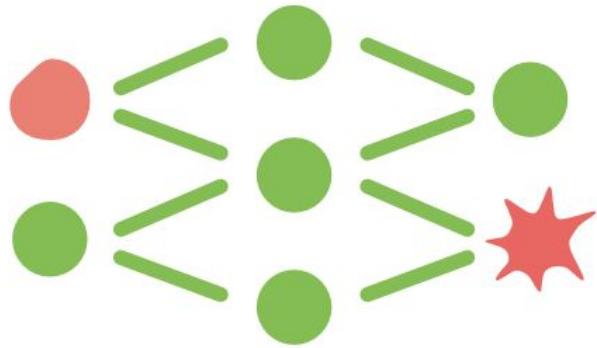


# Intro to Biolimage Analysis and Deep Learning Utilization



Martin Schätz, PhD  
Biolimage Analyst  
Viničná Microscopy Core Facility, Charles University



CHARLES  
UNIVERSITY



SORBONNE  
UNIVERSITÉ



UNIVERSITÄT  
HEIDELBERG  
ZUKUNFT  
SEIT 1386



UNIVERSITY  
OF WARSAW



UNIVERSITÀ  
DEGLI STUDI  
DI MILANO



EUROPEAN  
UNIVERSITY  
ALLIANCE

# About me

- Charles University
  - BiolImage Analyst, under [Core Facility](#)
  - [GloBIAS](#), [CzechBIAS](#)
- National Library of Technology
  - Data Stewardship & PIDs specialist
- University of Chemistry and Technology
  - Assistant Professor: Advanced signal and image processing
  - Institutional Data Steward

more at: [www.schaetz.cz](http://www.schaetz.cz) or

<https://www.schaetz.cz/bia-overview/intro.html>



# Overview

(Super Quick) Introduction to BioImage Analysis

AI4LIFE

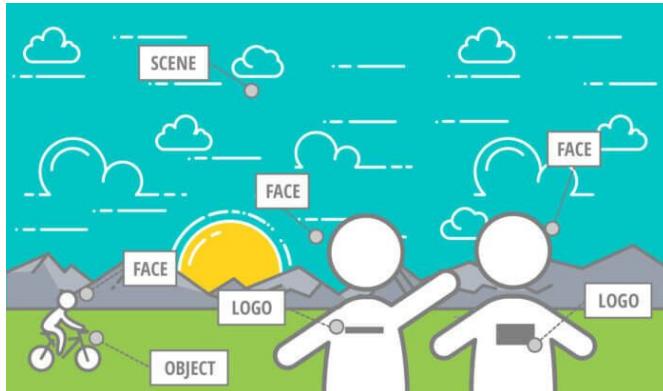
Noise2Void

StarDist

# Term Definition

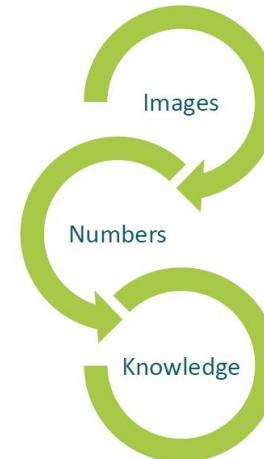
## Image Analysis?

Image analysis (also known as “computer vision” or image recognition) is the ability of computers to **recognize attributes** within an image.



## BioImage Analysis?

Understanding and quantifying microscopy, medical or any other calibrated image data.

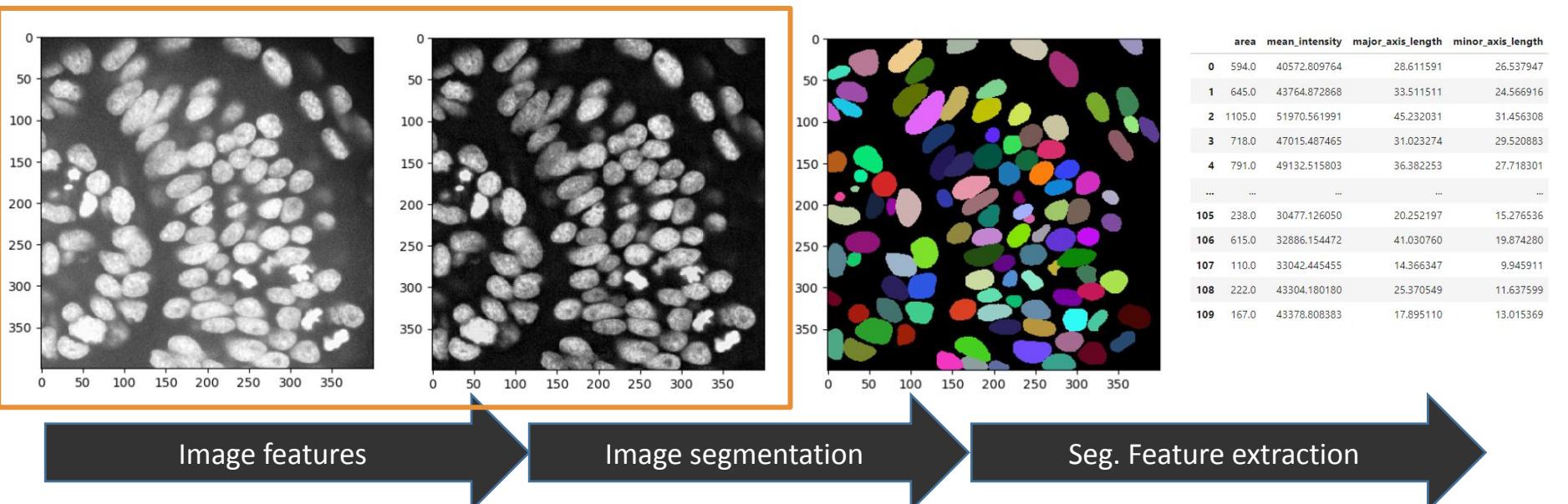


- Objective
- Reliable
- Reproducible
- Replicable
- Repeatable

# Bio-image Analysis

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

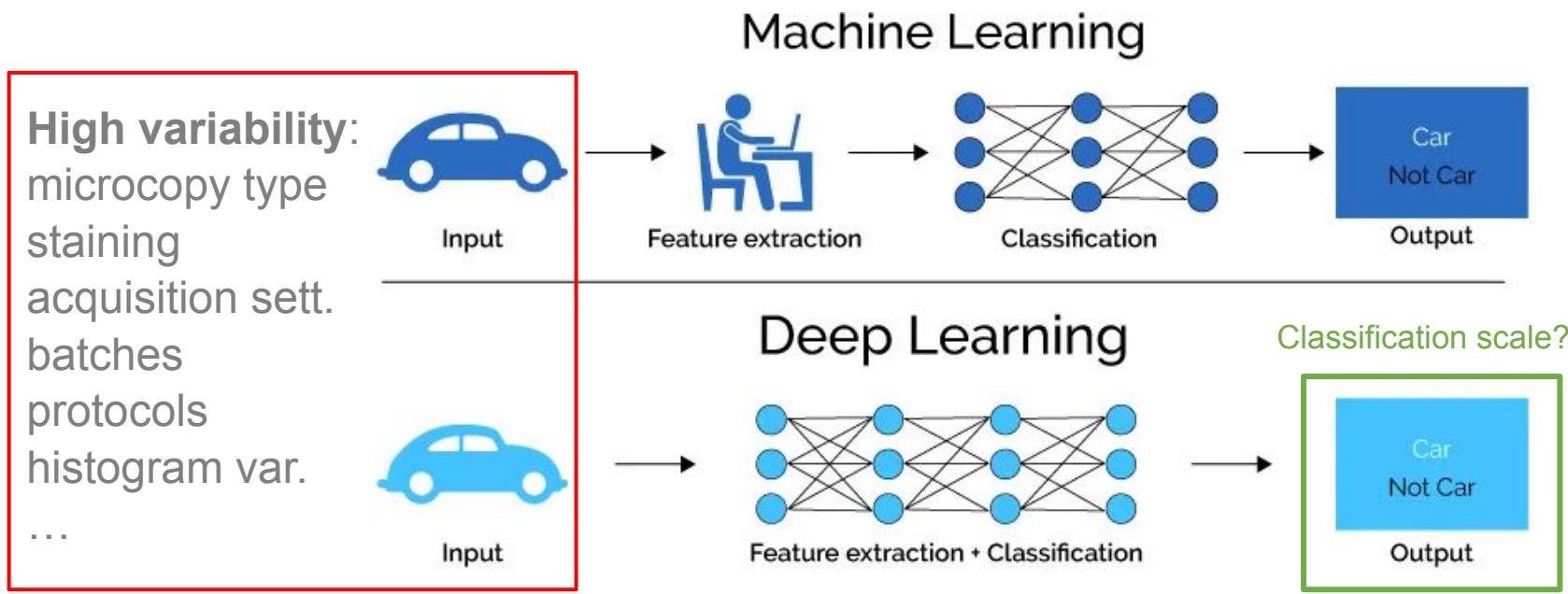
## DL Segmentation models



# Deep Learning & BioImage Analysis



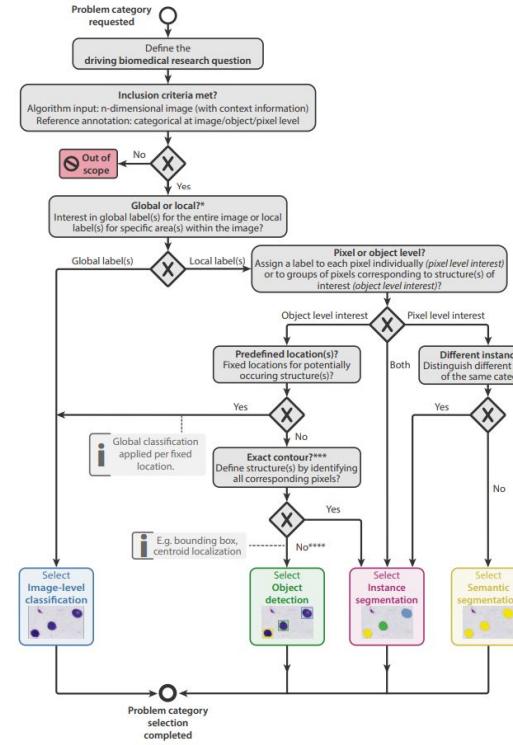
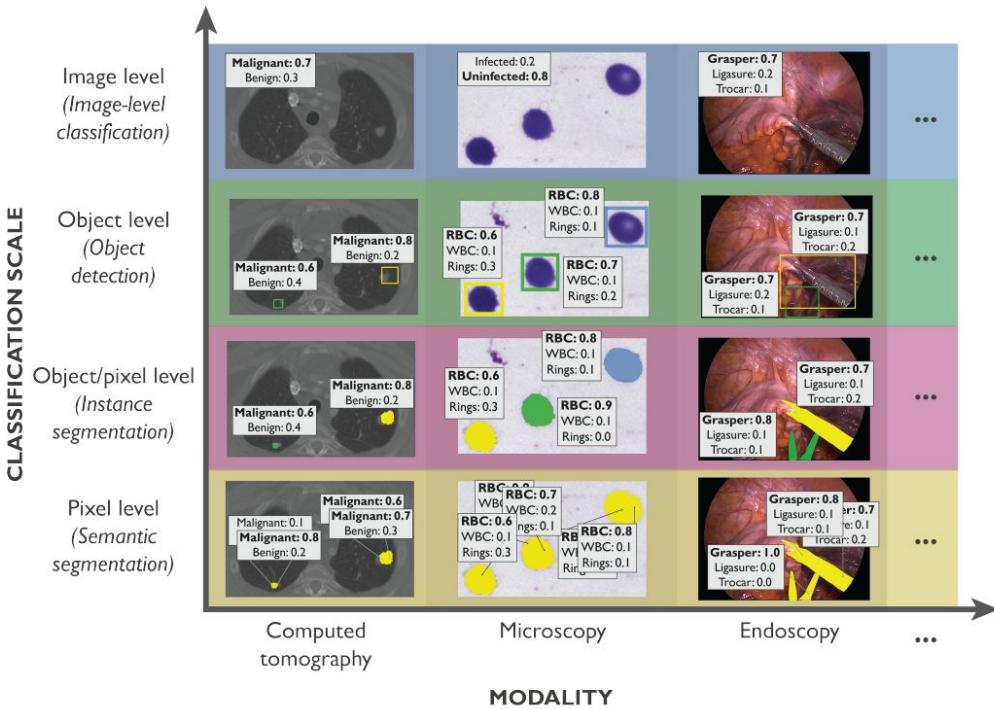
Training models from scratch is super expensive. We need hundreds to thousands of high quality annotated images/instances. So transfer learning, fine tuning or task specific models are used. Changing one parameter in your experiment might mean retraining part of your model!



StarDist, N2V, Unet, Cellpose, ...

