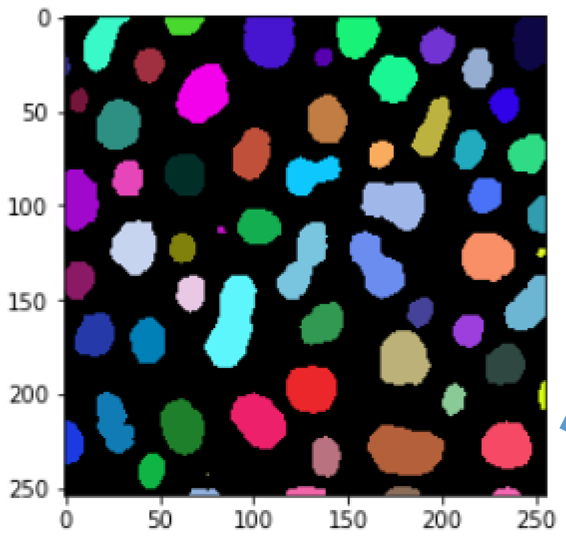
- Compute architecture (CPU, GPU, TPU, …)
- Hardware vendor
- Software / driver versions

Lecture in 2 weeks

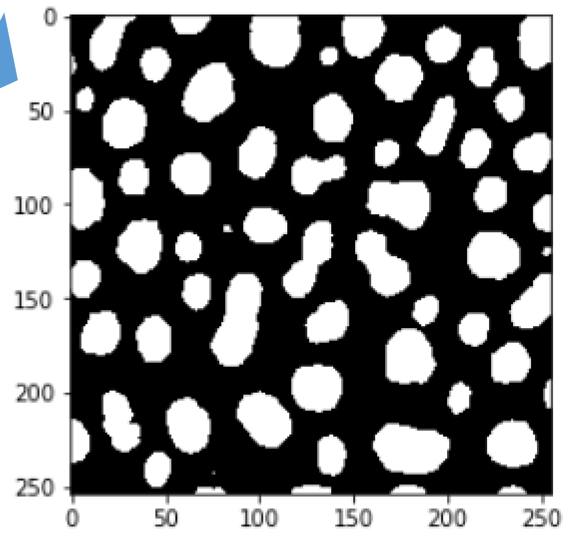


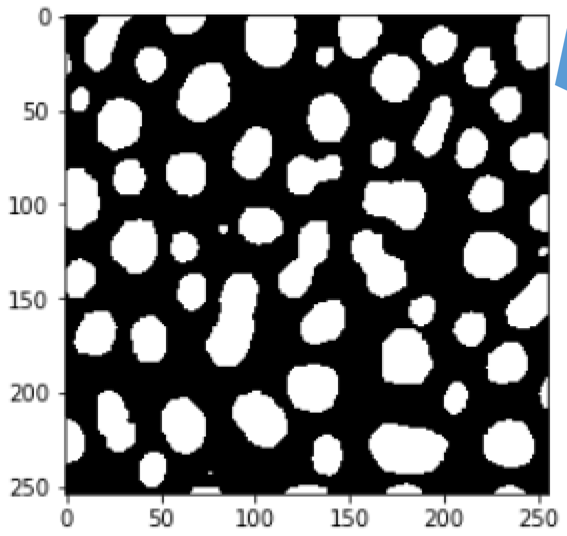
Visual comparison

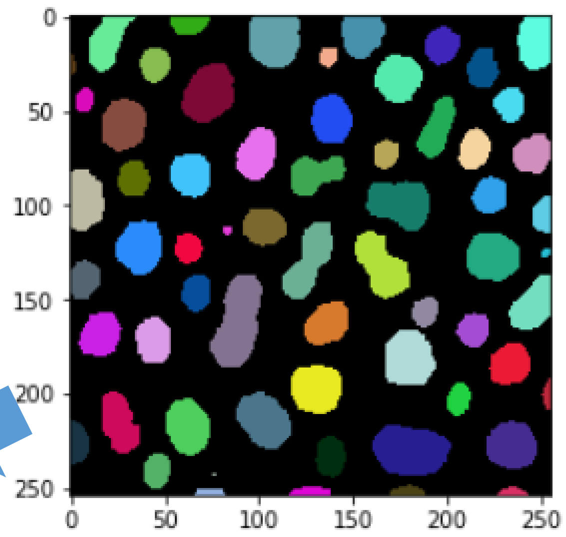
- The order in label images may be different. To compare them visually, we need to turn them into binary images first.




```
binary_blobs_imagej = imread(filenames[0]) > 0
binary_blobs_skimage = imread(filenames[1]) > 0
imshow(binary_blobs_imagej)
imshow(binary_blobs_skimage)
```

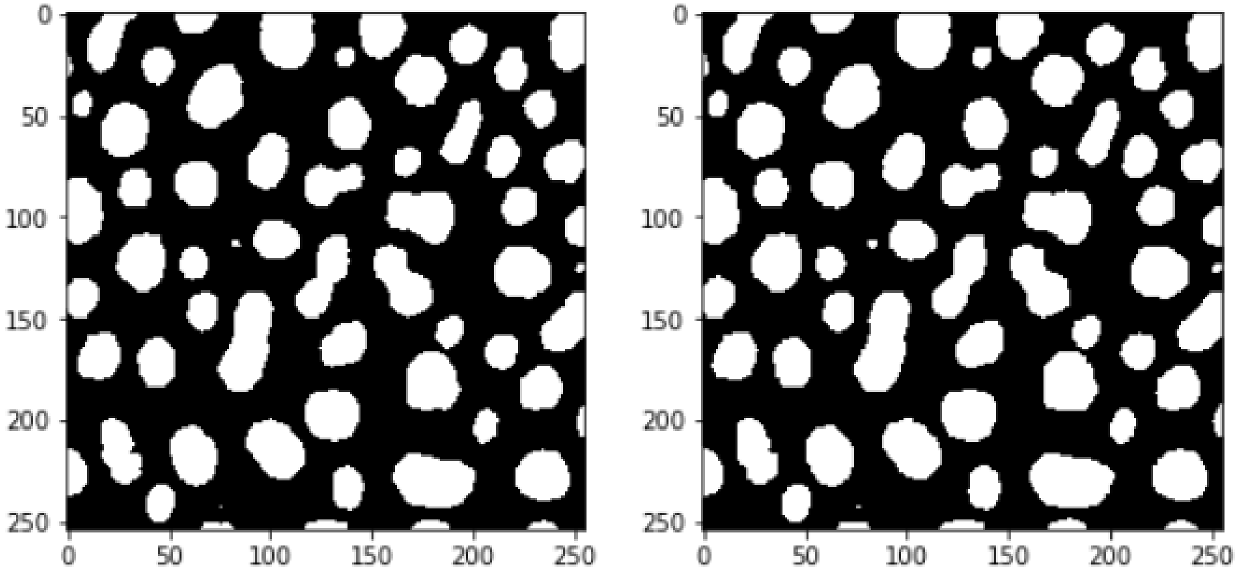






Quiz

- How many pixels n in these two images are different?



n=0

0<n<100

100<n<1000

n>1000

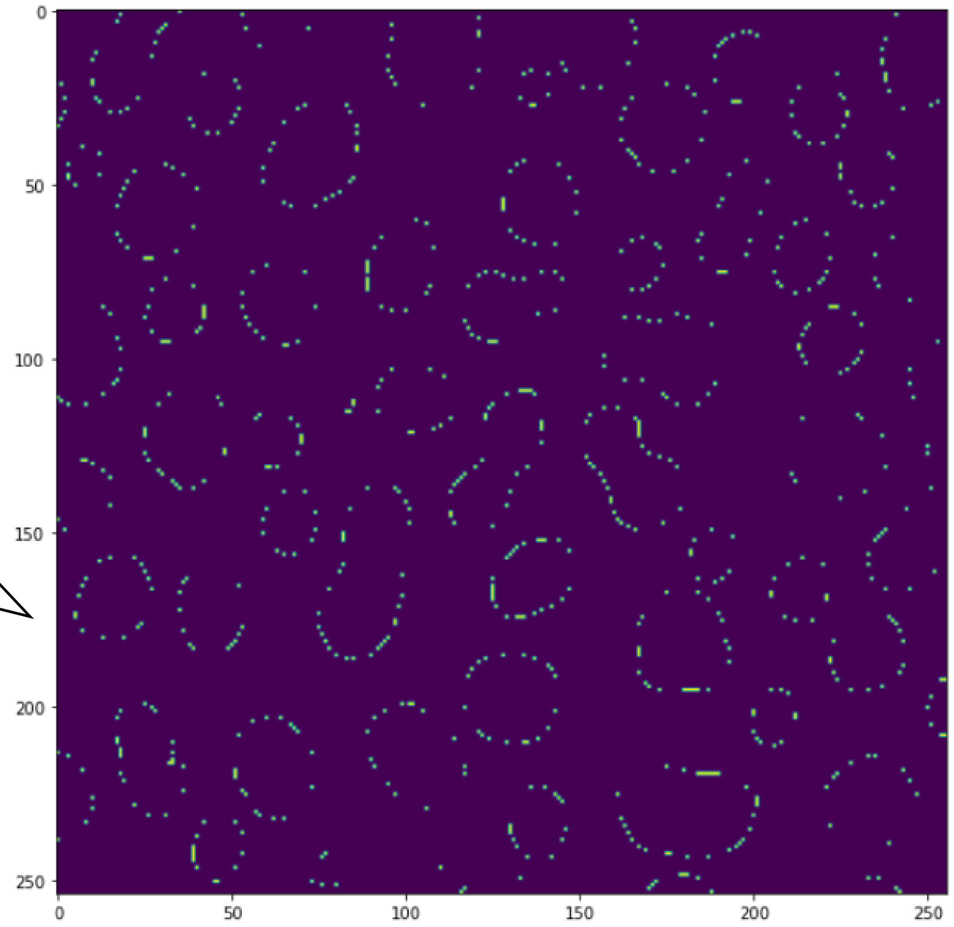
Visual comparison

- Binary image comparison: difference or XOR

```
difference = np.logical_xor(binary_blobs_imagej, binary_blobs_skimage)

fig, axs = plt.subplots(figsize=(10,10))
axs.imshow(difference)
```

Number of different pixels:

```
[5]: np.sum(difference)

[5]: 830
```



- Does not work well if labels are close-by

Quantitative comparison

- Voxel-wise Youden-Index

$$YI = p_{TP} + p_{TN} - 1$$

- Volume error

$$\Delta_V = V_A - V_B$$

$$\delta_V = \frac{\Delta_V}{V_B}$$

- Dice Index

$$DI(A,B) = \frac{2\,|A \cap B|}{|A| + |B|}$$

- Jaccard Index

$$JI(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{DI}{2 - DI}$$

- Contour distance

$$d_{e,min}(a,B) = \min(d_e(a,b)|b \in B)$$

$$\bar{d}_c(A,B) = \frac{\sum_{\forall a \in C(A)} d_{e,min}(a, C(B))}{|C(A)|}$$

$$\bar{d}_{bil,c}(A,B) = \frac{\bar{d}_c(A,B) + \bar{d}_c(B,A)}{2}$$

- Hausdorff distance

$$d_H(A,B) = \max(d_{e,min}(a,B)|a \in A)$$

$$d_{bil,H}(A,B) = \max(d_H(A,B), d_H(B,A))$$

- Simplified Hausdorff distance

$$d_H(A,B) = \max(d_{e,min}(a, C(B))|a \in C(A))$$

- Volume standard deviation

$$\delta_{\bar{V}} = 2\frac{|V_A - V_B|}{|V_A + V_B|}$$

- Classification error

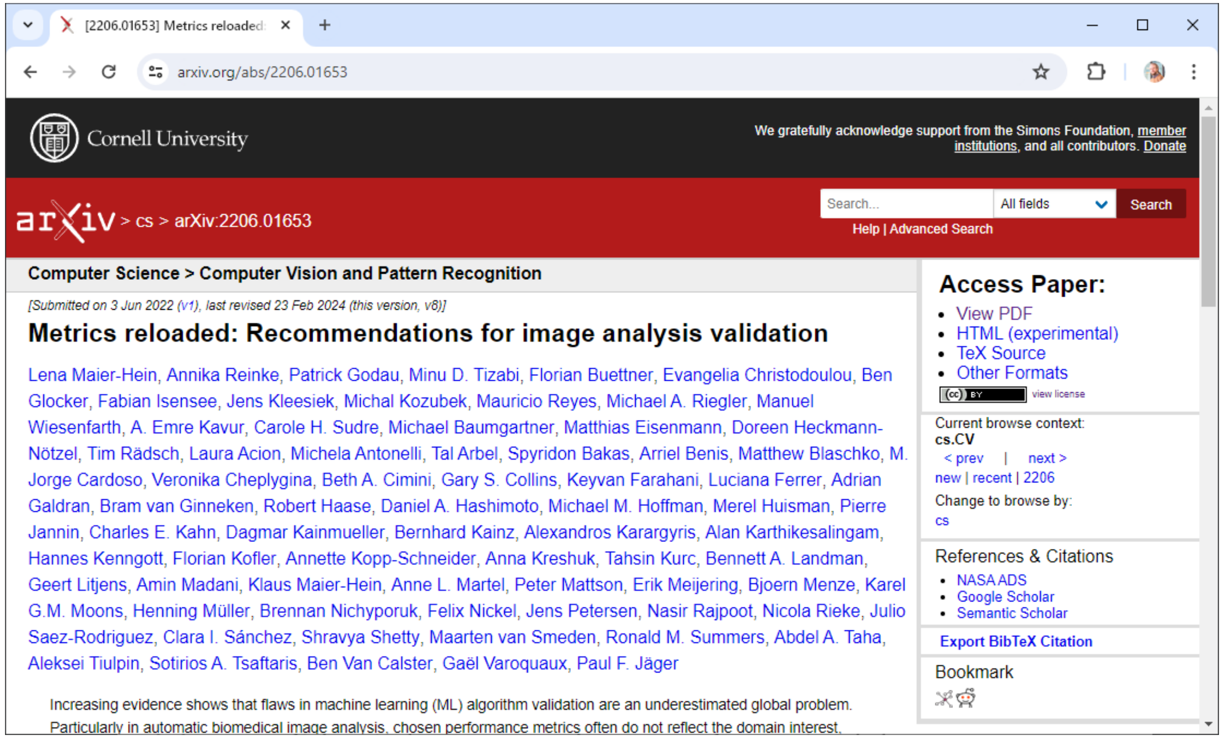
$$e_{Class} = \frac{H}{|TP| + |FN|}$$

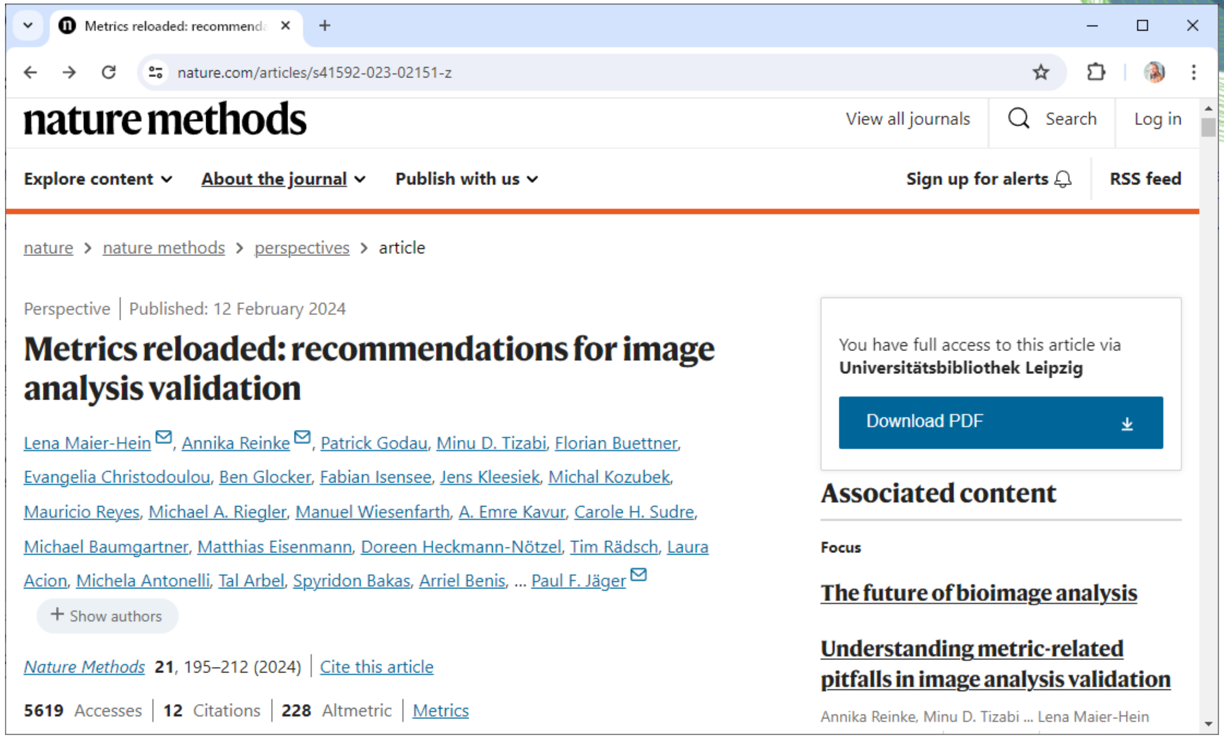
- Hamming distance

$$d_h = |A \cup B| - |A \cap B|$$
$$= |FP| + |FN|$$

Choosing the right metric is key

- Systematic overview by Maier-Hein, Reinke at al.

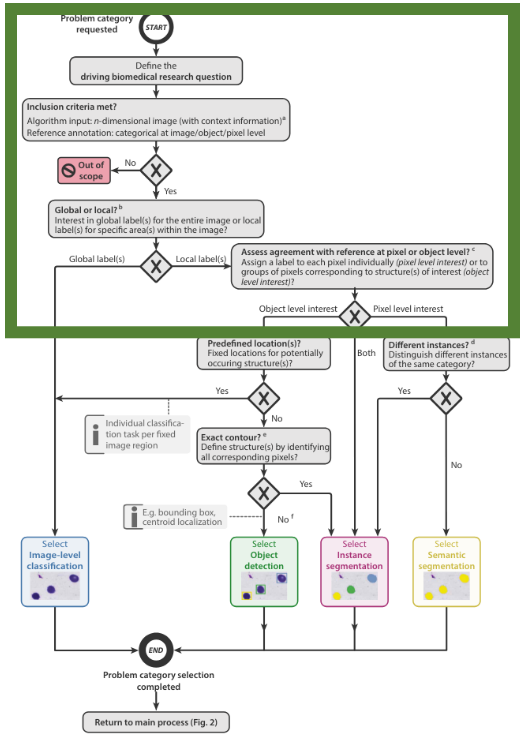


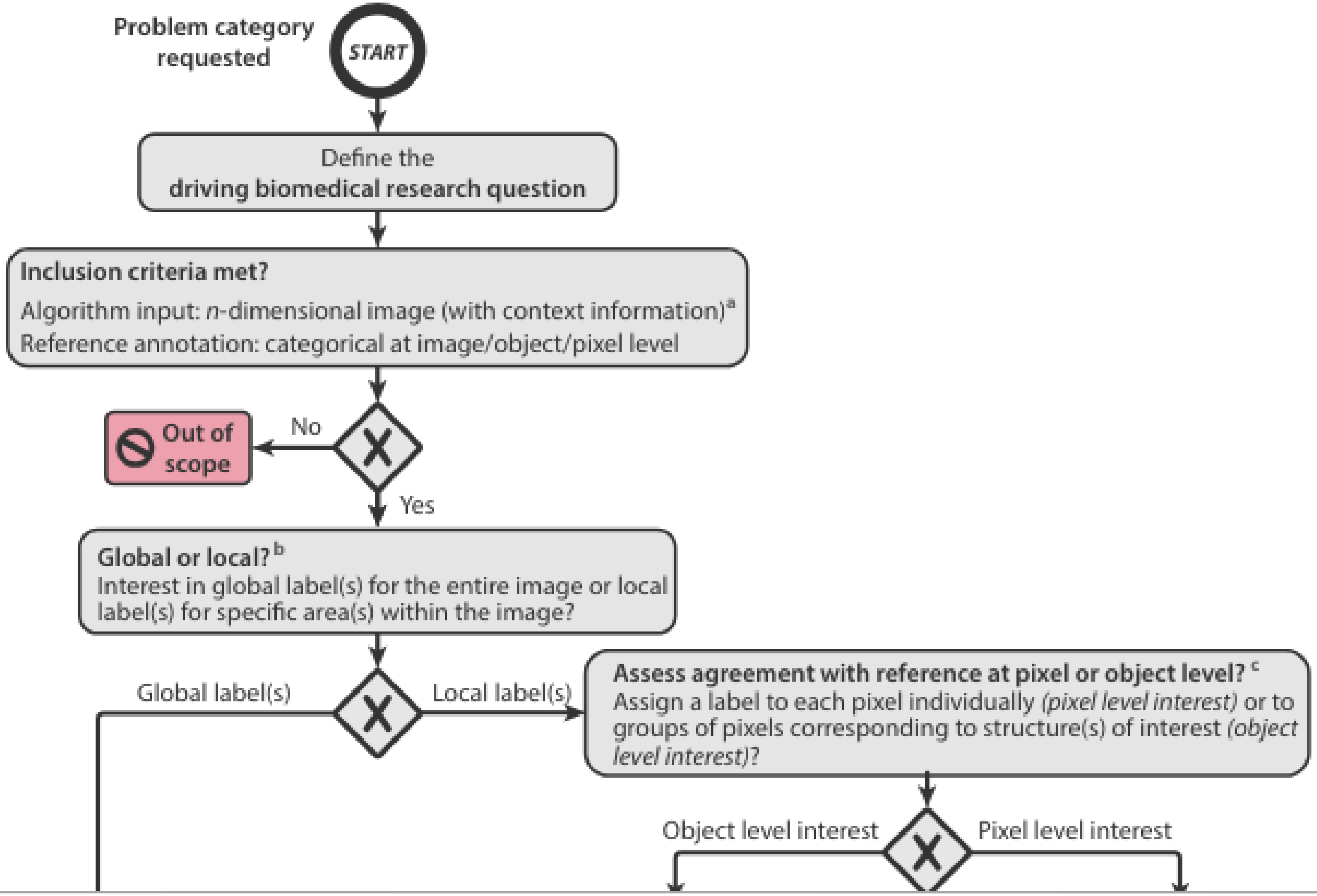


Maier-Hein, Reinke et al. Arxiv (2023)
 https://arxiv.org/pdf/2206.01653.pdf
https://www.nature.com/articles/s41592-023-02151-z

Choosing the right metric is key

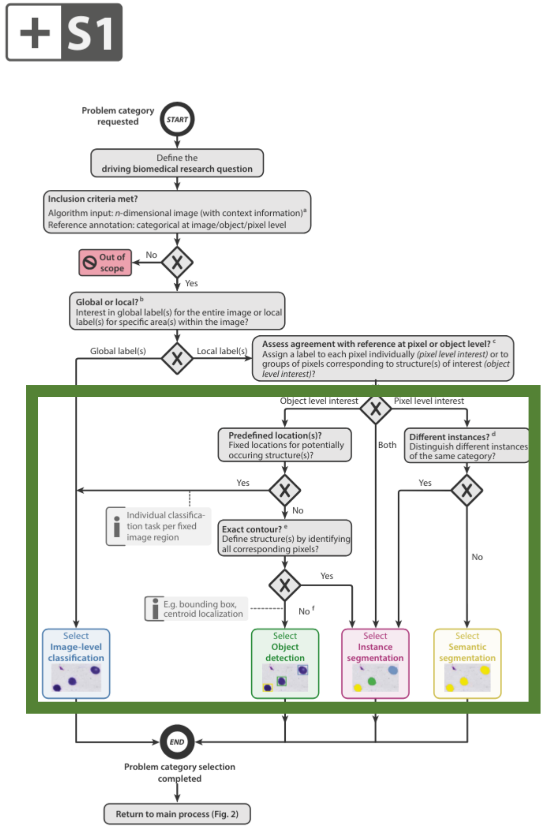
+ S1

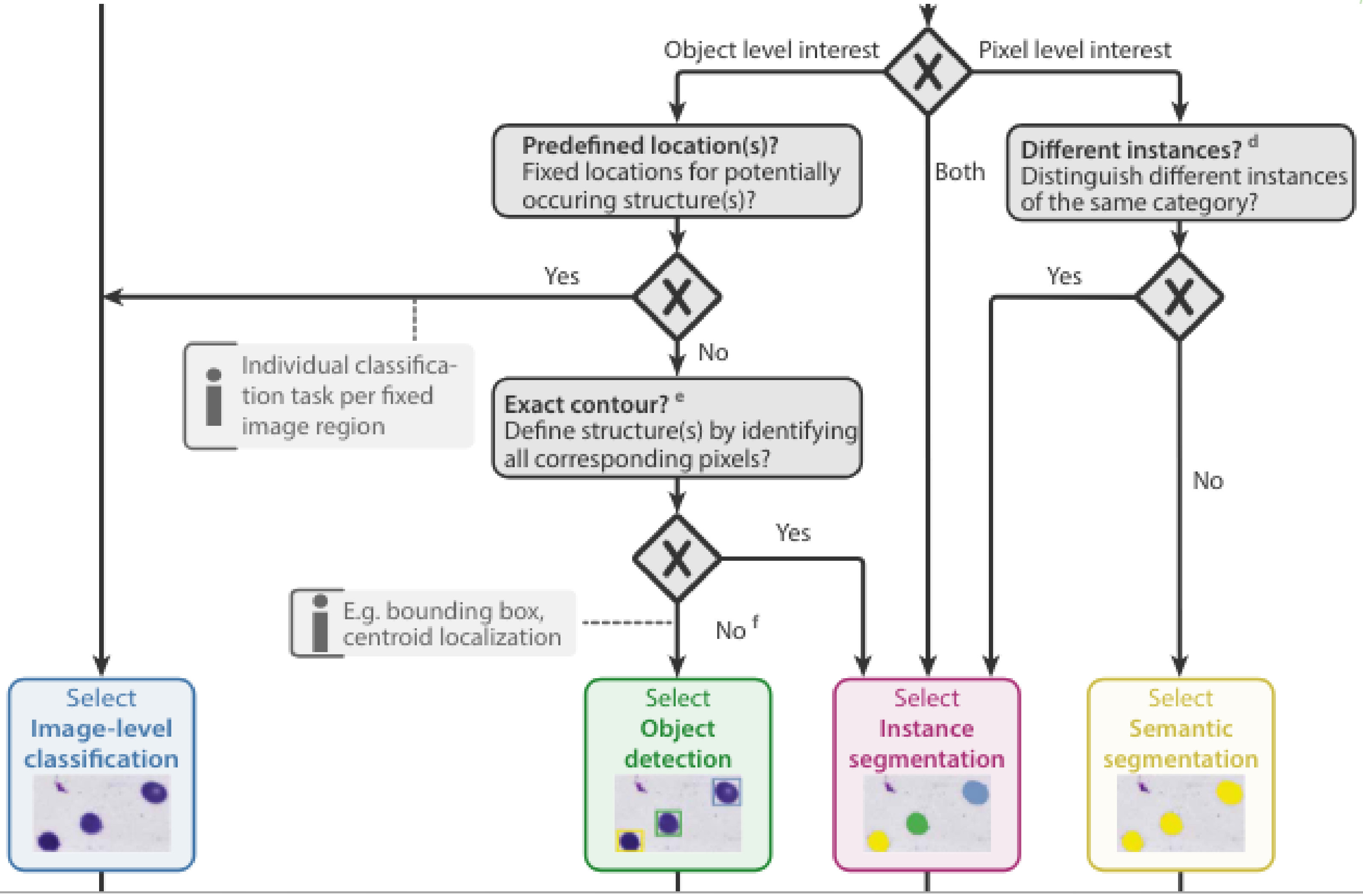




Source: Cropped from Ext. Data Fig 1 in Maier-Hein, Reinke, et al. licensed [CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/) https://arxiv.org/pdf/2206.01653.pdf

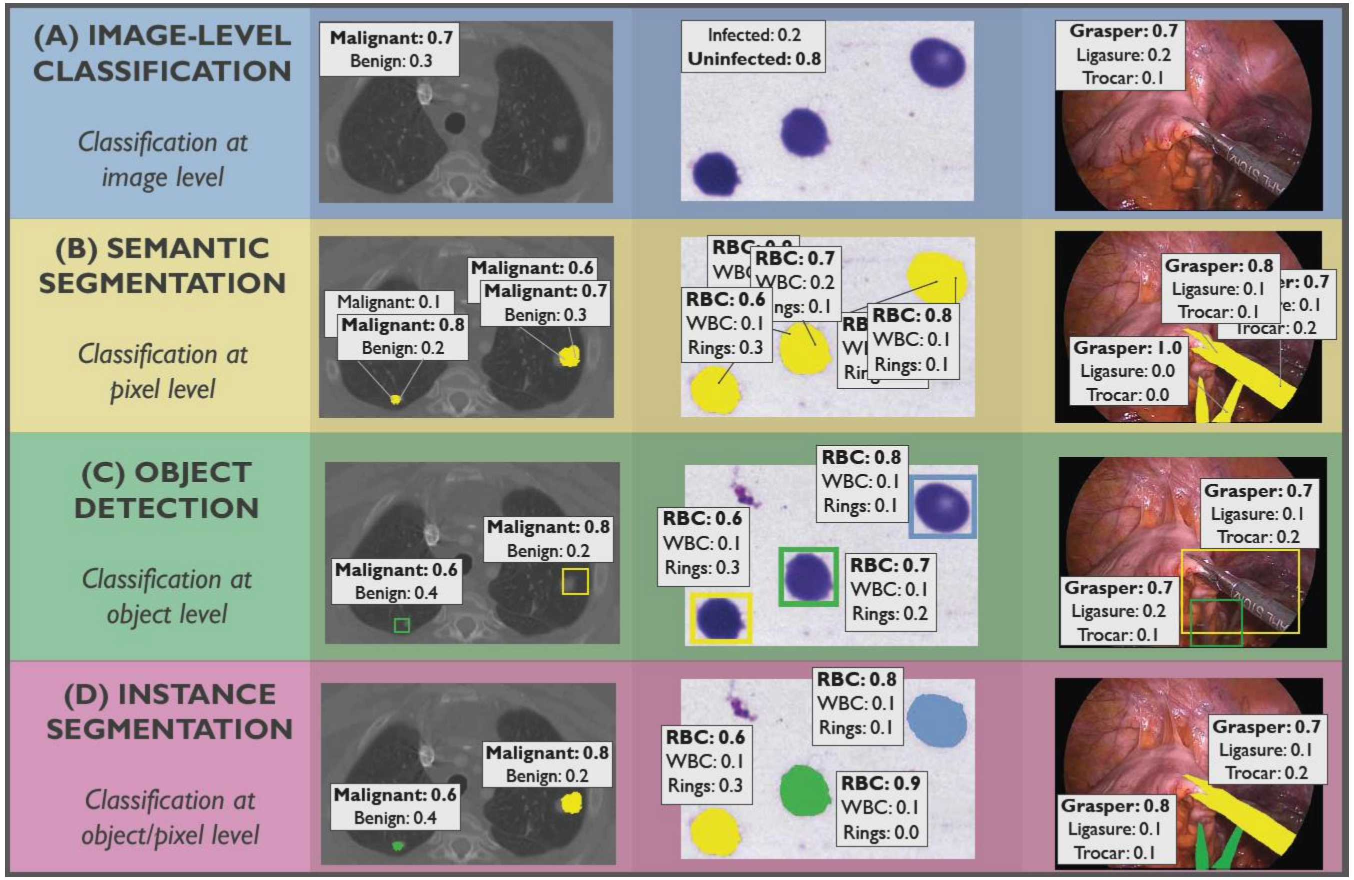
Choosing the right metric is key





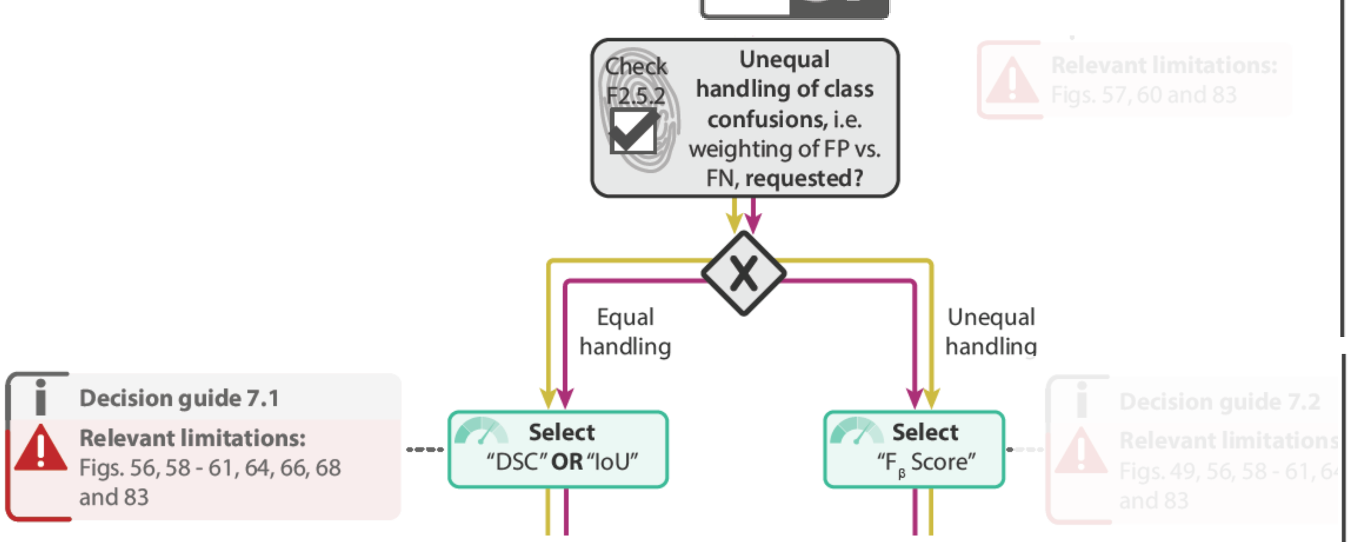
Choosing the right metric is key

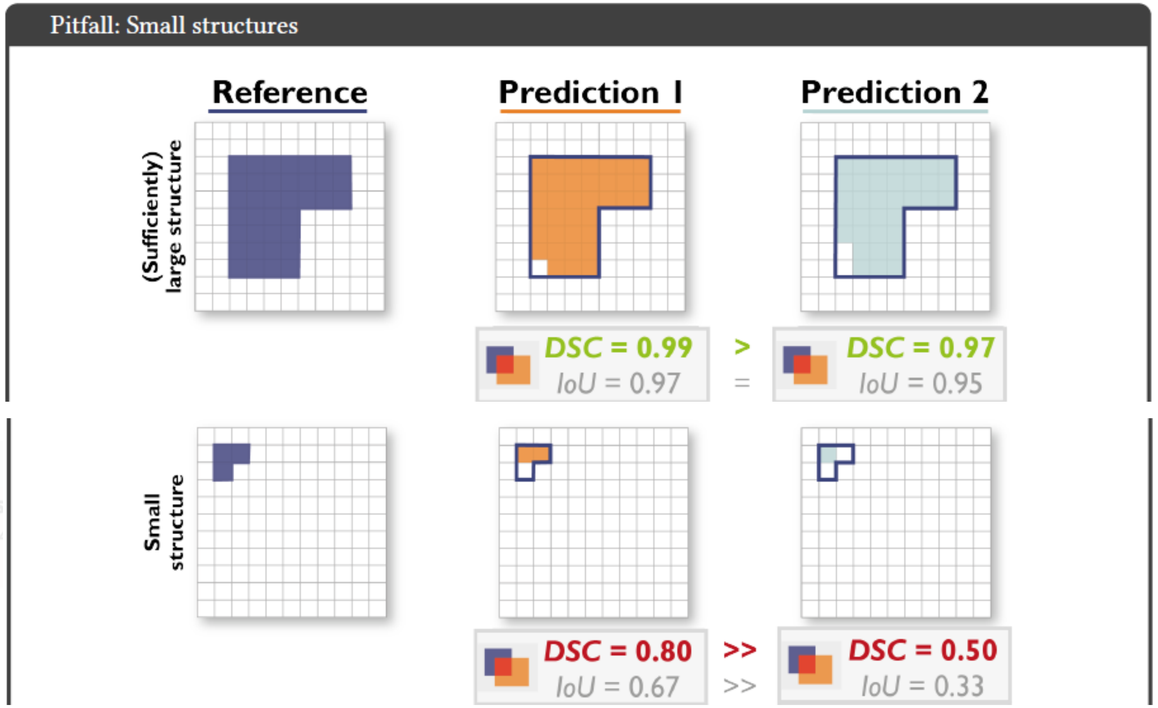
- Define your question



Overlap metrics

- Consider pitfalls and limitations





(From Figure 56)

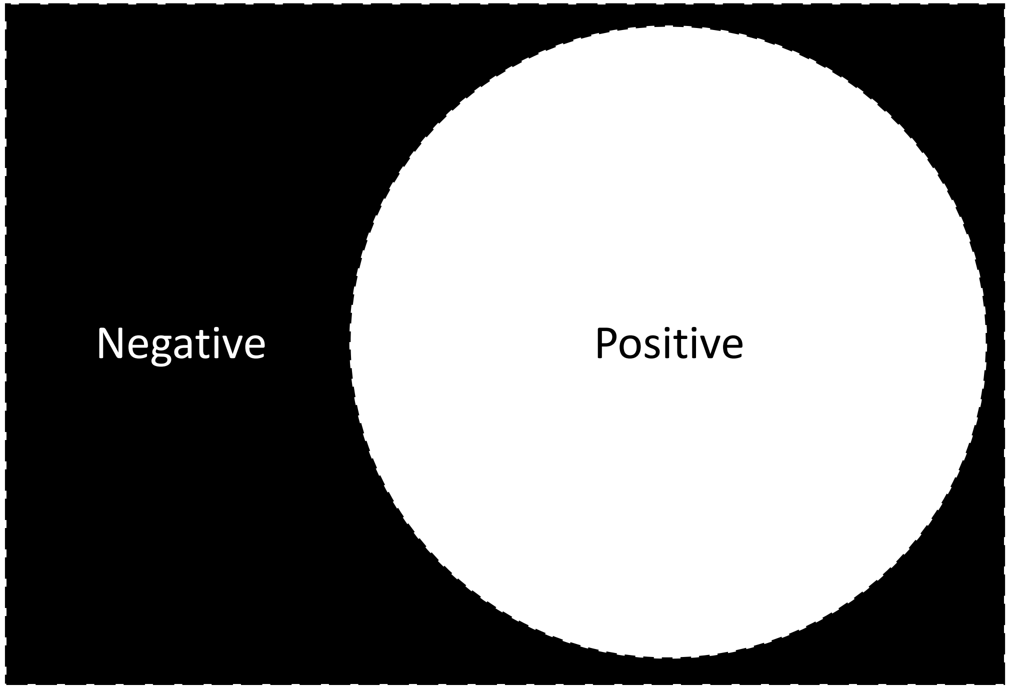
E.6 Decision guide S7.

D7.1: Dice vs IoU

The DSC is identical to the F_1 Score on pixel level and closely related to the IoU, which, in turn, is identical to the Jaccard Index (see equations 5 and 6). The two metrics will yield the same ranking (of aggregated metric values) in most applications (theoretically, deviations are possible), such that there is no value in combining them. Commonly, the computer vision community prefers the IoU, while the medical image community favors the DSC.

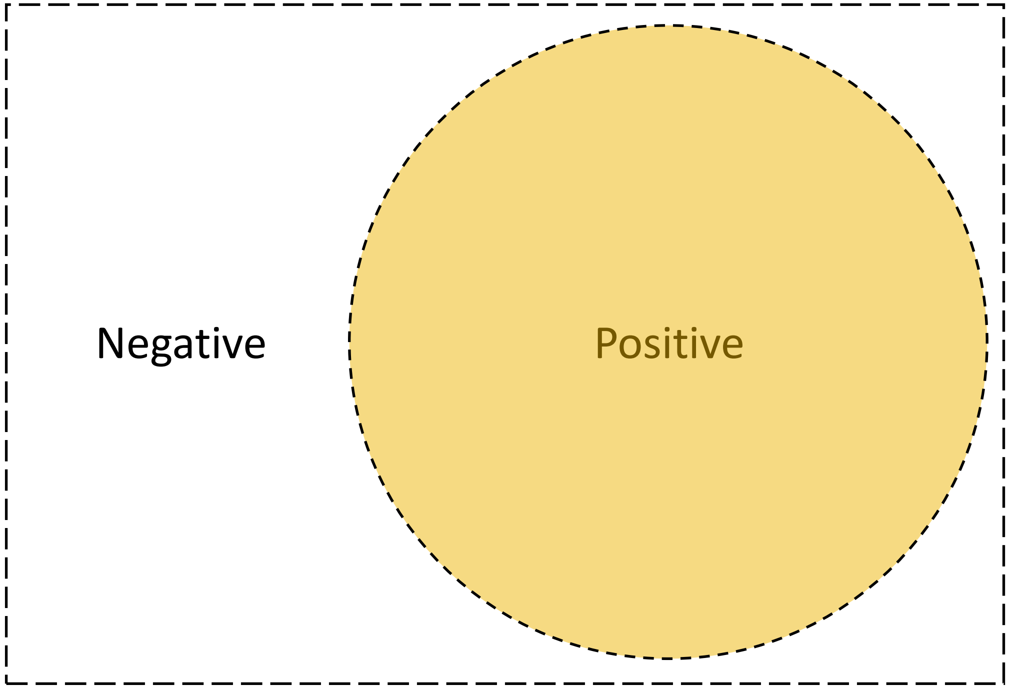
Segmentation quality estimation

- In general
  - Define what's positive and what's negative.



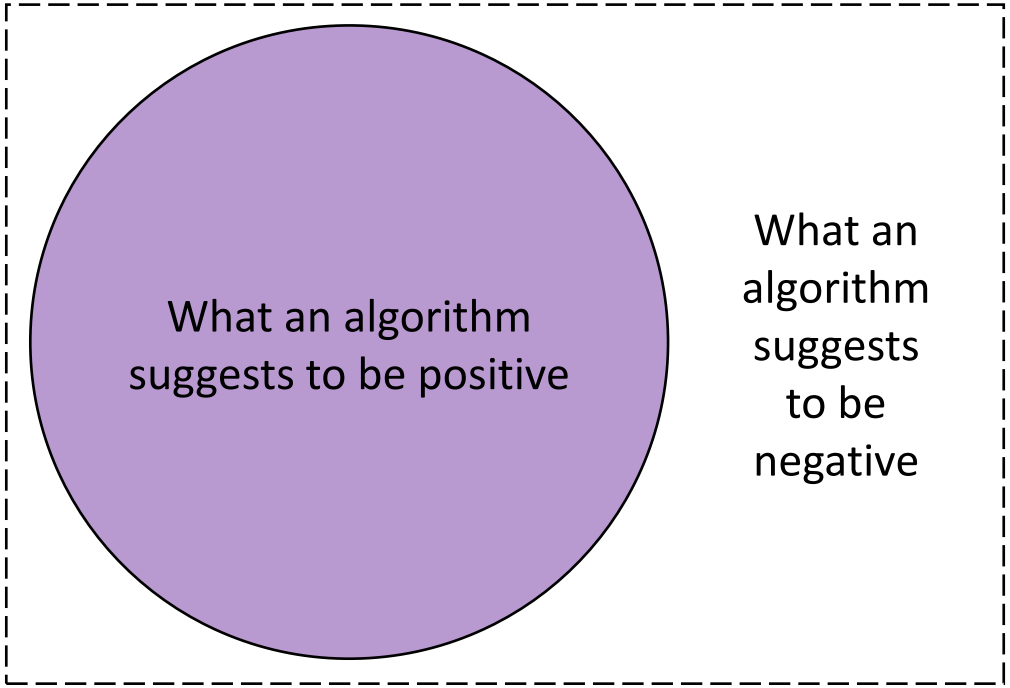
Segmentation quality estimation

- In general
  - Define what's positive and what's negative.



Segmentation quality estimation

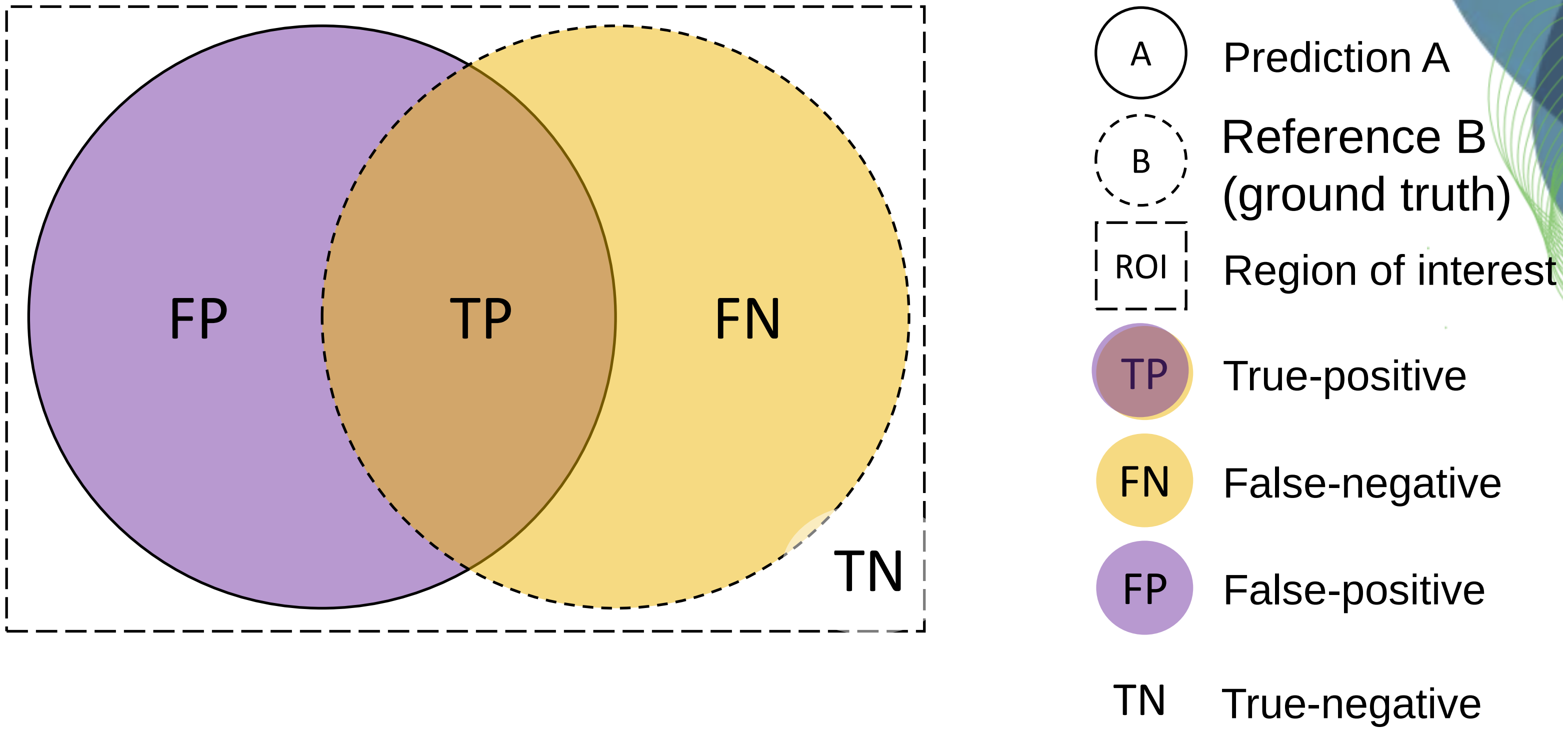
- In general
  - Define what's positive and what's negative.



Segmentation quality estimation

- In general
  - Define what's positive and what's negative.
  - Compare with a reference to figure out what was true and false

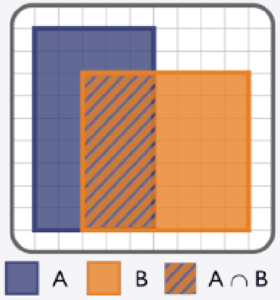
  - Welcome to the Theory of Sets



Dice versus Jaccard Index (IoU)

DICE SIMILARITY COEFFICIENT (DSC)

Synonyms: Dice, Dice Coefficient, Sørensen–Dice Coefficient, F_1 Score, Balanced F Score



$$DSC(A,B) = \frac{2\,|A \cap B|}{|A| + |B|} = \frac{2\,PPV \cdot Sensitivity}{PPV + Sensitivity}$$

VALUE RANGE: [0, 1] ↑

DESCRIPTION
DSC measures the overlap between two structures.