# How to decide on metrics for performance?

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



# Bio-image Analysis

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

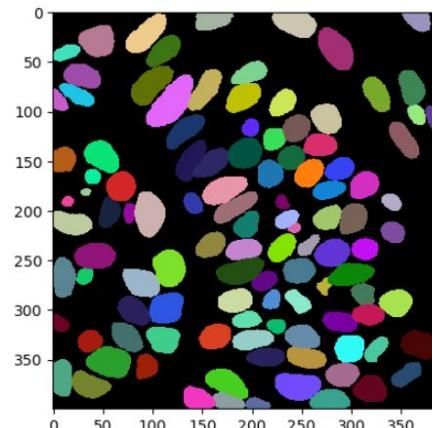
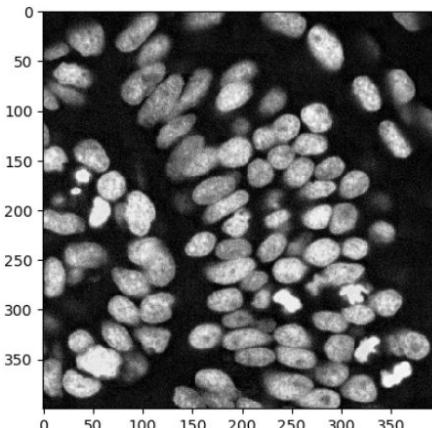
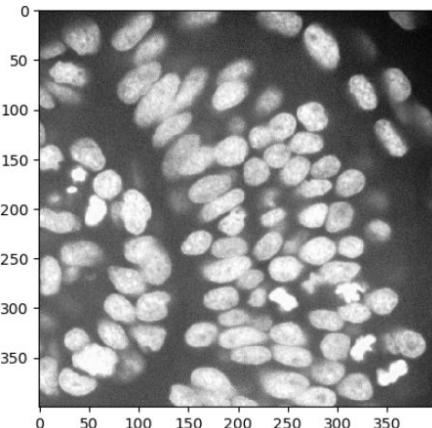


Image filtering

Image segmentation

Feature extraction

	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	29.520883
4	791.0	49132.515803	36.382253	27.718301
...	...	...	...	...
105	238.0	30477.126050	20.252197	15.276536
106	615.0	32886.154472	41.030760	19.874280
107	110.0	33042.445455	14.366347	9.945911
108	222.0	43304.180180	25.370549	11.637599
109	167.0	43378.808383	17.895110	13.015369

# Quantitative bio-image analysis



Deriving quantitative information from images of biological samples taken with microscopes

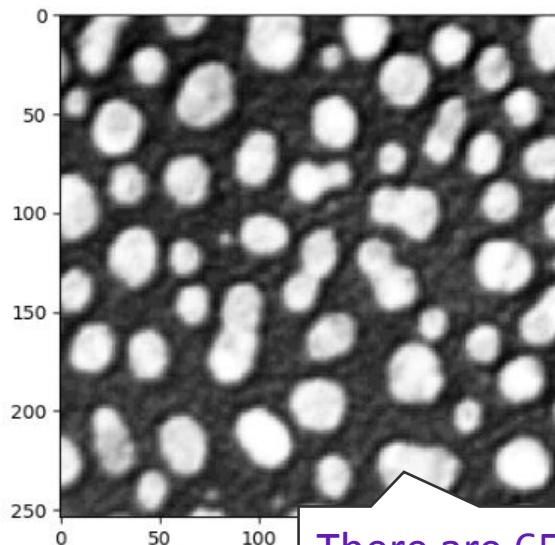


cat height = 1.5 x microscope height

# Reliable bio-image analysis

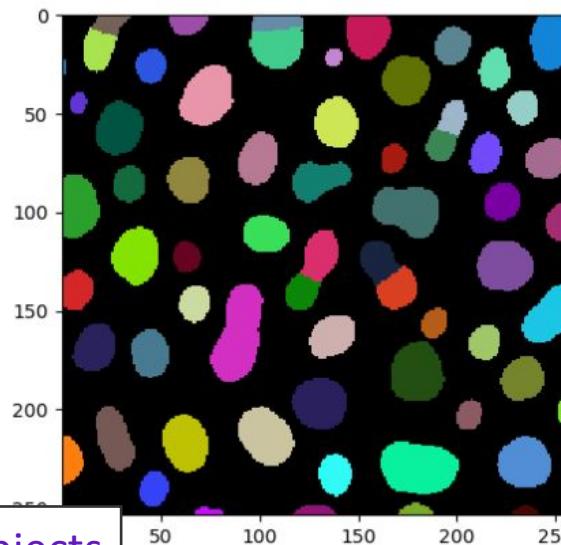
- Algorithms must be reliable (trustworthy).
- Visualization helps gaining trust in automated methods.

Original image

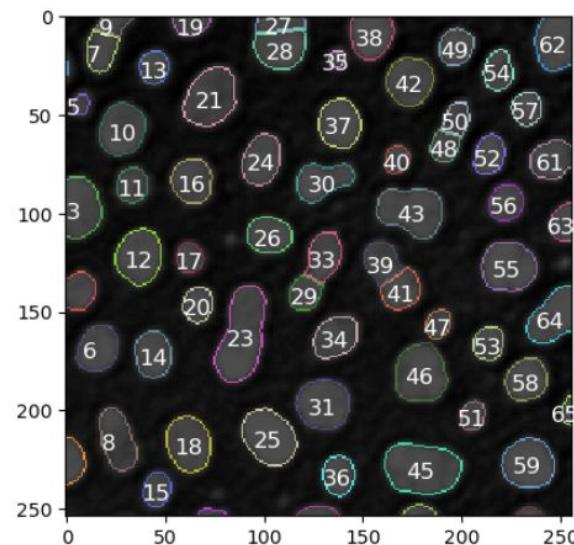


There are 65 objects  
in this image.

Label image

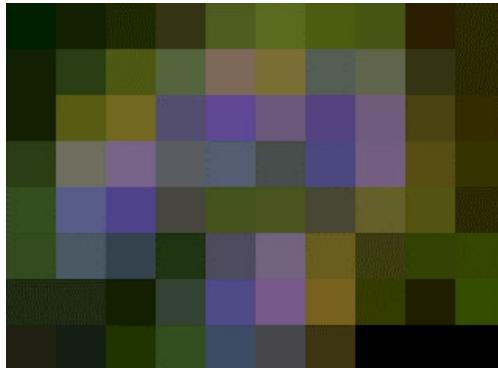


Overlay

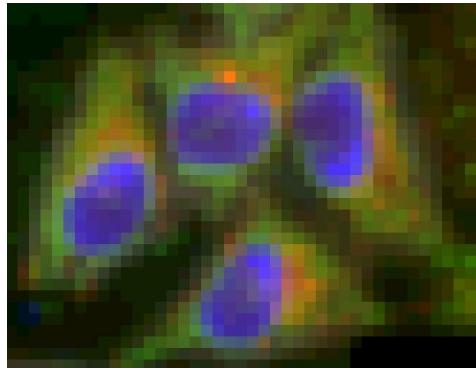


# Pixel size versus resolution

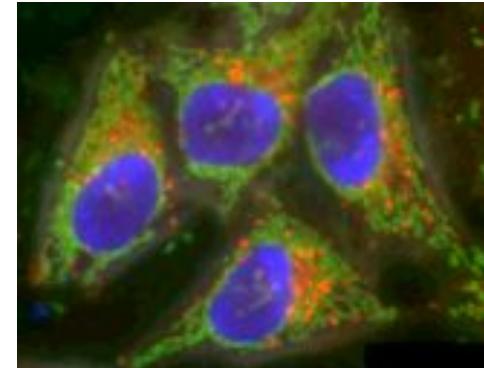
- Pixel size is a digital property of an image.
- You configure it during the imaging session at the microscope/imaging device.



Pixel size: 3.3  $\mu\text{m}$



Pixel size: 0.8  $\mu\text{m}$

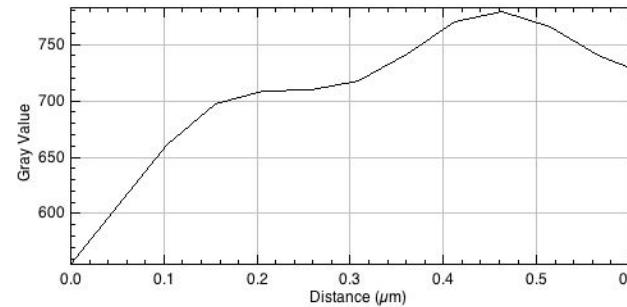
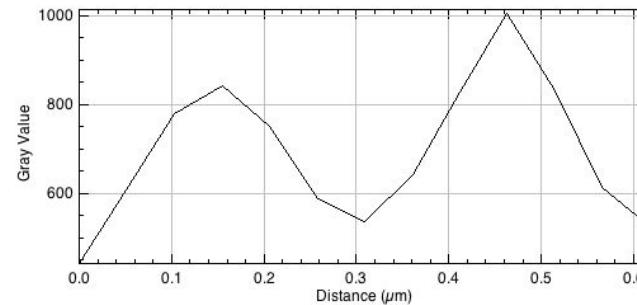
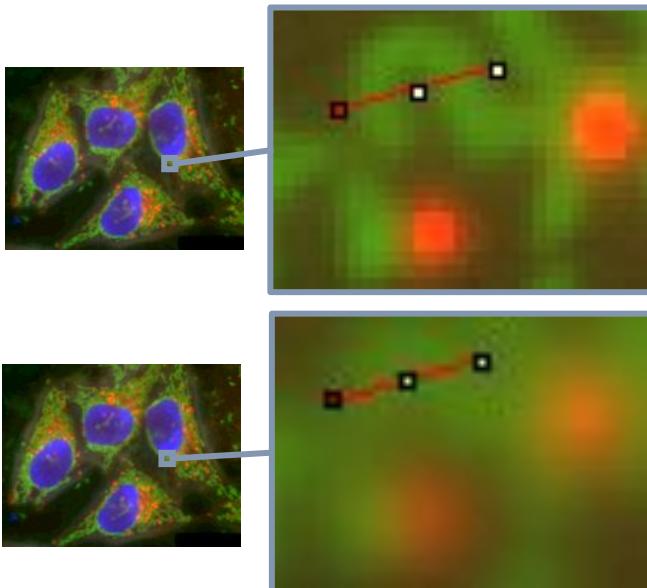


Pixel size: 0.05  $\mu\text{m}$

- We are not talking about resolution!

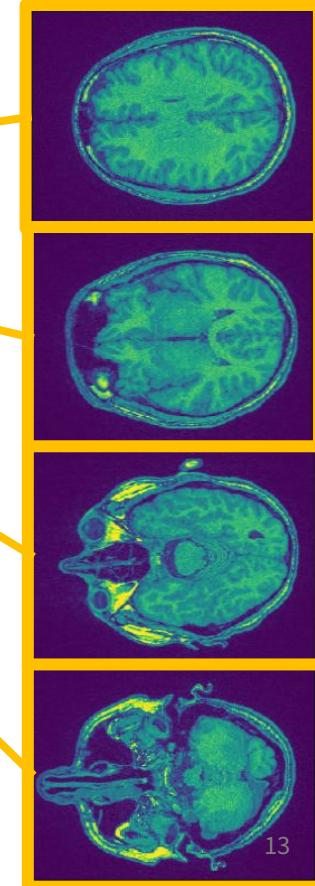
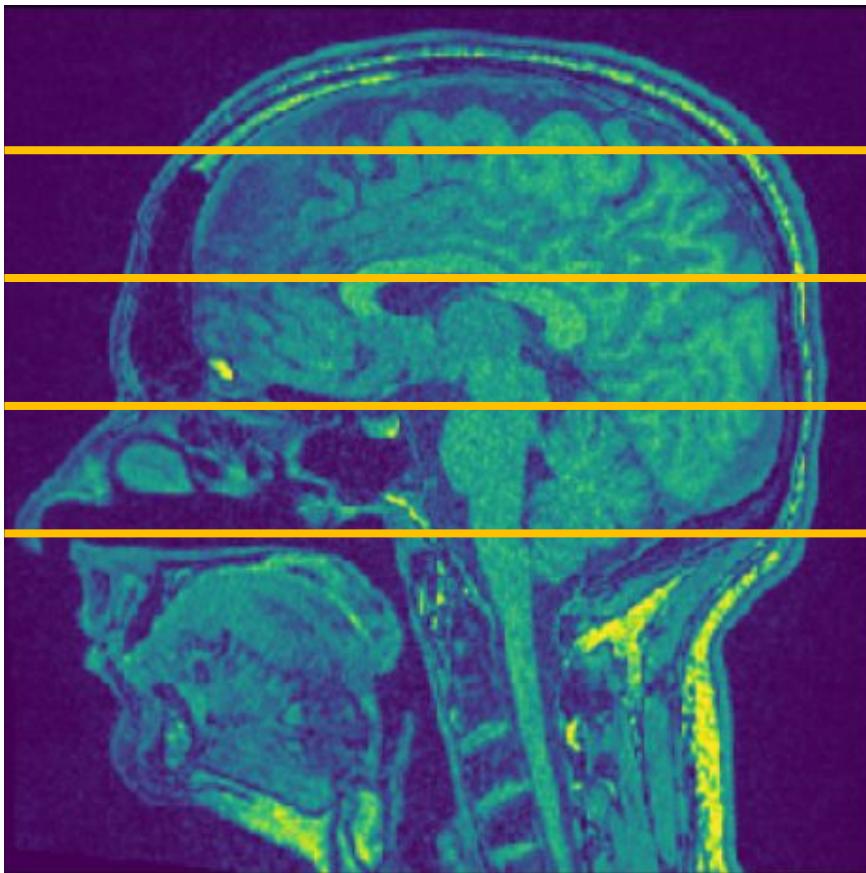
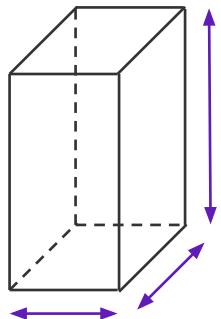
# Pixel size versus resolution

- Resolution is a property of your imaging system.
- The measure of how close object can be in an image while still being differentiable, is called spatial resolution.