DEFINITION
[Dice, 1945]

RECOMMENDED FOR

ImLC	SemS	ObD	InS
●	●	●	●

IMPORTANT RELATIONS

DSC is closely related to the IoU = Jaccard index:

$$DSC = \frac{2\,IoU}{1 + IoU}$$

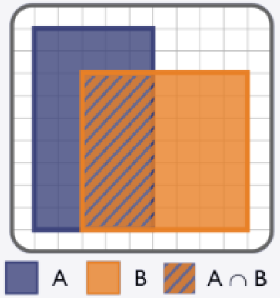
DSC is equal to the F_1 Score ($\beta = 1$ in F_β Score) at pixel level.

RECOMMENDATIONS

- An overlap-based metric (by default the DSC or IoU) should be used in most cases of segmentation assessment. An exception is the case of consistently tiny structures along with a noisy reference.
- DSC should generally be used in combination with a boundary-based metric if boundaries are of interest.
- DSC should generally not be considered if…
  - … there is a high variability of structure sizes within an image or across images.
  - … inter-rater variability is requested to be compensated.
  - … over- and undersegmentation should be treated similarly.
- DSC should be considered as a metric in the medical community rather than in the computer vision and biology communities (where the almost identical IoU is preferred).

INTERSECTION OVER UNION (IoU)

Synonyms: Jaccard Index, Tanimoto Coefficient



$$IoU(A,B) = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} = \frac{|A \cap B|}{|A \cup B|} = \frac{PPV \cdot Sensitivity}{PPV + Sensitivity - PPV \cdot Sensitivity}$$

VALUE RANGE: [0, 1] ↑

DESCRIPTION
IoU measures the overlap between two structures. It is often referred to as **Box IoU** when comparing bounding boxes, **Mask IoU** when comparing segmentation masks, or **Approx IoU** when comparing approximations of objects beyond bounding boxes.

DEFINITION
[Jaccard, 1912]

RECOMMENDED FOR

ImLC	SemS	ObD	InS
○	●	○	●

IMPORTANT RELATIONS

$$IoU = \frac{DSC}{2 - DSC} \qquad IoU = \frac{F_\beta}{2 - F_\beta} \text{ for } \beta = 1$$

RECOMMENDATIONS

- An overlap-based metric (by default DSC or IoU) should be used in most cases for segmentation assessment. An exception is the case of consistently tiny structures along with a noisy reference.
- IoU should generally be used in combination with a boundary-based metric if boundaries are of interest.
- IoU should generally not be considered if…
  - … there is a high variability of structure sizes within an image or across images.
  - … inter-rater variability is requested to be compensated.
  - … over- and undersegmentation should be treated similarly.
- IoU should be considered as a metric in the computer vision and biology communities rather than in the medical community (which prefers the almost identical DSC).

Source: Cropped from Ext. Data Fig SN 3.5 and 3.9 in Maier-Hein, Reinke, et al. licensed CC-BY 4.0 https://arxiv.org/pdf/2206.01653.pdf

Dice versus Jaccard Index (IoU)

2.7.5 Decision guide S6.

DG6.1: Dice Similarity Coefficient (DSC) versus Intersection over Union (IoU)

Summary of DG6.1: DSC versus IoU

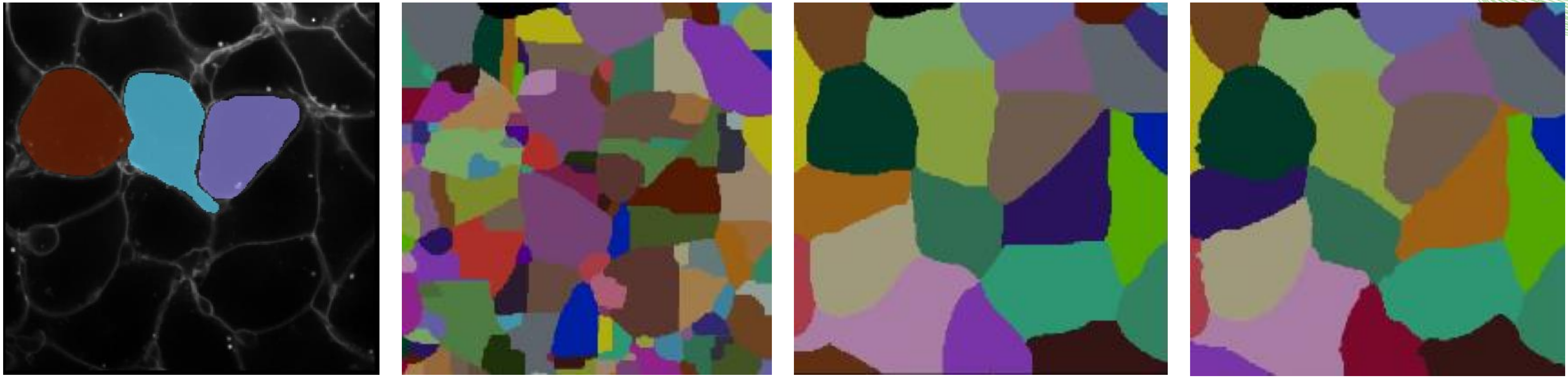
DSC
- Identical to F_1 Score
- Close relation to IoU (see Eq. 5)
- Preference in medical community

IoU
- Identical to Jaccard Index
- Close relation to DSC (see Eq. 4)
- Preference in computer vision community

Extended Data Tab. SN 2.18. Comparison of Dice Similarity Coefficient (DSC) and Intersection over Union (IoU) in the context of the decision guide DG6.1 for Subprocess S6. Context: no exclusive interest in the center line of structures (FP2.3 = FALSE, FP3.3 = FALSE) and equal severity of class confusions (FP2.5.2 = FALSE).

Sparse Jaccard Index

- For every annotated object, we compute the maximum IoU with any segmented object.
- We average this value over all annotated objects



Sparse instance annotation

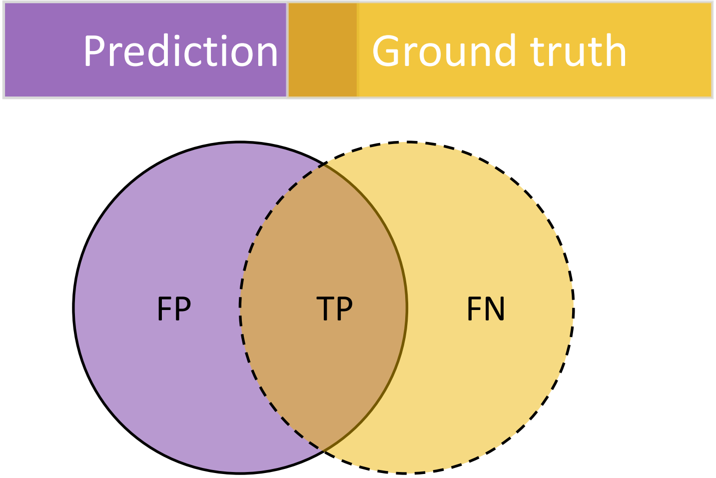
IoU = 0.35 IoU = 0.66 IoU = 0.69

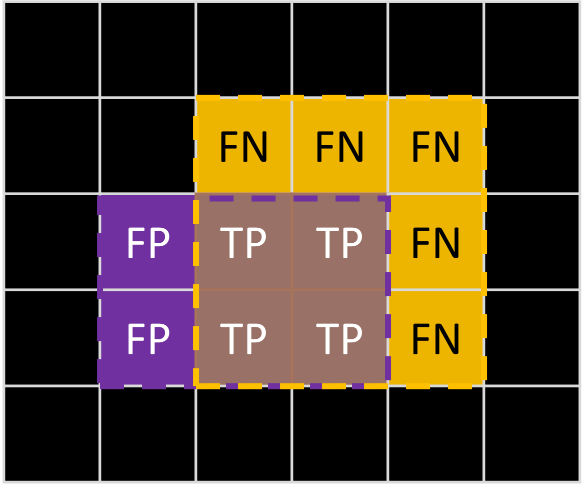
Instance segmentation candidates

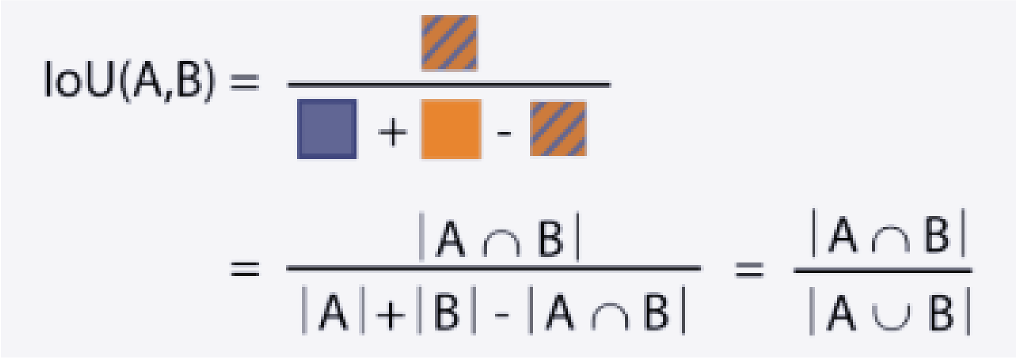
Source: https://github.com/haesleinhuepf/napari-workflow-optimizer

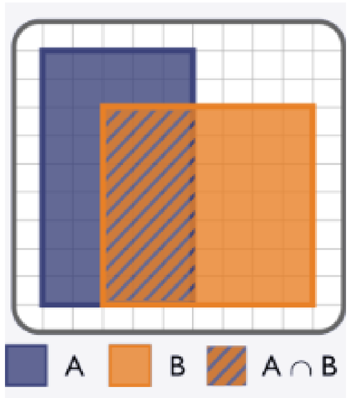
Pixel-wise versus Object-wise evaluation

- Pixel wise: Segmentation quality









True-positive: 4

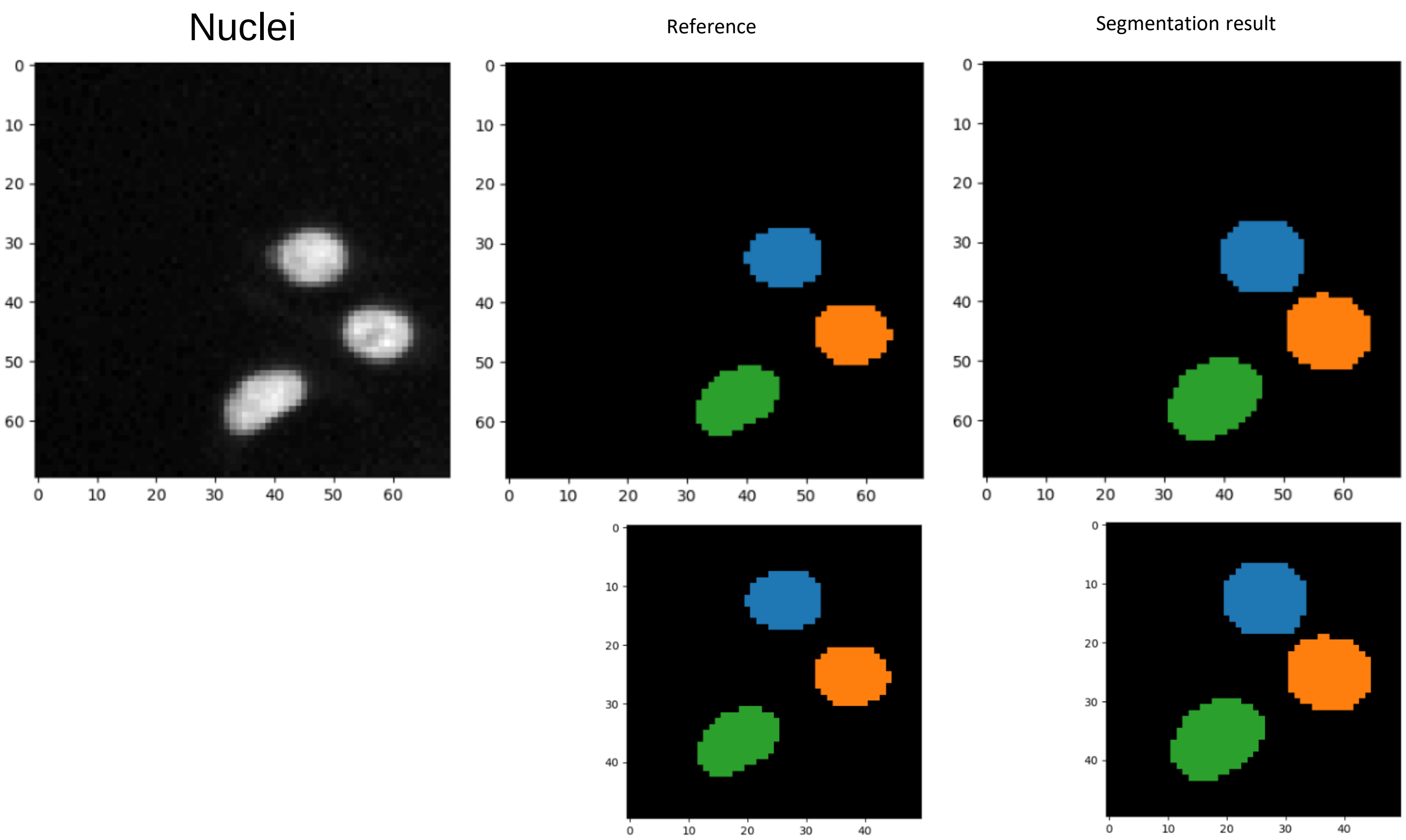
False-negative: 5

False-positive: 2

IoU = 4 / 11

Accuracy versus Jaccard Index (IoU)

- Side-effects of image size and number of nuclei



$$A = \frac{TP + TN}{FN + FP + TP + TN}$$

$$J = \frac{TP}{FN + FP + TP}$$

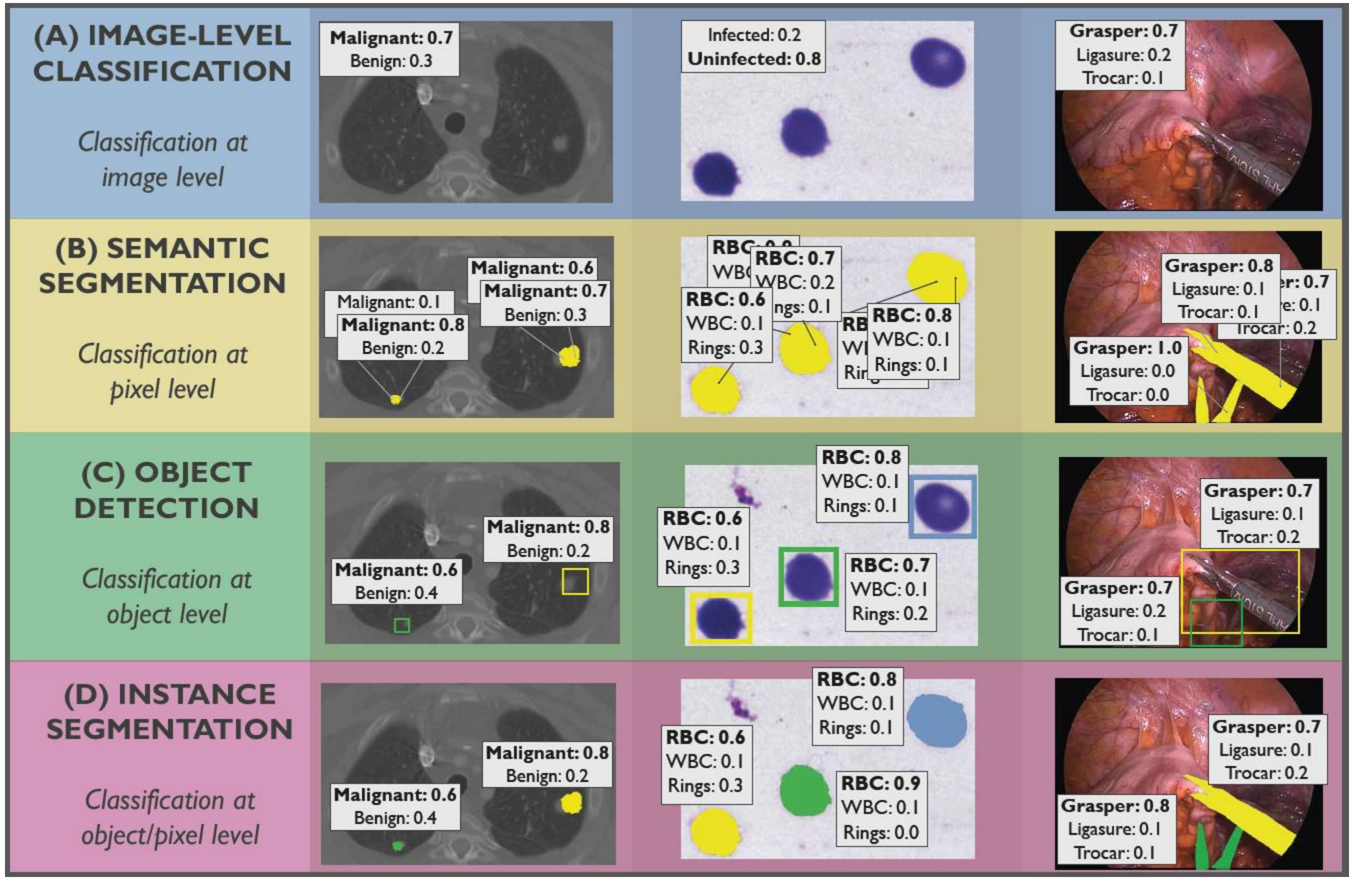
Accuracy: 0.97
Jaccard Index: 0.73

Accuracy decreases because there are less correct black pixels (TN)

Accuracy: 0.95
Jaccard Index: 0.73

Choosing the right metric is key

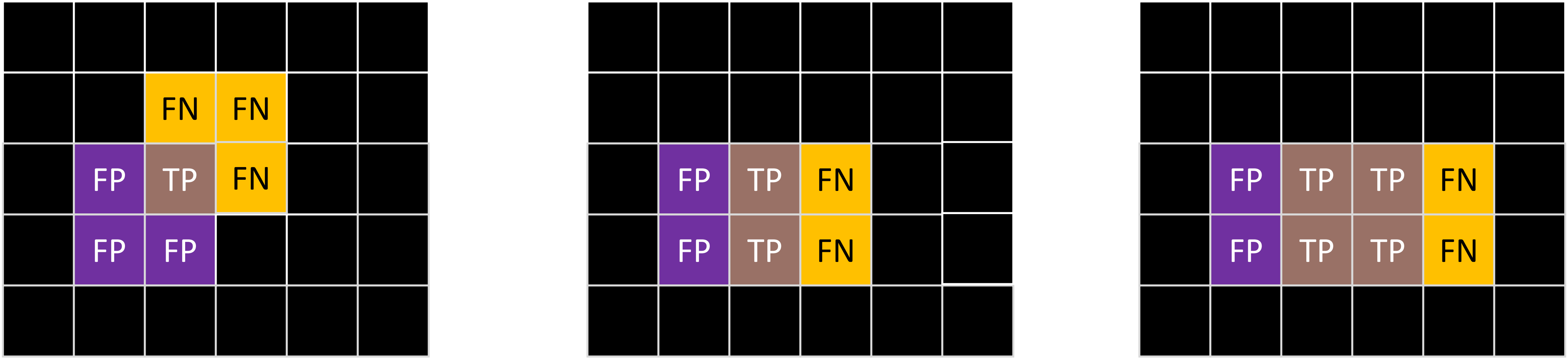
- Define your question



Source: Cropped from Fig 4 in Maier-Hein, Reinke, et al. licensed [CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/) https://arxiv.org/pdf/2206.01653.pdf

Pixel-wise versus Object-wise evaluation

- Are these objects overlapping?