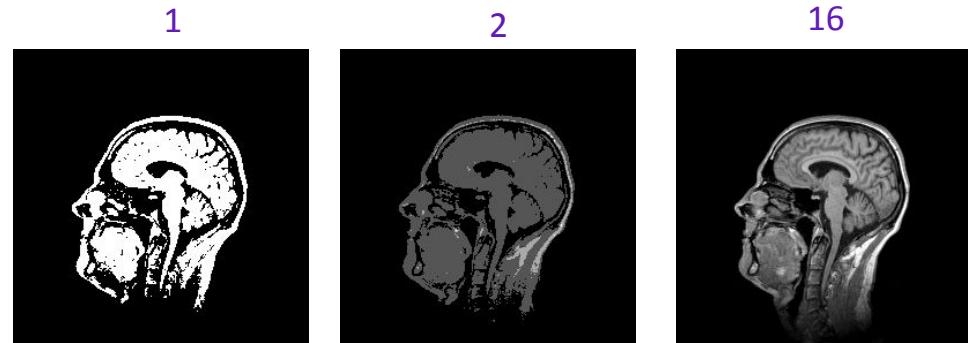
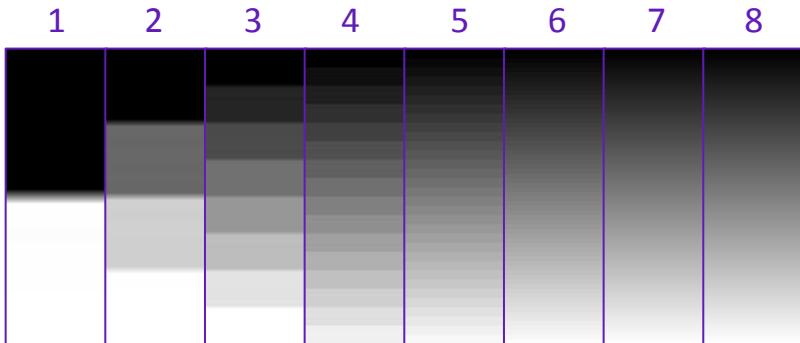
# Image stacks and voxels

- 3-dimensional images consisting of voxels
- “Image stack”
- Often anisotropic (not equally large in all directions)



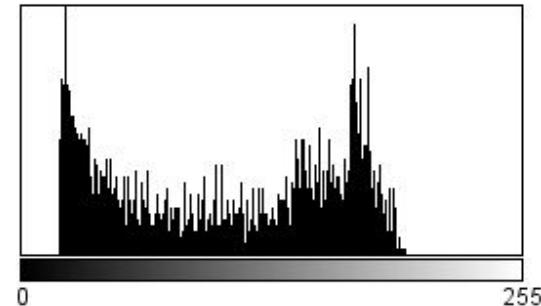
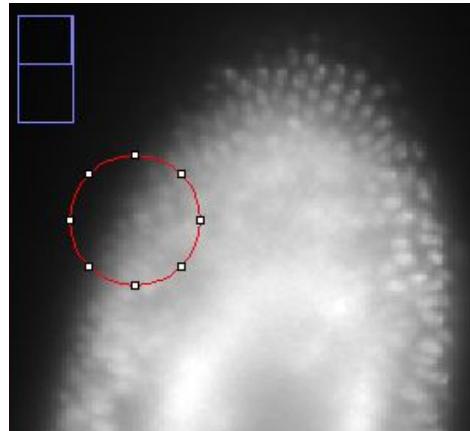
# Bit-depth

- A bits is the smallest memory unit in computers, *atomic data*.
- The bit-depth  $n$  enumerates how many different intensity values are present in an image:
  - $2^n$  grey values
- In microscopy, images are usually stored as **8, 12 or 16-bit** images.
- Lowering bit depth means significant information loss!

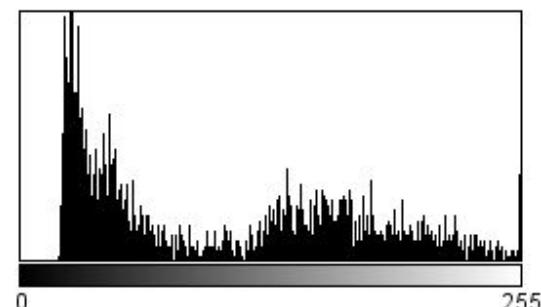
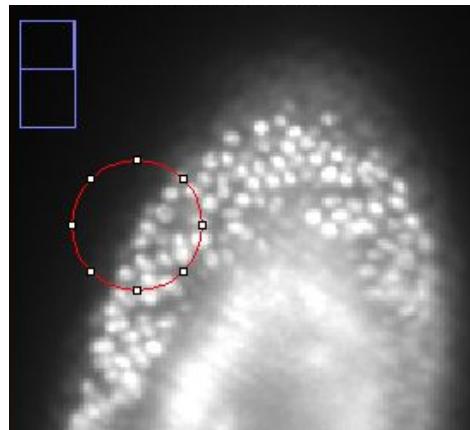


# Histograms

- Histograms are summaries of images
- Tell stories, e.g. about image quality



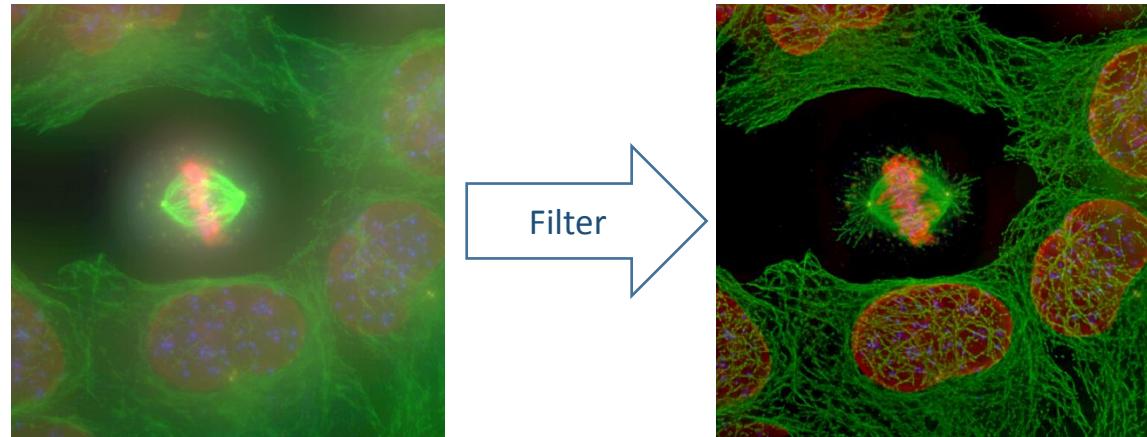
Count: 2053      Min: 19  
Mean: 103.818      Max: 196  
StdDev: 57.093      Mode: 22 (41)



Count: 2053      Min: 19  
Mean: 103.370      Max: 255  
StdDev: 70.260      Mode: 25 (49)

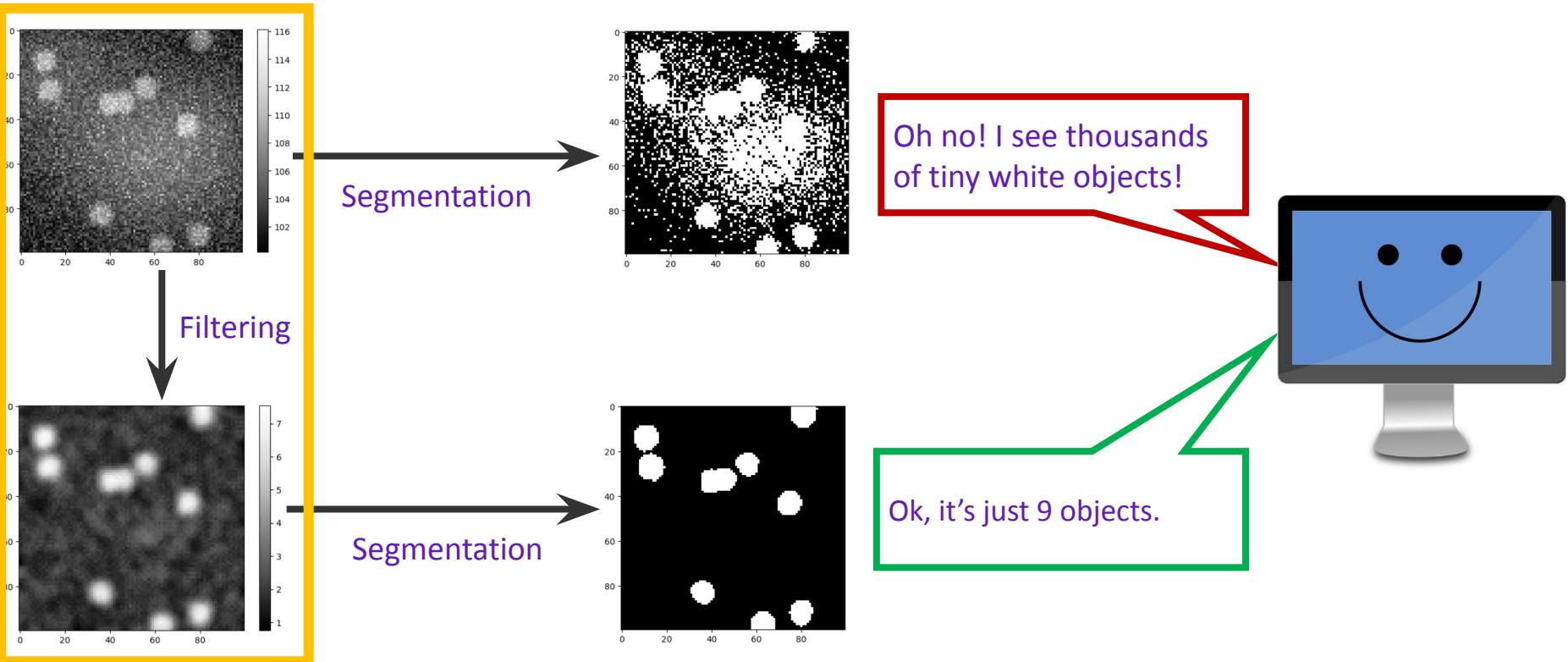
# Filters

- An image processing filter is an operation on an image.
- It takes an image and produces a new image out of it.
- Filters change pixel values.
- There is no “best” filter. Which filter fits your needs, depends on the context.
- Filters do not do magic. They can not make things visible which are not in the image.
- Application examples
  - Noise-reduction
  - Artefact-removal
  - Contrast enhancement
  - Correct uneven illumination



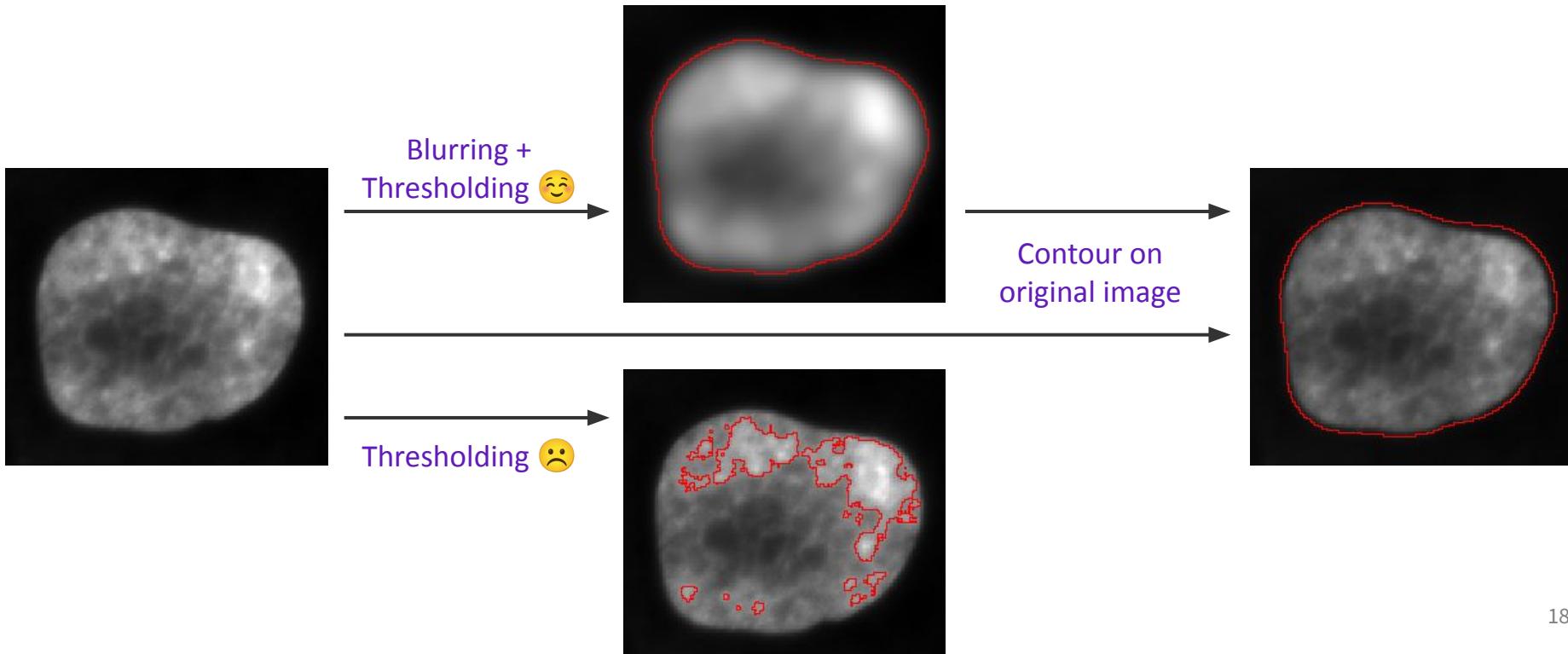
# Image filtering

- We need to remove the noise to help the computer *interpreting* the image



# Filtering for improving thresholding results

- In case thresholding algorithms outline the wrong structure, blurring in advance may help.
- However: **Do not** continue processing the blurred image, continue with the original!

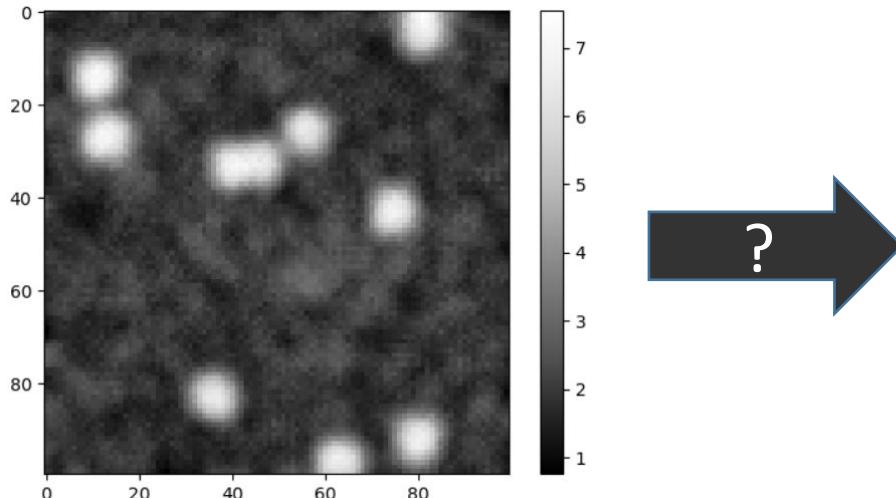


# Segmentation / binarization

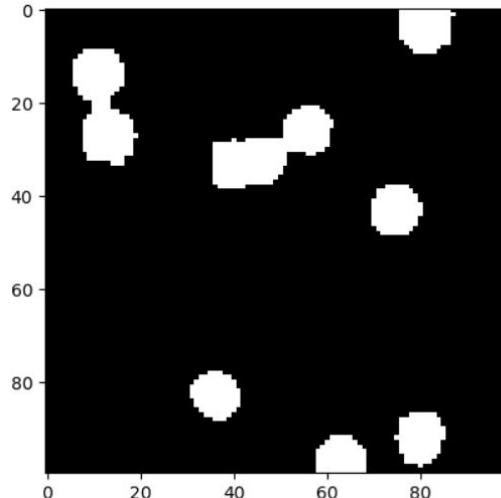
## Thresholding

- Very basic and yet efficient segmentation technique
- Histogram based, to determine an intensity threshold
- Not state-of-the-art in many fields (anymore)

Intensity image

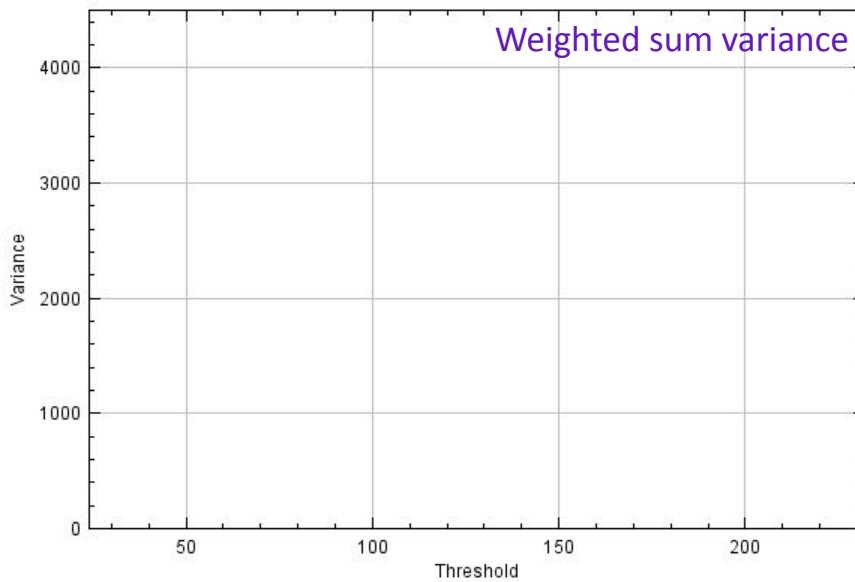
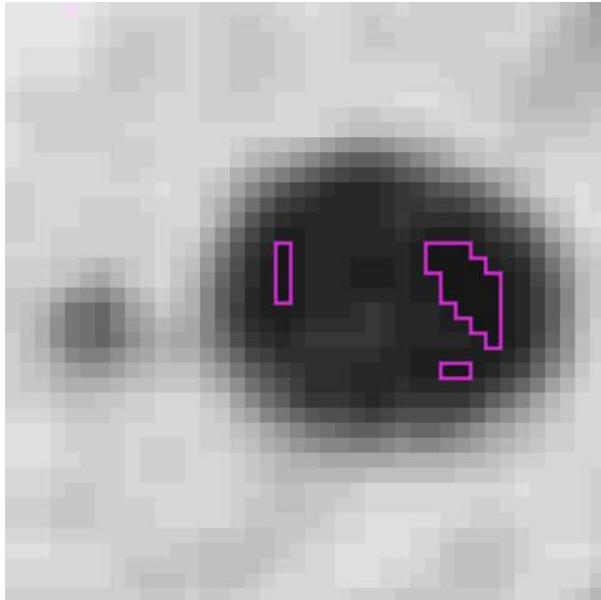


Binary image



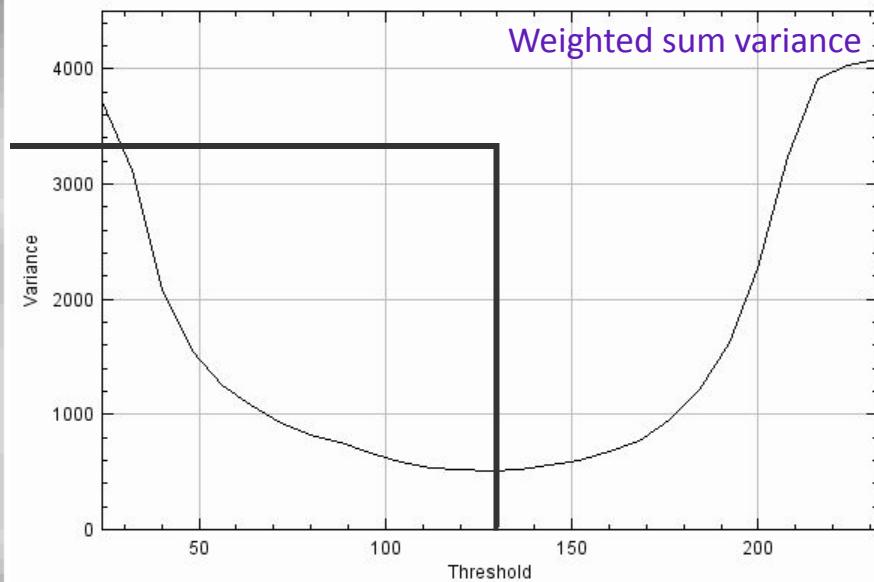
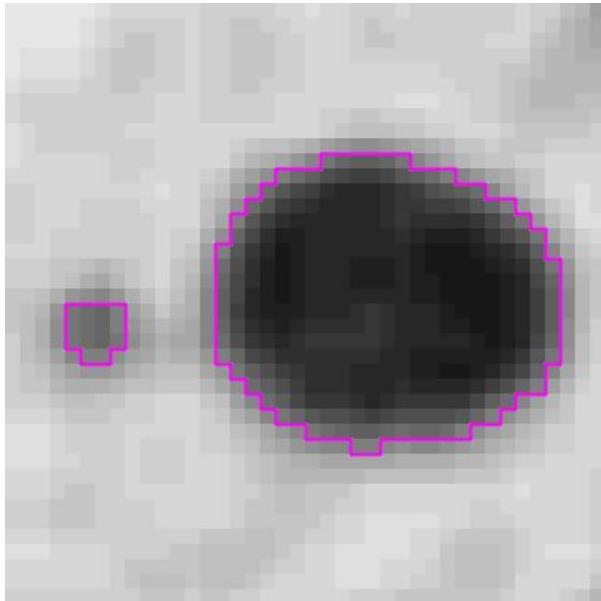
# Thresholding: Otsu's method

- Searching for a threshold where the variance in both classes (above/below threshold) becomes minimal.
- Weighted (!) sum variance



# Thresholding: Otsu's method

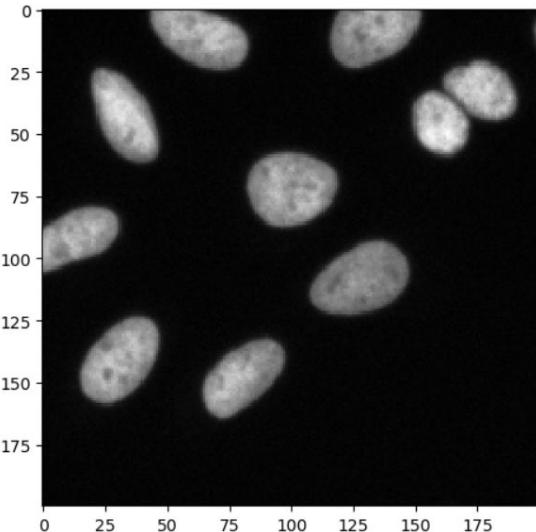
- Searching for a threshold where the variance in both classes (above/below threshold) becomes minimal.
- Weighted (!) sum variance



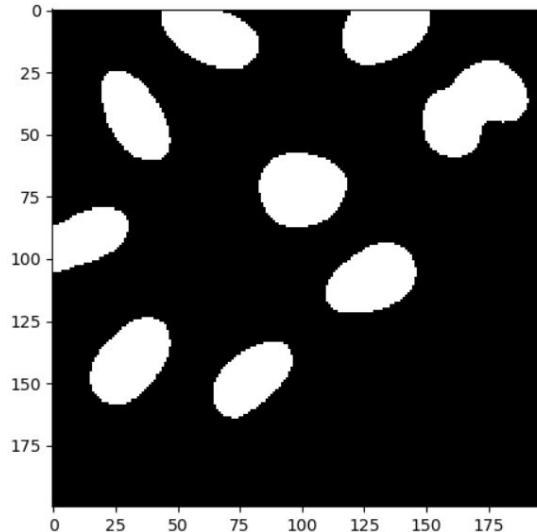
See also: <http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html>

# Terminology

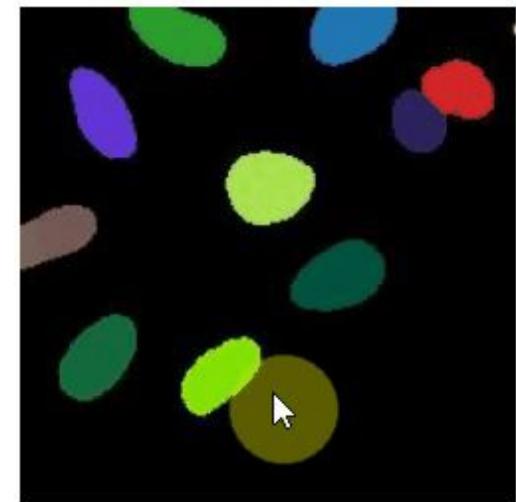
Intensity image



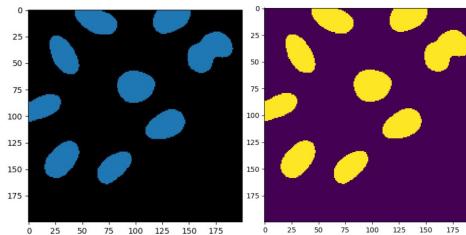
Binary image



Label image



No matter how they are displayed



[y=152, x=92] = 0

# Terminology

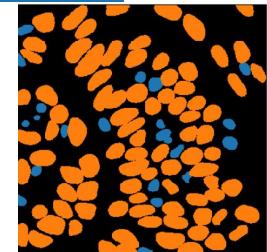
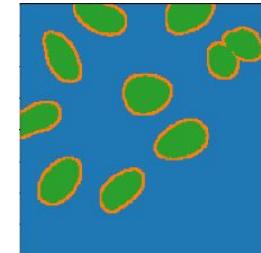
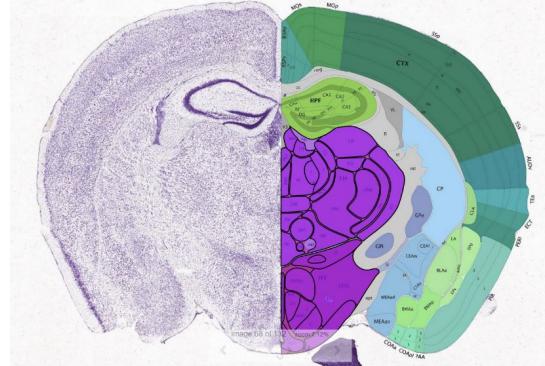
## Instance



### Instances:

- Cells, nuclei, cats, dogs, cars, trees

## Semantic segmentation

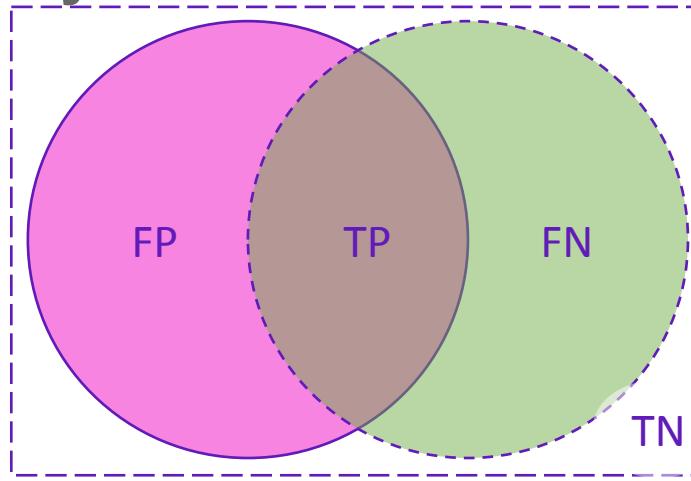


### Regions:

- Anatomical, geographical
- All pixels belonging to the same type of object have the same value

# 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



Overlap  
(a.k.a. Jaccard index)

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

How much do A and B overlap?

Precision

$$\frac{TP}{TP + FP}$$

What fraction of points that were predicted as positives were really positive?