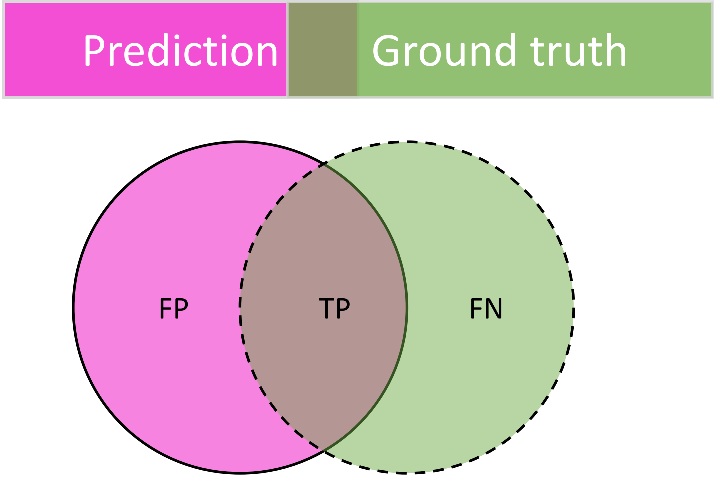
IoU = 1 / 7

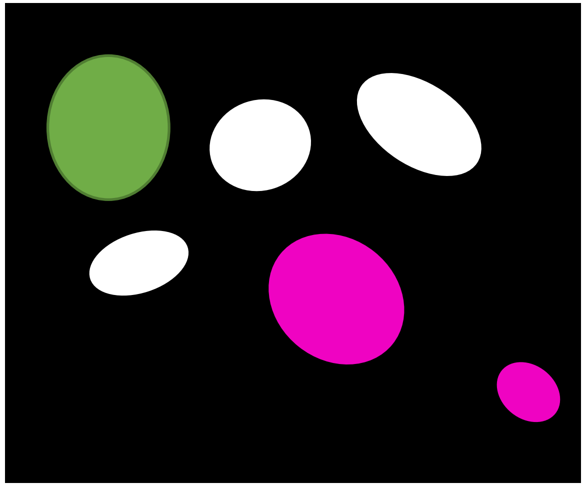
IoU = 1 / 3

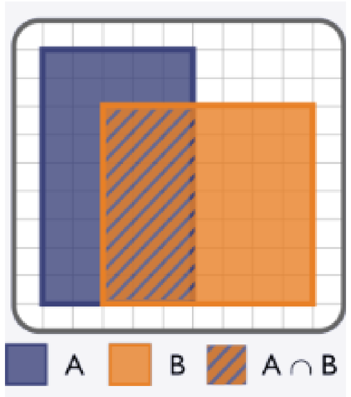
IoU = 1 / 2

Pixel-wise versus Object-wise evaluation

- Object wise: Detection quality



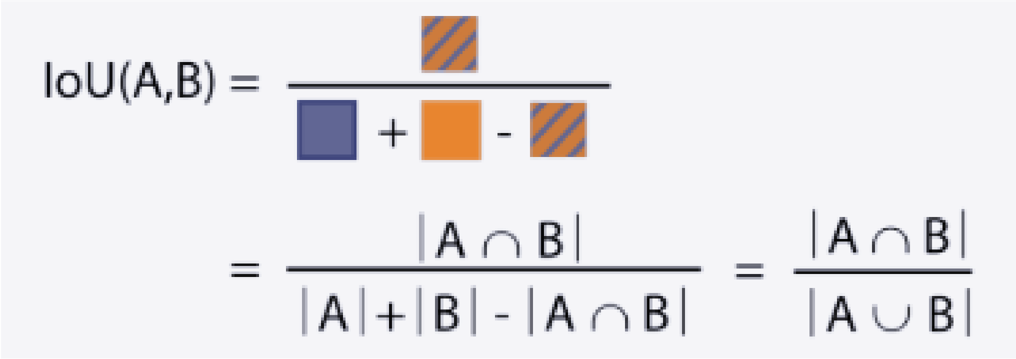




True-positive: 3

False-negative: 1

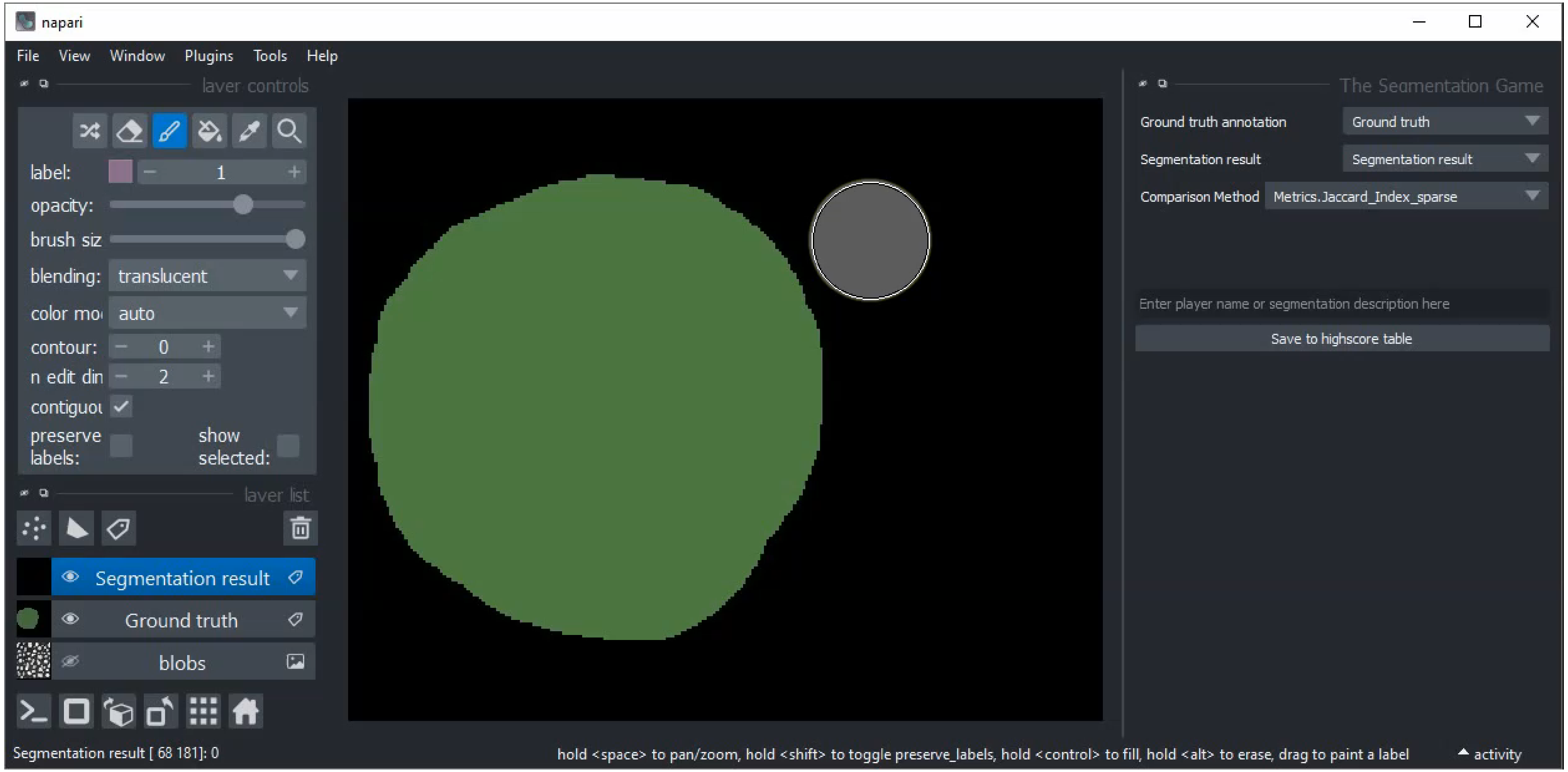
False-positive: 2



IoU = 1 / 2

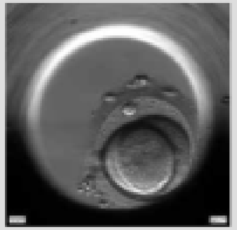
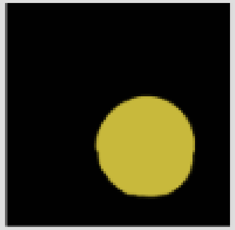
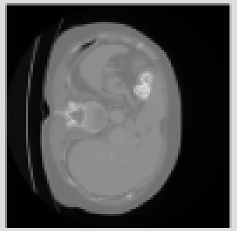
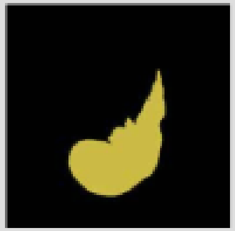
Pixel-wise versus Object-wise evaluation

- Play with metrics to gain understanding



With Martin Schätz (Charles Uni, Prague)
@schatzcz

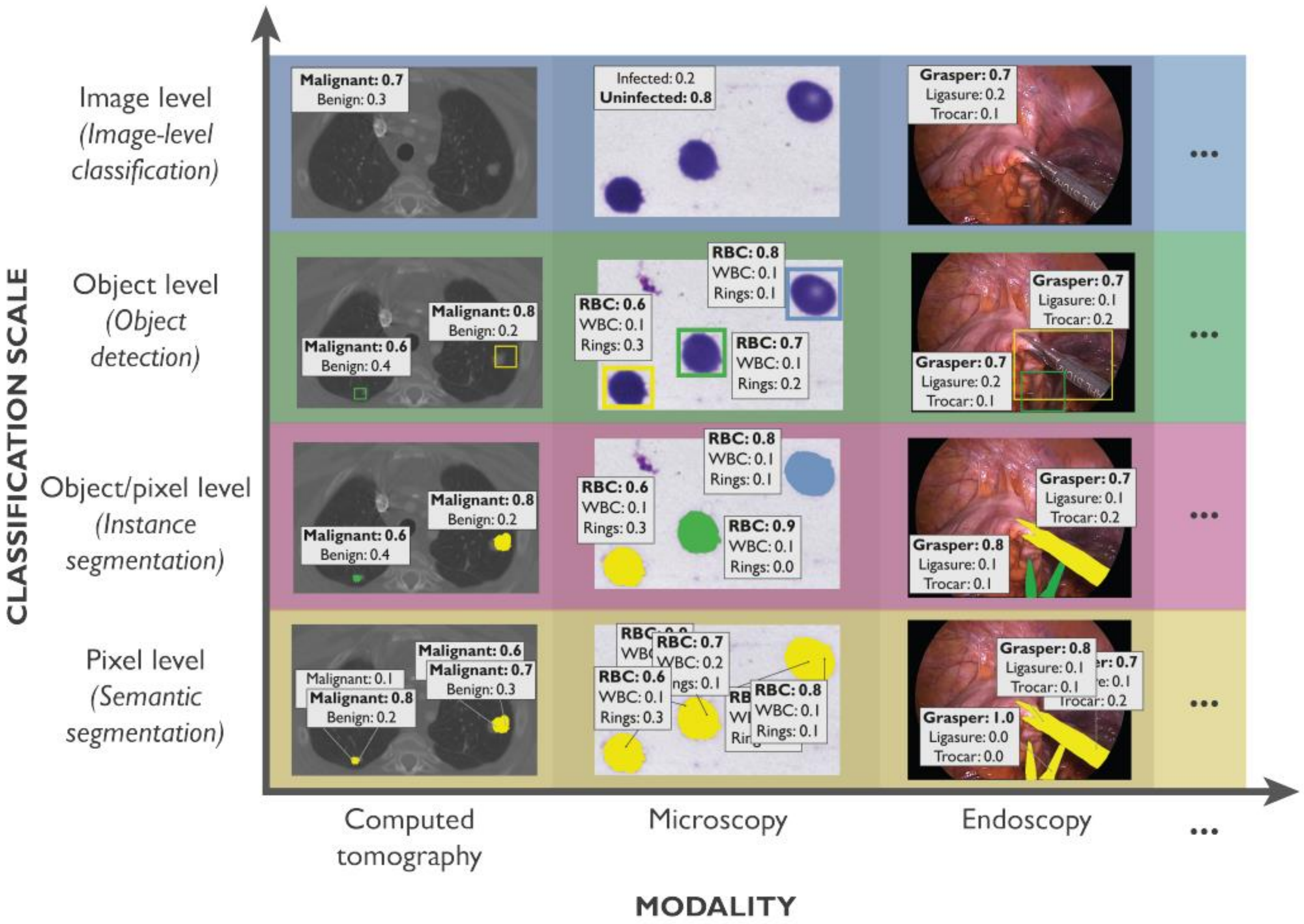
- Special case: We elaborate segmentation quality of one / large object:

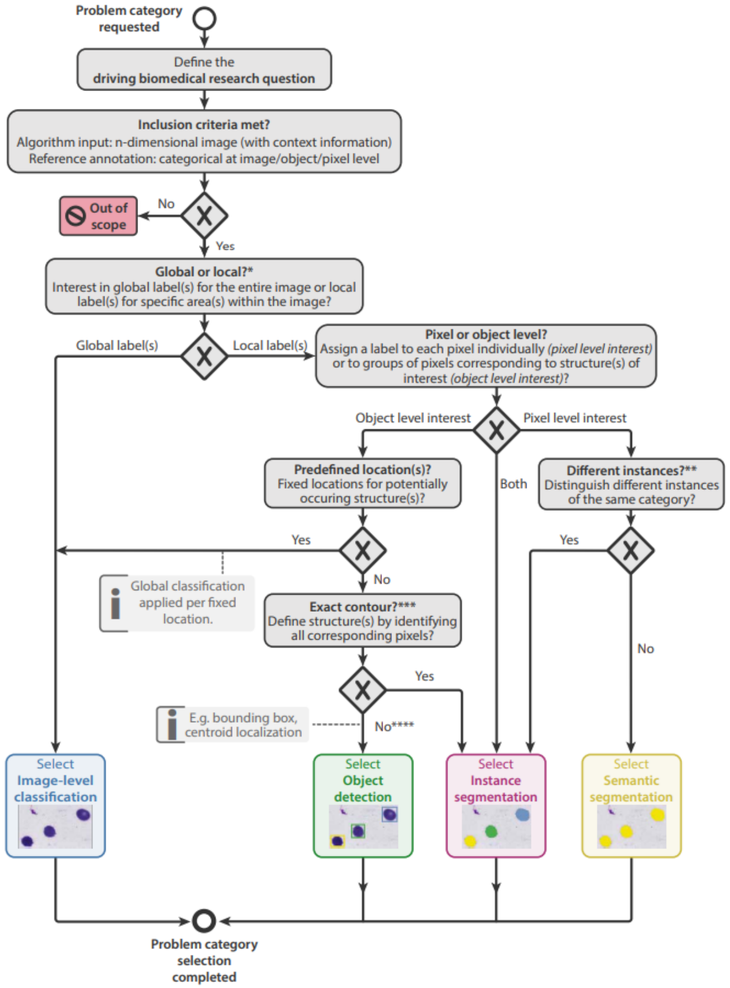
PROBLEM DESCRIPTION	ID	SCENARIO	SAMPLE INPUT IMAGE	RECOMMENDED OUPUT	RECOMMENDATION
Segmentation of large objects	**SemS-1**	Embryo segmentation from microscopy images			**Problem category:** Semantic segmentation **Overlap-based metric (S6):** Dice Similarity Coefficient (DSC) **Boundary-based metric (S7):** Normalized Surface Distance (NSD) **Specific property-related metric:** Liver segmentation: Absolute Volume Difference
	SemS-2	Liver segmentation in computed tomography (CT) images			

Source: Cropped from Fig 5 in Maier-Hein, Reinke, et al. licensed CC-BY 4.0 https://arxiv.org/pdf/2206.01653.pdf

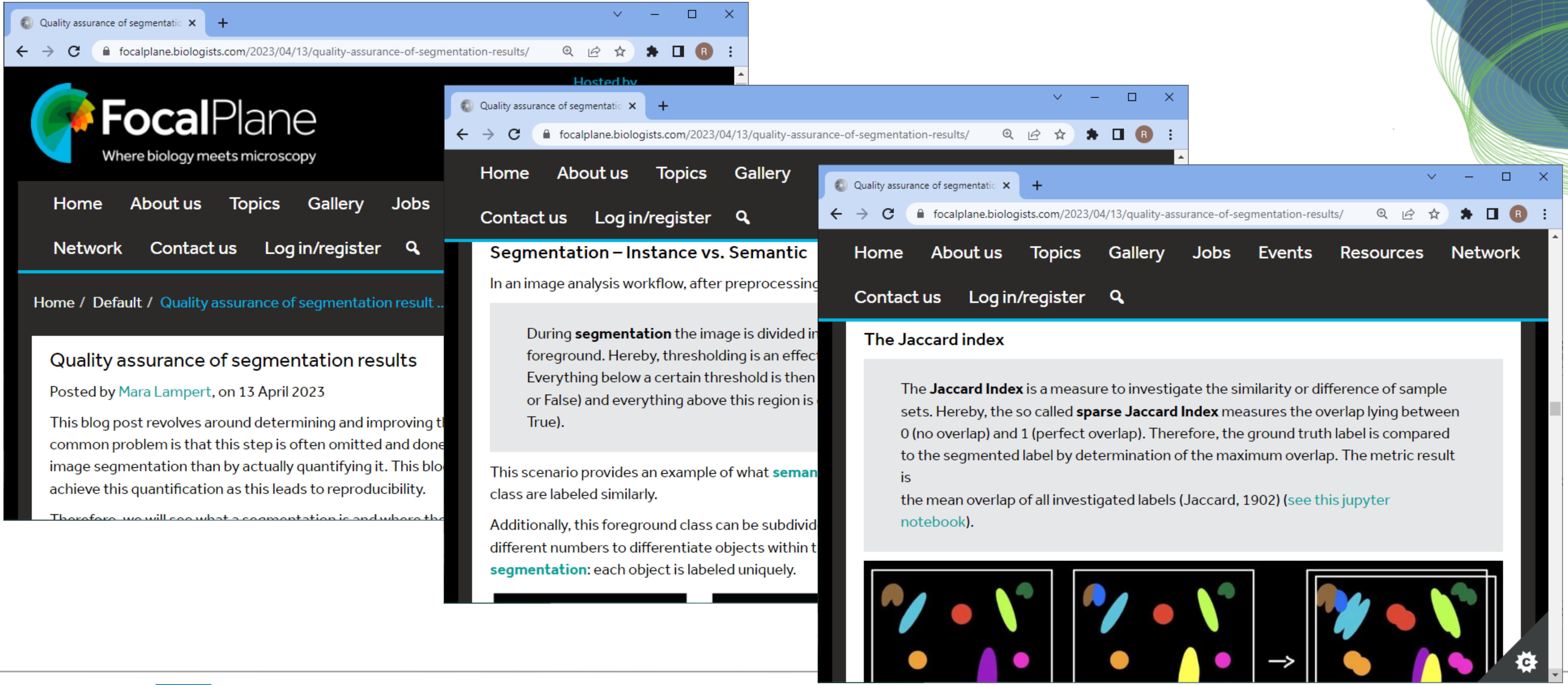
What metric to use when?

- "Metrics reloaded: Pitfalls and recommendations for image analysis validation"
  Maier-Hein, Reinke et al. https://arxiv.org/abs/2206.01653





Further reading



https://focalplane.biologists.com/2023/04/13/quality-assurance-of-segmentation-results/

ScaDS.AI
DRESDEN LEIPZIG
CENTER FOR SCALABLE DATA ANALYTICS
AND ARTIFICIAL INTELLIGENCE

Feature extraction

Robert Haase

Using materials from Johannes Soltwedel, PoL, TU Dresden

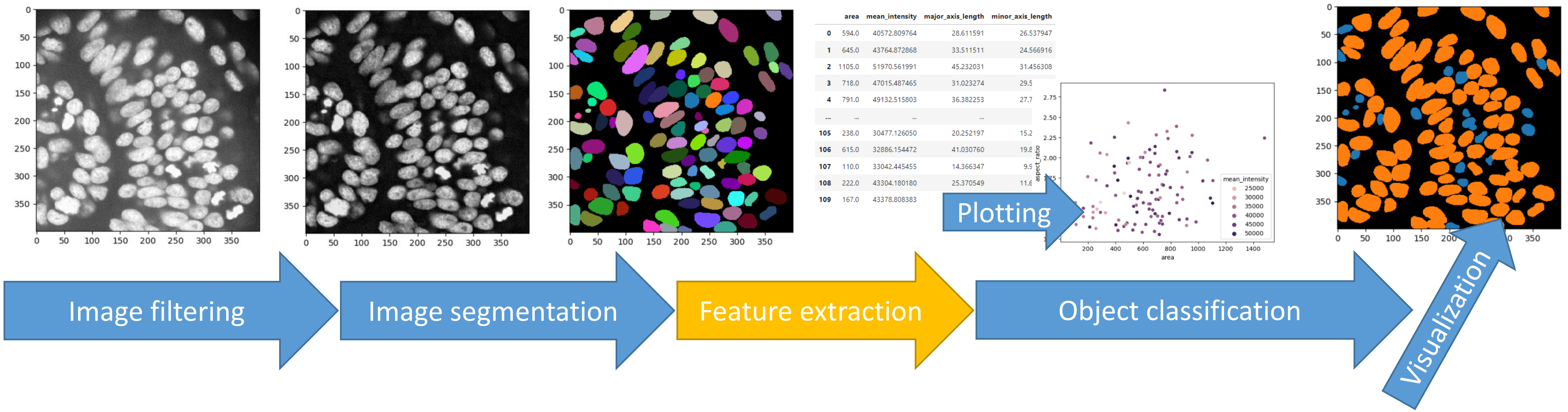
GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung

Diese Maßnahme wird gefördert durch die Bundesregierung aufgrund eines Beschlusses des Deutschen Bundestages. Diese Maßnahme wird mitfinanziert durch Steuermittel auf der Grundlage des von den Abgeordneten des Sächsischen Landtags beschlossenen Haushaltes.

Lecture overview: Bio-image Analysis

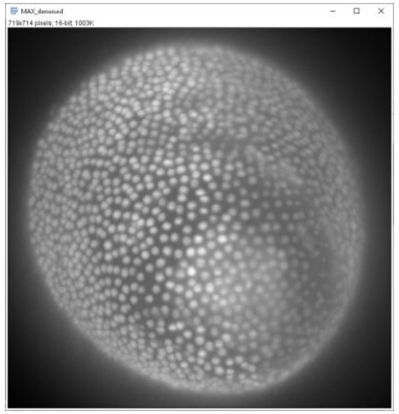
- Image Data Analysis workflows
- Goal: **Quantify observations, substantiate conclusions with numbers**



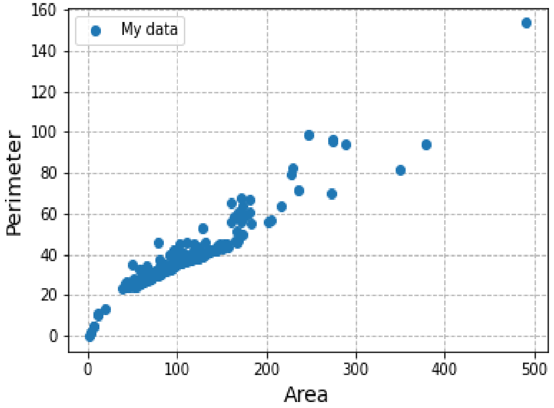
	area	mean_intensity	major_axis_length	minor_axis_length
0	594.0	40572.809764	28.611591	26.537947
1	645.0	43764.872868	33.511511	24.566916
2	1105.0	51970.561991	45.232031	31.456308
3	718.0	47015.487465	31.023274	29.5
4	791.0	49132.515803	36.382253	27.7
...	...	...	...	
105	238.0	30477.126050	20.252197	15.2
106	615.0	32886.154472	41.030760	19.8
107	110.0	33042.445455	14.366347	9.9
108	222.0	43304.180180	25.370549	11.6
109	167.0	43378.808383		

Feature extraction

- Feature extraction is a *late* processing step in image analysis.
- It can be used for images or

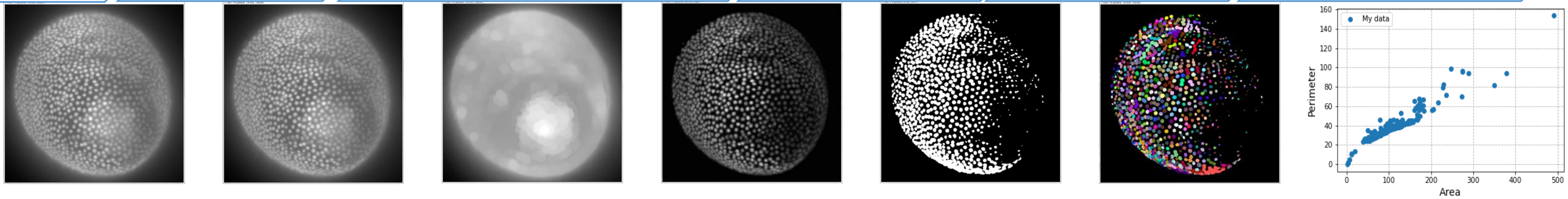


Feature Extraction



- or segmented/labelled images

Acquisition → Denoising → Background subtraction → Segmentation → Labeling → Feature Extraction



Feature extraction

- A *feature* is a countable or measurable property of an image or object.
- Goal of feature extraction is finding a minimal set of features to describe an object well enough to differentiate it from other objects.