Recall  
(a.k.a.  
sensitivity)

$$\frac{TP}{TP + FN}$$

What fraction of positives points were predicted as positives?

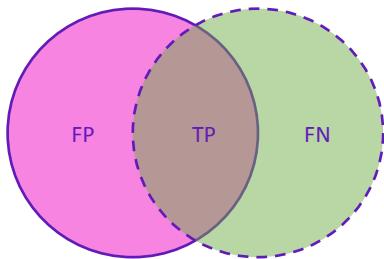
A	Prediction A
B	Reference B (ground truth)
ROI	Region of interest
TP	True-positive
FN	False-negative
FP	False-positive
TN	True-negative

# Pixel-wise versus Object-wise evaluation

- Object wise: Detection quality

- Pixel wise: Segmentation quality

Prediction	Ground truth
------------	--------------

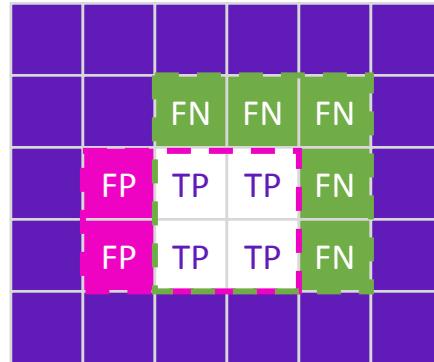


Precision

$$\frac{TP}{TP + FP}$$

Recall  
(a.k.a.  
sensitivity)

$$\frac{TP}{TP + FN}$$

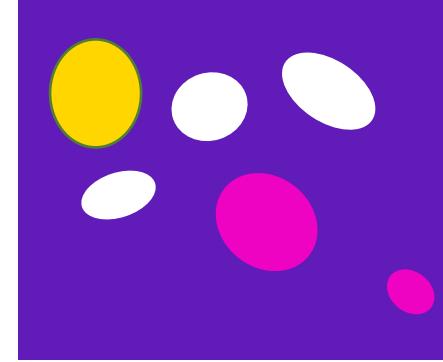


True-positive: 4

False-negative: 5

False-positive: 2

Precision:  $4/6 = 66\%$   
Recall:  $4/9 = 44\%$



True-positive: 3

False-negative: 1

False-positive: 2

Precision:  $3/4 = 75\%$   
Recall:  $3/5 = 60\%$

- 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

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

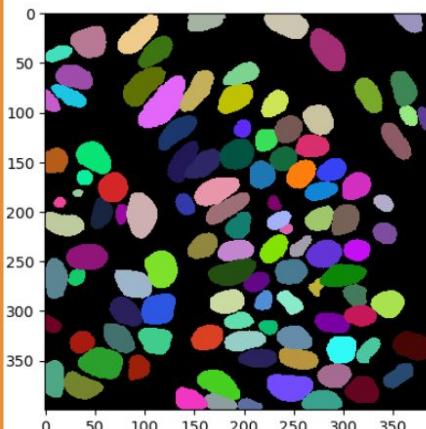
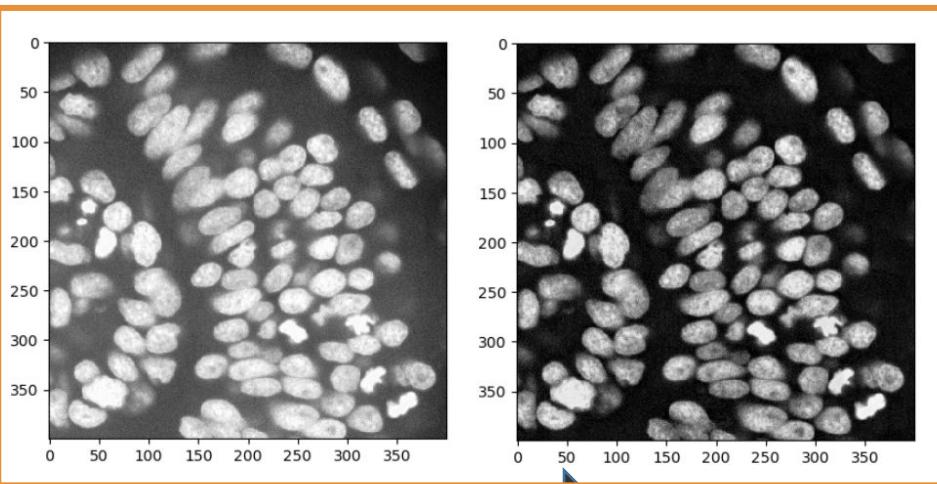
- Hamming distance

$$\begin{aligned} d_h &= |A \cup B| - |A \cap B| \\ &= |FP| + |FN| \end{aligned}$$

# Bio-image Analysis

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

StarDist, N2V, Unet, Cellpose, ...



	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	29.520883
4	791.0	49132.515803	36.382253	27.718301
...	...	...	...	...
105	238.0	30477.126050	20.252197	15.276536
106	615.0	32886.154472	41.030760	19.874280
107	110.0	33042.445455	14.366347	9.945911
108	222.0	43304.180180	25.370549	11.637599
109	167.0	43378.808383	17.895110	13.015369

Image filtering

Image segmentation

Feature extraction

# Acknowledgements



## BiAPoL team

- Robert Haase
  - Mara Lampert
  - Marcelo Zoccoler
  - Johannes Soltwedel
  - Maleeha Hassan
  - Allyson Ryan
  - Till Korten
  - Stefan Hahmann
  - Somashekhar Kulkarni
- Former lab members:
- Ryan George Savill
  - Laura Zigutyte

## Networks



CENTER FOR  
SYSTEMS BIOLOGY  
DRESDEN



NFDIA  
BIOIMAGE



## Funding



Chan  
Zuckerberg  
Initiative



<https://physics-of-life.tu-dresden.de/bia>

# AI4LIFE & Biolmage Model Zoo



AI4Life



---

Advanced AI models in one click



The BioImage Model Zoo and FAIR data principles are core facets of the AI4Life project.

#### OBJECTIVES

and our goals

**1**

**Democratized** availability of  
AI-based image analysis methods

**3**

**Simple model deployment, sharing, and  
dissemination** through a new developer-  
facing service

**5**

Empower **common image analysis**  
platforms with **AI integration**

**2**

**Establish standards** for the submission,  
storage and FAIR access

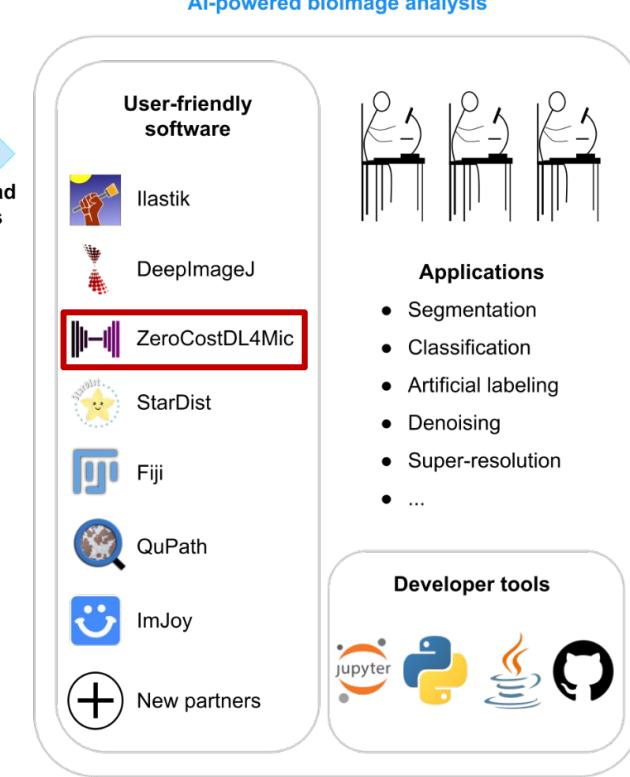
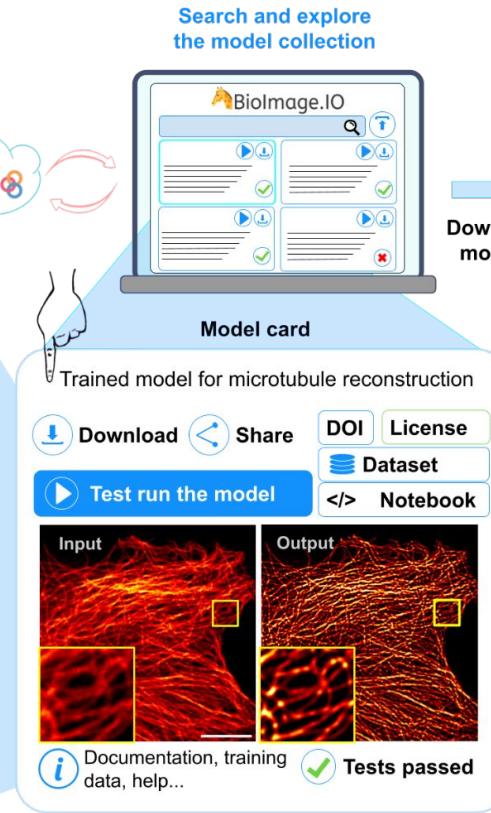
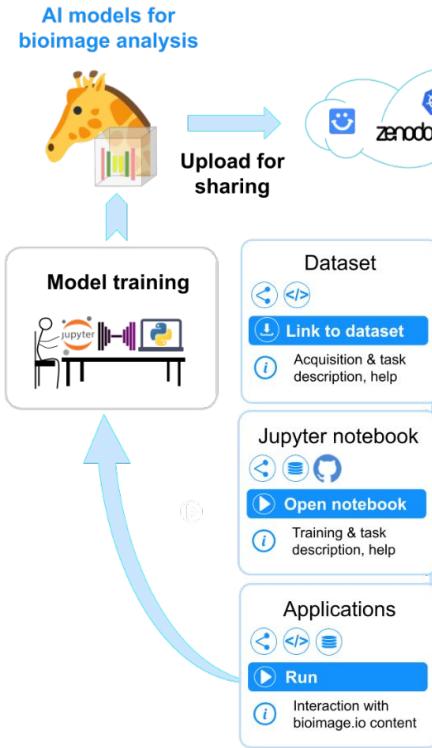
**4**

Organize **Open Calls and Challenges** for  
image analysis problems

**6**

**Organizing outreach and training** events  
i.e. image analysis courses/workshops and  
participation in international conferences

Advanced AI models in one click



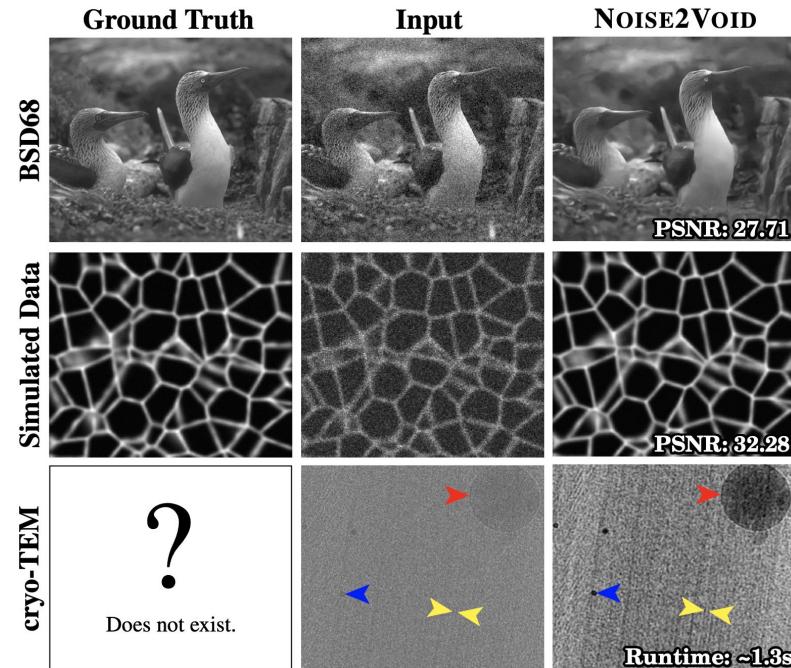
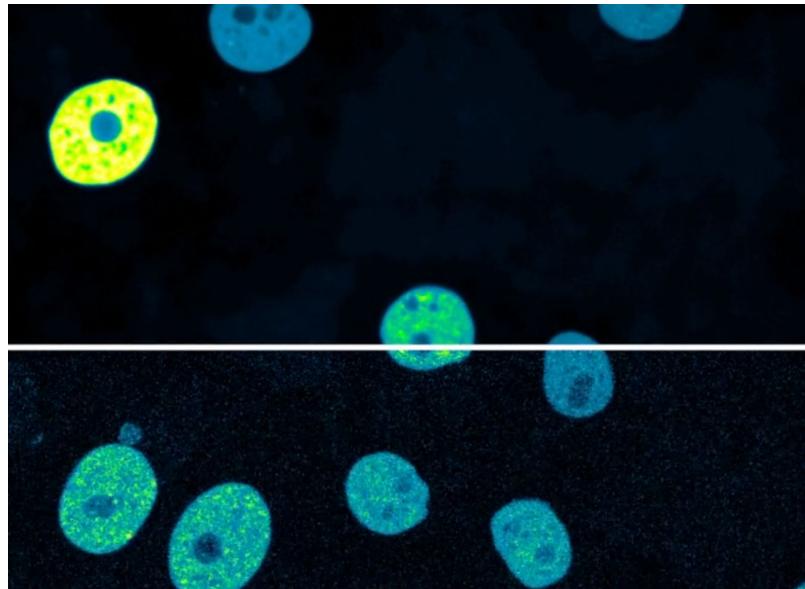
# Smart image denoising with noise2void

 Alexander Krull, Tim-Oliver Buchholz, Florian Jug.  
[Noise2Void - Learning Denoising from Single Noisy Images.](https://arxiv.org/abs/1811.10980)  
Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019.

 Alexander Krull, Tim-Oliver Buchholz, Florian Jug.  
<https://arxiv.org/abs/1811.10980>.  
GitHub repository.

# Noise2Void (N2V)

Machine learning model for denoising. Designed based on the search for statistical dependencies in the image.

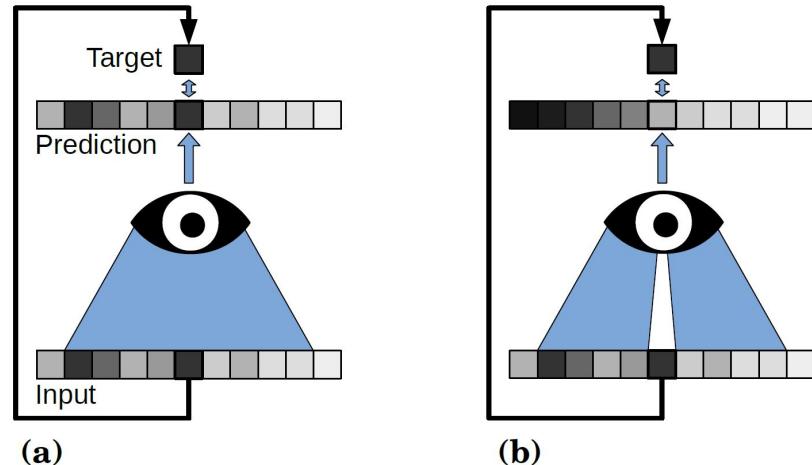


# Motivation

- Smart and universal denoising of various image types for visualization.
- Ready-to-use tool in ImageJ, open source code on GitHub
- The original article also described the limitations of the method.
- Availability of test data from the article.
- Deep Learning model (black box)
  - with very well described dependencies
  - focused on re-usability
- Possibility to export BioImage Model Zoo package for publication.

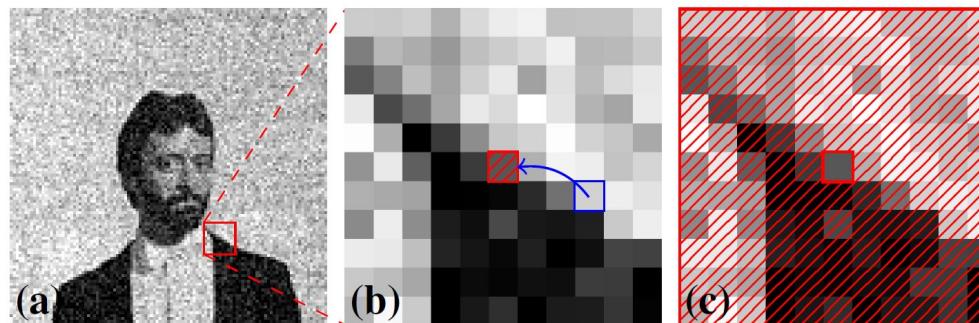
# Implementation of Noise2Void

- Assumption: noise changes but image information remains.
  - The signal is statistically dependent on itself in the image.
  - Noise is statistically independent of the signal.
- Creating blind spots in the training image **(b)** will help the model recognize dependencies in the image.



# Supervised Learning

A classical learning approach: the input (an image with noise) is compared against a target (ground truth)



N2V innovation: input **(a)** is divided into subsections **(b)**. In training part b, a random pixel is copied to the center. Target **(c)** has no center pixel.

# Using the N2V model

It is necessary to take into account:

- DL model is a "black box"
- It will only be as good as the data we use to train it ("copying the teacher")

The following must be observed:

- Apply to same data as training (bit depth, luminance distribution)
- To share the model, it is good to keep the training data + description.

Practical use:

- Visualization, not the basis for segmentation. Not yet usable for brightness quantification!

# Limitations

## Structured noise

N2V assumes that the noise has no statistical dependence, and therefore does not remove structured noise (checker/mosquito noise, bad frequency filter).

## Seemingly easy to use

DL methods are relatively new, very effective, but there is still no standardized description of their use (mainly for scientific data). It is very easy to misuse them.

## Ethical rules of image analysis

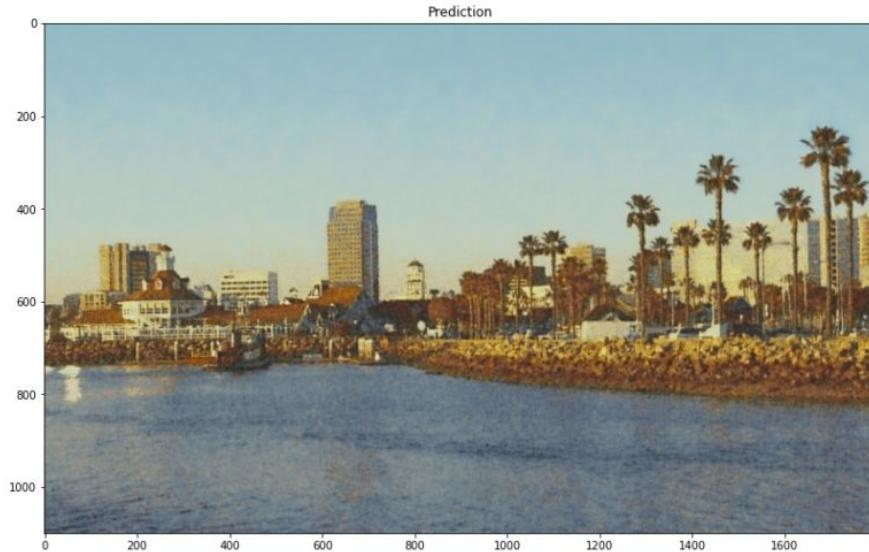
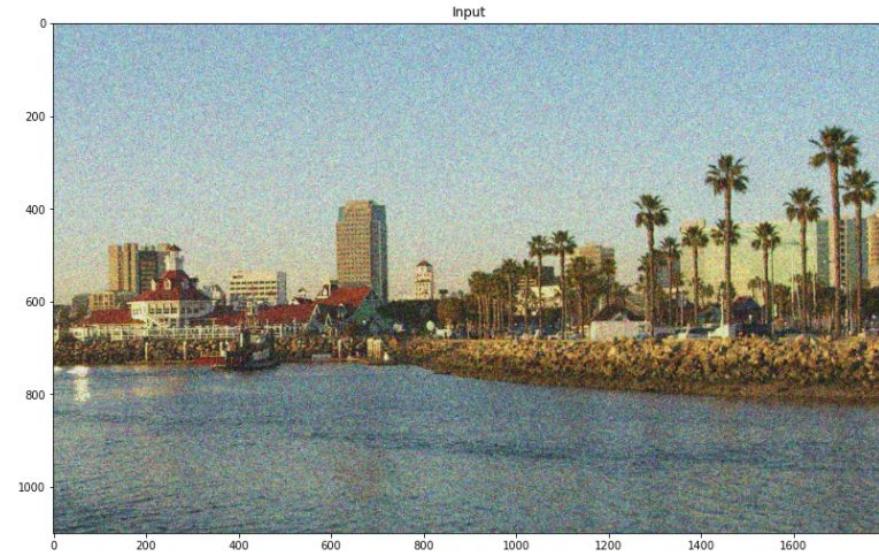
Due to the complexity of Deep Learning, it is necessary to be careful for now.

Cromey, D. W. (2010, June 22). **Avoiding Twisted Pixels: Ethical Guidelines for the Appropriate Use and Manipulation of Scientific Digital Images**. Science and Engineering Ethics. Springer Science and Business Media LLC.

<http://doi.org/10.1007/s11948-010-9201-y>

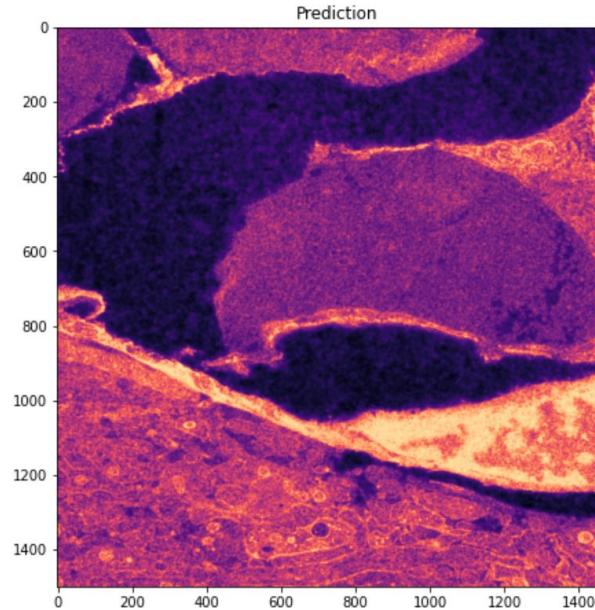
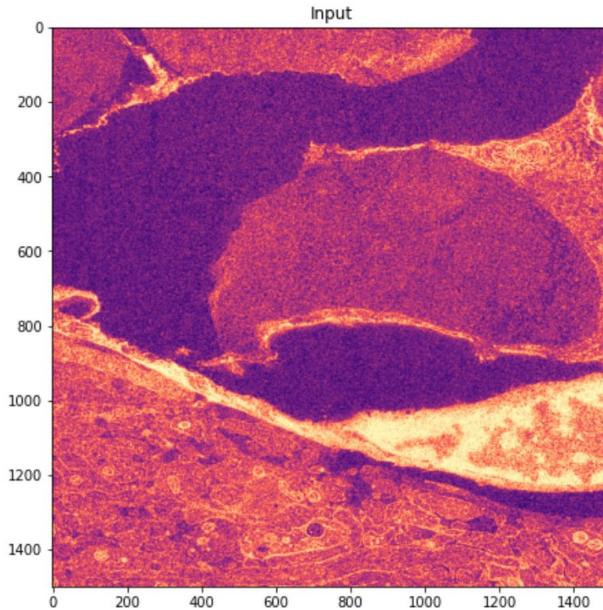
# Python - Jupyter notebook

<https://github.com/juglab/n2v/blob/master/examples/> 2D RGB



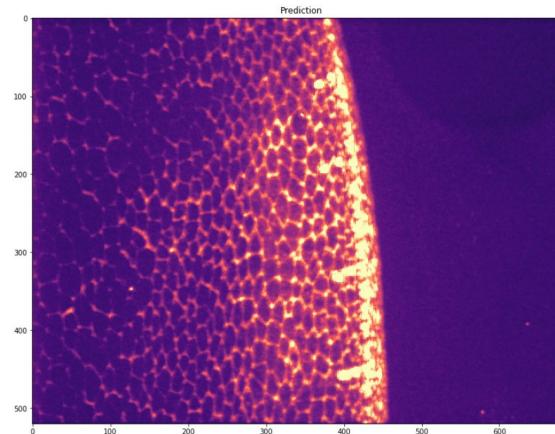
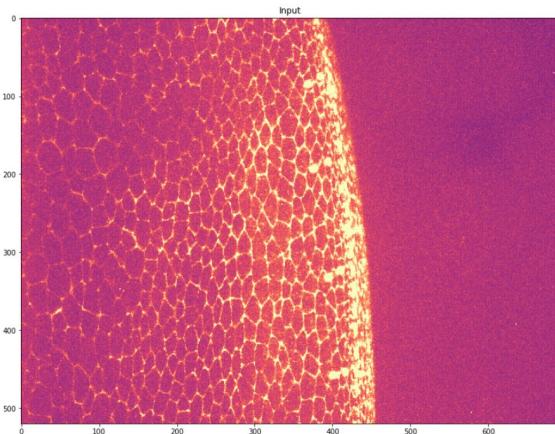
# Python - Jupyter notebook

<https://github.com/juglab/n2v/blob/master/examples/> 2D SEM



# Python - Jupyter notebook

<https://github.com/juglab/n2v/blob/master/examples/> 3D

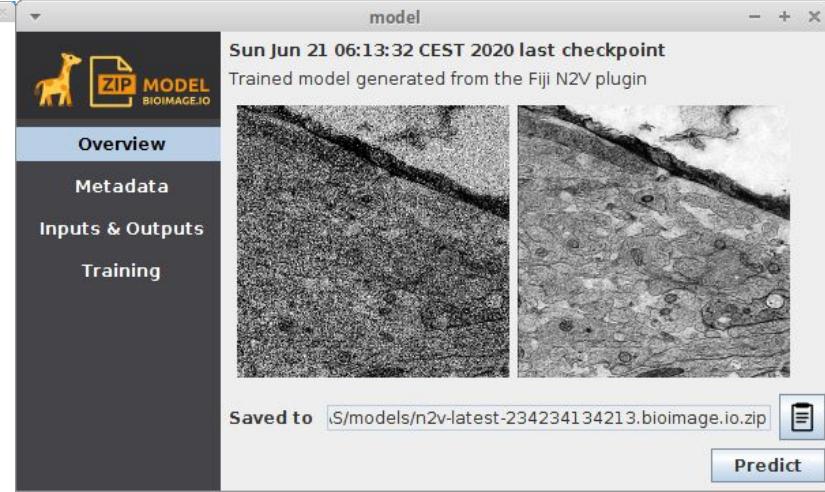
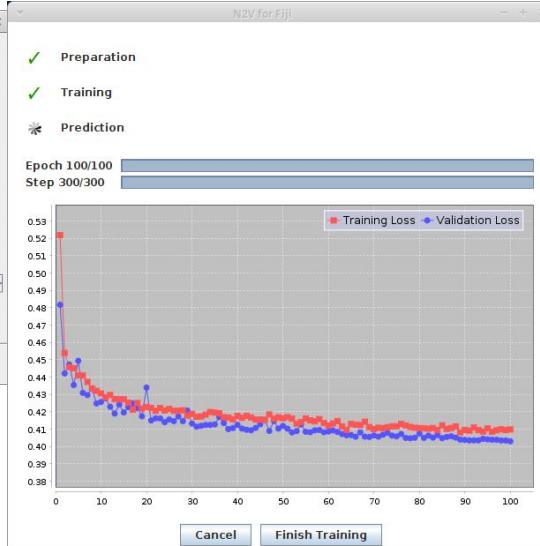
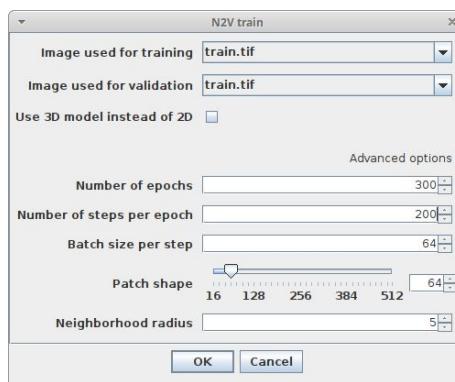


Save results

```
save_tiff_imagej_compatible('prediction.tif', pred, 'ZYX')
```

# ImageJ plugin

N2V Fiji plugin - <https://imagej.net/plugins/n2v>  
 Dependent on CSBDeep



# Summary

Ideal for:

- Visualization
- Preprocessing for segmentation
- Preprocessing for counting objects

Simple use through ImageJ - both training and application. Or use in other Python-enabled applications.

But:

- Black box
- May produce artifacts  
(statistically dependent noise)



# StarDist

## Object Detection with Star-convex Shapes

 Uwe Schmidt, Martin Weigert, Coleman Broaddus, and Gene Myers.

*Cell Detection with Star-convex Polygons.*

International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI),  
Granada, Spain, September 2018.

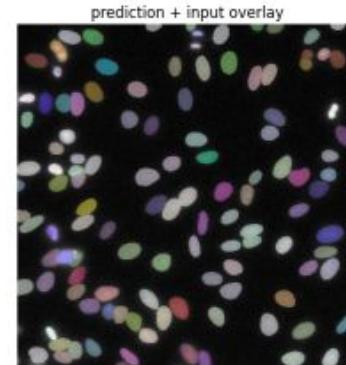
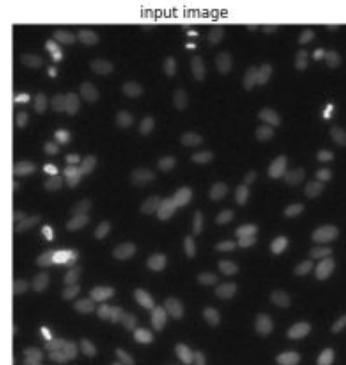
 Martin Weigert, Uwe Schmidt, Robert Haase, Ko Sugawara, and Gene Myers.

*Star-convex Polyhedra for 3D Object Detection and Segmentation in Microscopy.*

The IEEE Winter Conference on Applications of Computer Vision (WACV), Snowmass Village, Colorado, March 2020

# What is StarDist?

- Deep learning tool designed to localize convex objects (cell nuclei).
- Available as:
  - Package for training custom prediction model (Python).
  - Pretrained model ready to use.
  - Plugin(s) using pre-trained models.

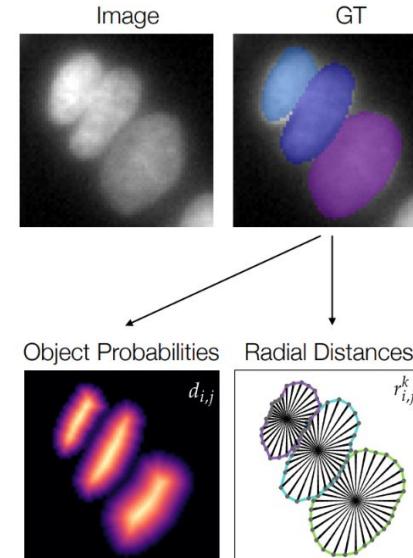
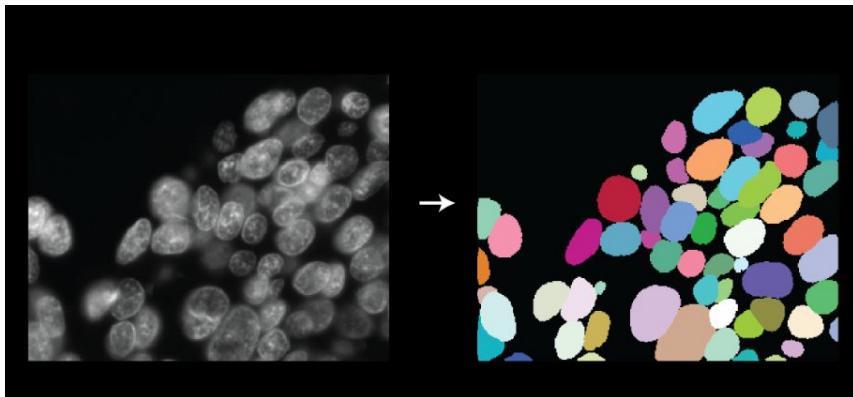


# Motivation & what to learn

- Easy to use.
- Can detect overlapping objects.
- Robust to intensity changes.
- Usable for both 2D and 3D data.
- What is StarDist?
- The main idea
- Why to use StarDist
- Other software plugins

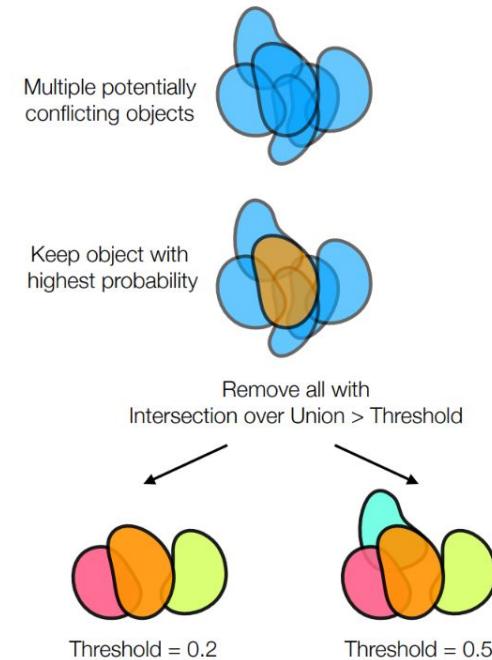
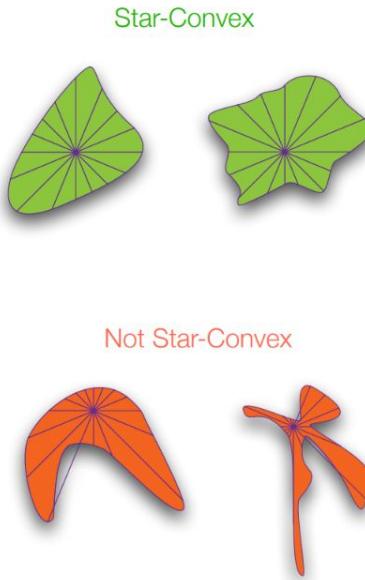
# How it works

- Tool designed to localize cell nuclei via star-convex polygons.
- Similar to methods that directly predict shapes for each object of interest.



# How it works

- Segmentation based on Star-Convex objects.
- Capability to handle intersection/overlapping objects.

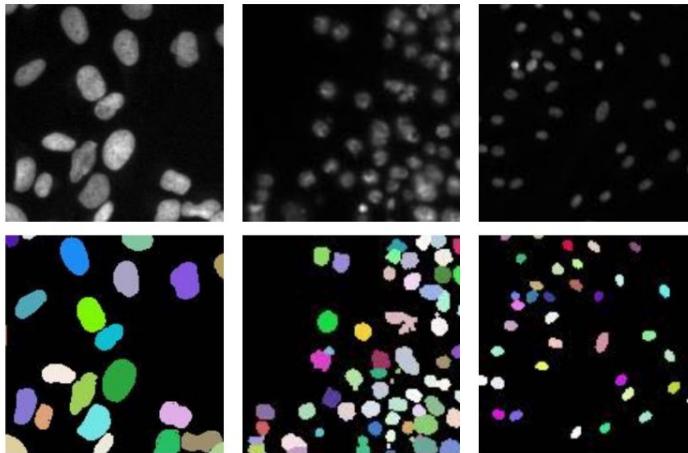


# Why to use StarDist

- Easy to use.
- Can detect overlapping objects.
- Robust to intensity changes.
- Usable for both 2D and 3D data.
- Available as plugin.
- Available models are widely usable.
- Possibility to retrain model for specific data.

# Pretrained models

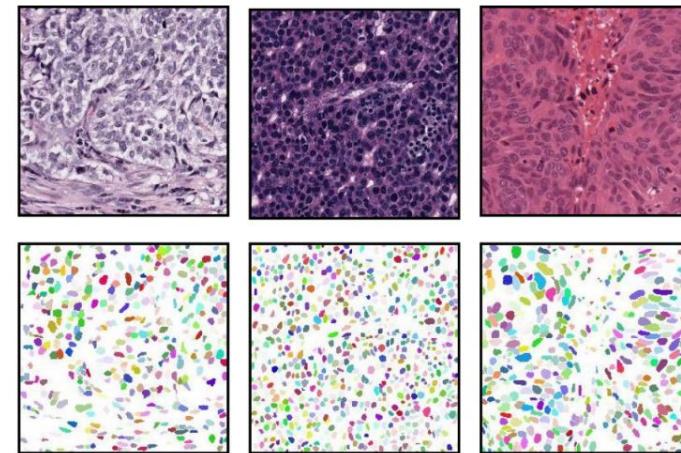
Fluorescence Microscopy  
Single Channel



Data Science Bowl 2018  
Caicedo et al. (2018)

~ 600 images (2D)  
~ 20k annotations

Histopathology  
RGB H&E



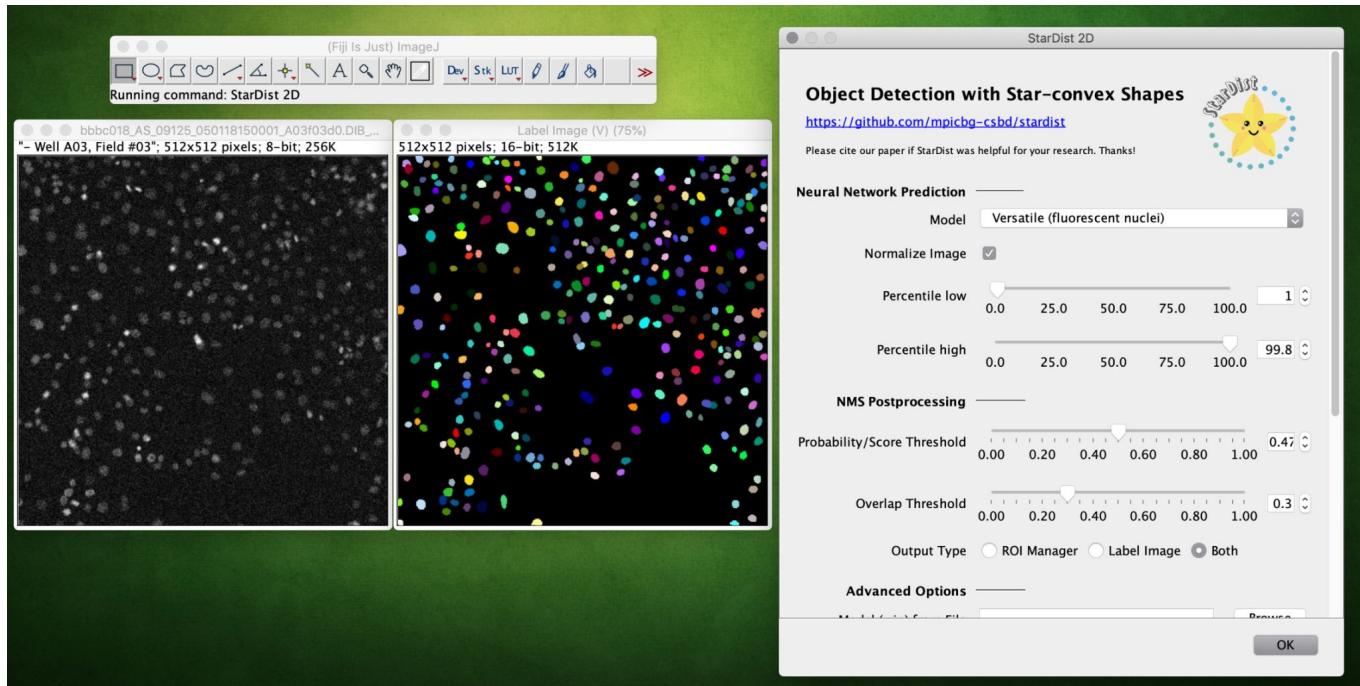
MoNuSeg  
Kumar et al (2017)

~ 30 Images (2D)  
~ 22k annotations

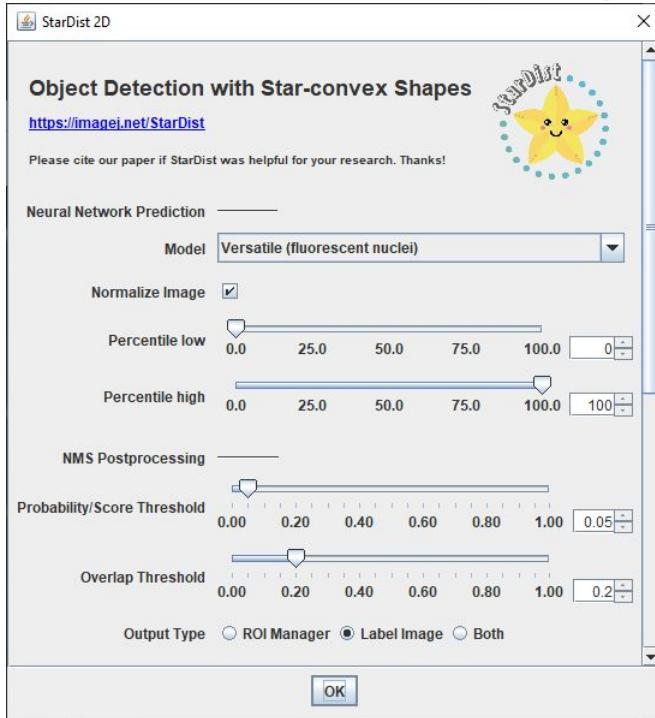
# Examples - Overview

Plugin currently supports only 2D images.

2 models:  
**Versatile**  
**(fluorescent nuclei)**  
Versatile (H&E nuclei)

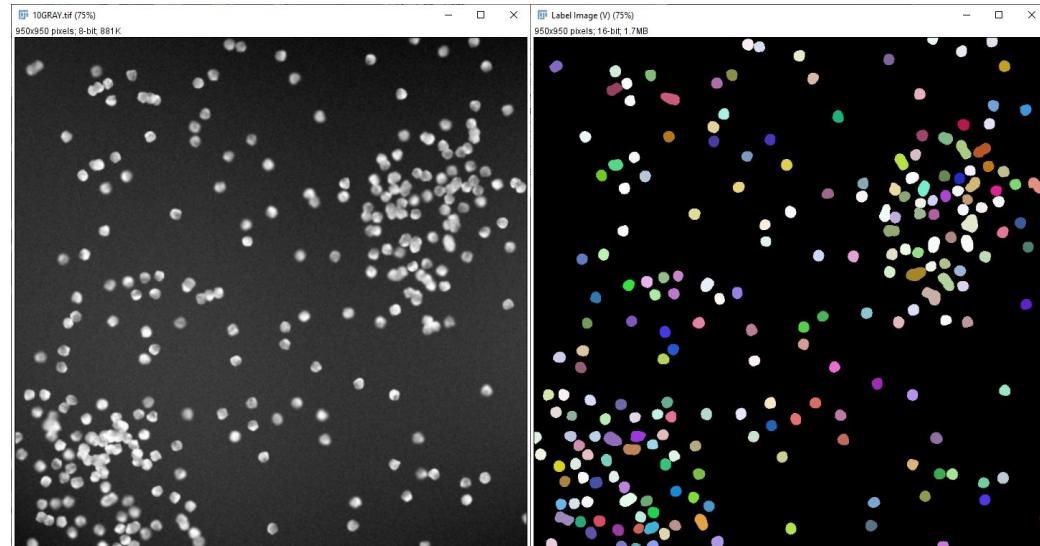
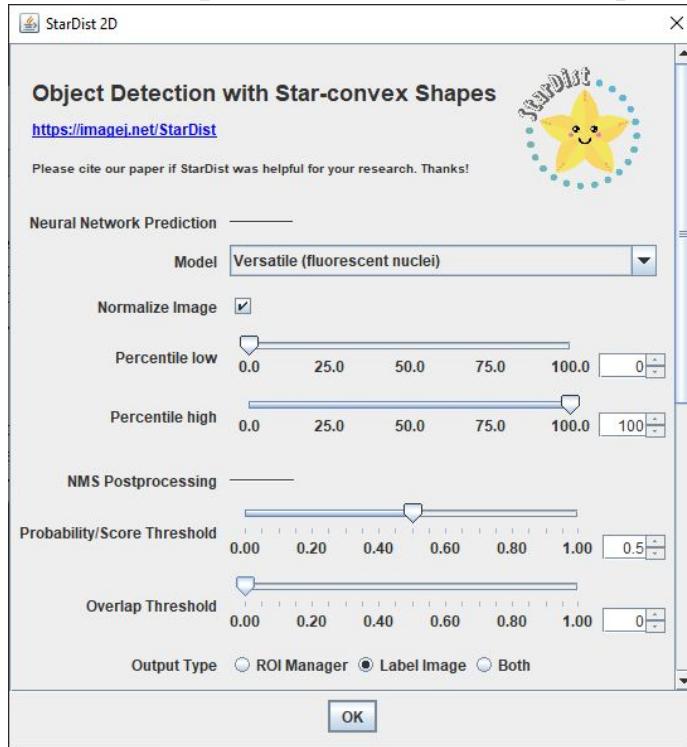


# Examples - Settings

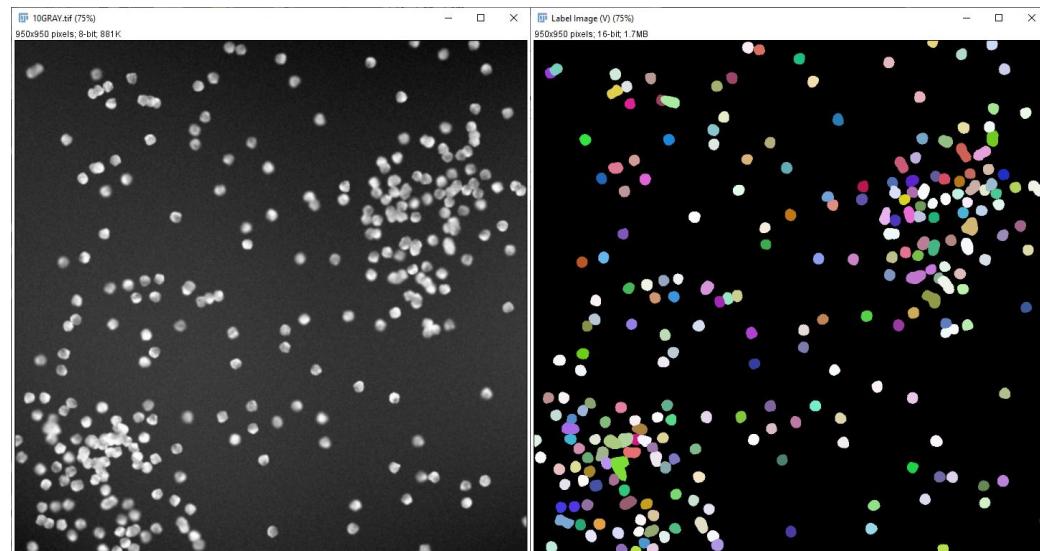
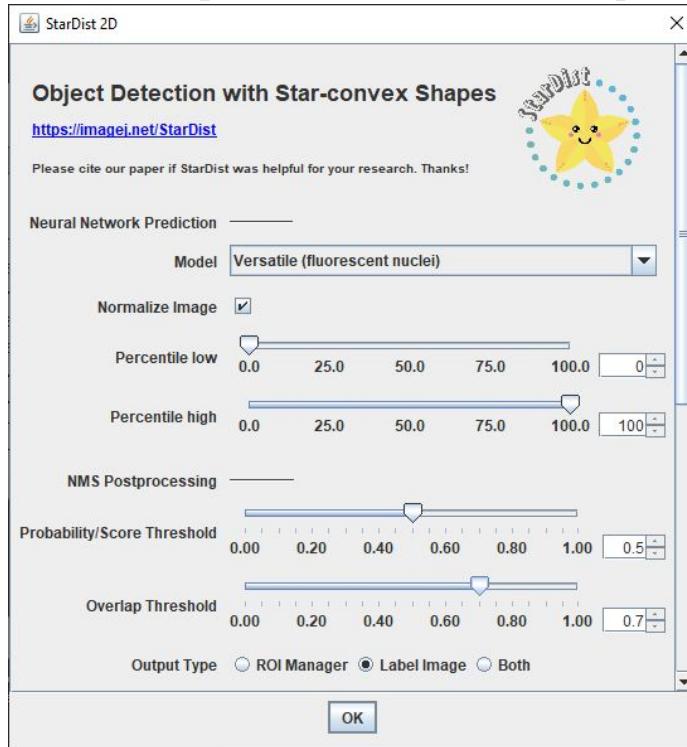


- Preprocessing settings
  - Normalization of image values.
  - Correction of “underexposure”.
  - Correction of “overexposure”.
- StarDist settings
  - Probability Threshold - how sure we want to be in detection of object.
  - Overlap Threshold - how much overlap we want to allow.

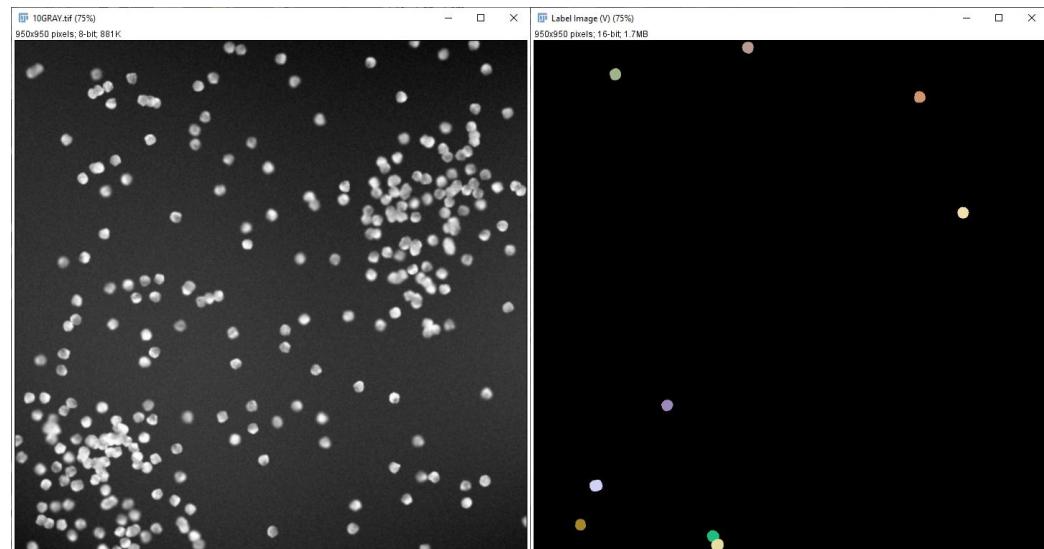
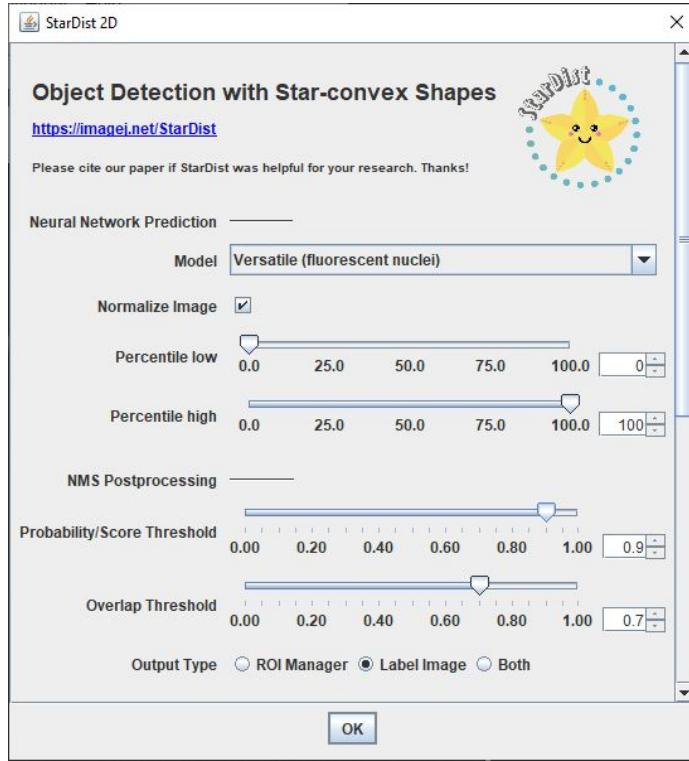
# Examples - Overlap



# Examples - Overlap



# Examples - Overlap + Probability



# Resources

- Haase, R. (2023). Introduction to bio-image analysis I. F1000 Research Limited.  
<http://doi.org/10.7490/F1000RESEARCH.1119427.1>
- Haase, R. Biolimage Analysis Lectures,  
<https://www.youtube.com/playlist?list=PL5ESQNFM5lc7SAMstEu082ivW4BDMvd0U>
- Haase, R. Biolimage Analysis Notebooks,  
<https://haesleinhuepf.github.io/BiolimageAnalysisNotebooks/intro.html>
- NEUBIAS Academy Lectures and Tools: <https://www.youtube.com/c/NEUBIAS/videos>
- Image to Knowledge Conference workshops: <https://www.youtube.com/@I2KConference>
- Schätz, M., Azevedo, M., & Sampaio, P. (2023). Internal ALM Biolimage Analysis workshop 2023. Zenodo.  
<https://doi.org/10.5281/zenodo.10205578>

# Exercises

# Exercises

Building and training new models is time consuming, so focus will be on **exploring, understanding and reusing**.

Two models to tackle:

- Noise2Void - denoising single channel images
  - Train and apply denoising on provided images with high and low noise.
- StarDist - training and evaluating custom model
  - Explore num. of training epochs and vs **Quality Control**
  - Test it on high overlap data (0.6 probability): <https://bbbc.broadinstitute.org/BBBC004>

Both are heavily GPU dependent! CPU will take even 80-100x more time.

# References

- Uwe Schmidt, Martin Weigert, Coleman Broaddus, and Gene Myers.  
*Cell Detection with Star-convex Polygons.*  
International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Granada, Spain, September 2018.
- Martin Weigert, Uwe Schmidt, Robert Haase, Ko Sugawara, and Gene Myers.  
*Star-convex Polyhedra for 3D Object Detection and Segmentation in Microscopy.*  
The IEEE Winter Conference on Applications of Computer Vision (WACV),  
Snowmass Village, Colorado, March 2020
- ImageJ/Fiji plugin for StarDist: <https://imagej.net/plugins/stardist>