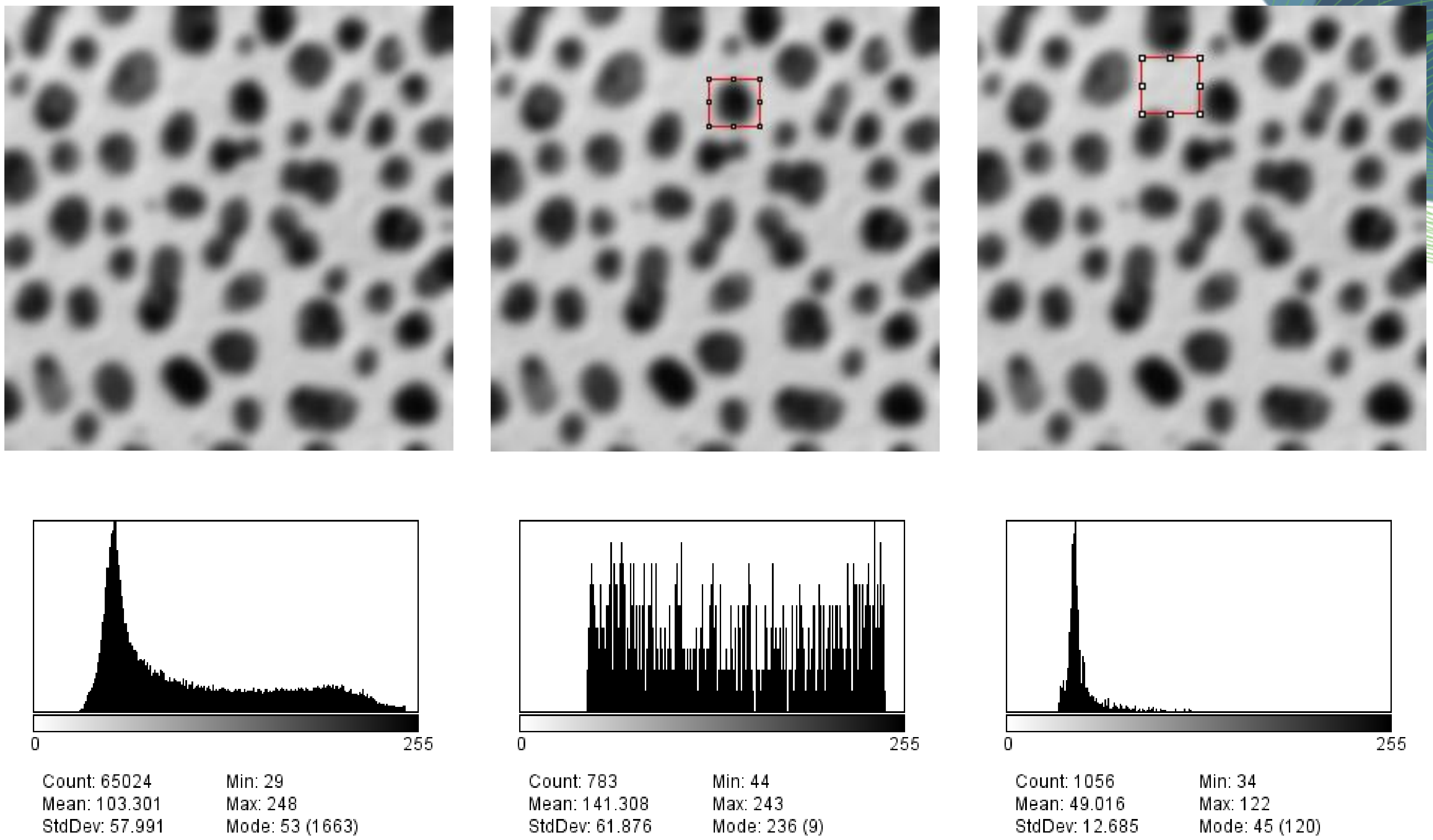
- **Intensity based**
  - Mean intensity
  - Standard deviation
  - Total intensity
  - Textures
- **Mixed features**
  - Center of mass
  - Local minima / maxima
  - Distance to neighbors
  - Average intensity in neighborhood
- **Shape based /spatial**
  - Area / Volume
  - Roundness
  - Solidity
  - Circularity / Sphericity
  - Elongation
  - Centroid
  - Bounding box
- **Spatio-temporal**
  - Displacement,
  - Speed,
  - Acceleration
- **Topological**
  - Number of neighbors
- **Others**
  - Overlap
  - Colocalization

Intensity based features

- Min / max
- Median
- Mean
- Mode
- Variance
- Standard deviation

- Can be derived from pixel values
- Don’t take spatial relationship of pixels into account

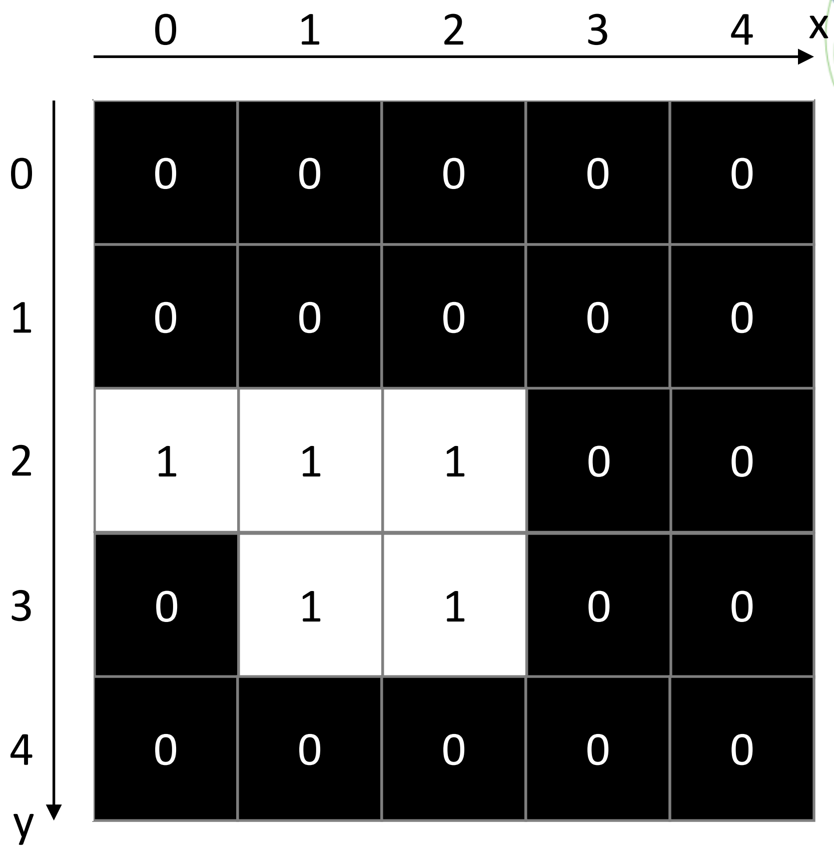
- See also:
  - descriptive statistics
  - histogram



Bounding rectangle / bounding box

- Position and size of the smallest rectangle containing all pixels of an object
  - x_b, y_b … position of the bounding box
  - w_b … width of the bounding box
  - h_b … height of the bounding box

variable	value
x_b	0
y_b	2
w_b	3
h_b	2



Center of mass

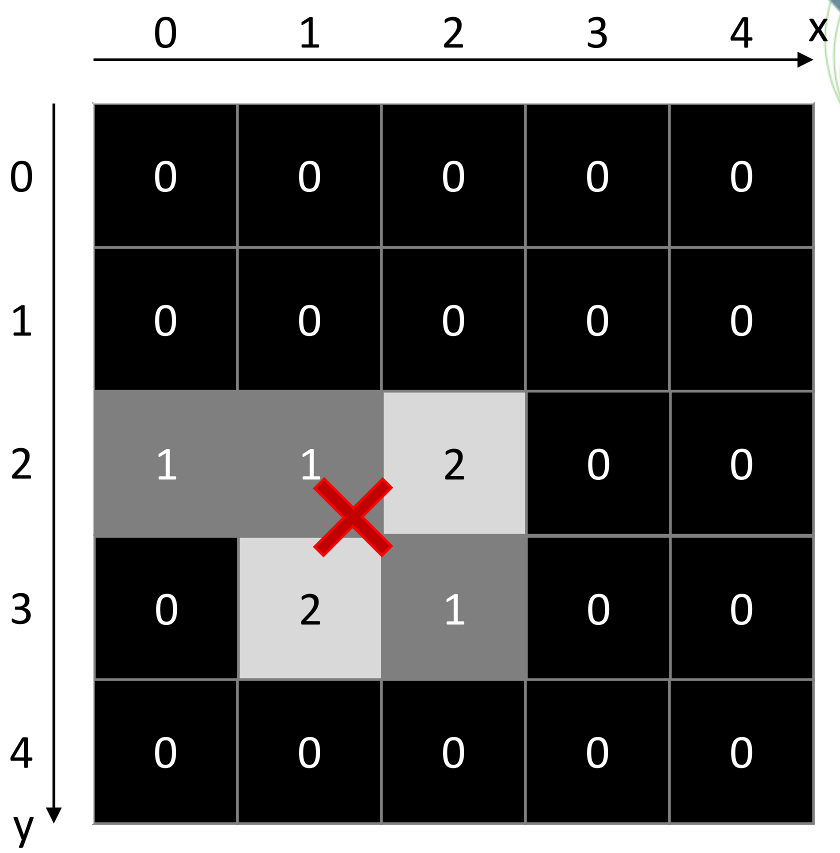
- Relative position in an image weighted by pixel intensities
  - x, y … pixel coordinates
  - w … image width
  - h … image height
  - µ … mean intensity
  - $g_{x,y}$ … pixel grey value
  - x_m, y_m … center of mass coordinates

$$\mu = \frac{1}{wh} \sum_{y=0}^{h-1} \sum_{x=0}^{w-1} g_{x,y}$$

$$x_m = \frac{1}{wh\mu} \sum_{y=0}^{h-1} \sum_{x=0}^{w-1} x\, g_{x,y}$$

$$y_m = \frac{1}{wh\mu} \sum_{y=0}^{h-1} \sum_{x=0}^{w-1} y\, g_{x,y}$$

"sum intensity"
"total intensity"



x_m = 1/7 (1·0 + 1·1 + 2·2 + 2·1 + 1·2) = 1.3

y_m = 1/7 (1·2 + 1·2 + 2·3 + 2·2 + 1·3) = 2.4

Center of geometry / centroid

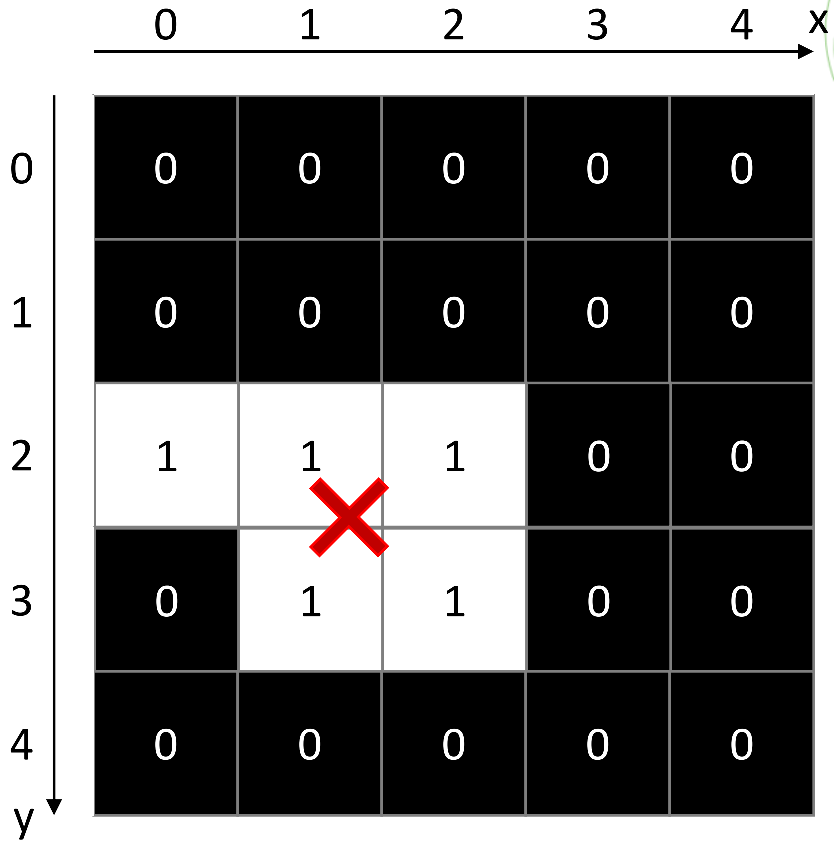
- Relative position in an image weighted by pixel intensities
- Special case of center of mass for binary images
  - x, y … pixel coordinates
  - w … image width
  - h … image height
  - µ … mean intensity
  - $g_{x,y}$ … pixel grey value, integer in range [0;1]
  - x_m, y_m … center of mass coordinates

$$\mu = \frac{1}{wh} \sum_{y=0}^{h-1} \sum_{x=0}^{w-1} g_{x,y}$$

$$x_m = \frac{1}{wh\mu} \sum_{y=0}^{h-1} \sum_{x=0}^{w-1} x \, g_{x,y}$$

$$y_m = \frac{1}{wh\mu} \sum_{y=0}^{h-1} \sum_{x=0}^{w-1} y \, g_{x,y}$$

Number of white pixels

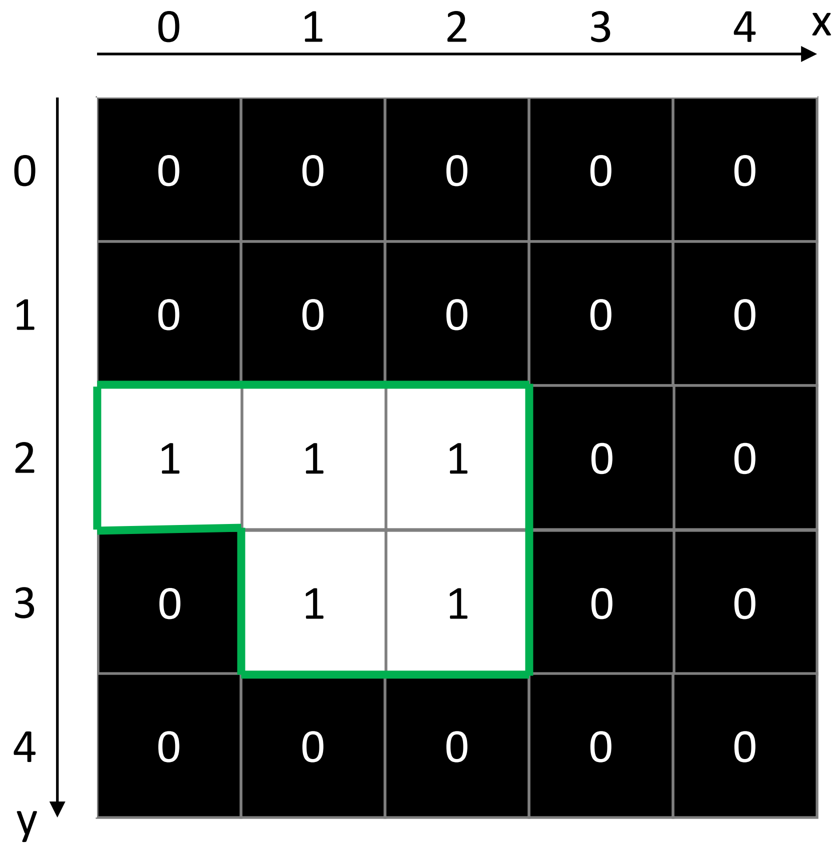


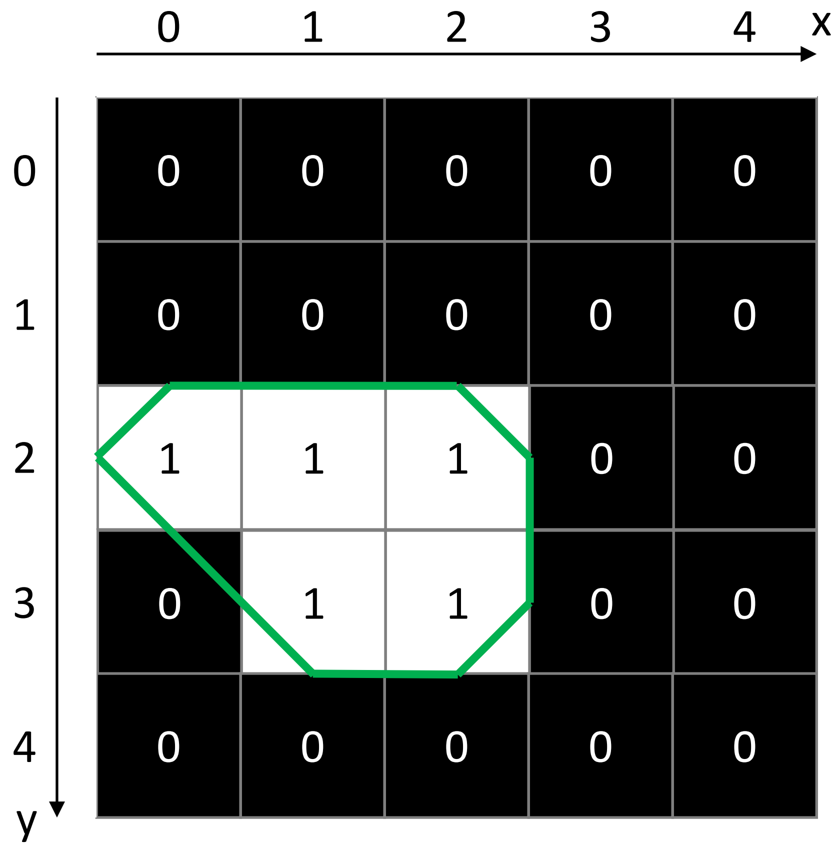
x_m = 1/5 (1·0 + 1·1 + 1·2 + 1·1 + 1·2) = 1.2

y_m = 1/5 (1·2 + 1·2 + 1·3 + 1·2 + 1·3) = 2.4

Perimeter

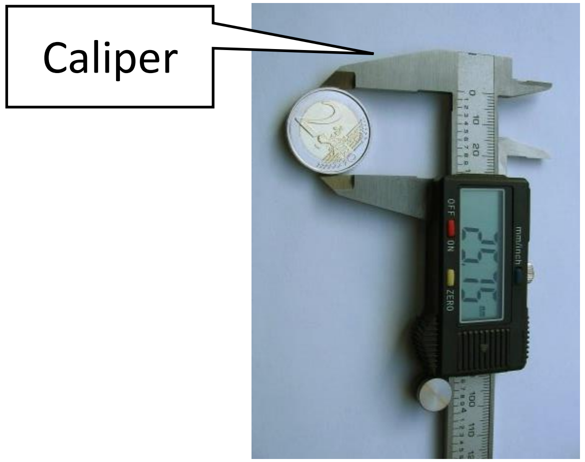
- Length of the outline around an object
- Depends on the actual implementation

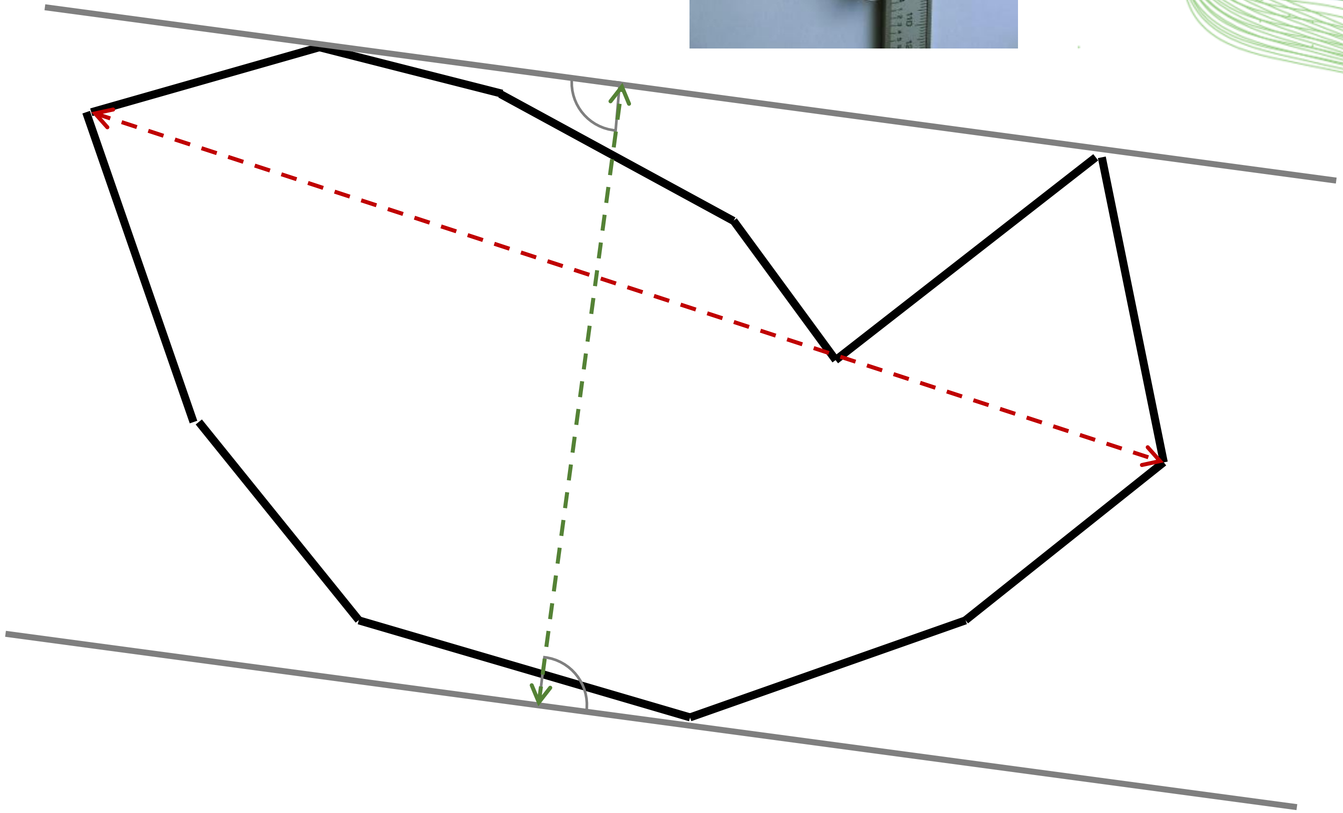




Feret's diameter

- Feret's diameter describes the maximum distance between any two points of an outline.
- The minimum caliper ("Minimum Feret") describes the shortest distance, the object would fit through.
- Feret and Minimum Feret do not need to be perpendicular to each other!

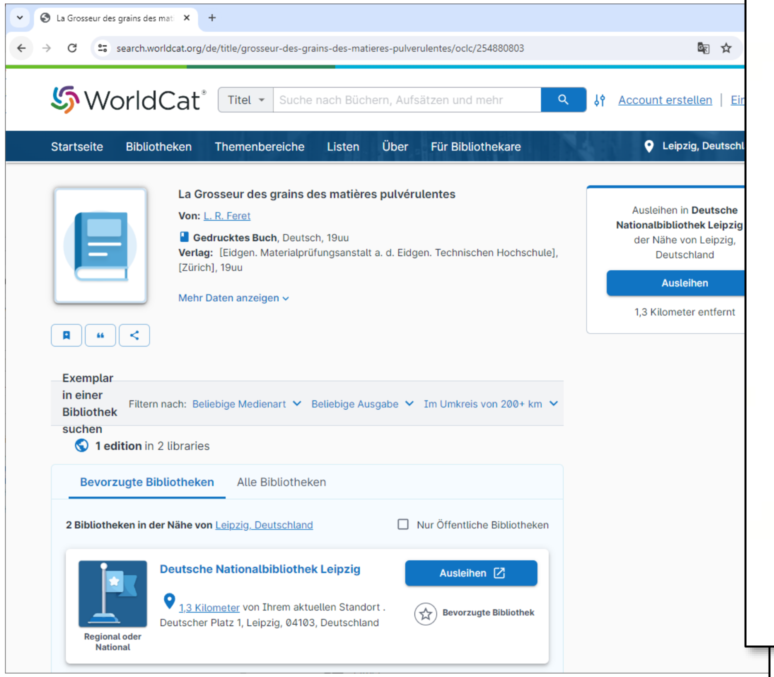


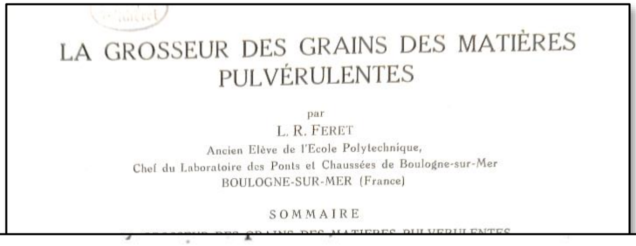


https://en.wikipedia.org/wiki/Feret_diameter#/media/File:DigitalCaliperEuro.jpg

Feret's diameter

- Feret's diameter (L.R. Feret, 1931) is often cited, but impossible to read online …
- The term "Feret's Diameter" was established in the 1970s





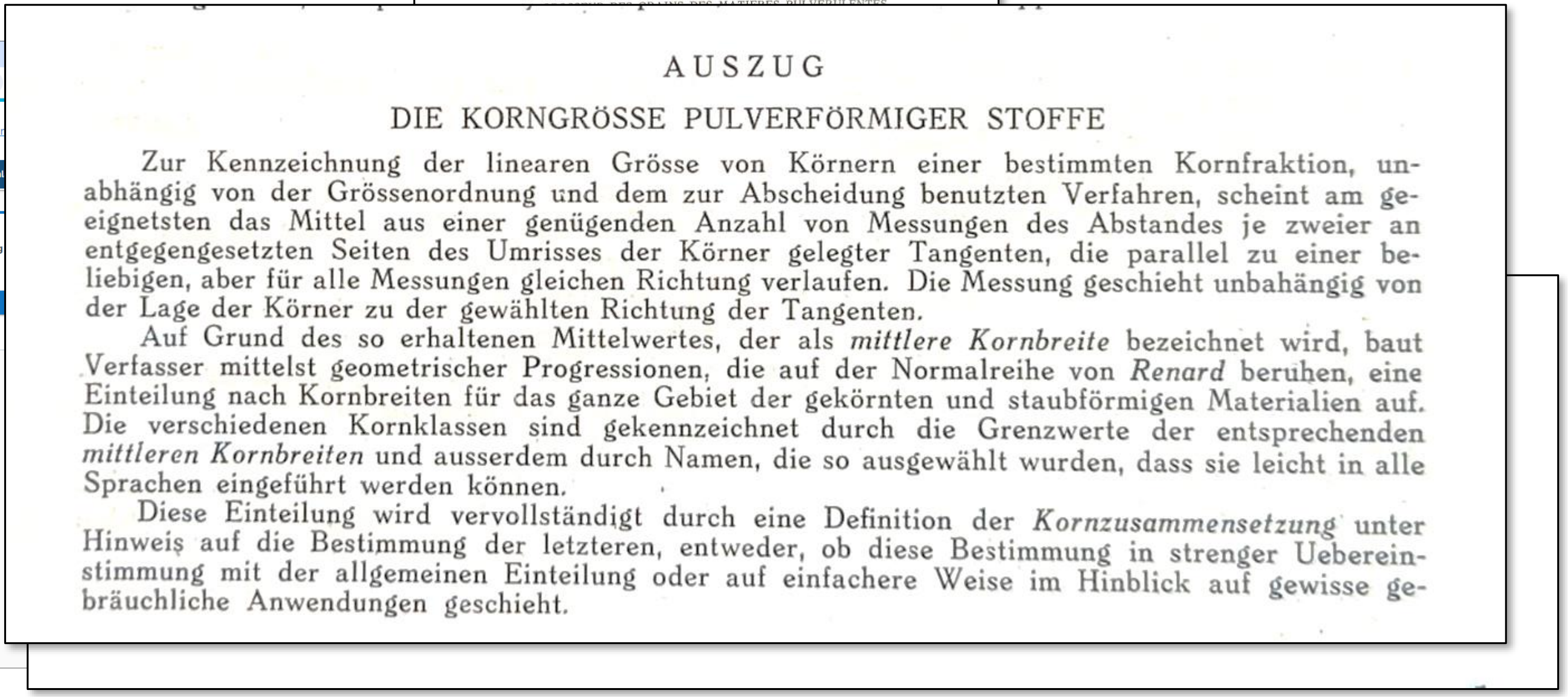
LA GROSSEUR DES GRAINS DES MATIÈRES PULVÉRULENTES

par

L. R. FERET

Ancien Elève de l'Ecole Polytechnique,
Chef du Laboratoire des Ponts et Chaussées de Boulogne-sur-Mer
BOULOGNE-SUR-MER (France)

SOMMAIRE



AUSZUG

DIE KORNGRÖSSE PULVERFÖRMIGER STOFFE

Zur Kennzeichnung der linearen Grösse von Körnern einer bestimmten Kornfraktion, unabhängig von der Grössenordnung und dem zur Abscheidung benutzten Verfahren, scheint am geeignetsten das Mittel aus einer genügenden Anzahl von Messungen des Abstandes je zweier an entgegengesetzten Seiten des Umrisses der Körner gelegter Tangenten, die parallel zu einer beliebigen, aber für alle Messungen gleichen Richtung verlaufen. Die Messung geschieht unbahängig von der Lage der Körner zu der gewählten Richtung der Tangenten.

Auf Grund des so erhaltenen Mittelwertes, der als *mittlere Kornbreite* bezeichnet wird, baut Verfasser mittelst geometrischer Progressionen, die auf der Normalreihe von *Renard* beruhen, eine Einteilung nach Kornbreiten für das ganze Gebiet der gekörnten und staubförmigen Materialien auf. Die verschiedenen Kornklassen sind gekennzeichnet durch die Grenzwerte der entsprechenden *mittleren Kornbreiten* und ausserdem durch Namen, die so ausgewählt wurden, dass sie leicht in alle Sprachen eingeführt werden können.

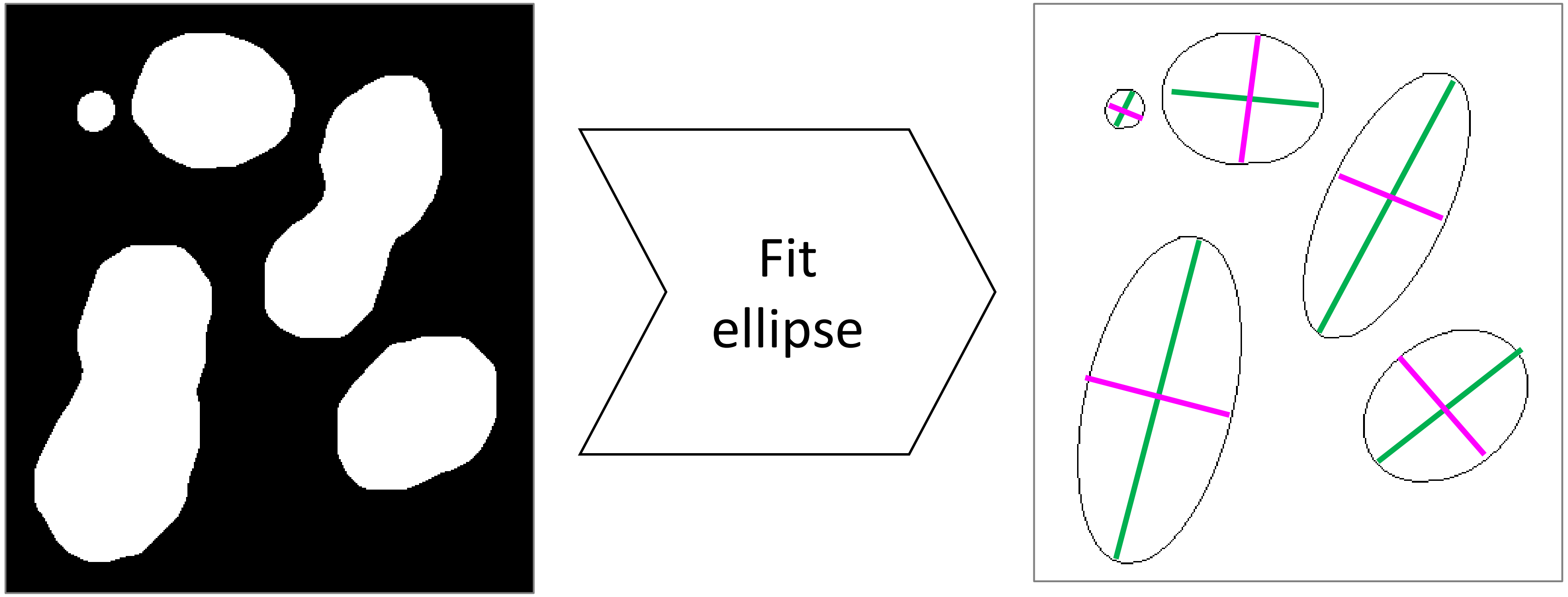
Diese Einteilung wird vervollständigt durch eine Definition der *Kornzusammensetzung* unter Hinweis auf die Bestimmung der letzteren, entweder, ob diese Bestimmung in strenger Uebereinstimmung mit der allgemeinen Einteilung oder auf einfachere Weise im Hinblick auf gewisse gebräuchliche Anwendungen geschieht.

https://search.worldcat.org/de/title/grosseur-des-grains-des-matieres-pulverulentes/oclc/254880803

Minor / major axis

- For every object, find the optimal ellipse simplifying the object.
- Major axis … long diameter
- Minor axis … short diameter

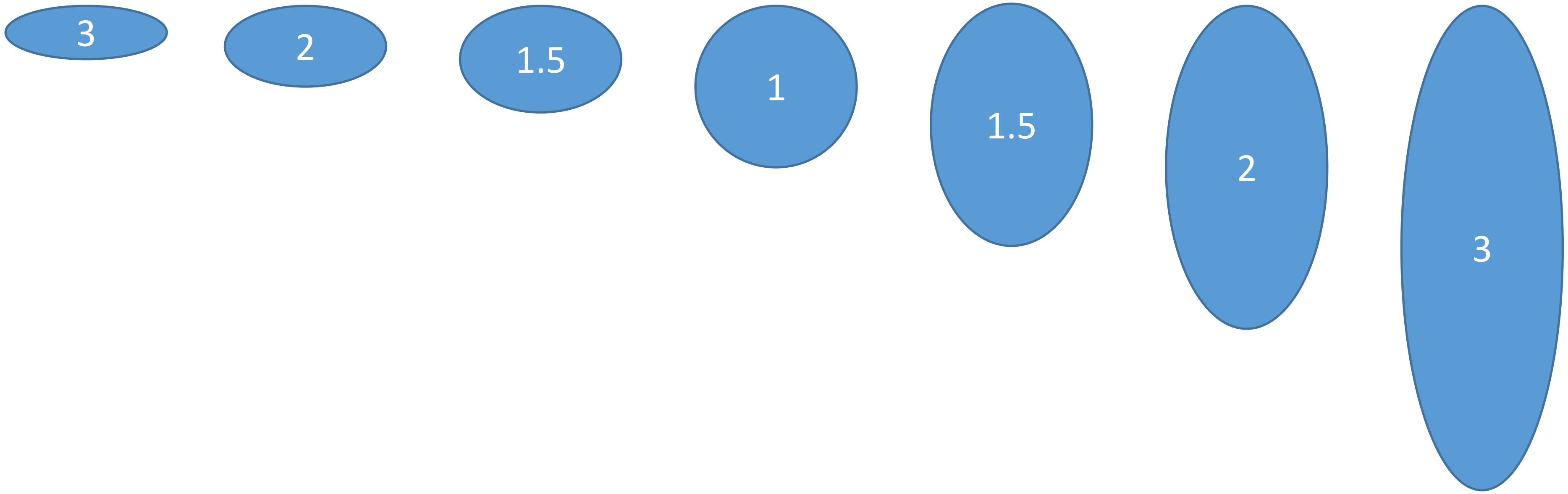
- Major and minor axis are perpendicular to each other



Aspect ratio

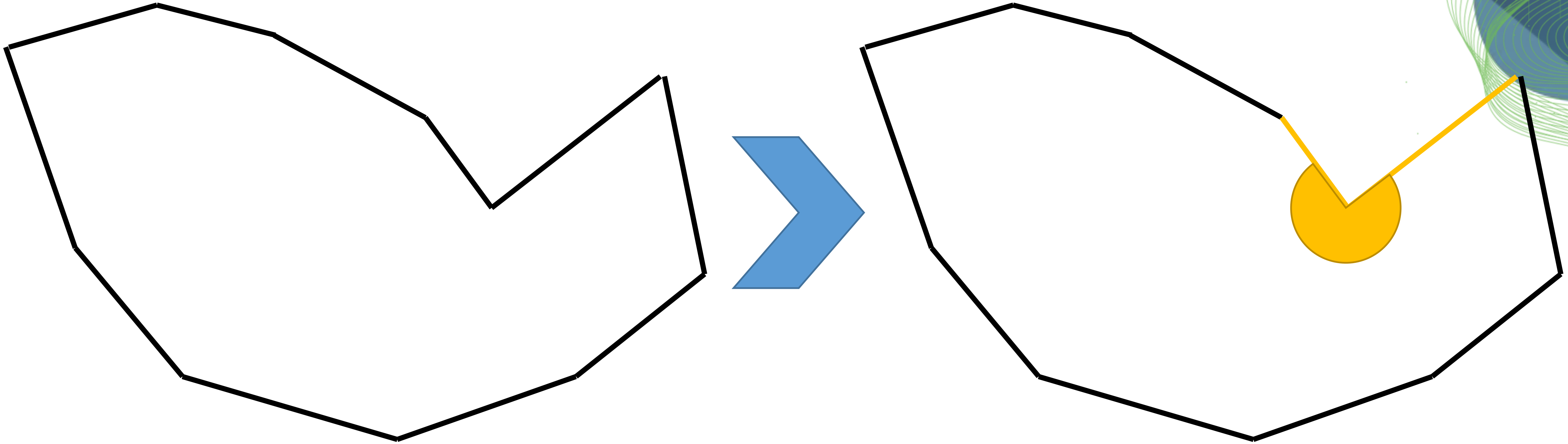
- The aspect ratio describes the elongation of an object.

AR = major / minor



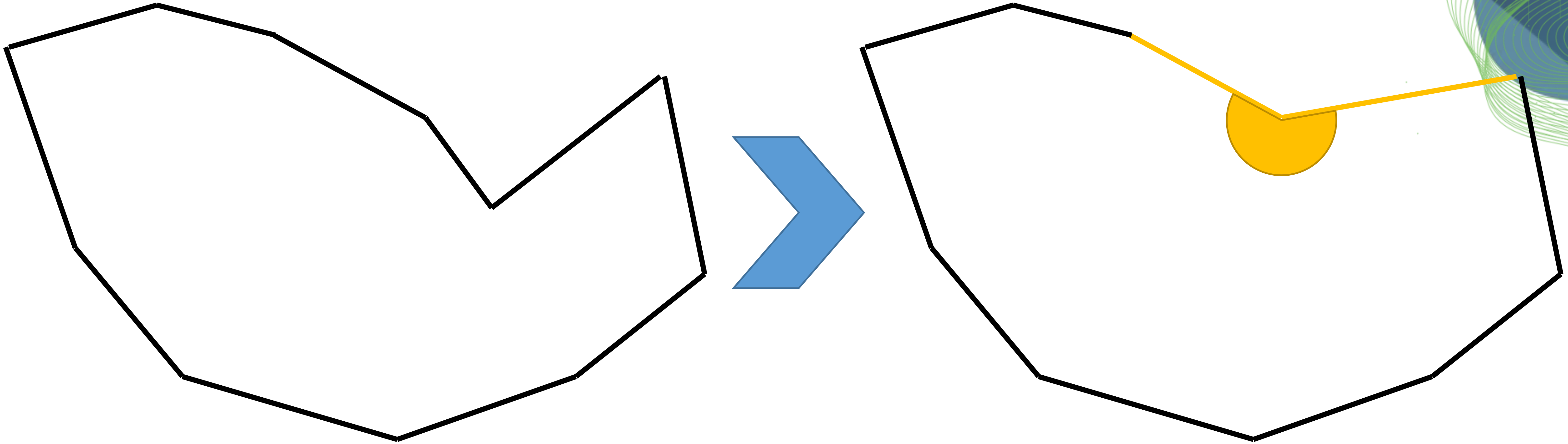
Convex hull

- By removing all concave corners of an object, we retrieve its convex hull.



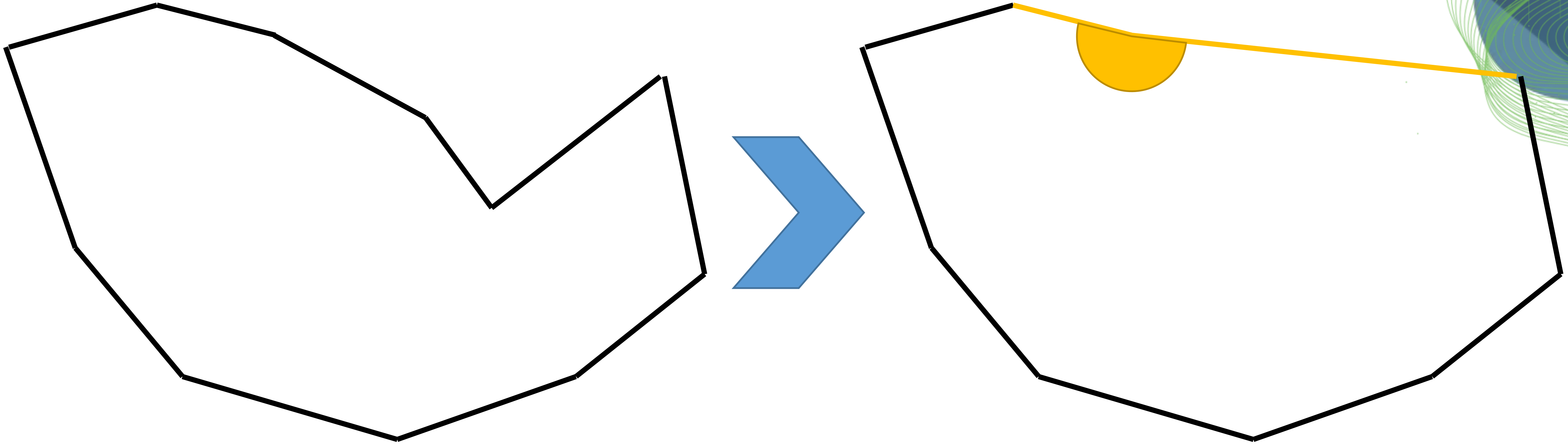
Convex hull

- By removing all concave corners of an object, we retrieve its convex hull.



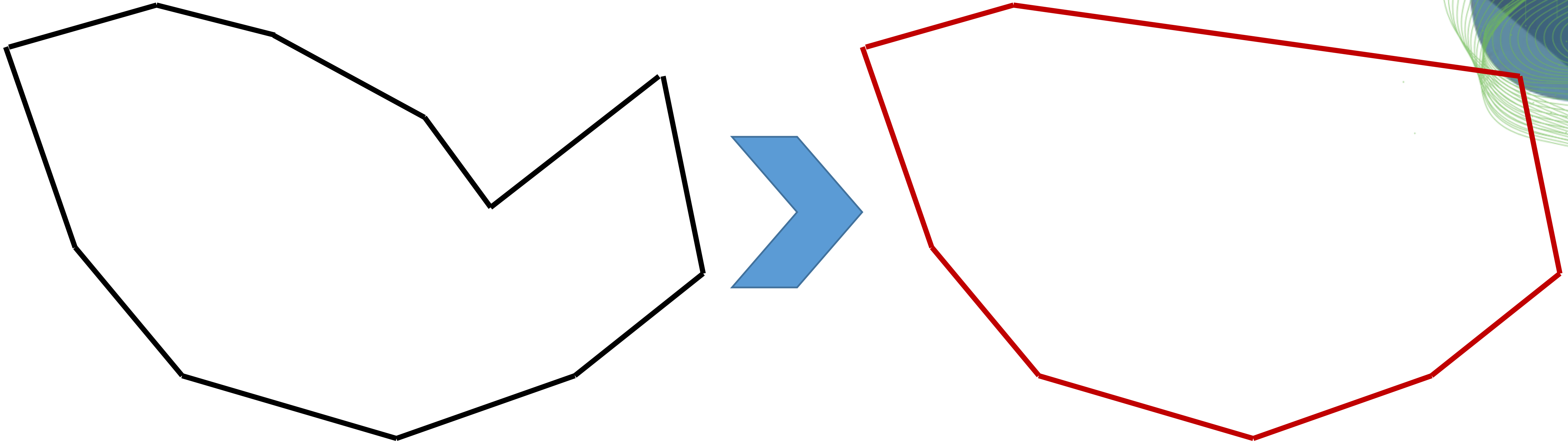
Convex hull

- By removing all concave corners of an object, we retrieve its convex hull.



Convex hull

- By removing all concave corners of an object, we retrieve its convex hull.



$$solidity = \frac{A}{A_{convexHull}}$$

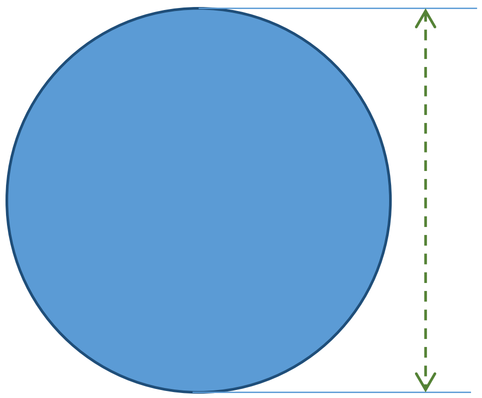
Roundness and circularity

- The definition of a circle leads us to measurements of circularity and roundness.
- In case you use these measures, define them correctly. They are not standardized!

Diameter d

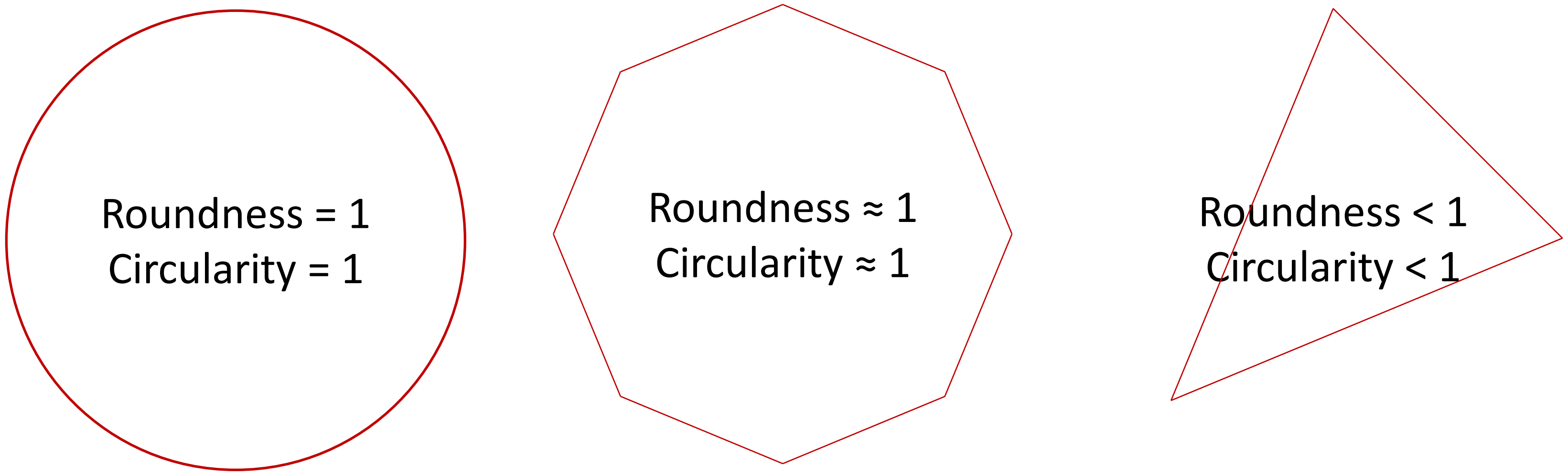
Circumference $C = \pi d$

Area $A = \frac{\pi d^2}{4}$

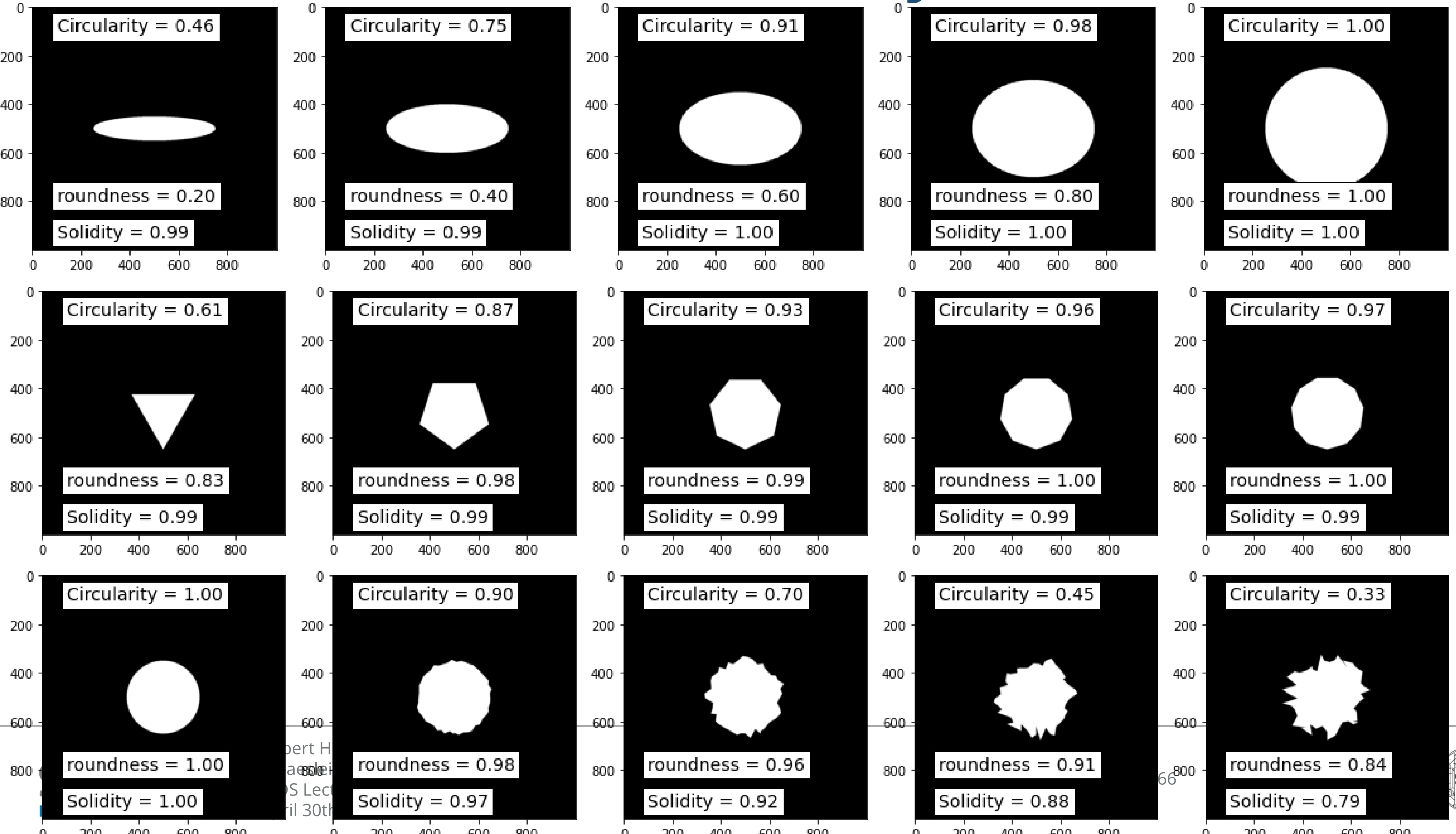


$$roundness = \frac{4 * A}{\pi \, major^2}$$

$$circularity = \frac{4\pi * A}{perimeter^2}$$



Roundness and circularity



$$roundness = \frac{4 \ * A}{\pi \ major^2}$$

$$circularity = \frac{4\pi \ * A}{perimeter^2}$$

$$solidity = \frac{A}{A_{convexHull}}$$

Feature extraction in Python

- **In Python:** `from skimage import measure`

  https://scikit-image.org/docs/stable/api/skimage.measure.html

`skimage.measure.regionprops` (label_image[, …])	Measure properties of labeled image regions.
`skimage.measure.regionprops_table` (label_image)	Compute image properties and return them as a pandas-compatible table.

area : int

Number of pixels of the region.

area_bbox : int

Number of pixels of bounding box.

area_convex : int

Number of pixels of convex hull image, which is the smallest convex polygon that encloses the region.

area_filled : int

Number of pixels of the region will all the holes filled in. Describes the area of the image_filled.

axis_major_length : float

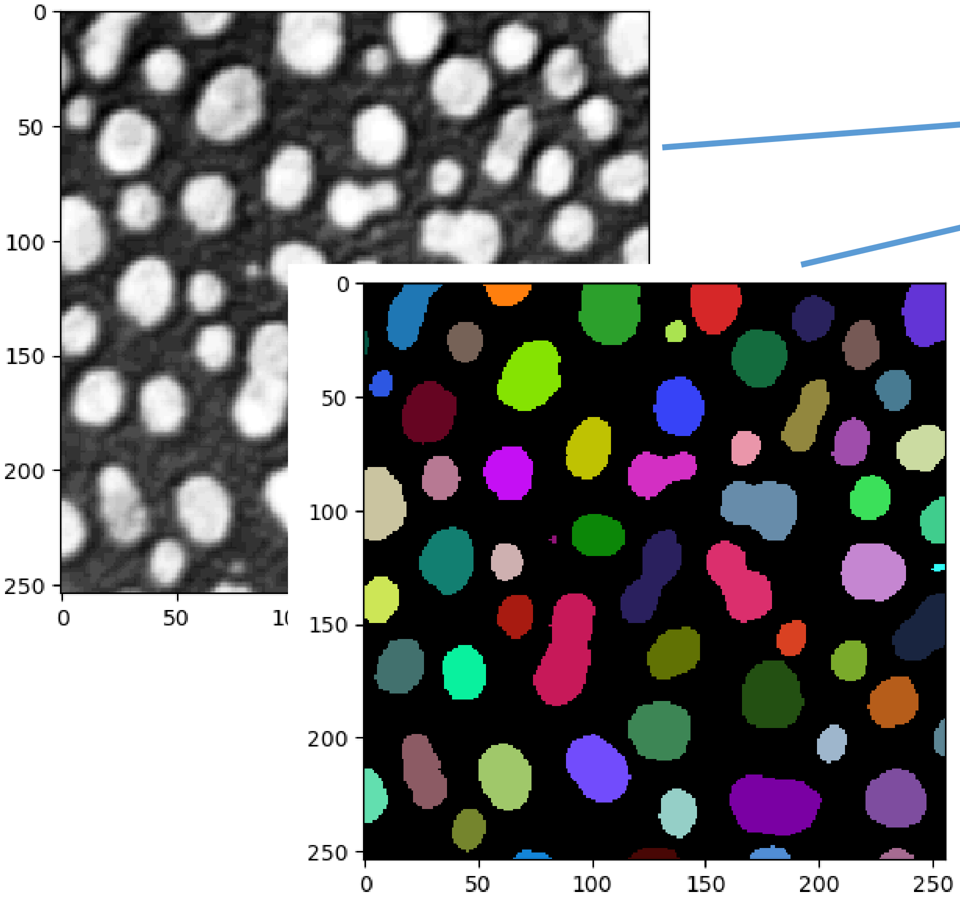
The length of the major axis of the ellipse that has the same normalized second central moments as the region.

axis_minor_length : float

The length of the minor axis of the ellipse that has the same normalized second central moments as the region.

Feature extraction in Python

- The transition from image data to tabular data / pandas DataFrames



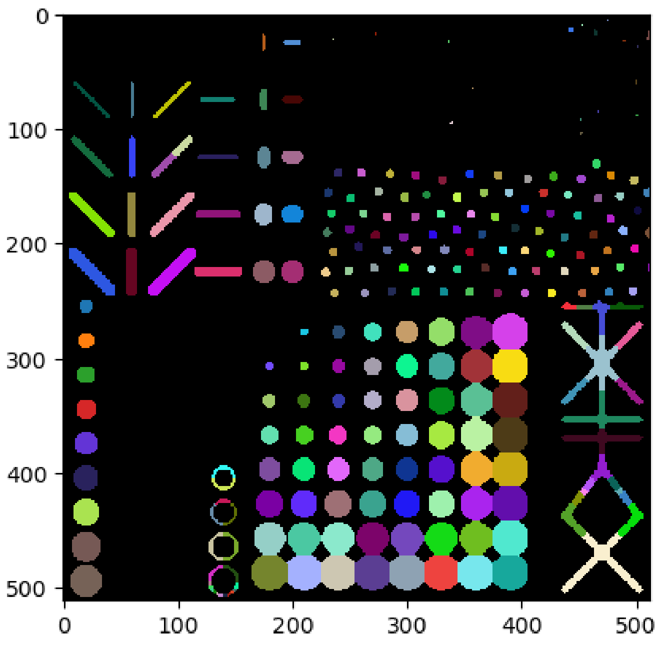
[4]:
```
df = pd.DataFrame(regionprops_table(image , label_image,
                                    perimeter = True,
                                    shape = True,
                                    position=True,
                                    moments=True))
df
```

[4]:

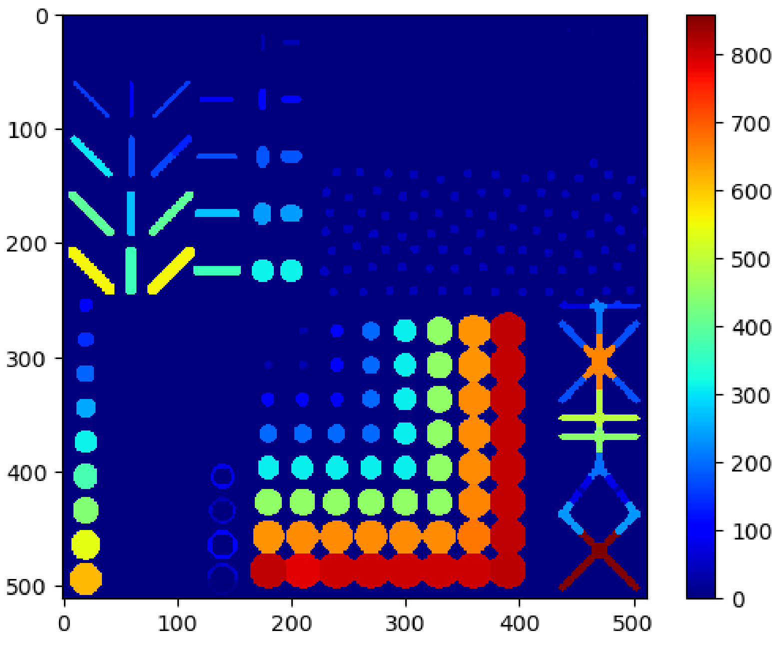
	label	area	bbox_area	equivalent_diameter	convex_area	max_intensity	mean_intensity	min_intensity	perimeter	perimete
0	1	429.0	750.0	23.371345	479.0	232.0	191.440559	128.0	89.012193	
1	2	183.0	231.0	15.264430	190.0	224.0	179.846995	128.0	53.556349	
2	3	658.0	756.0	28.944630	673.0	248.0	205.604863	120.0	95.698485	
3	4	433.0	529.0	23.480049	445.0	248.0	217.515012	120.0	77.455844	
4	5	472.0	551.0	24.514670	486.0	248.0	213.033898	128.0	83.798990	
...	...	...	...	...	...	...	...	...	...	
57	58	213.0	285.0	16.468152	221.0	224.0	184.525822	120.0	52.284271	
58	59	79.0	108.0	10.029253	84.0	248.0	184.810127	128.0	39.313708	
59	60	88.0	110.0	10.585135	92.0	216.0	182.727273	128.0	45.692388	
60	61	52.0	75.0	8.136858	56.0	248.0	189.538462	128.0	30.692388	
61	62	48.0	68.0	7.817640	53.0	224.0	173.833333	128.0	33.071068	

Parametric images

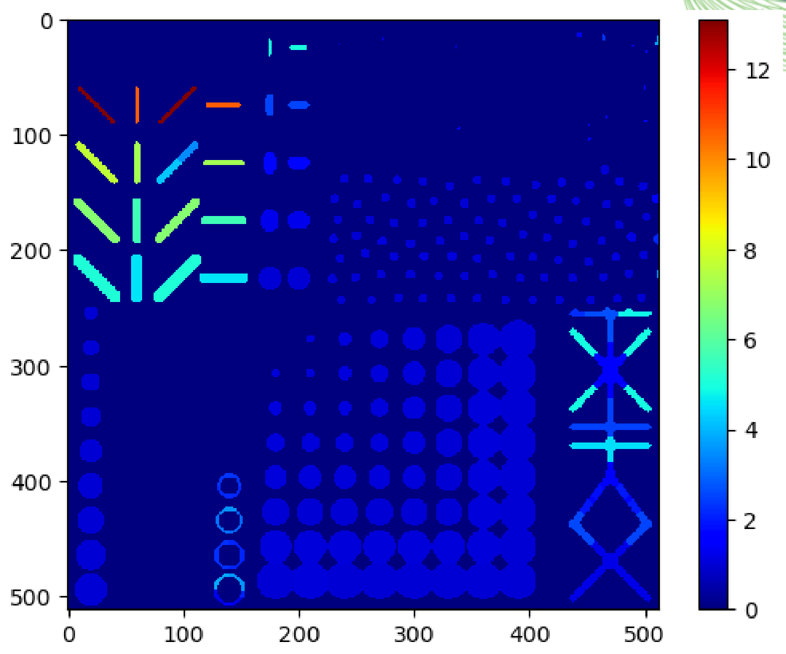
- The way back: Visualizing quantitative measurements



Label image



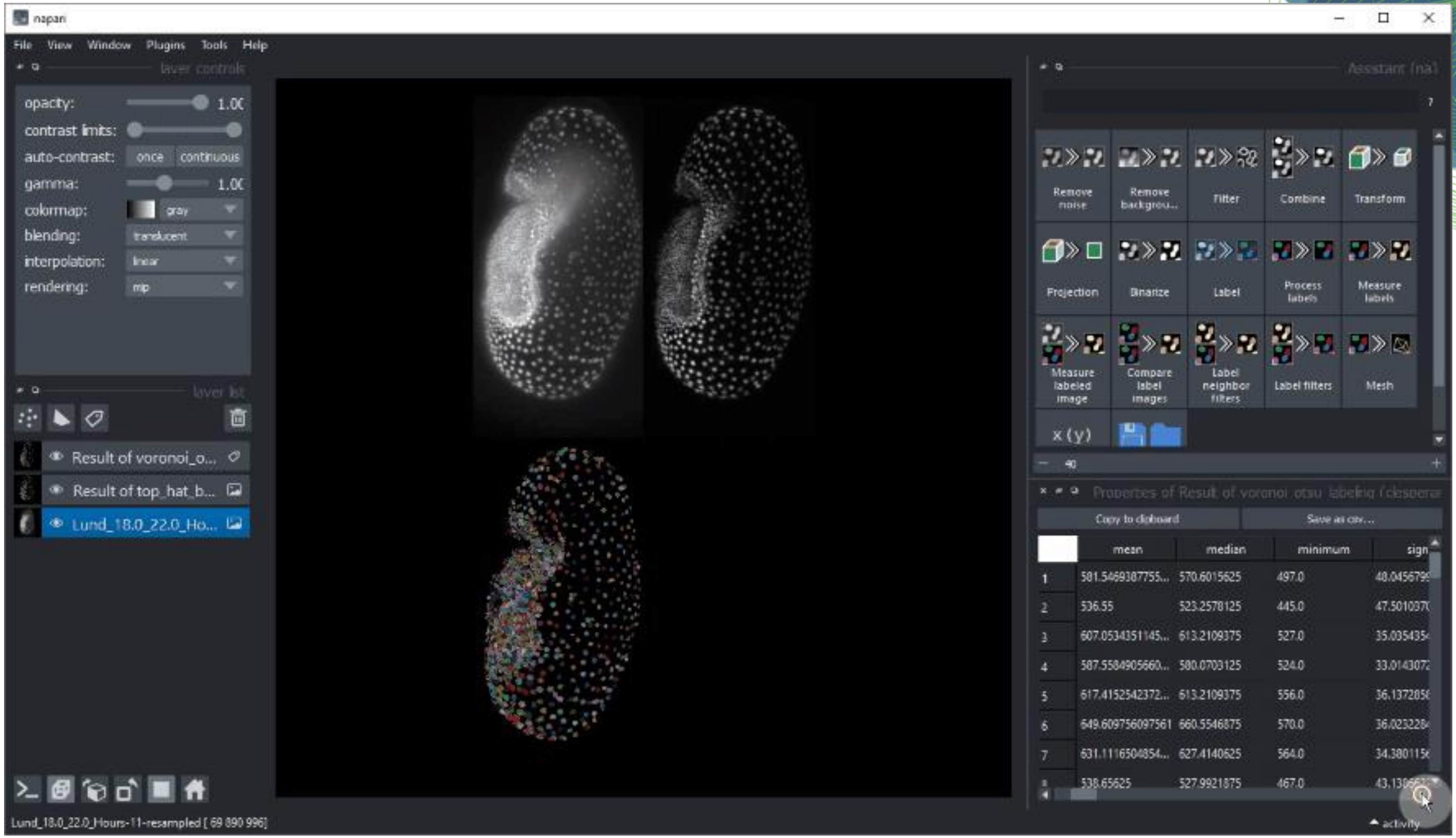
Pixel count image



Aspect ratio image

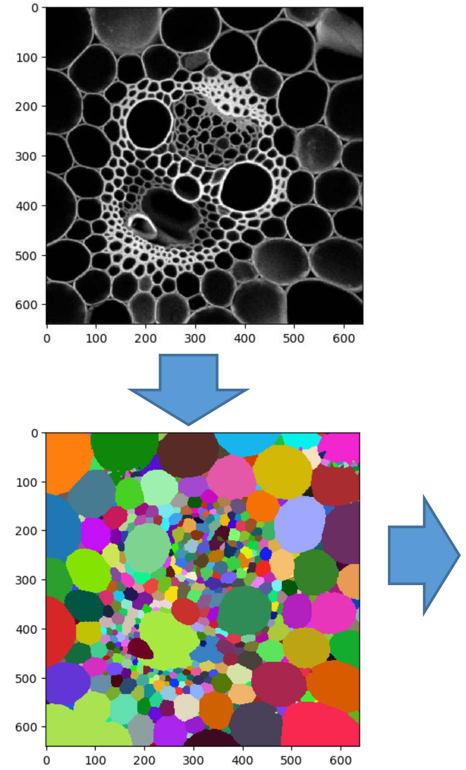
Exploring features in Napari

- Double-click on table column to retrieve a parametric map image

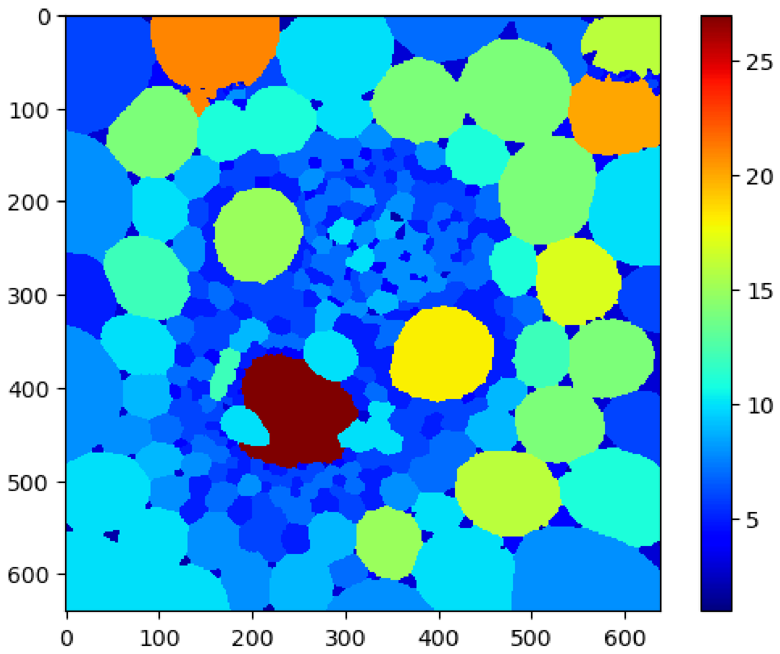


Exploring neighborhood relationships between cells

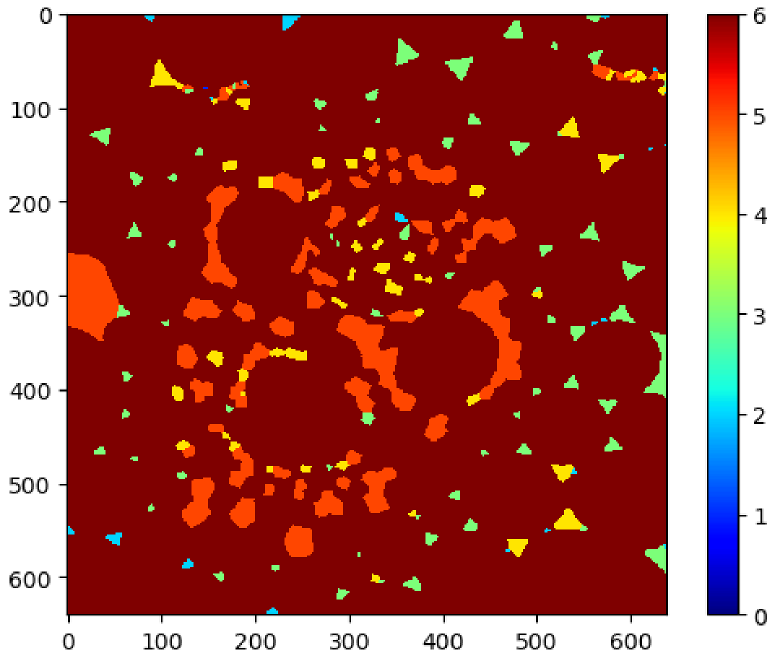
- Study how many neighbours objects have.
- How likely is it that an object with 3 neighbors is a cell?




```python
num_neighbors_map = cle.touching_neighbor_count_map(objects)

cle.imshow(num_neighbors_map,
           color_map='jet',
           colorbar=True)
```



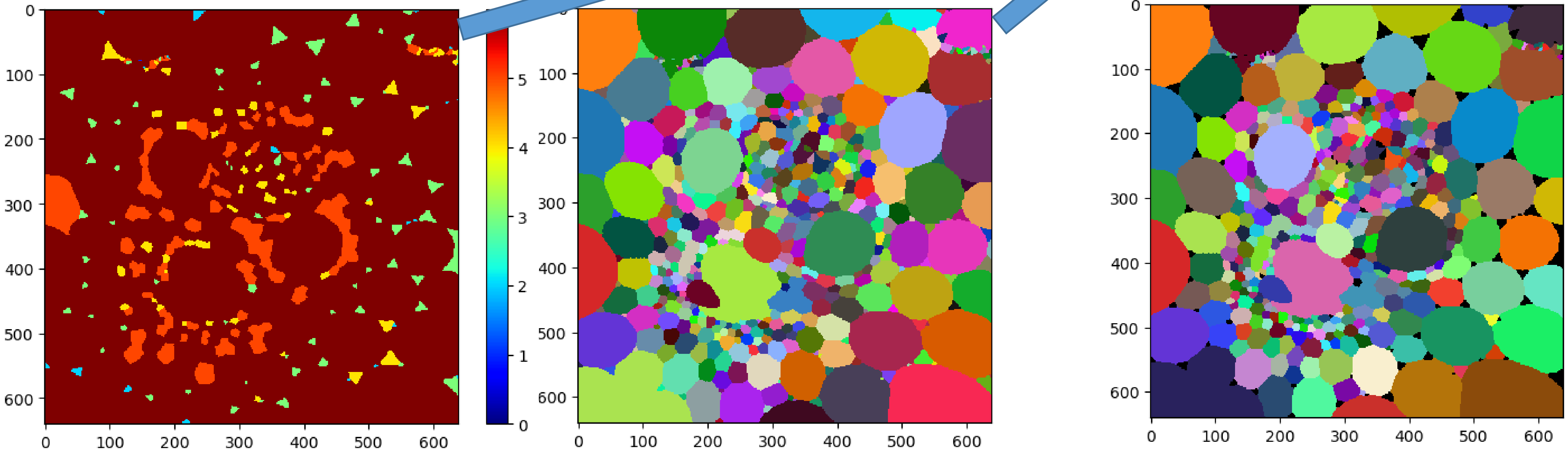

```python
cle.imshow(num_neighbors_map,
           min_display_intensity=0,
           max_display_intensity=6,
           color_map='jet',
           colorbar=True)
```



Neighborhood-based label filters

- Filter out objects which have an unreasonable number of neighbors

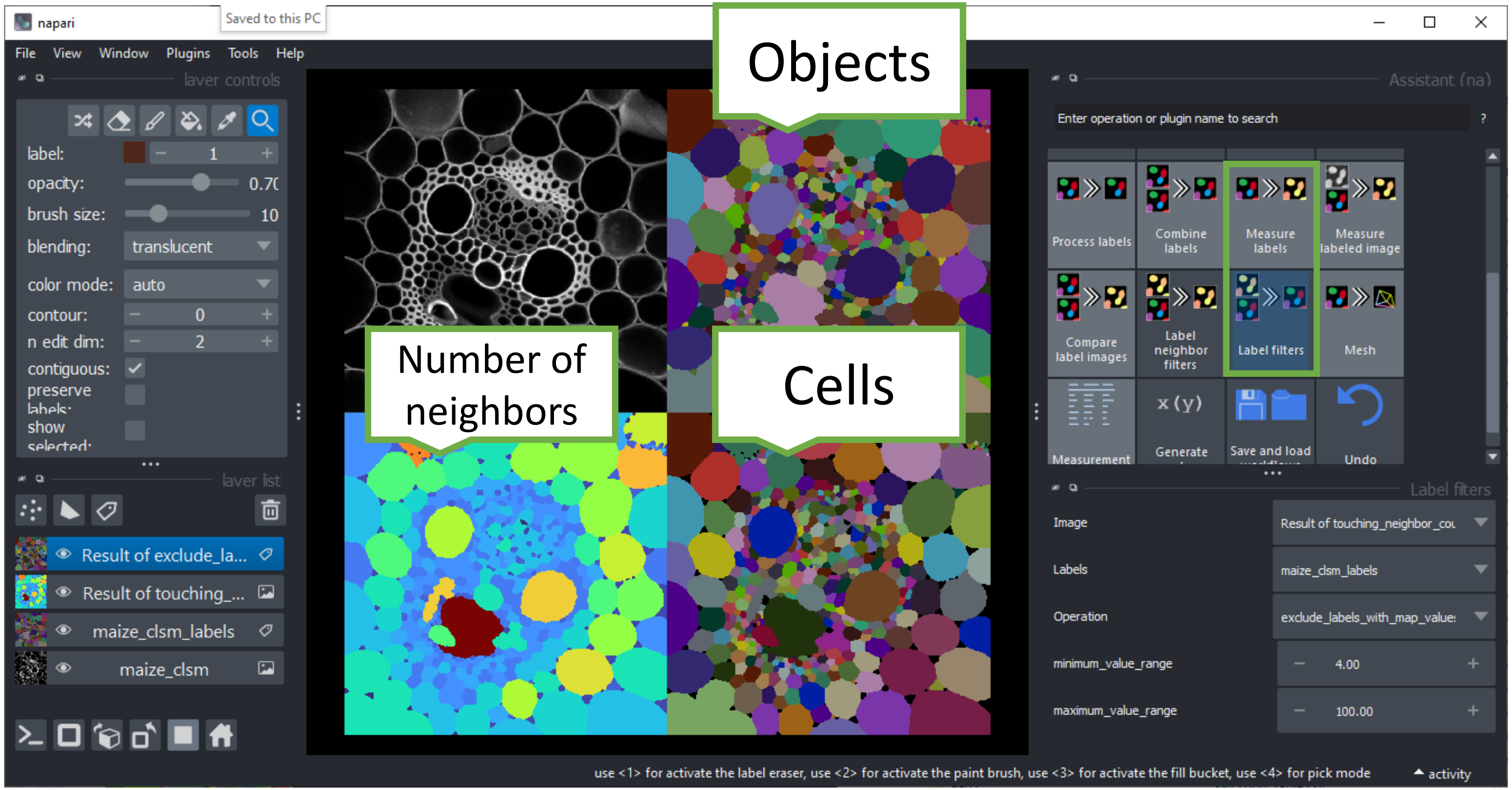
```
cells = cle.exclude_labels_with_map_values_out_of_range(num_neighbors_map, objects, minimum_value_range=4)

cle.imshow(cells, labels=True)
```



neighbor count objects cells

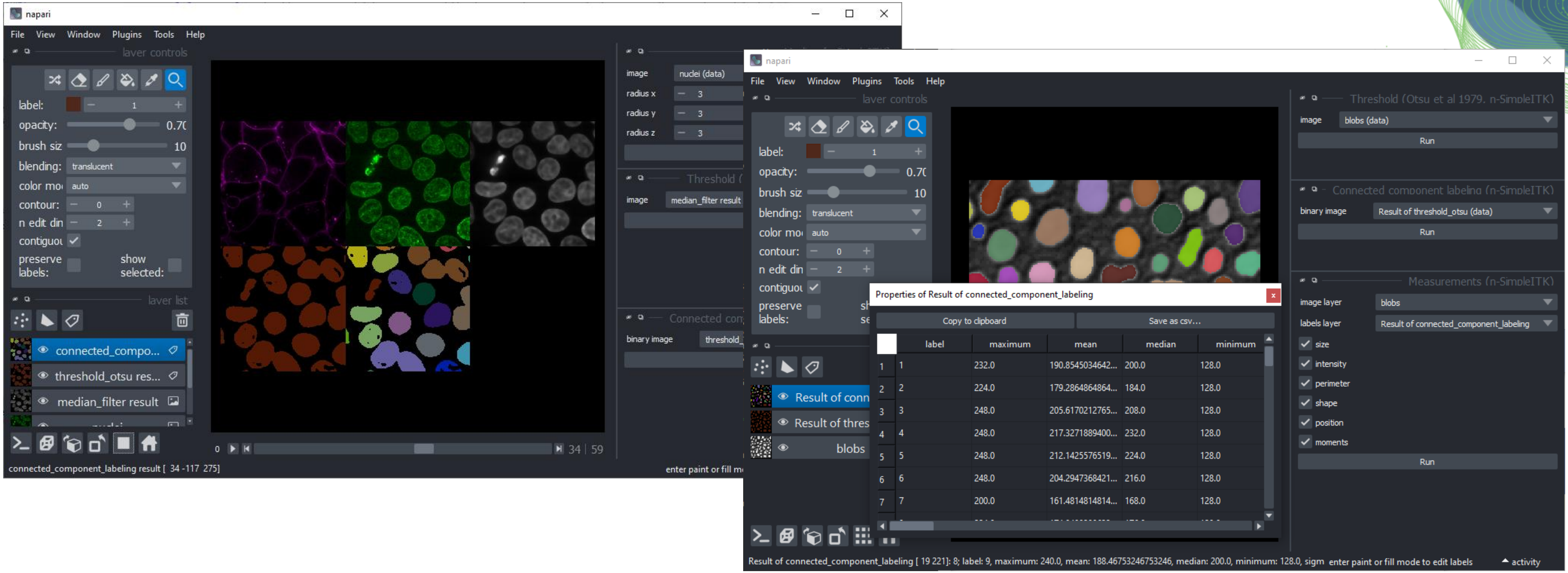
Neighborhood-based label filters

- Filter labeled objects using Measure Labels and Label Filters in Napari.



SimpleITK

- Recommended for 3D-measurements, based on the SimpleITK-project



SimpleITK

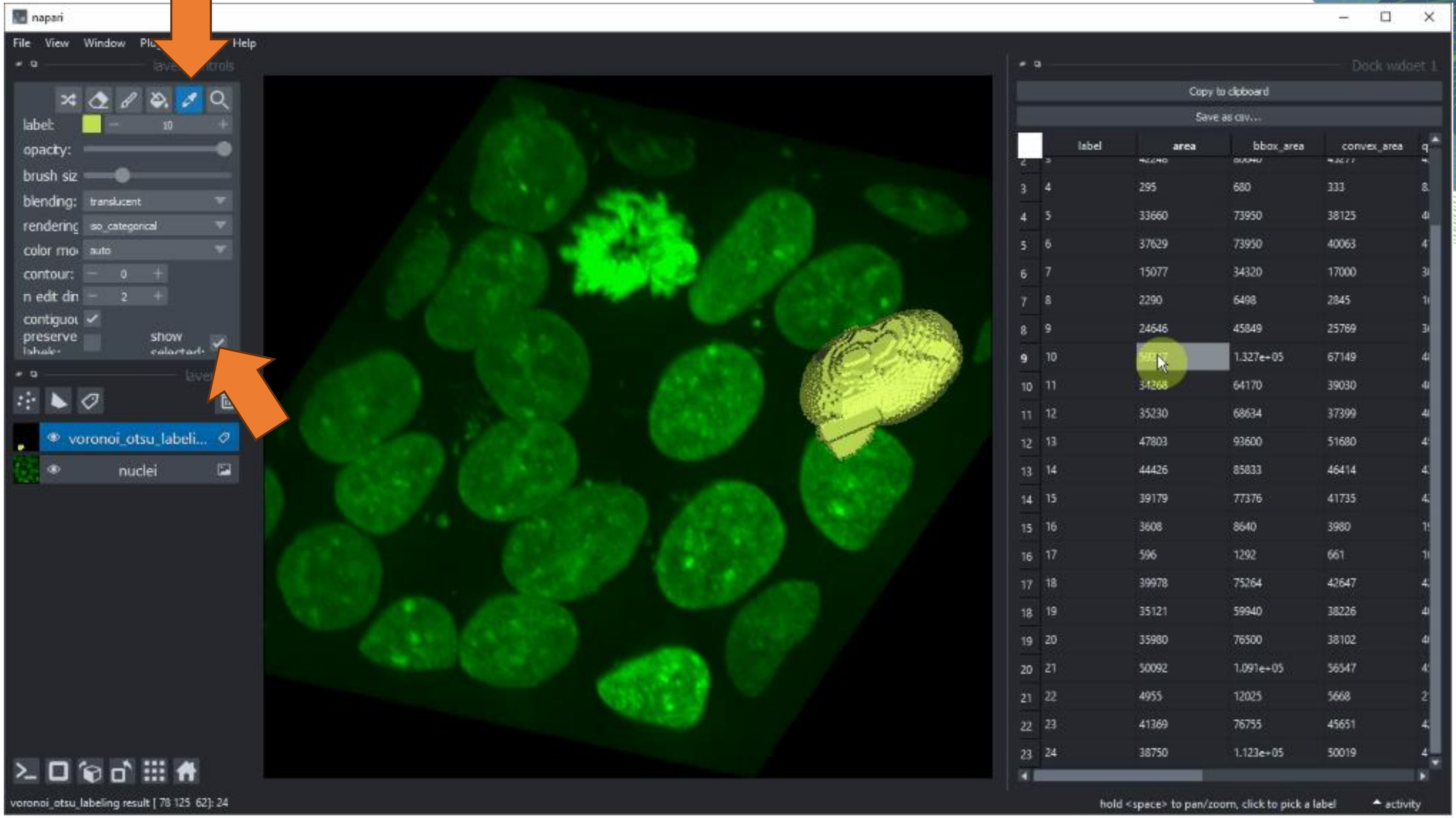
- Many Napari plugins for feature extraction can also be called from Python

```
statistics = label_statistics(blobs, labels,
                              intensity=True,
                              size=True,
                              shape=True,
                              perimeter=True,
                              position=True,
                              moments=True)

df = pd.DataFrame(statistics)
df
```

	label	maximum	mean	median	minimum	sigma	sum	variance	bbox_0	bbox_1
0	1	224.0	137.526132	136.0	112.0	13.360739	157880.0	178.509343	0	0
1	2	232.0	193.014354	200.0	128.0	28.559077	80680.0	815.620897	11	0
2	3	224.0	179.846995	184.0	128.0	21.328889	32912.0	454.921516	53	0
3	4	248.0	207.082171	216.0	120.0	27.772832	133568.0	771.330194	95	0
4	5	248.0	223.146402	232.0	128.0	30.246515	89928.0	914.851647	144	0
5	6	248.0	214.906725	224.0	128.0	26.386796	99072.0	696.263020	238	0
6	7	248.0	211.565891	224.0	136.0	30.197236	54584.0	911.873073	189	7
7	8	200.0	166.171429	168.0	136.0	16.466894	11632.0	271.158592	133	17

Exploring features in Napari

- Select table rows and view corresponding object in 2D/3D space



CENTER FOR SCALABLE DATA ANALYTICS
AND ARTIFICIAL INTELLIGENCE

Complex exercise

Robert Haase

GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung

SACHSEN
Diese Maßnahme wird gefördert durch die Bundesregierung aufgrund eines Beschlusses des Deutschen Bundestages. Diese Maßnahme wird mitfinanziert durch Steuermittel auf der Grundlage des von den Abgeordneten des Sächsischen Landtags beschlossenen Haushaltes.

Complex exercise

- Scenario: Imagine a biologist sent you some data (images + corresponding label image). They ask you to write an image-analysis workflow for processing these images + more images of similar kind.
- You will receive a link to data in-person
  - You can return the link and exchange it with another link 2 times.
- Scientific tasks
  - Develop an image-segmentation workflow, which produces label images
  - Extract features from these images
  - Visualize relationships between these features
  - Find out which features are strongly correlated and which not.

Complex exercise

- Engineering tasks
  - Setup a software environment
  - Setup an image processing workflow
  - Setup a data analysis / visualization workflow
  - Setup a quality assurance procedure
- Documentation tasks
  - Installation instructions
  - User guide
  - Documentation of used data
  - Explanation of the used algorithms

Act as if you would communicate with a biologist, with limited image-analysis, conda and programming skills.

Complex exercise

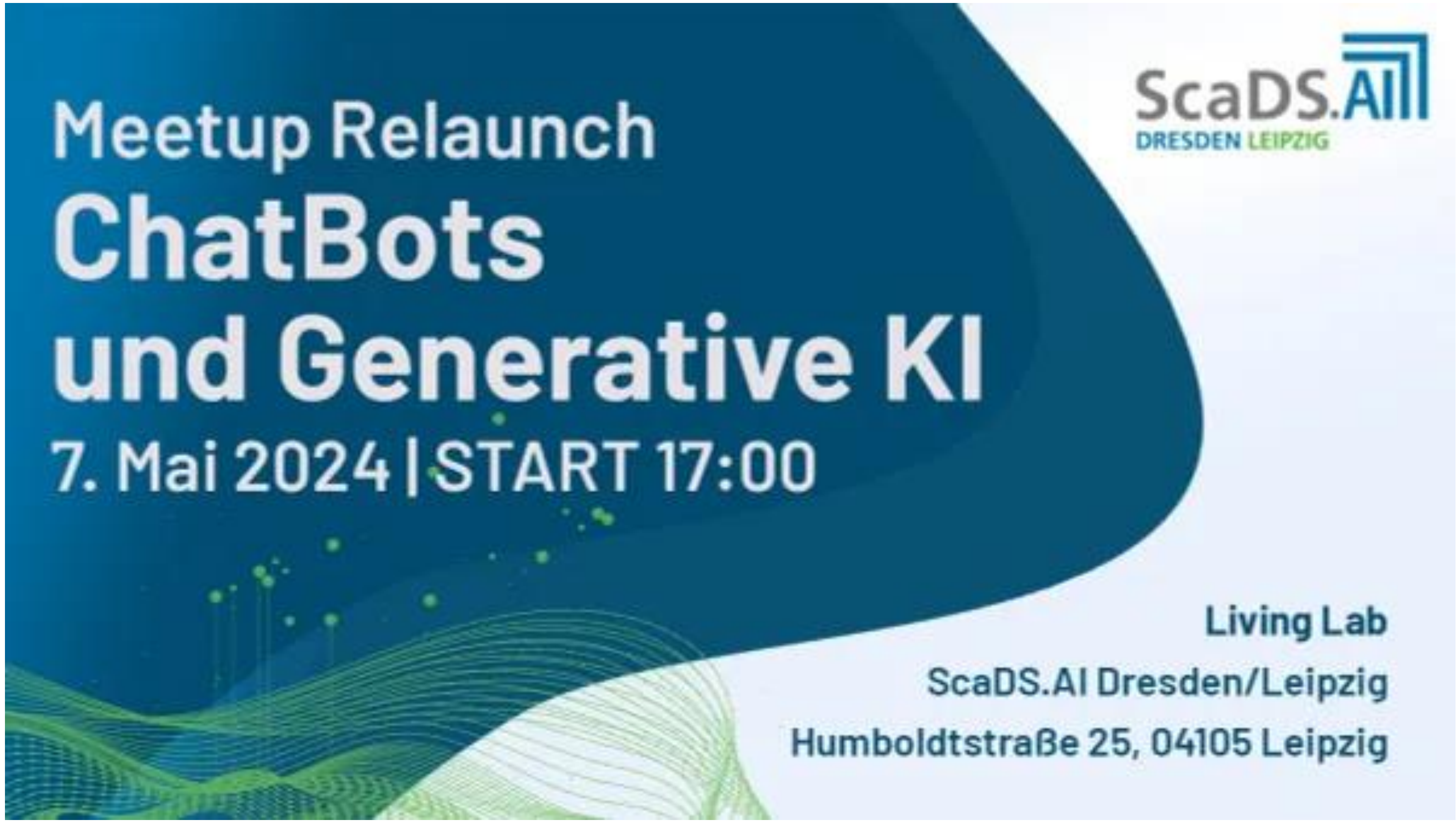
- Submission
  - Submit a password-protected ZIP file to robert.haase@uni-leipzig.de (Why password protected: The virus scanner cannot reject python files in encrypted zip-files)
  - Allowed file formats: ipynb, py, docx, pdf, md, csv, yml, json, xml, txt
  - Deadline: June 24th 2024
- Hint
  - Send this ZIP file to a friend and ask them to run the analysis. If they can follow your instructions successfully, without communicating with you, proceed to final submission.

Complex exercise

- Checklist
  - The software environment is reproducible
  - The example data is available in the right directory (note: you cannot submit a 500MB ZIP file via email) The image/data analysis code is executable
  - The code is well documented / commented
  - Segmentation results are visualized
  - Segmentation results are stored to disc as label images
  - The quality of the segmentation result is measured
  - Used algorithms are cited, and well explained
  - Extracted features / measurements are saved as CSV-file in a way that one can associate an entry in the CSV file with the corresponding segmented object
  - Resulting plots and visualizations have reasonable axis labels and are well explained
  - The copyright of re-used data and code are respected

Next week:

- 15:15-16:30 Short lecture + short practicals (SG 312)
- Afterwards:



CENTER FOR SCALABLE DATA ANALYTICS AND ARTIFICIAL INTELLIGENCE

Exercises

Robert Haase

GEFÖRDERT VOM

Bundesministerium für Bildung und Forschung

SACHSEN

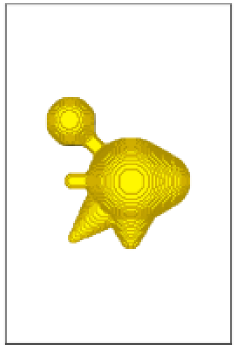
Diese Maßnahme wird gefördert durch die Bundesregierung aufgrund eines Beschlusses des Deutschen Bundestages. Diese Maßnahme wird mitfinanziert durch Steuermittel auf der Grundlage des von den Abgeordneten des Sächsischen Landtags beschlossenen Haushaltes.

Exercise: Surface meshes

- Creating, storing, processing surface mesh data

```
[9]: new_surface = nppas.to_napari_surface_data(new_mesh)
     new_surface
```

[9]:

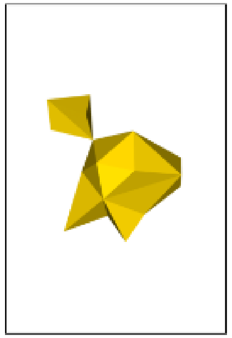


nppas.SurfaceTuple

origin (z/y/x)	[0. 0. 0.]
center of mass(z/y/x)	50.000,46.575,42.589
scale(z/y/x)	1.000,1.000,1.000
bounds (z/y/x)	25.500...74.500 2.500...88.500 2.500...83.500
average size	31.277
number of vertices	19040
number of faces	38076

```
[11]: simplified_surface2 = nppas.decimate_quadric(smoothed_surface, fraction=0.001)
      simplified_surface2
```

[11]:



nppas.SurfaceTuple

origin (z/y/x)	[0. 0. 0.]
center of mass(z/y/x)	49.959,46.174,40.689
scale(z/y/x)	1.000,1.000,1.000
bounds (z/y/x)	25.809...76.552 3.858...93.107 3.385...83.730
average size	35.058
number of vertices	22
number of faces	38

Exercise

Load the `skimage.data.cells3d` dataset, extract the nuclei channel and create a surface mesh for every individual nucleus. Store all these surface meshes to disc.

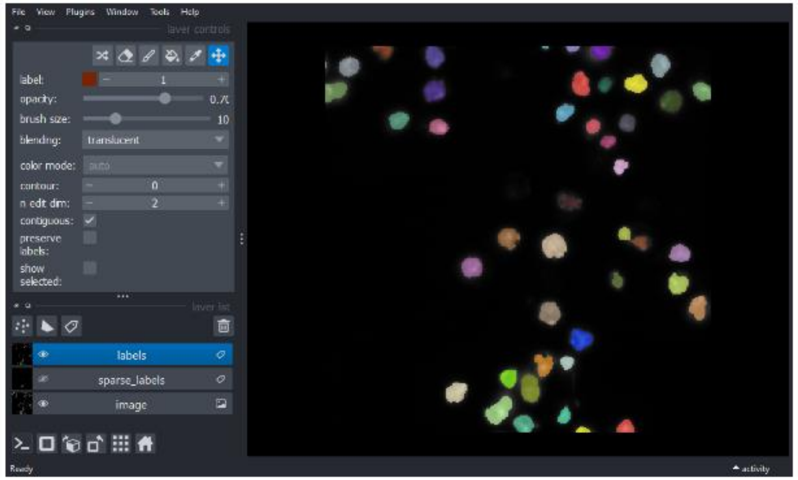
Exercise

Store the branchoid as 8-bit integer image to disc. Compare the file size to the differently simplified meshes above.

Segmentation quality

- Measure segmentation quality of a given algorithm applied to a folder of images.

```
[7]: def my_segmentation_algorithm(input_image):

         # background subtraction
         background_subtracted = nsbatwm.white_tophat(input_image, radius = 10

         # instance segmenation / labeling
         labels_result = nsbatwm.voronoi_otsu_labeling(background_subtracted,

         return labels_result
```



Quality estimation: Sparse Jaccard Index

From the two label images loaded and produced above we can compute the sparse Jaccard Index.

```
[9]: metrics.jaccard_index_sparse(sparse_labels, labels)

[9]: 0.8357392602053431
```

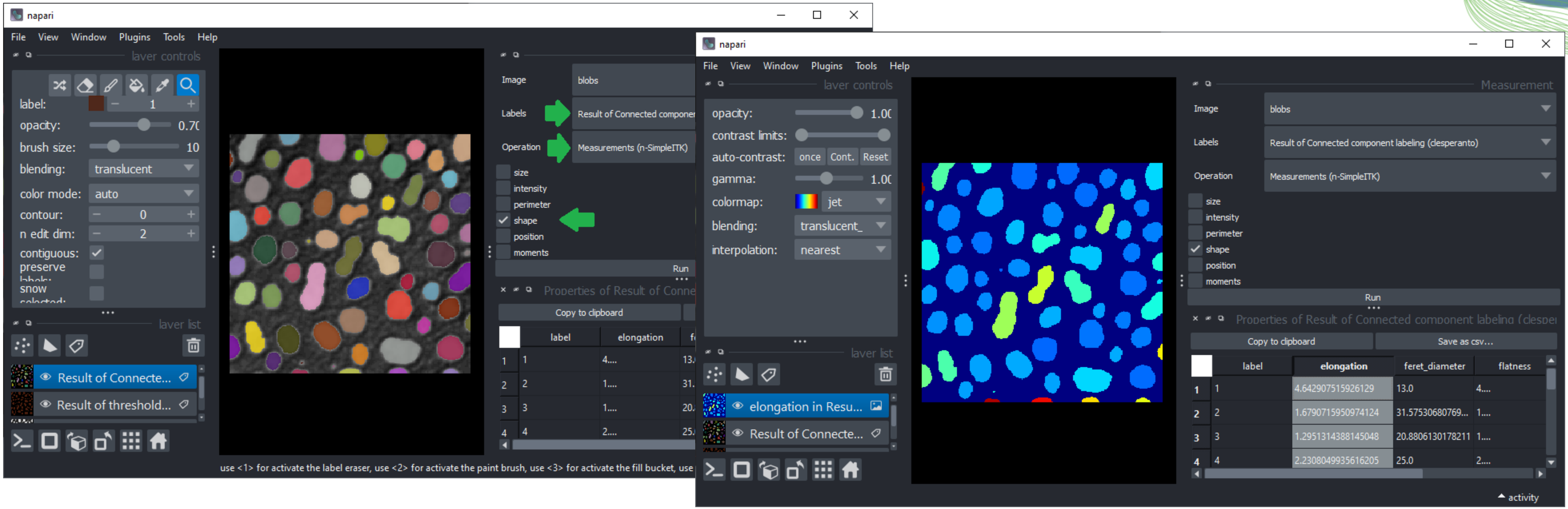
Exercise

Use the following for-loop and code snippets from above to compute the segmentation quality of all images in the folder. Provide the average quality over all images.

```
[10]: for image_filename in os.listdir(image_folder):
          print(image_folder + image_filename)

data/BBBC007_batch/17P1_POS0013_D_1UL.tif
data/BBBC007_batch/20P1_POS0005_D_1UL.tif
data/BBBC007_batch/20P1_POS0007_D_1UL.tif
data/BBBC007_batch/20P1_POS0010_D_1UL.tif
data/BBBC007_batch/A9 p7d.tif
data/BBBC007_batch/AS_09125_040701150004_A02f00d0.tif
```

Exercise: Parametric maps

- Produce a parametric map representing ‘elongation’ in Napari.
- Reproduce the same map using Python



Exercise: Quantitative measurements

- Use the given feature extraction notebook to apply some basic statistics to measurements

[5]:
```
df = pd.DataFrame(regionprops_table(image , label_image,
                                    perimeter = True,
                                    shape = True,
                                    position=True,
                                    moments=True))
df
```

[5]:

	label	area	bbox_area	equivalent_diameter	convex_area	max_intensity	mean_intensity	min_intensity	perimeter	perimete
0	1	429.0	750.0	23.371345	479.0	232.0	191.440559	128.0	89.012193	
1	2	183.0	231.0	15.264430	190.0	224.0	179.846995	128.0	53.556349	
2	3	658.0	756.0	28.944630	673.0	248.0	205.604863	120.0	95.698485	
3	4	433.0	529.0	23.480049	445.0	248.0	217.515012	120.0	77.455844	
4	5	472.0	551.0	24.514670	486.0	248.0	213.033898	128.0	83.798990	
...	...	...	...	...	...	...	...	...	...	
57	58	213.0	285.0	16.468152	221.0	224.0	184.525822	120.0	52.284271	
58	59	79.0	108.0	10.029253	84.0	248.0	184.810127	128.0	39.313708	
59	60	88.0	110.0	10.585135	92.0	216.0	182.727273	128.0	45.692388	
60	61	52.0	75.0	8.136858	56.0	248.0	189.538462	128.0	30.692388	
61	62	48.0	68.0	7.817640	53.0	224.0	173.833333	128.0	33.071068	

62 rows × 86 columns

Exercises

Make a table with only `area`, `mean_intensity`, `standard_deviation_intensity` and `label`.

[ ]:

How many object are in the dataframe?

[ ]:

How large is the largest object?

[ ]:

What is the mean intensity of the brightest object?

[ ]:

What are mean and standard deviation intensity of the image?

[ ]: