

Deep Learning for Bio-image Analysis

Robert Haase

Funded by



Bundesministerium
für Bildung
und Forschung

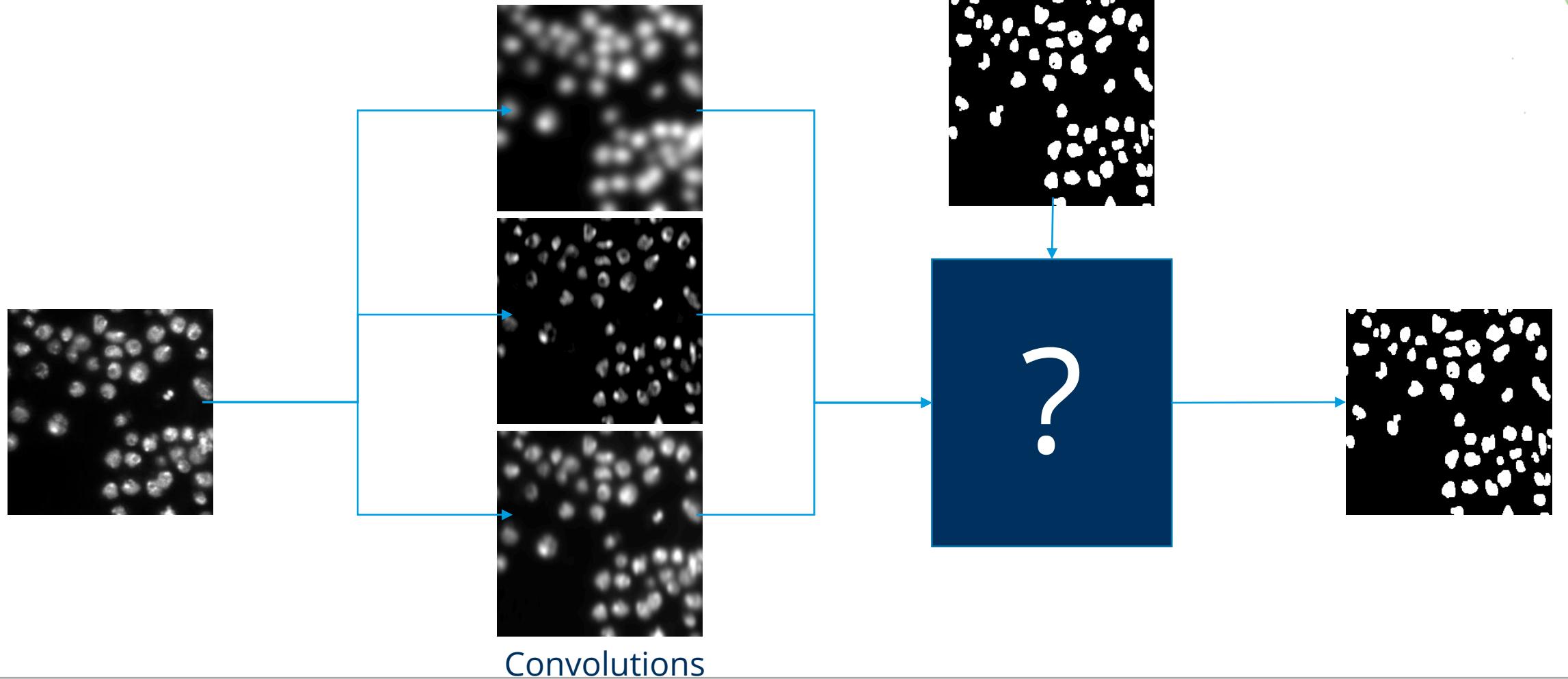
SACHSEN



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.

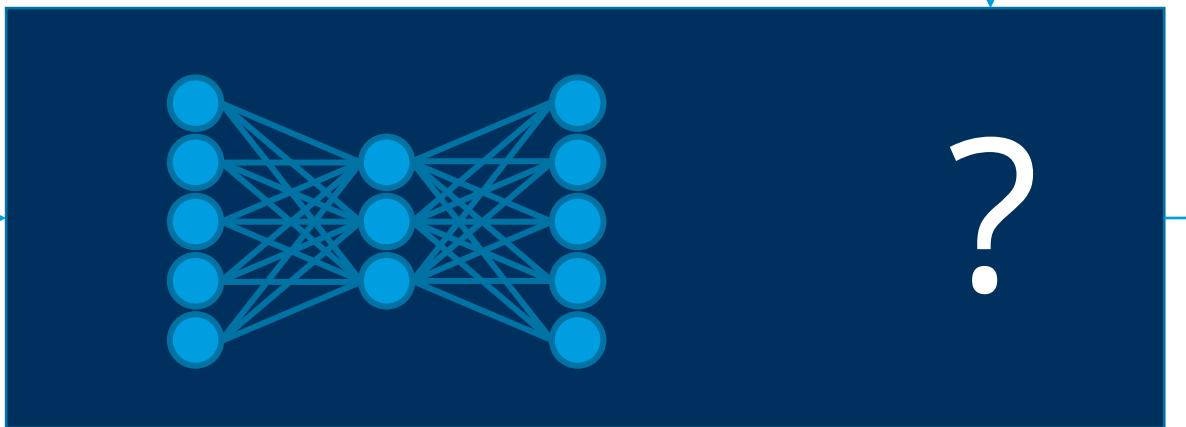
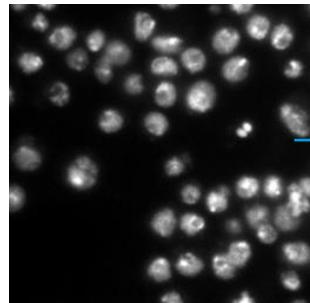
Machine learning for image analysis

In classical machine learning, we typically select features for training our classifier

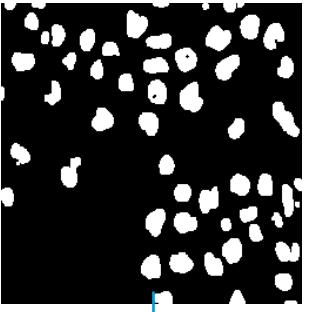


Outlook: Deep learning for image analysis

In deep learning, this selection becomes part of the black box

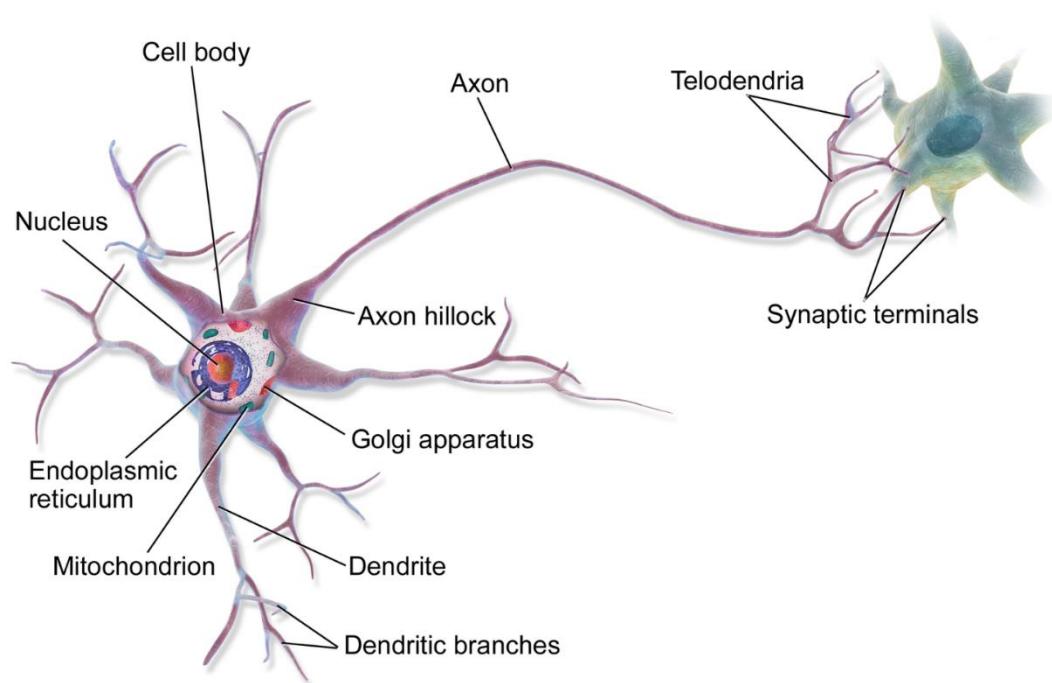


Convolutional neural
networks

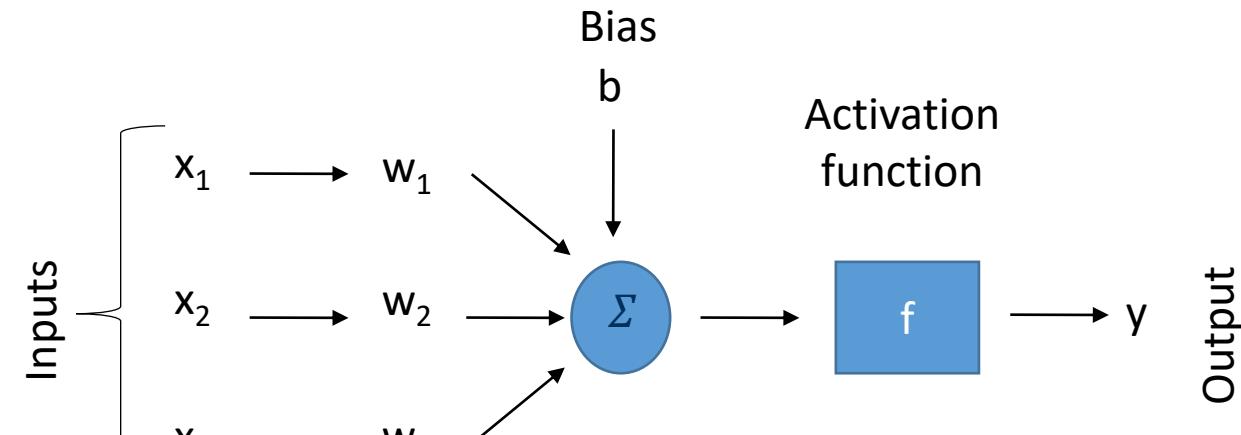


Neural networks

- How biologists see neurons



- How computer scientists see neurons
“perceptron”

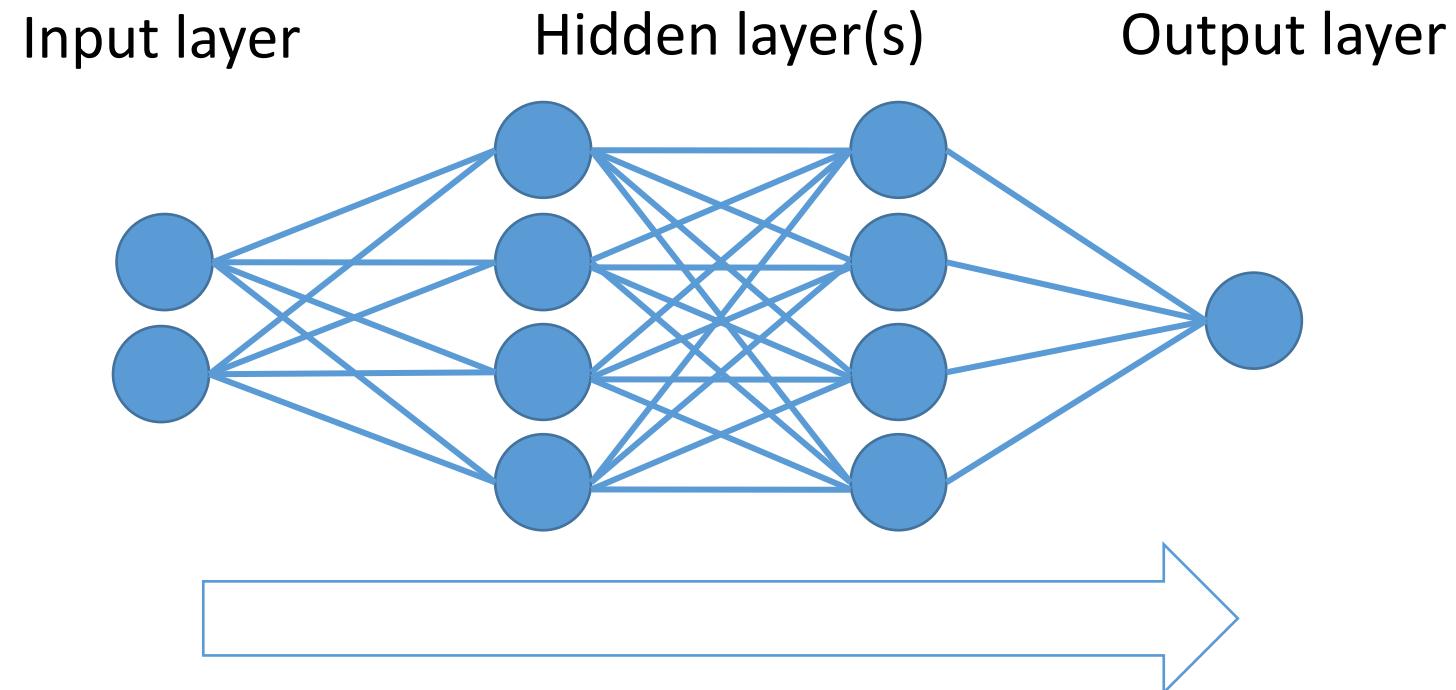


$$y = f(w_1 x_1 + w_2 x_2 + w_3 x_3 + b)$$

Neural Networks

- Early form: “Multilayer Perceptron”
- fully connected class of feedforward artificial neural network

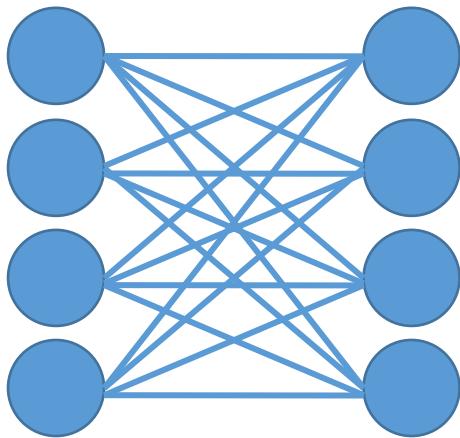
If there are *many* hidden layers, we speak of a *deep neural network*



Convolutional neural networks

- Layer types

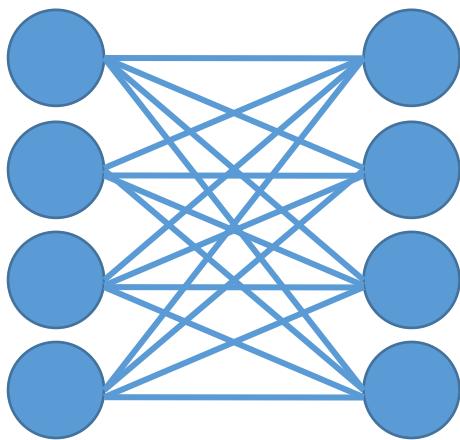
Fully connected layer



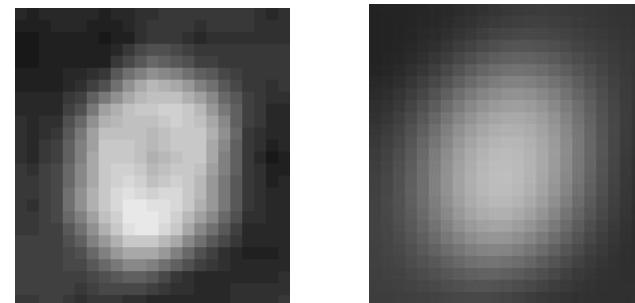
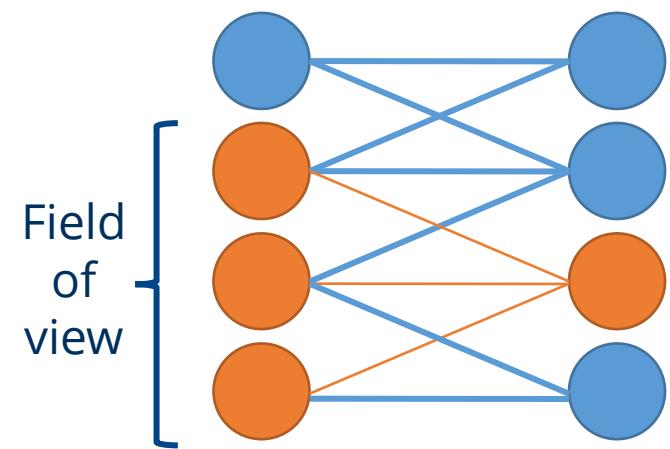
Convolutional neural networks

- Layer types

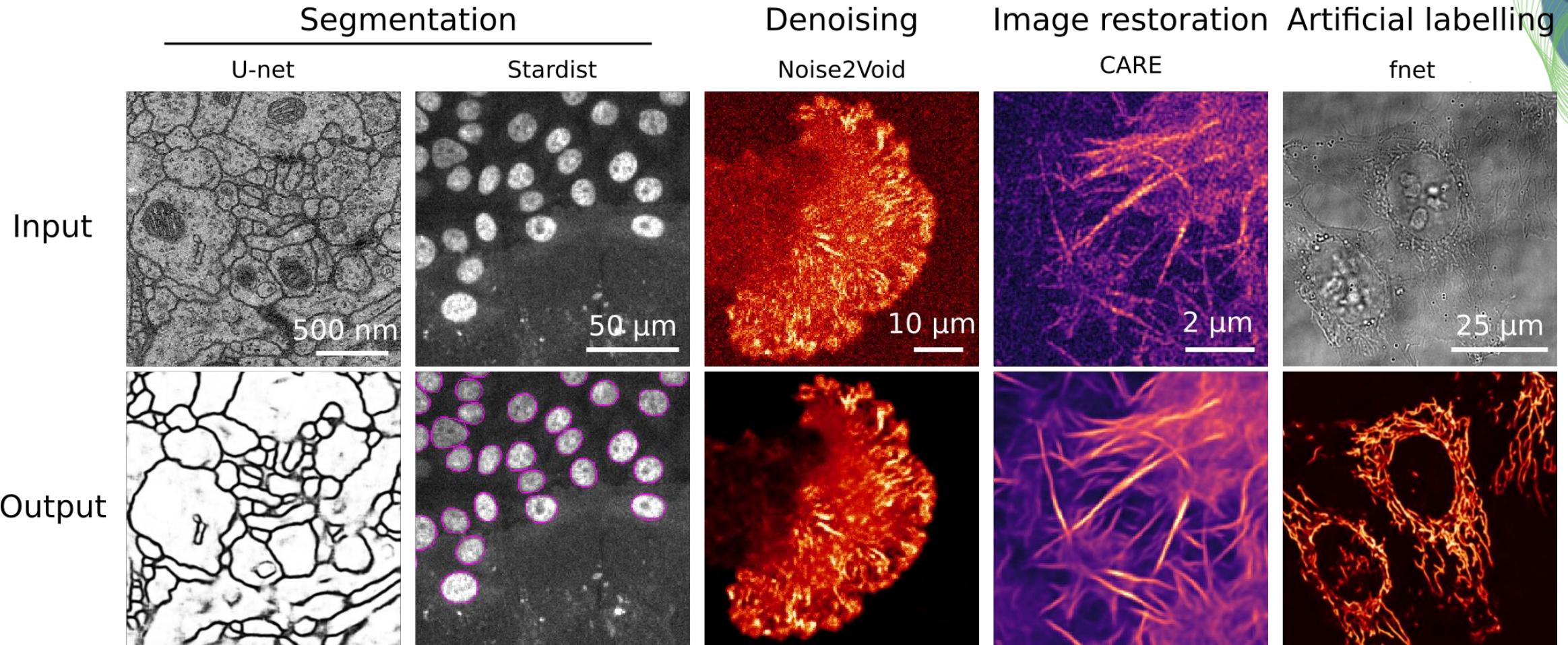
Fully connected layer



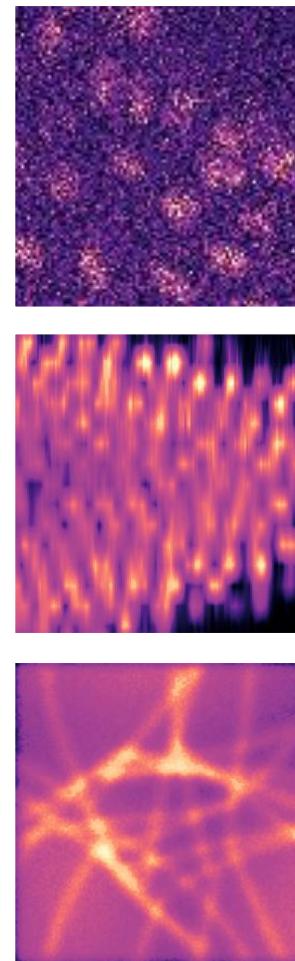
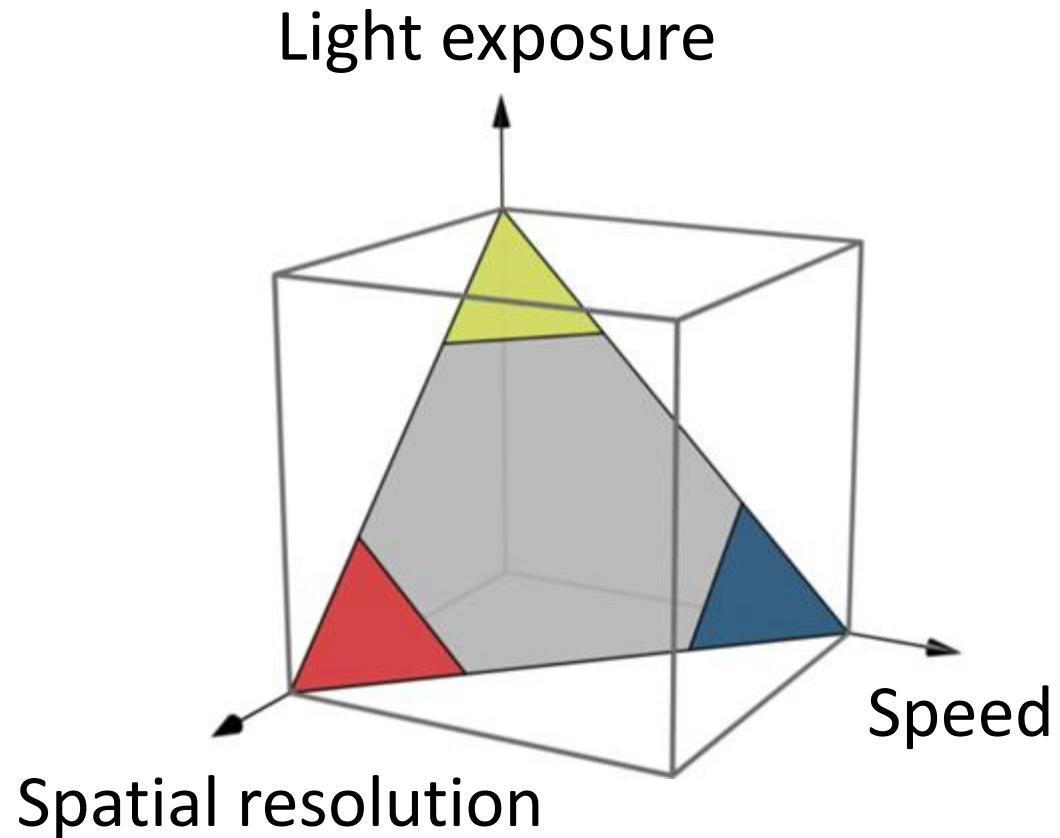
Convolutional layer



Deep Learning for Microscopy



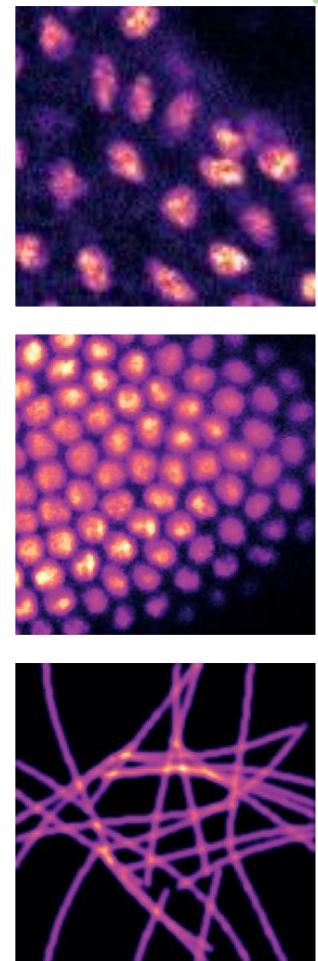
Trade-offs in live-cell imaging



Light Exposure

Speed

Spatial Resolution



The U-net

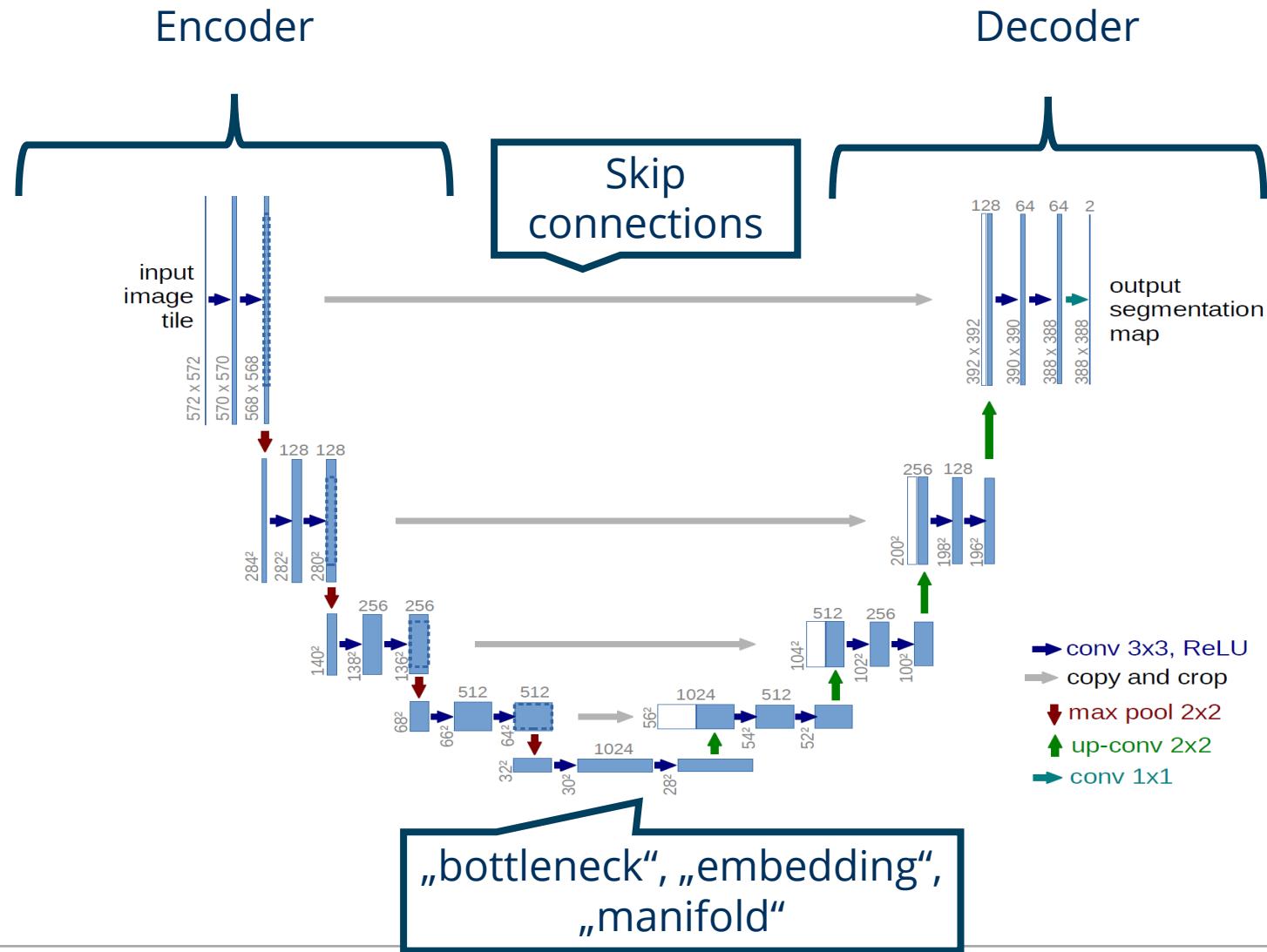


Image denoising: CARE

- Content aware image restoration (CARE)
- Image acquisition of pairs of images: A high-quality and a low-quality image.
- Problem: shot noise, biology moves!

Pair of images required!

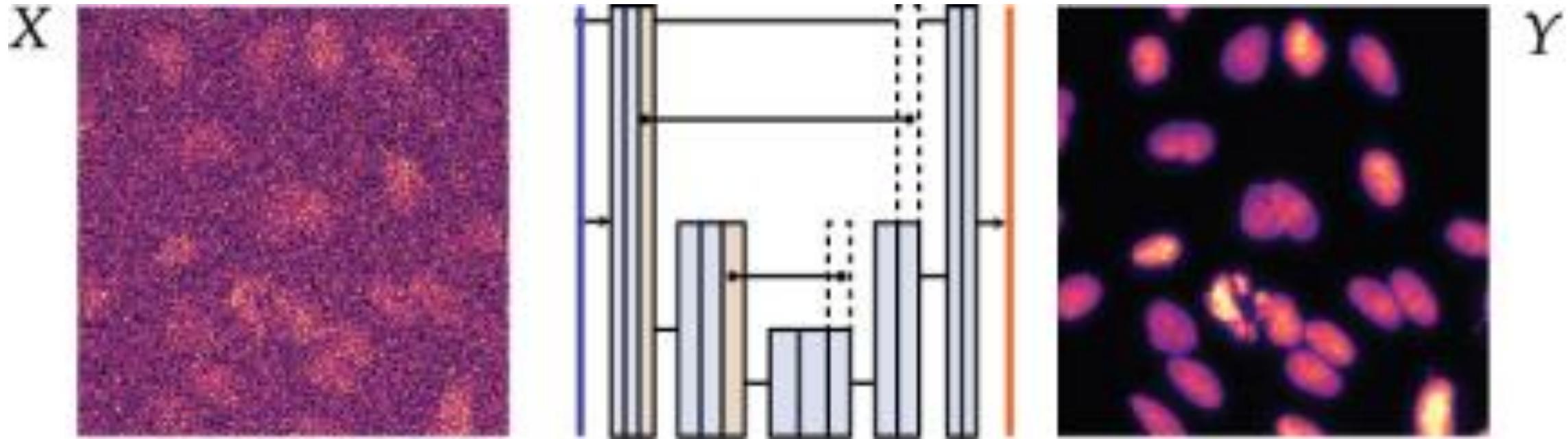
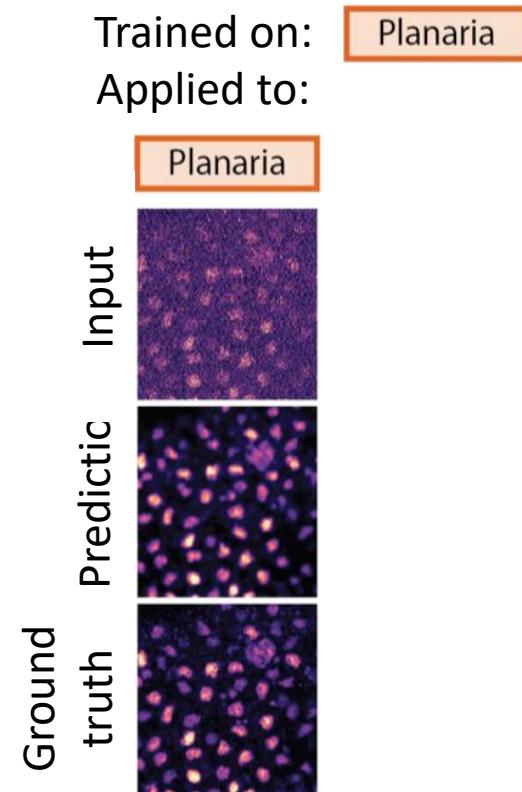


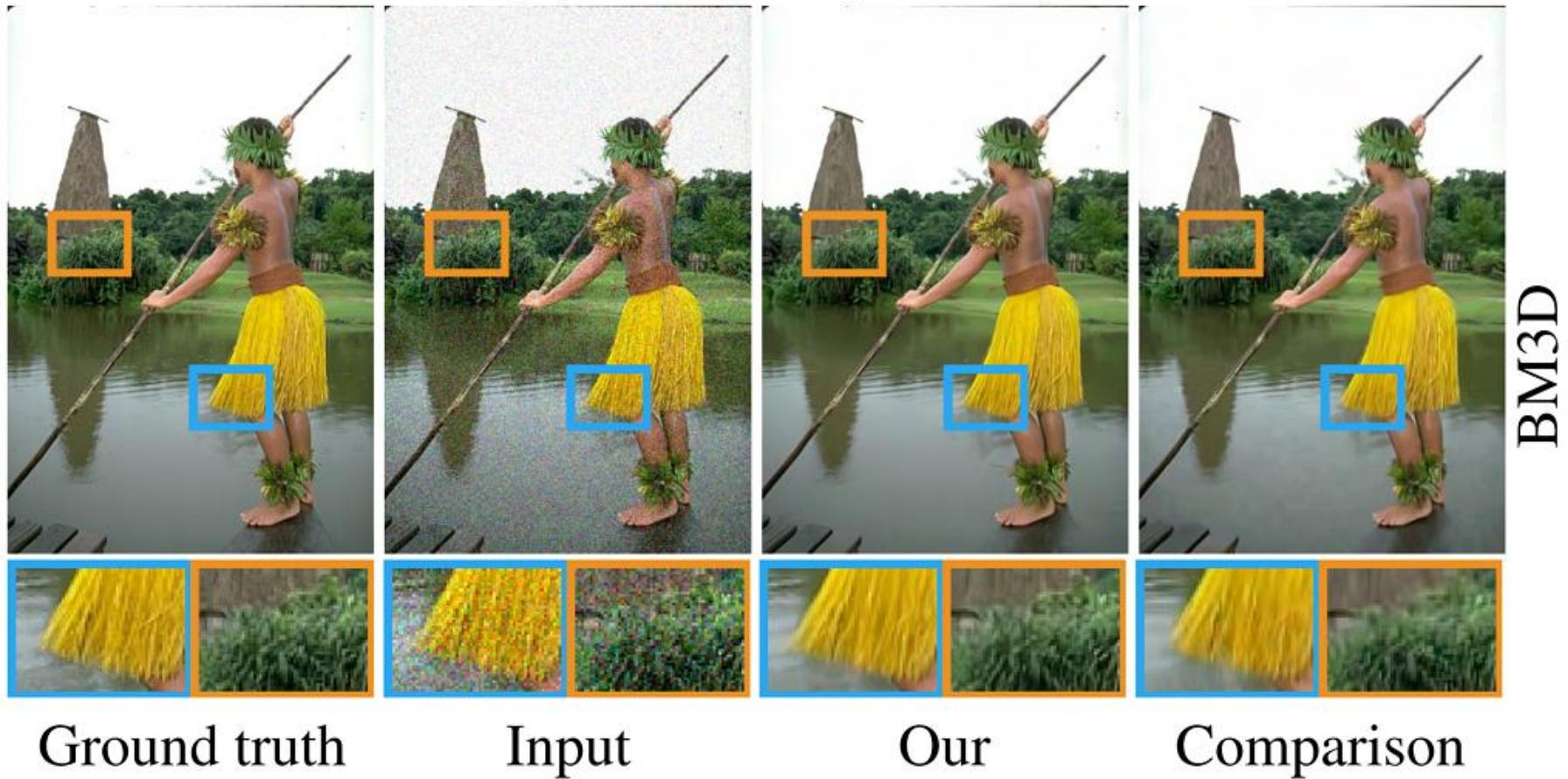
Image denoising: CARE

- Important to use on the same conditions/structures/staining that the networks were trained on!



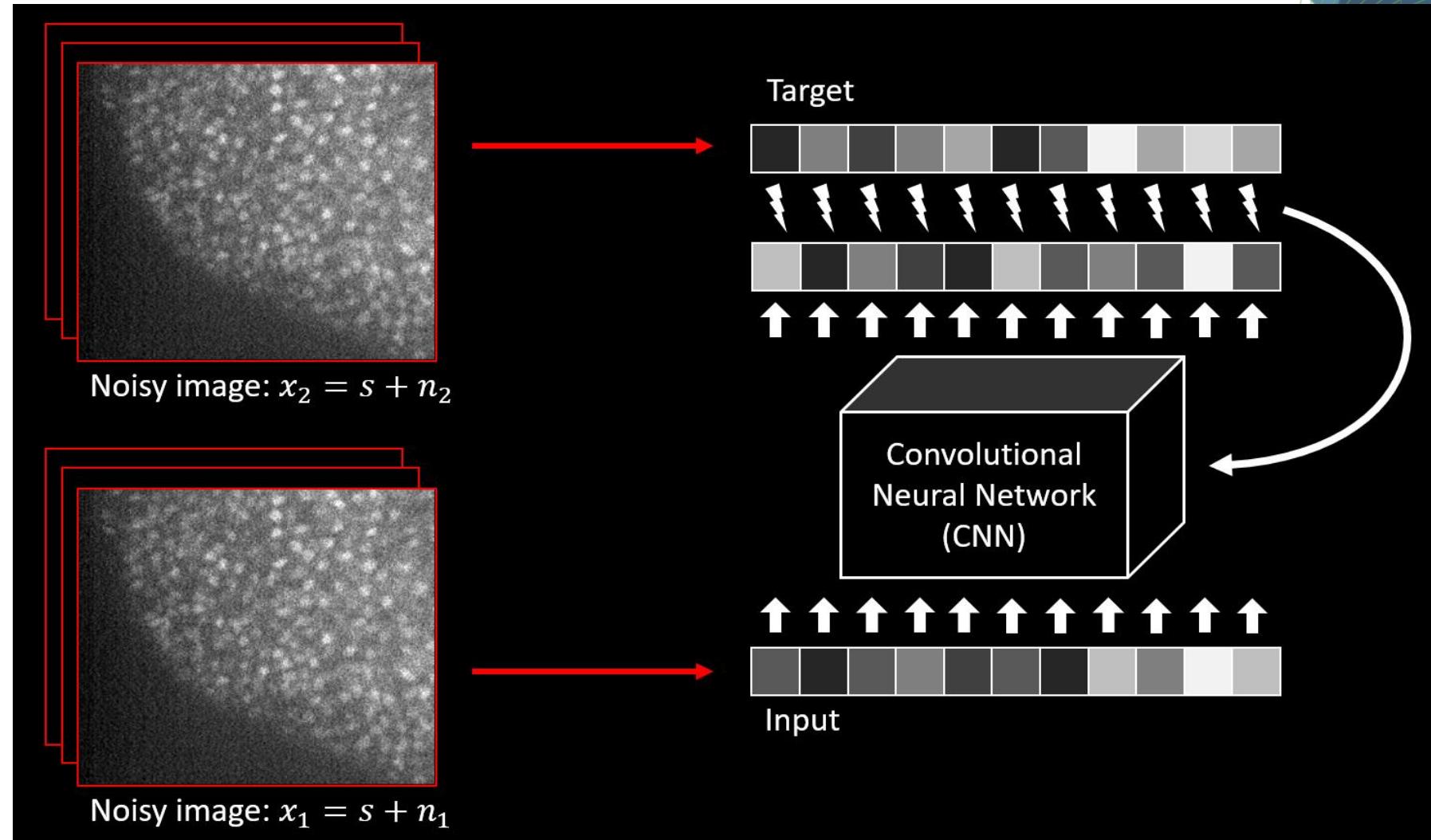
Noise2noise

(a) Gaussian ($\sigma = 25$)



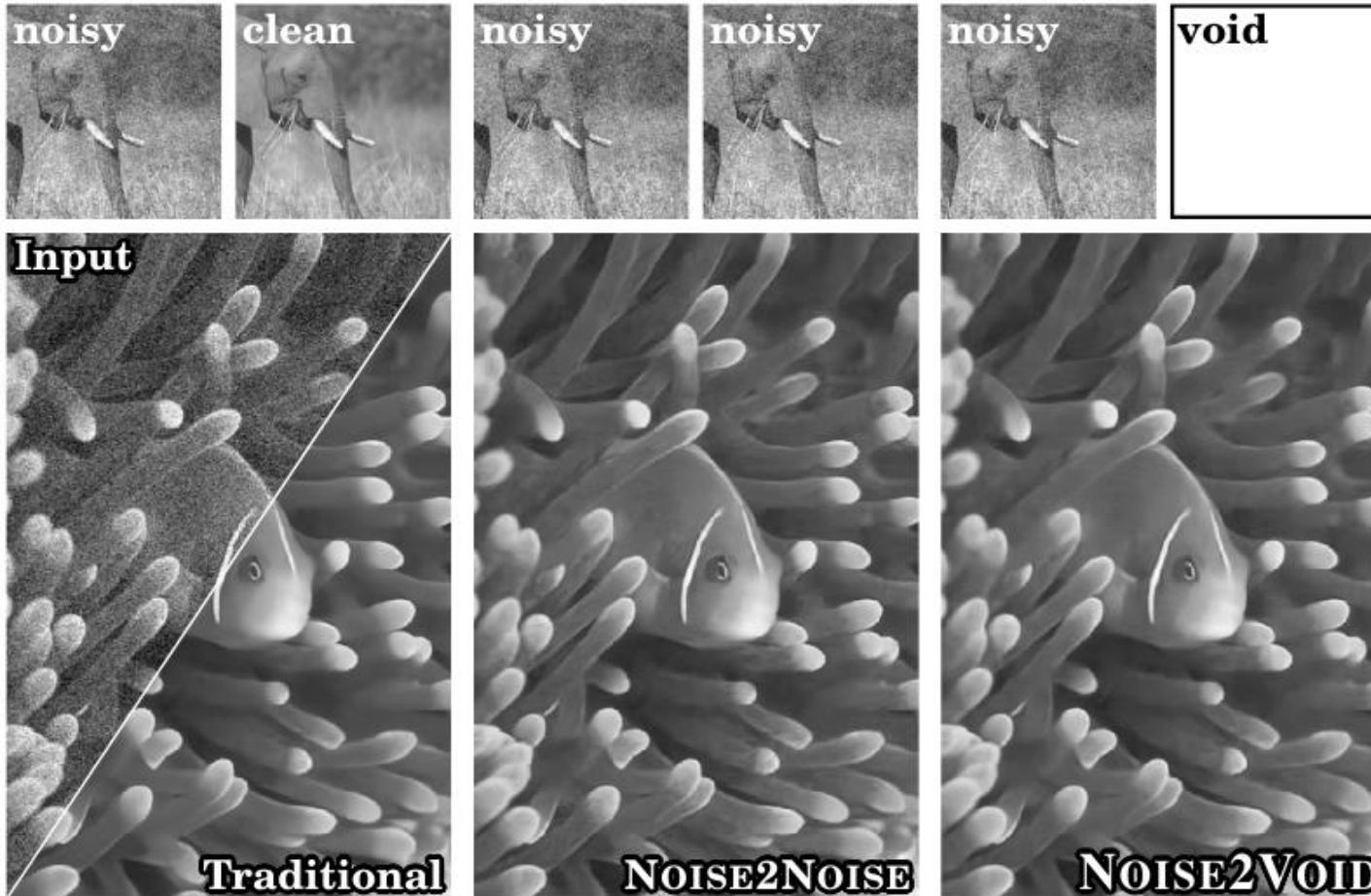
Noise2noise

- by Nvidia (Lehtinen 2018)
<https://arxiv.org/pdf/1803.04189.pdf>



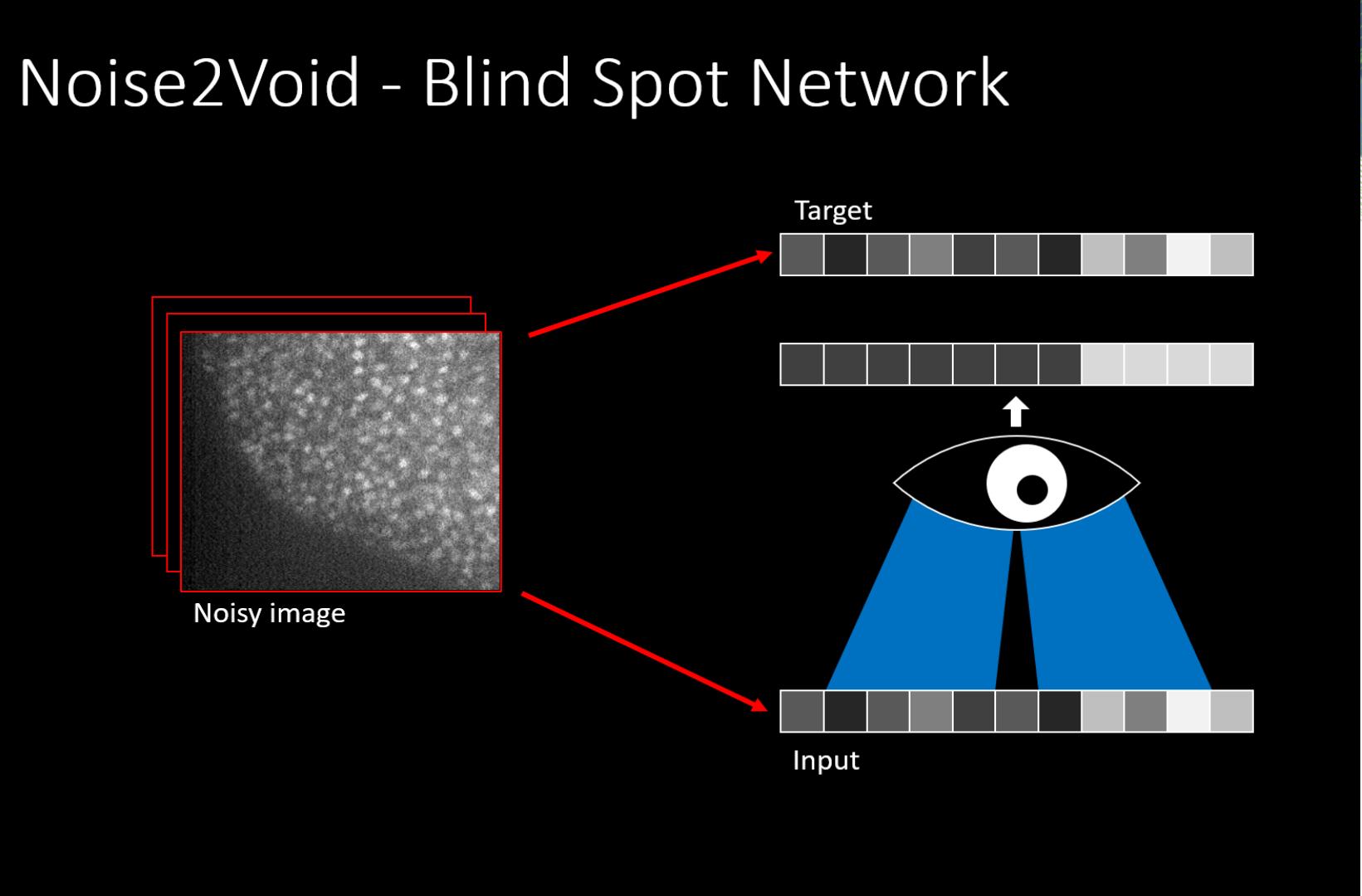
Noise2void

Image denoising without image pairs



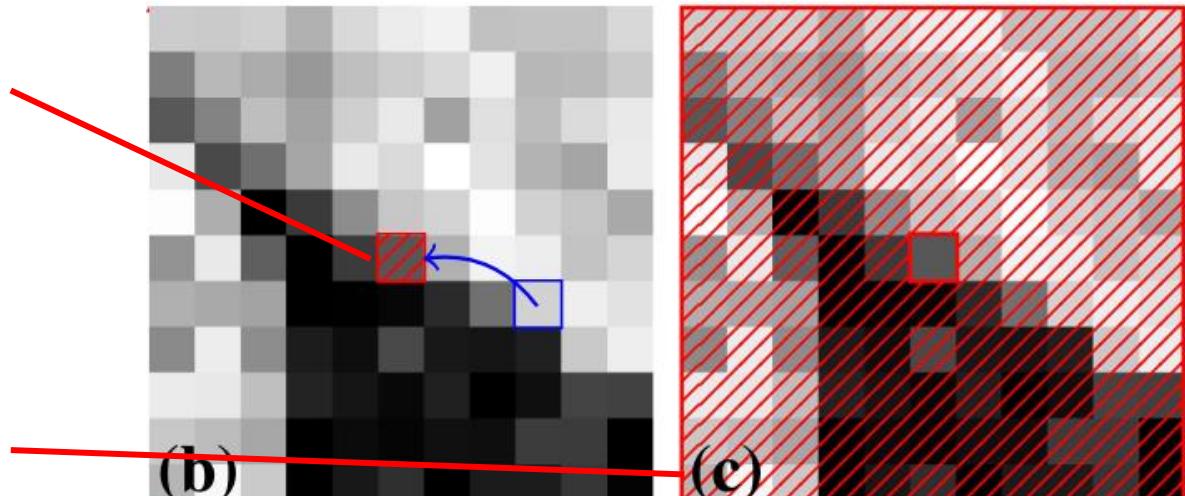
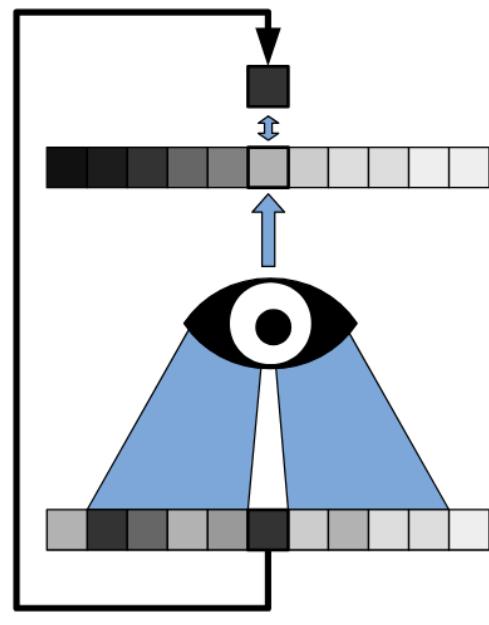
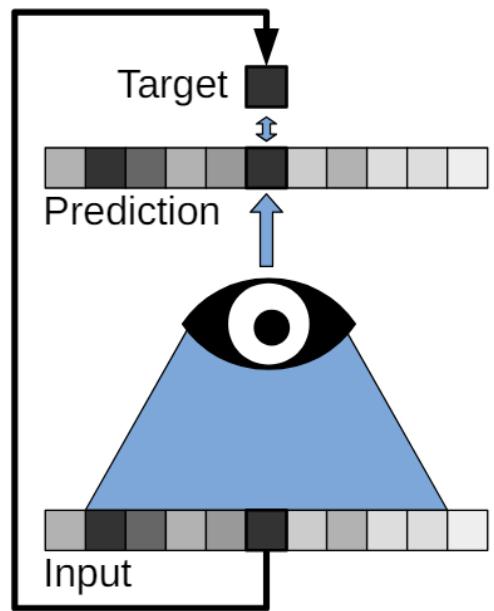
Noise2void

- Noise2void: Krull et al (2019)
<https://arxiv.org/abs/1811.10980>
- Noise2self: Batson and Royer (2019)
<https://arxiv.org/abs/1901.11365>



Noise2void

„Blind-spot-network“



Noise2void

Napari-plugin

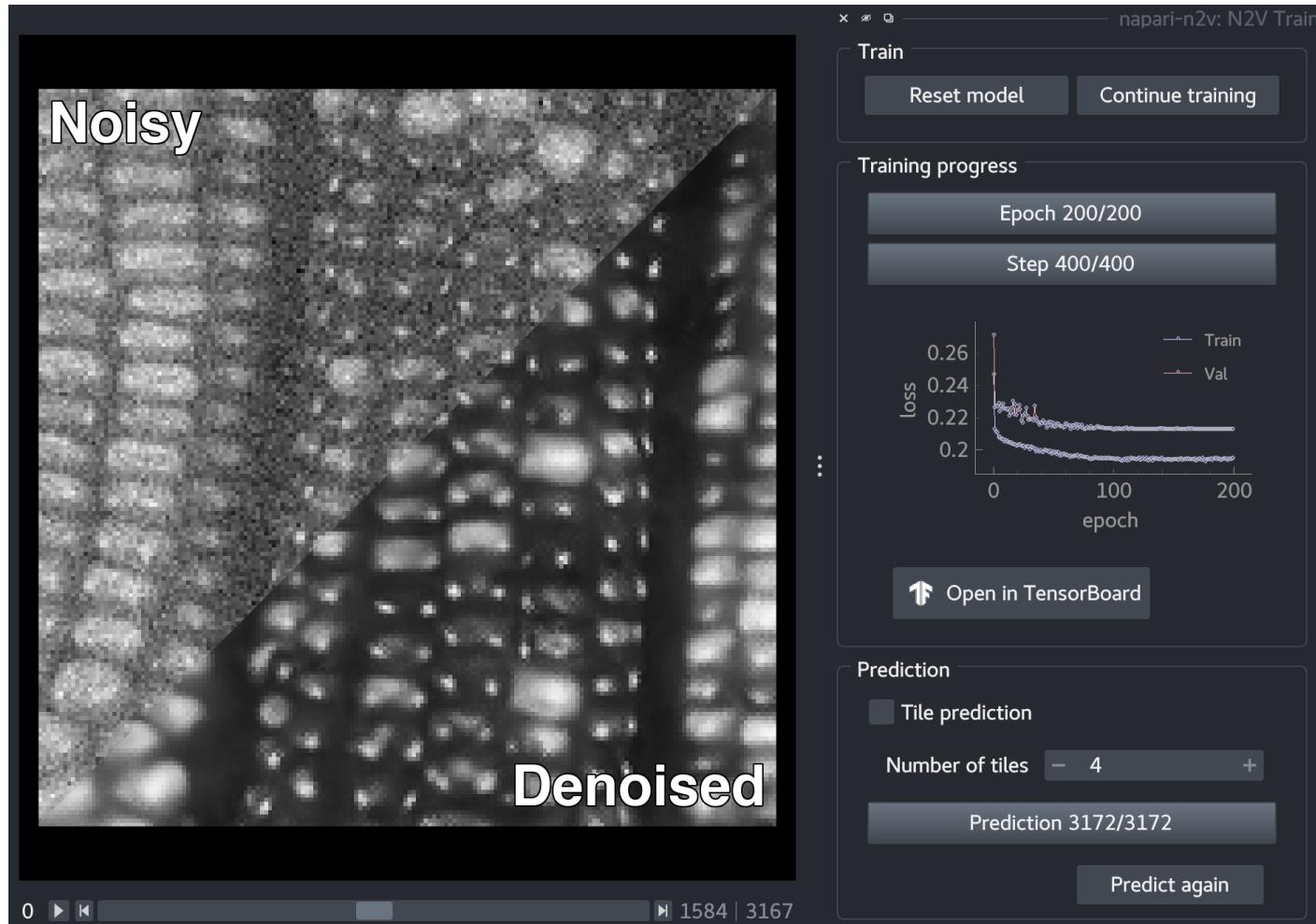
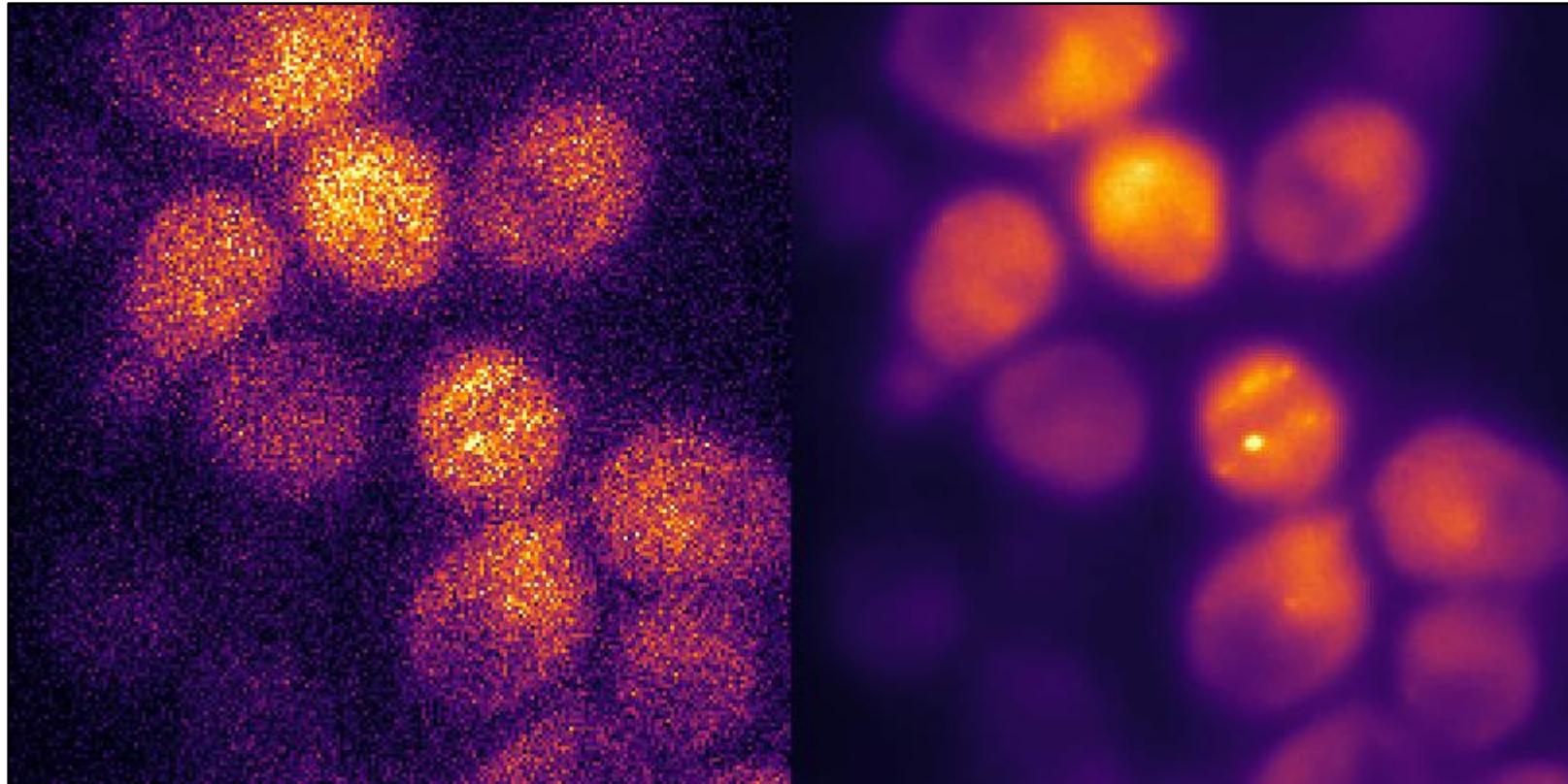


Image denoising

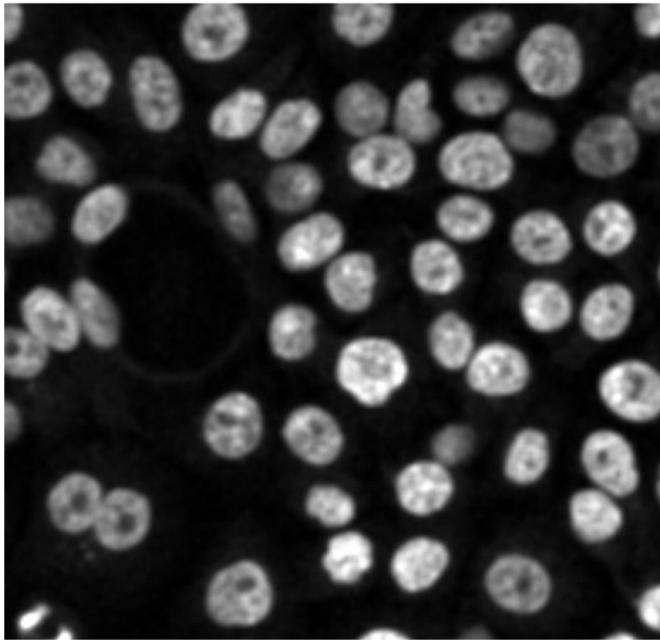
- Noise2Void



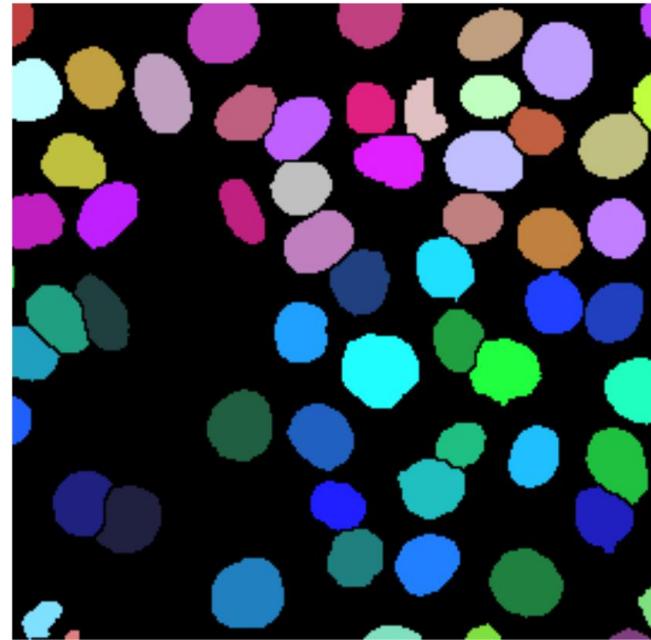
StarDist: Nuclei segmentation

Advanced algorithms are necessary when nuclei become too dense.

Image of nuclei



Corresponding masks



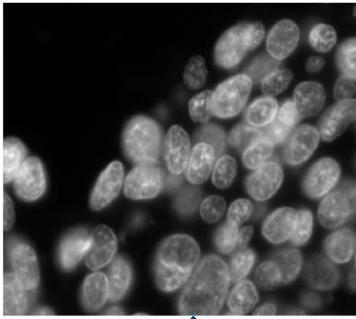
StarDist

- Prediction of probable object centers + polygon outlines
- Non-maximum-suppression of less likely polygons

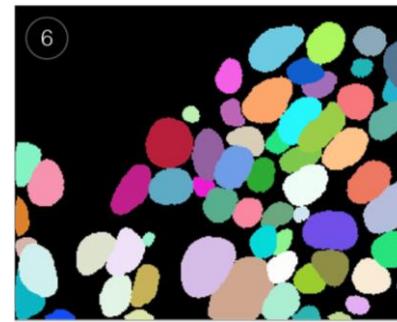
Class. Image proc.

Deep learning

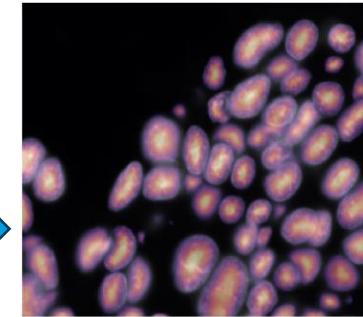
Input



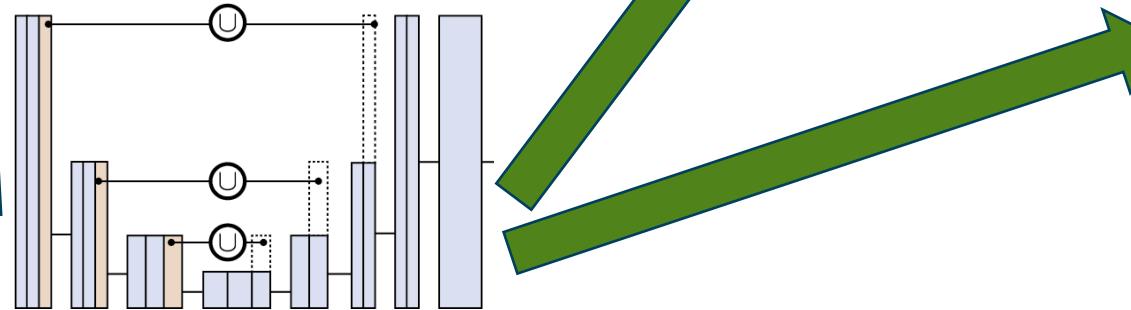
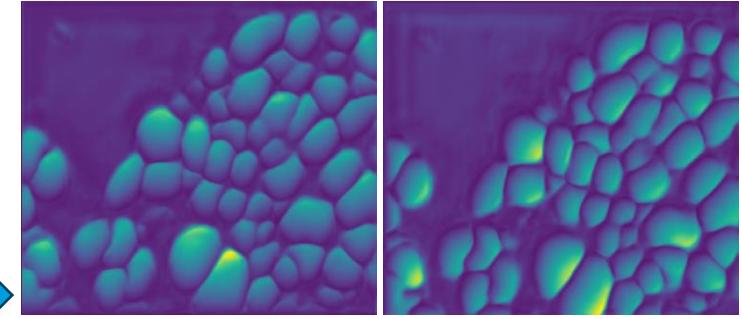
Ground truth



Object probabilities



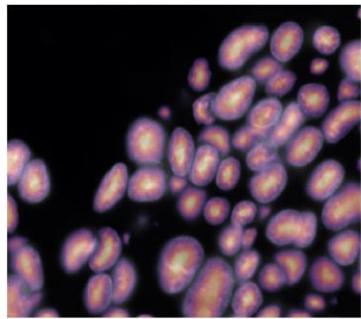
Directional distance maps (32x)



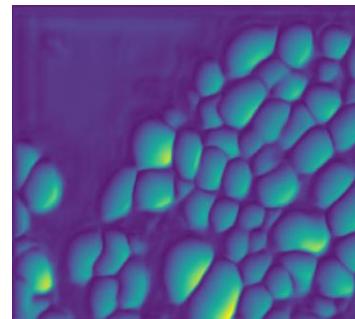
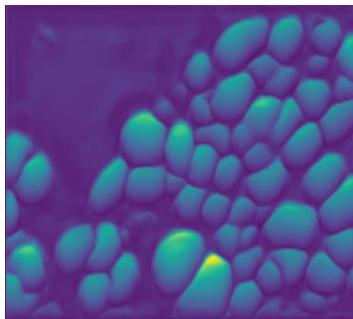
StarDist

- Prediction of probable object centers + polygon outlines
- Non-maximum-suppression of less likely polygons

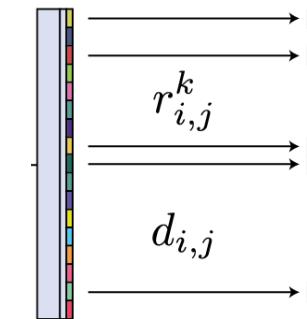
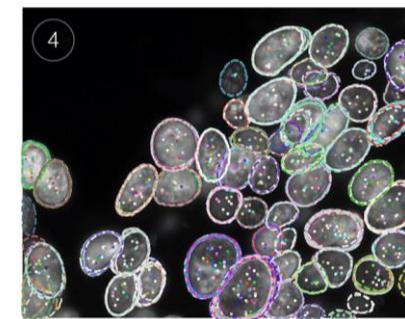
Object probabilities



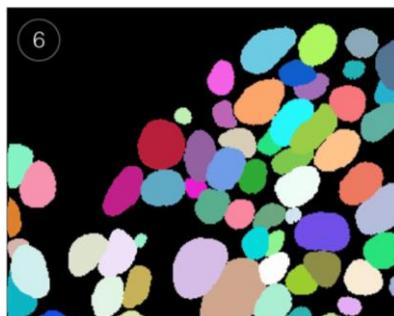
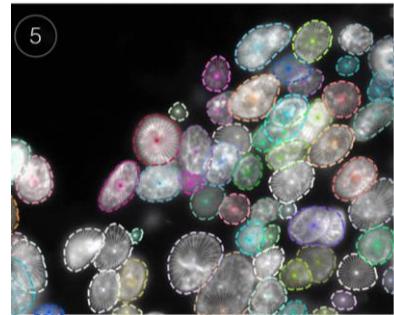
Directional distance maps



Polygon candidates

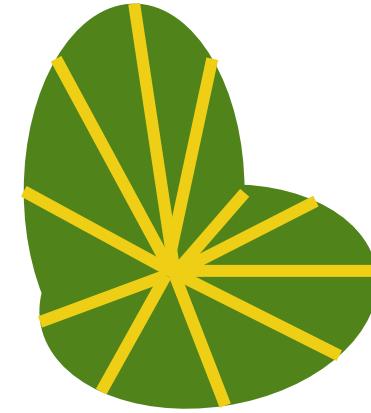
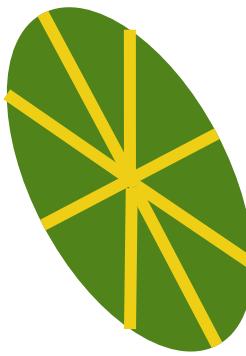
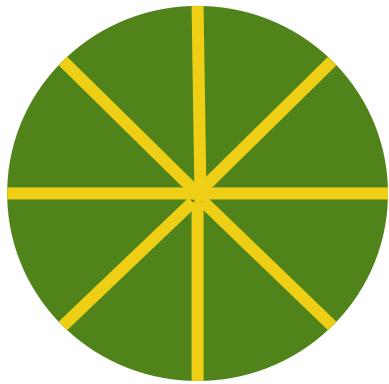


Final segmentation



StarDist: Limitations

Star-convex shapes!



Good for nuclei, bad for cells.

StarDist: Python

Initialize model / download pretrained model

```
[3]: # creates a pretrained model  
model = StarDist2D.from_pretrained('2D_versatile_fluo')  
  
Found model '2D_versatile_fluo' for 'StarDist2D'.  
Loading network weights from 'weights_best.h5'.  
Loading thresholds from 'thresholds.json'.  
Using default values: prob_thresh=0.479071, nms_thresh=0.3.
```

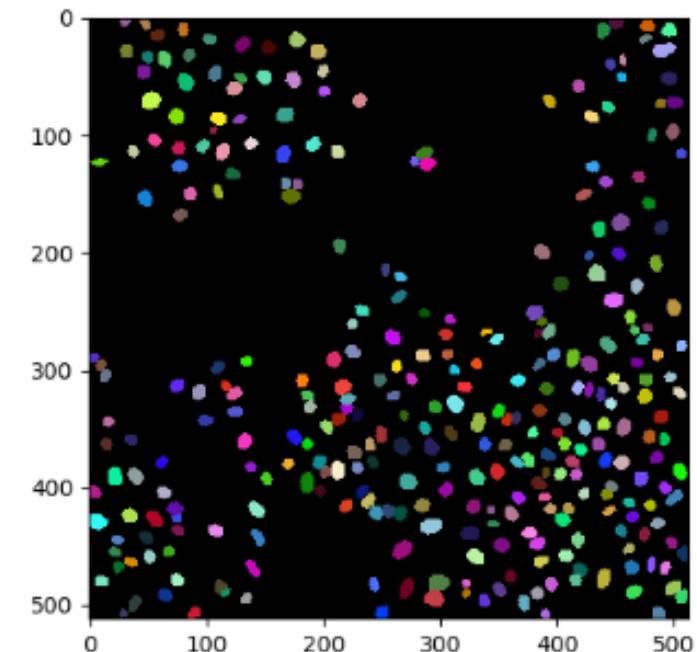
Normalize intensity to range [0, 1]

```
[4]: axis_norm = (0,1)  
image = normalize(image, 1,99.8, axis=axis_norm)
```

Apply model

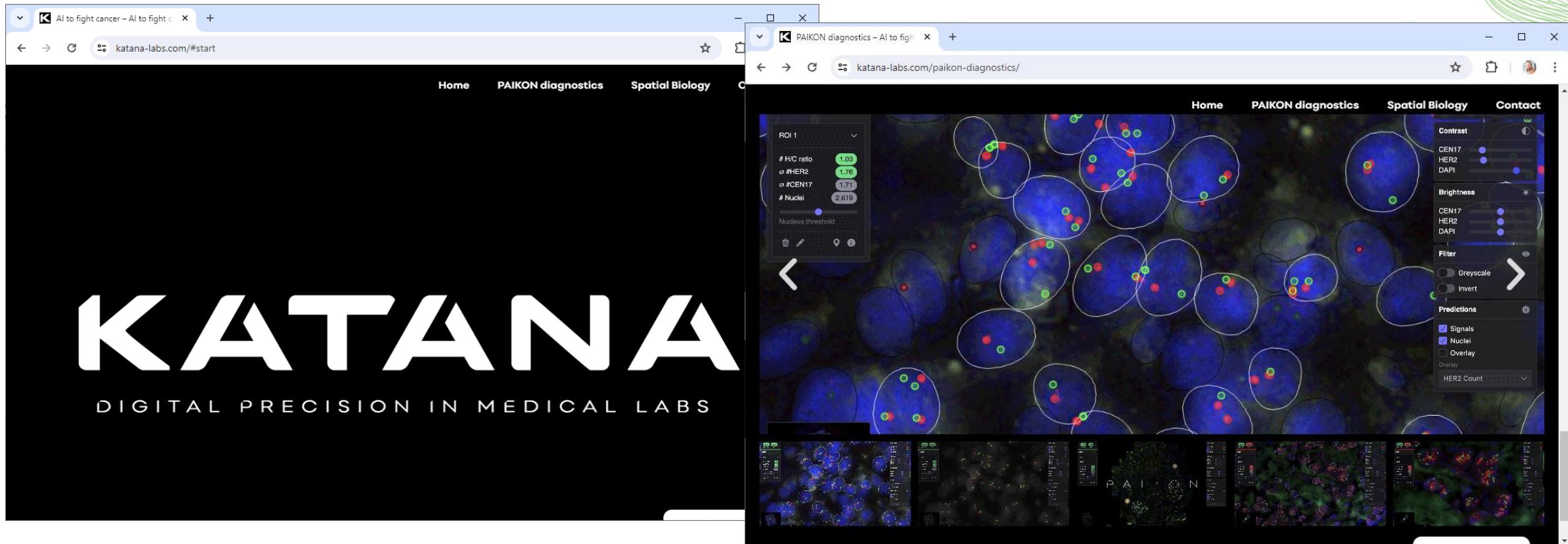
```
[5]: labels, details = model.predict_instances(image)  
  
stackview.insight(labels)
```

```
[5]:
```



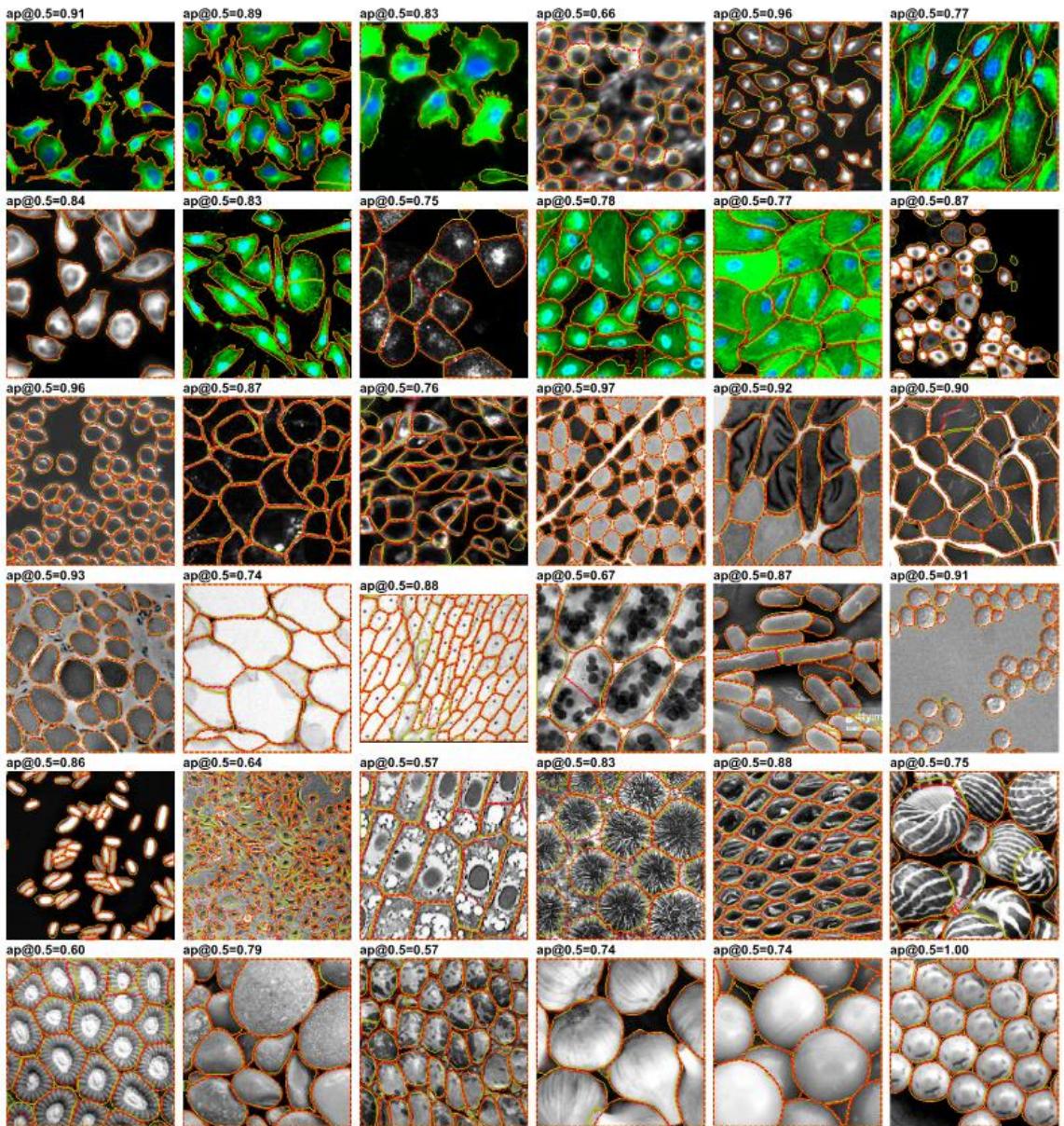
StarDist: Business model

Serving clinical scientists using software based on open-source code



CellPose

- Processing more diverse datasets and object shapes

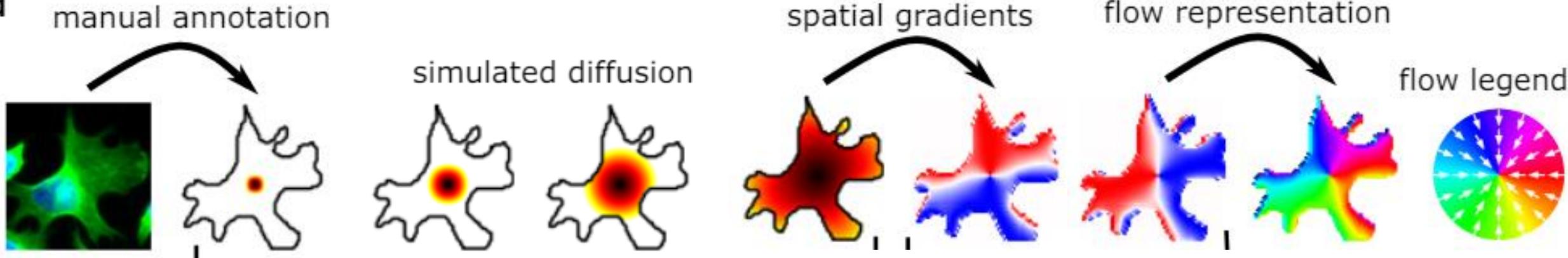


CellPose

Compute „flow fields“ from images

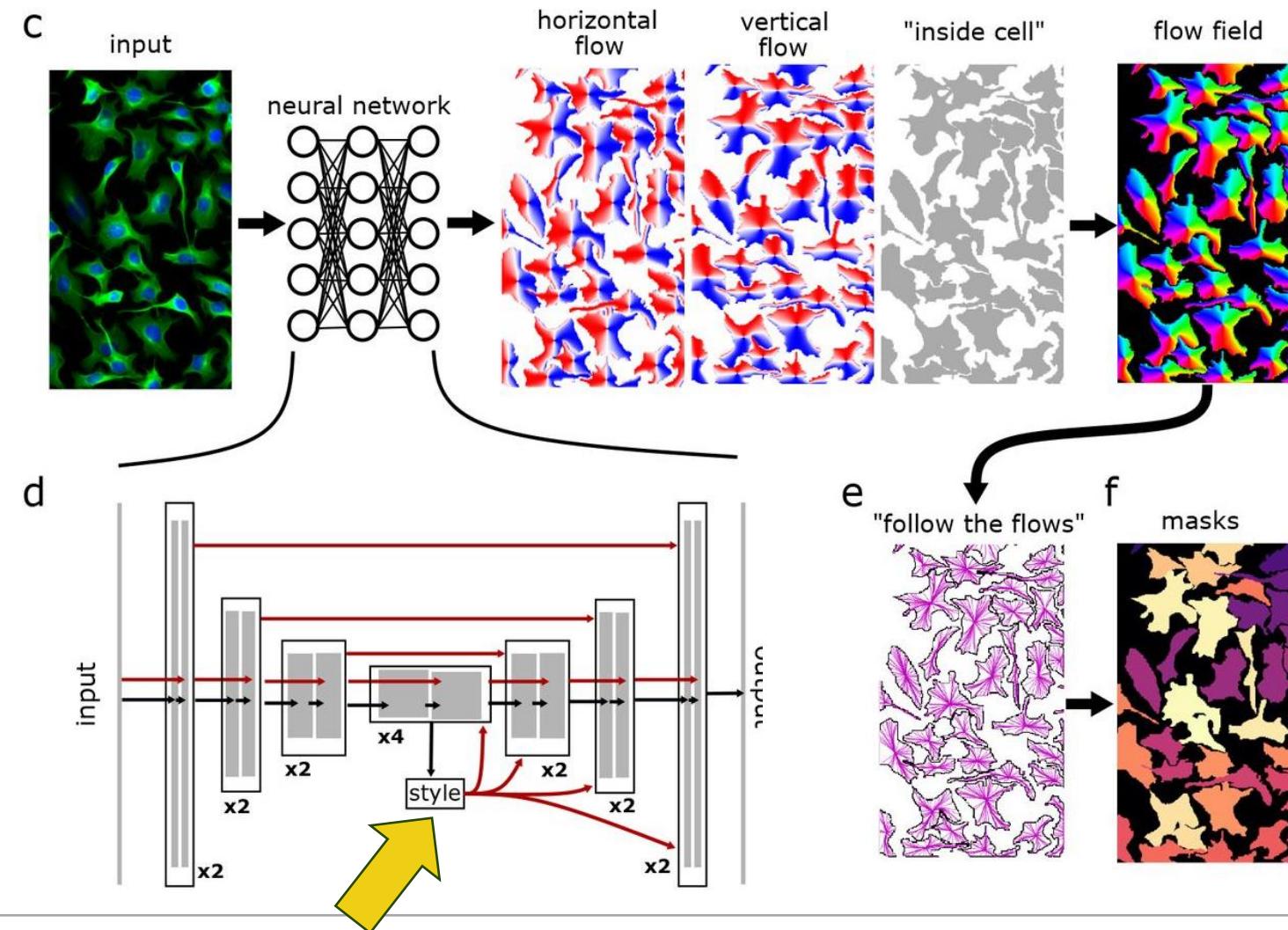
- using classical image processing during training
- using neural networks during prediction

a



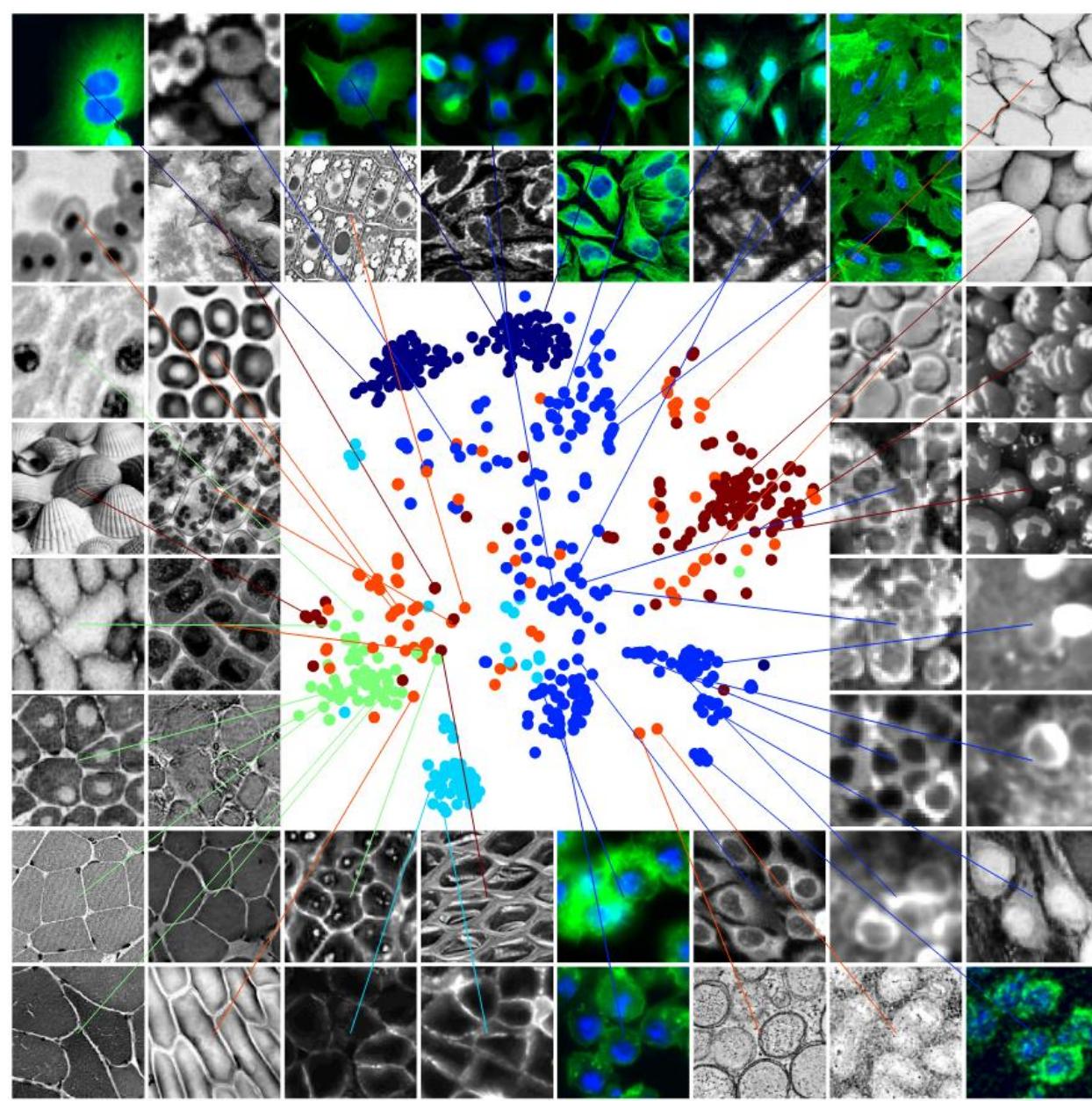
CellPose

- Cell/Nuclei – segmentation based on flow-fields
- Technically similar to Watershed, but with a deep-learning based altitude-image

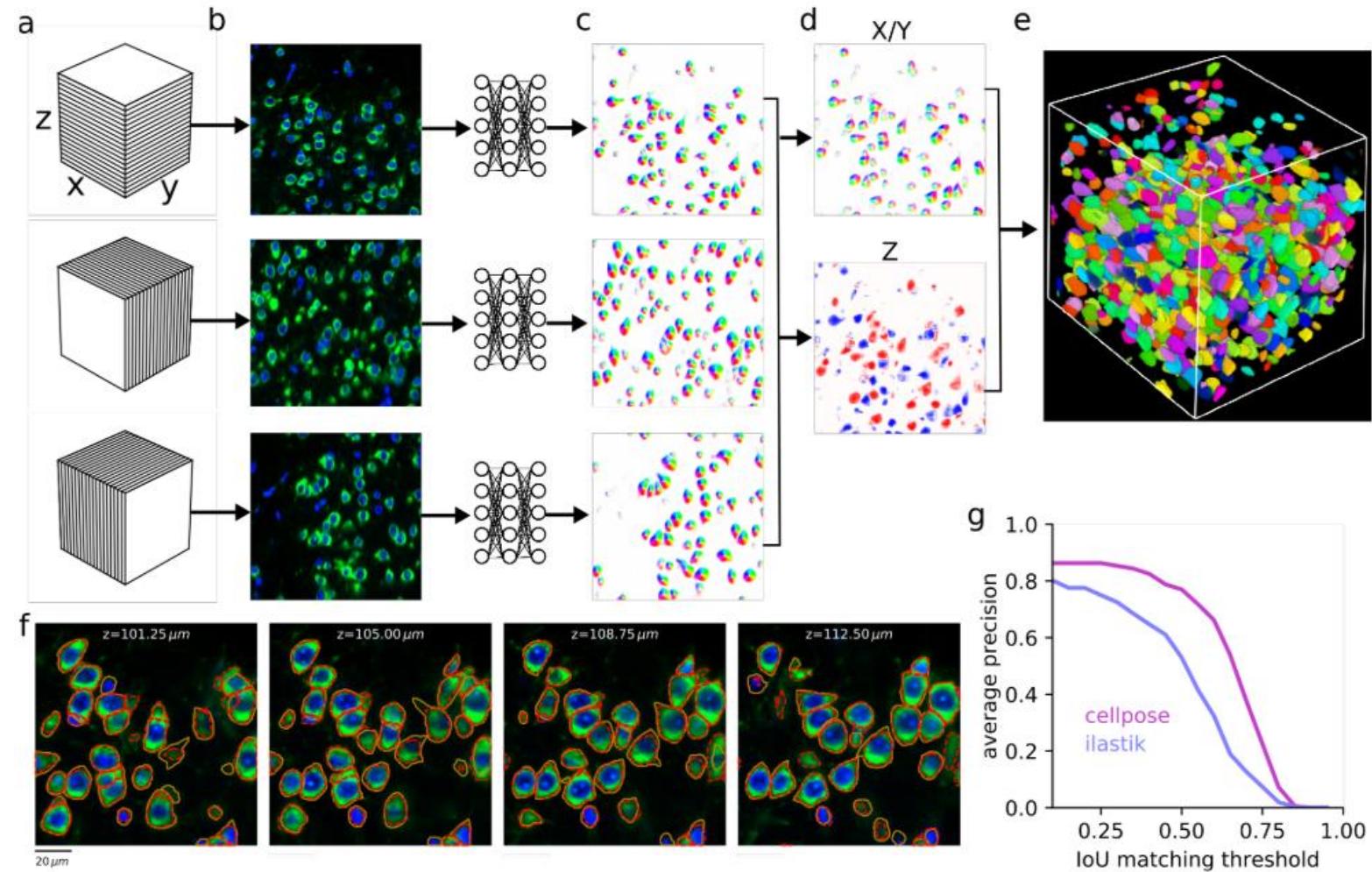


CellPose

- Image *style* is a parameter determined before prediction to guide segmentation.

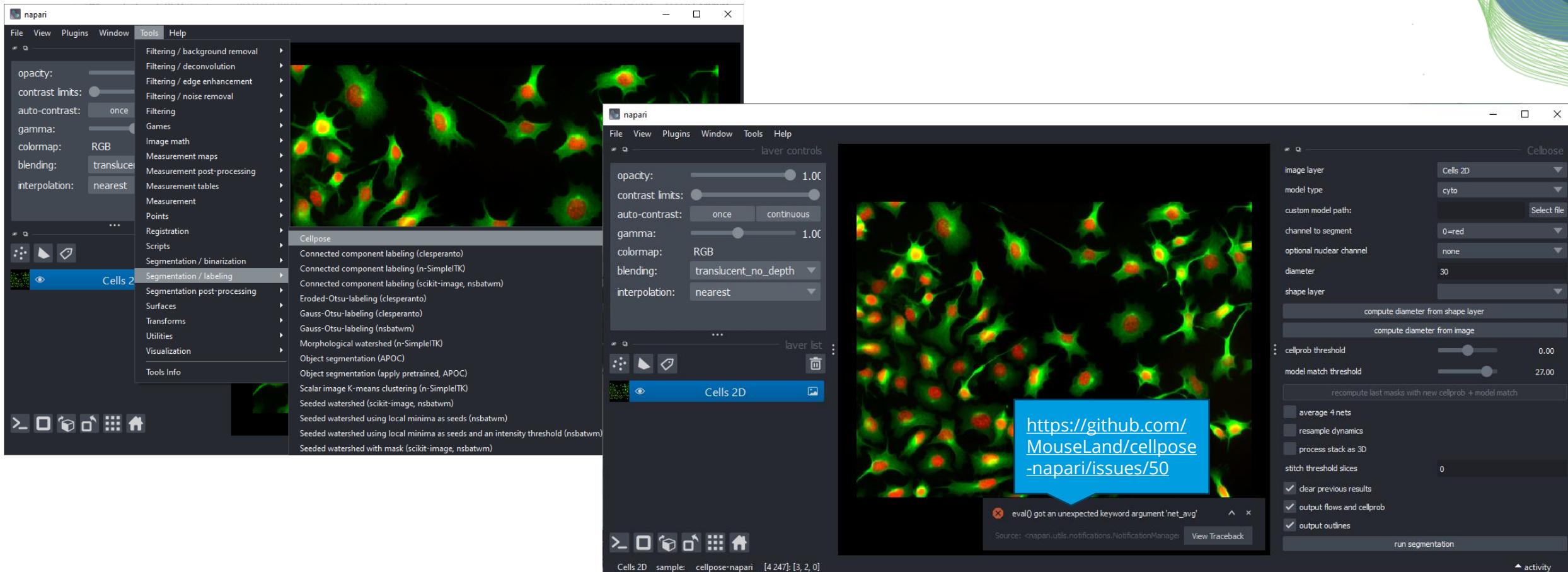


Cellpose 3D



Cellpose in Napari

mamba install cellpose-napari



Cellpose in Python

Initialize a pretrained model

```
[3]: model = cellpose.models.Cellpose(gpu=False, model_type='nuclei')
```

List available models

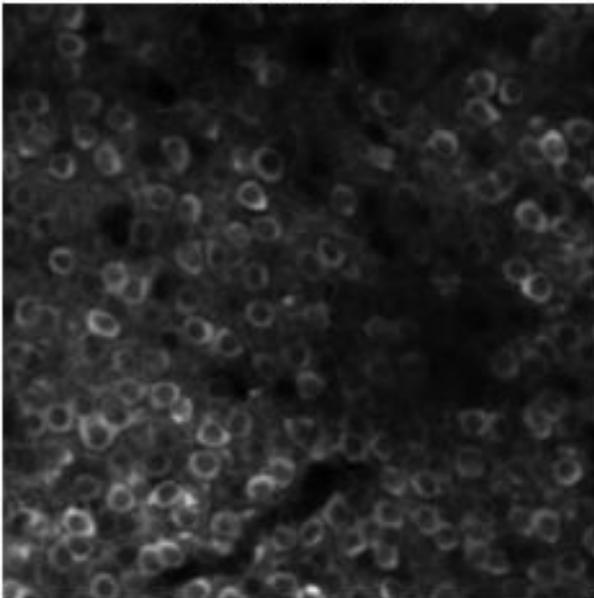
```
[8]: cellpose.models.MODEL_NAMES
```

```
[8]: ['cyto3',
      'nuclei',
      'cyto2_cp3',
      'tissuenet_cp3',
      'livecell_cp3',
      'yeast_PhC_cp3',
      'yeast_BF_cp3',
      'bact_phase_cp3',
      'bact_fluor_cp3',
      'deepbacs_cp3',
      'cyto2',
      'cyto',
      'transformer_cp3',
      'neurips_cellpose_default',
      'neurips_cellpose_transformer',
      'neurips_grayscale_cyto2']
```

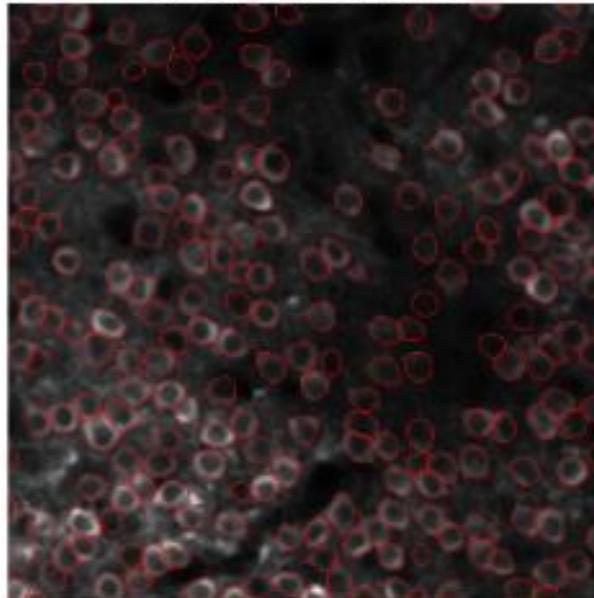
Cellpose in Python

```
[4]: channels = [0,0] # This means we are processing single-channel greyscale images.  
masks, flows, styles, diams = model.eval(image, diameter=None, channels=channels)
```

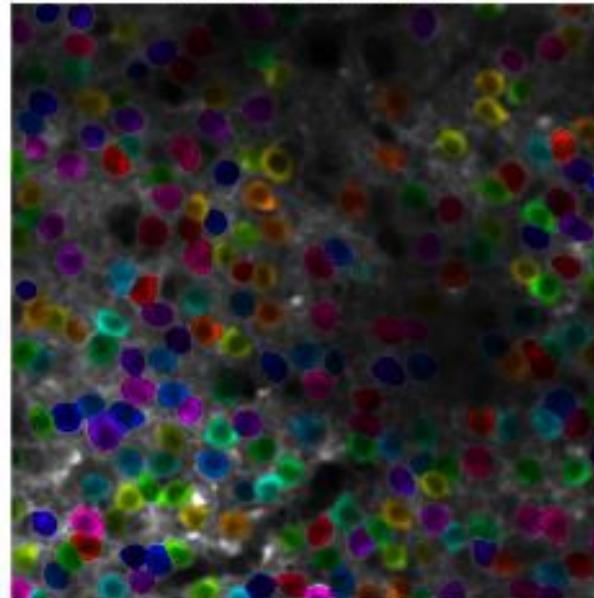
original image



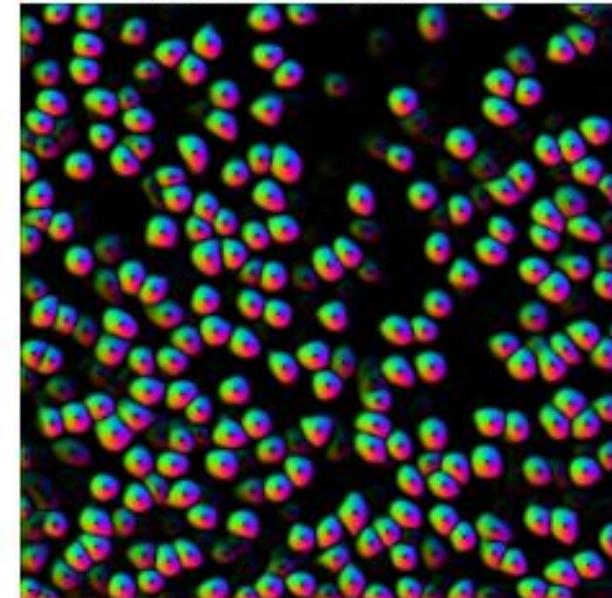
predicted outlines



predicted masks

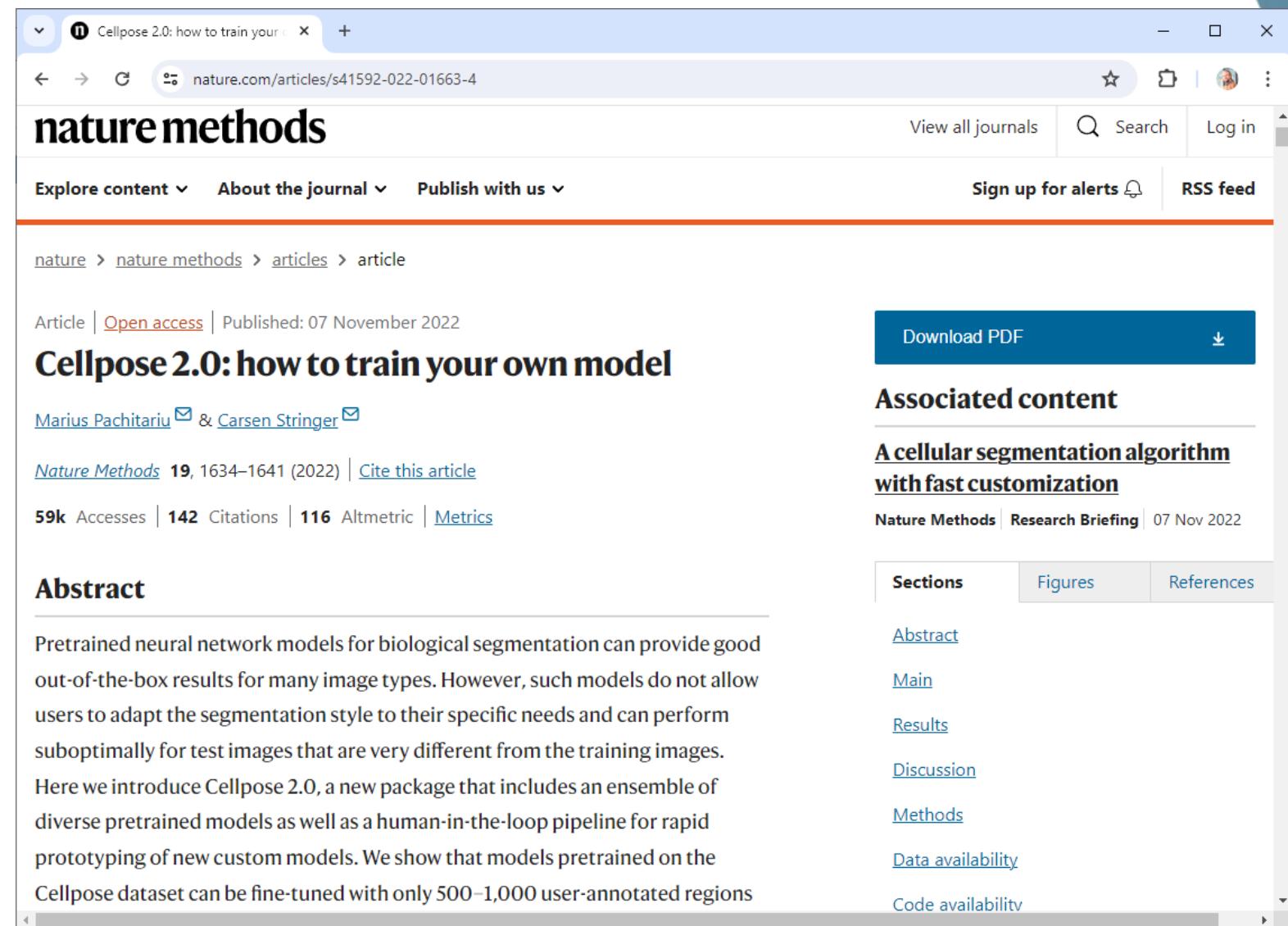


predicted cell pose



Cellpose 2

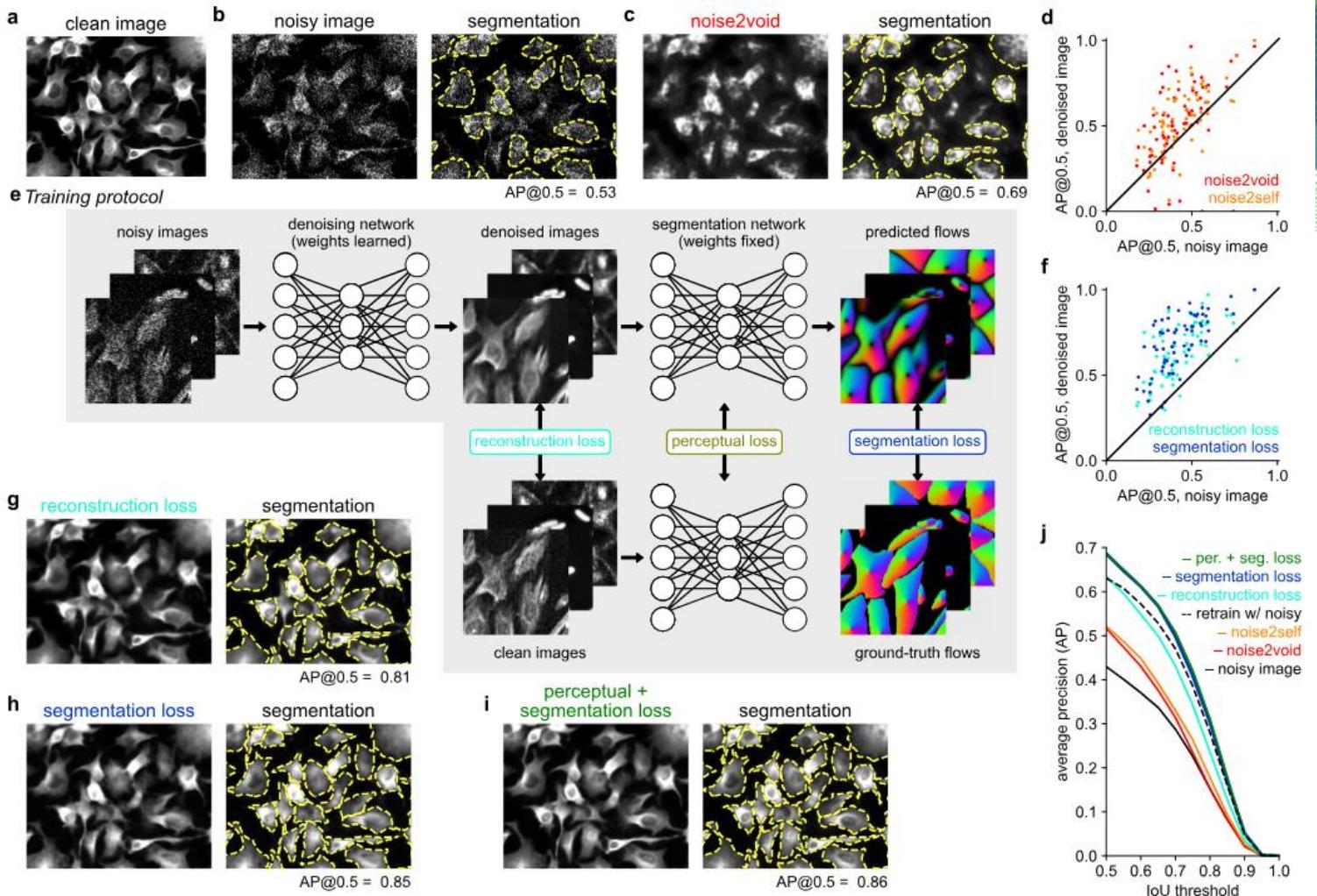
Train your own model



The screenshot shows a web browser displaying the article "Cellpose 2.0: how to train your own model" from the journal *nature methods*. The page includes the header with navigation links like "Explore content", "About the journal", "Publish with us", "Sign up for alerts", and "RSS feed". Below the header, the breadcrumb navigation shows "nature > nature methods > articles > article". The main content area features the title "Cellpose 2.0: how to train your own model" by Marius Pachitariu and Carsen Stringer, published in *Nature Methods* 19, 1634–1641 (2022). It also displays metrics: 59k Accesses, 142 Citations, 116 Altmetric, and Metrics. To the right, there is a "Download PDF" button and a sidebar titled "Associated content" which lists a "Research Briefing" titled "A cellular segmentation algorithm with fast customization". A vertical navigation bar on the right side lists sections: Abstract, Main, Results, Discussion, Methods, Data availability, and Code availability.

Cellpose 3

Cellpose 3 includes denoising and deblurring, to improve image segmentation quality



Cellpose

State-of-the-art despite the rise of transformers

 **bioRxiv**
THE PREPRINT SERVER FOR BIOLOGY

New Results Follow this preprint

Transformers do not outperform Cellpose

Carsten Stringer, Marius Pachitariu
[doi: https://doi.org/10.1101/2024.04.06.587952](https://doi.org/10.1101/2024.04.06.587952)

This article is a preprint and has not been certified by peer review [what does this mean?].



[Abstract](#) [Info/History](#) [Metrics](#) [Preview PDF](#)

Abstract

In a recent publication, Ma et al (2024) claim that a transformer-based cellular segmentation method called Mediar - which won a Neurips challenge - outperforms Cellpose (0.897 vs 0.543 median F1 score). Here we show that this result was obtained by artificially impairing Cellpose in multiple ways. When we removed these impairments, Cellpose outperformed Mediar (0.861 vs 0.826 median F1 score on the updated test set). To further investigate the performance of transformers for cellular segmentation, we replaced the Cellpose backbone with a transformer. The transformer-Cellpose model also did not outperform the standard Cellpose (0.848 median F1 test score). Our results suggest that transformers do not advance the state-of-the-art in cellular segmentation.

Competing Interest Statement

The authors have declared no competing interest.

Copyright The copyright holder for this preprint is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY-NC 4.0 International license.

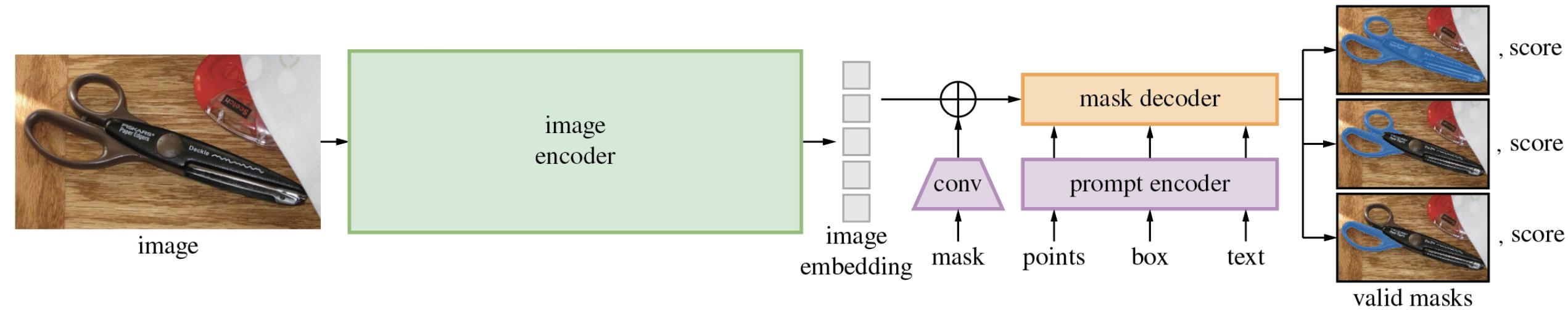
Segment Anything Model

New approach to DL-based image segmentation involving prompts



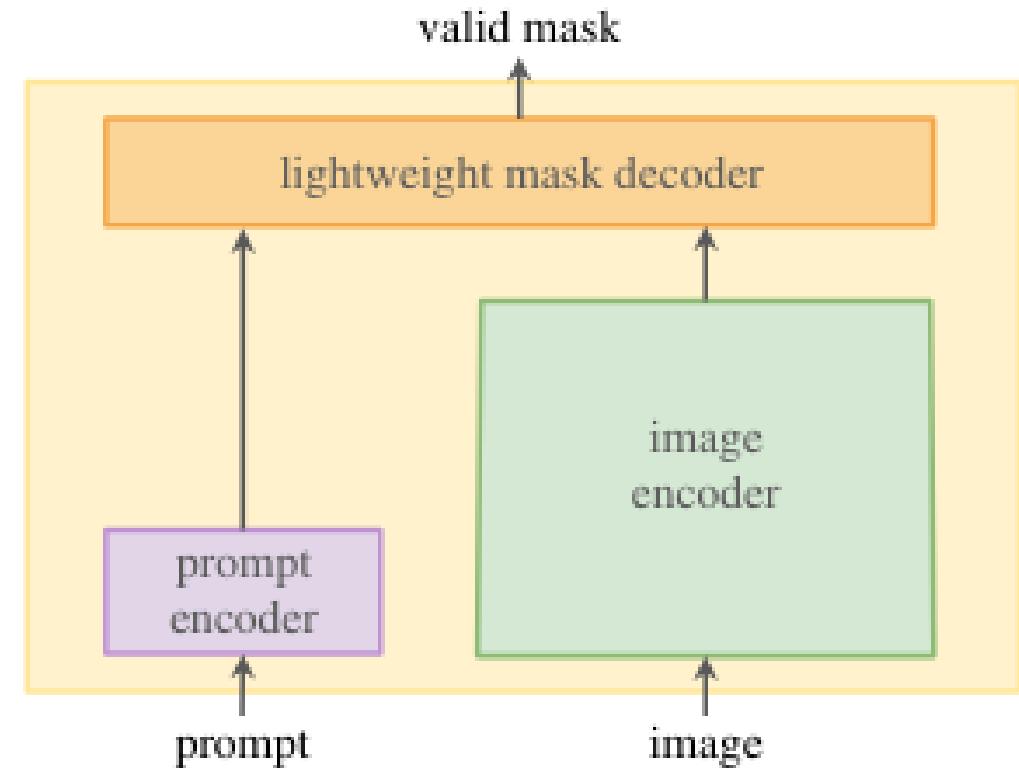
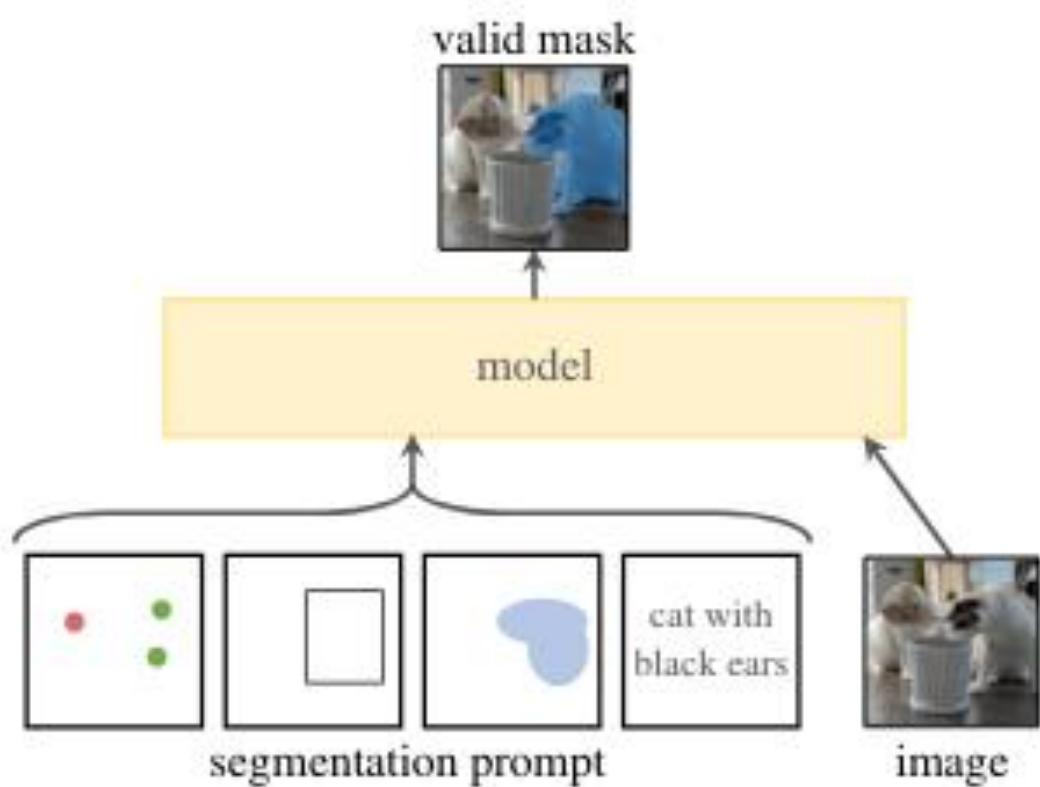
Segment Anything Model

New approach to DL-based image segmentation involving prompts



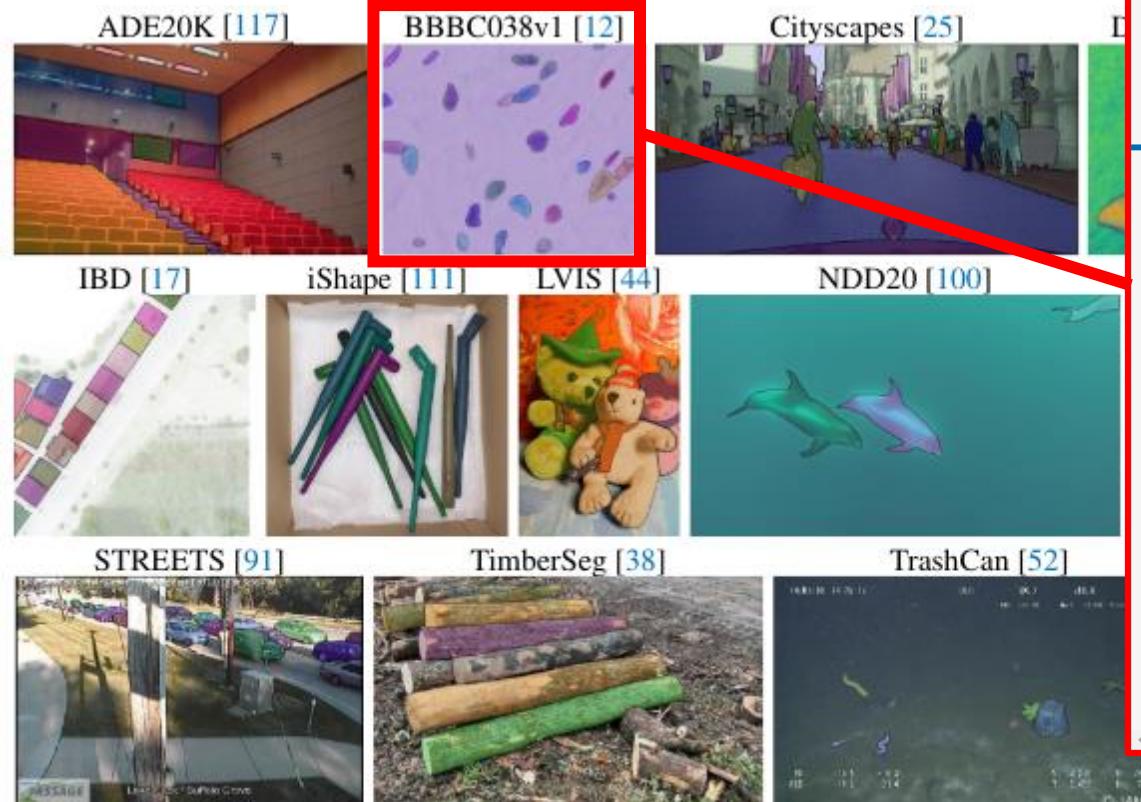
Segment Anything Model

New approach to DL-based image segmentation involving prompts



Segment Anything Model

Trained on mostly natural images



Kaggle 2018 Data Science Bowl

BROAD INSTITUTE

Broad Bioimage Benchmark Collection
Annotated biological image sets for testing and validation

Introduction Image Sets Benchmarking Contribute

LEGEND: KINDS OF GROUND TRUTH

- C COUNTS
- F FOREGROUND / BACKGROUND
- O OUTLINES OF OBJECTS
- B BIOLOGICAL LABELS
- L LOCATION
- U BOUNDING BOXES

HOME / KAGGLE 2018 DATA SCIENCE BOWL

Accession number [BBBC038](#) · Version 1

Example images

Figure 8: Samples from the 23 diverse segmentation datasets used to evaluate SAM's zero-shot transfer capabilities.

Segment Anything for Microscopy

Popping up napari plugins, some within 24h after SAM was published

The image shows four browser tabs side-by-side, each displaying a GitHub repository page for a napari plugin:

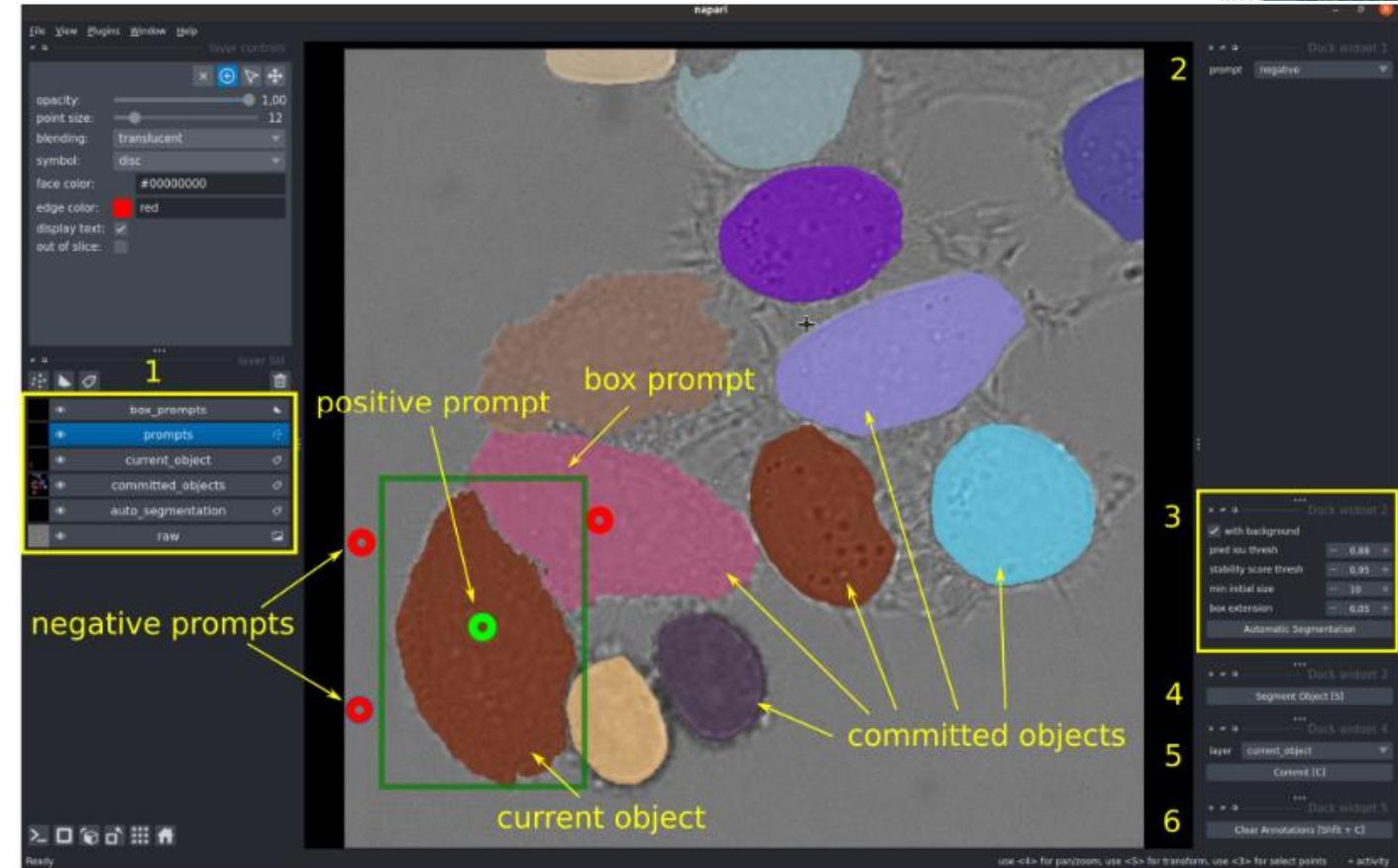
- napari-SAM4IS**: A plugin for instance and semantic segmentation using the Segment Anything Model (SAM). It includes a screenshot of a napari interface showing a heatmap over an image.
- napari-segmentanything**: A napari plugin of Segment Anything Model (SAM), featuring a video thumbnail of a napari-plugin anything-small.mp4.
- Segment Anything Model (SAM) in Napari**: A plugin that integrates the Segment Anything Model into Napari, supporting both 2D and 3D images.
- Segment Anything for Microscopy**: Tools for segmentation and tracking in microscopy build on top of Segment Anything, including a section for interactive cell tracking.

Each repository page includes standard GitHub metrics like license (Apache-2.0), pypi version, python versions, tests status, and code coverage.

Segment Anything for Microscopy

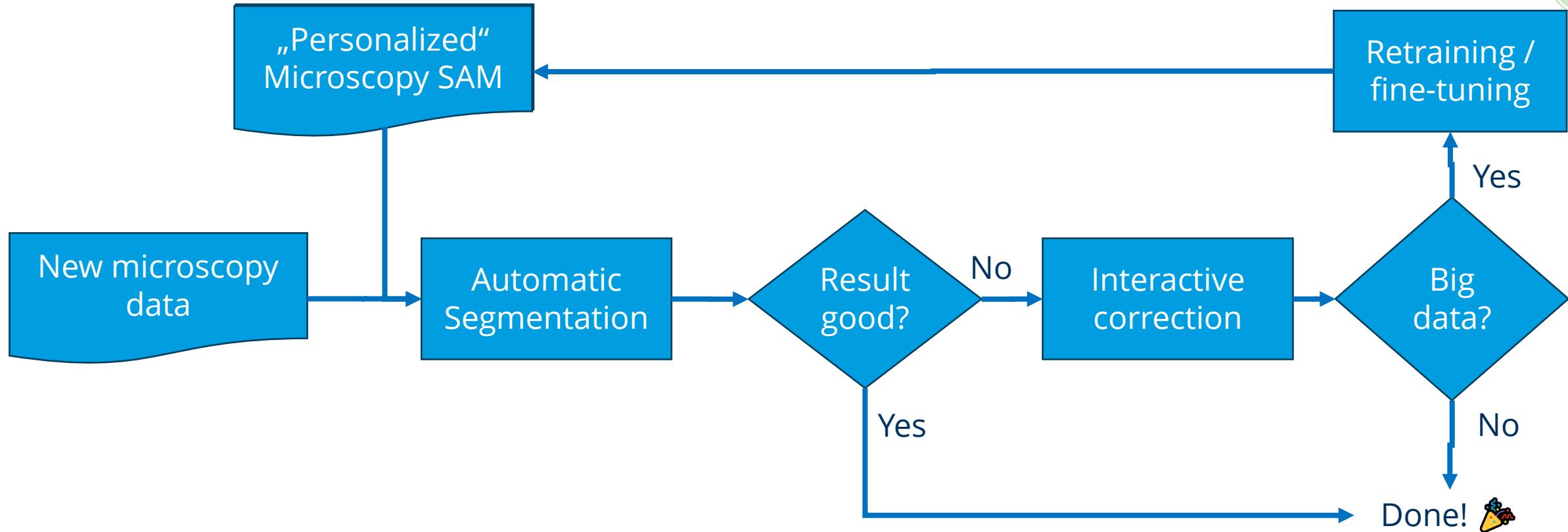
Downsides:

- Original code did not contain the procedure for iterative training
- Instance segmentation not ideal (watershed-implementation added in micro-sam)
- Fine-tuning for microscopy data necessary

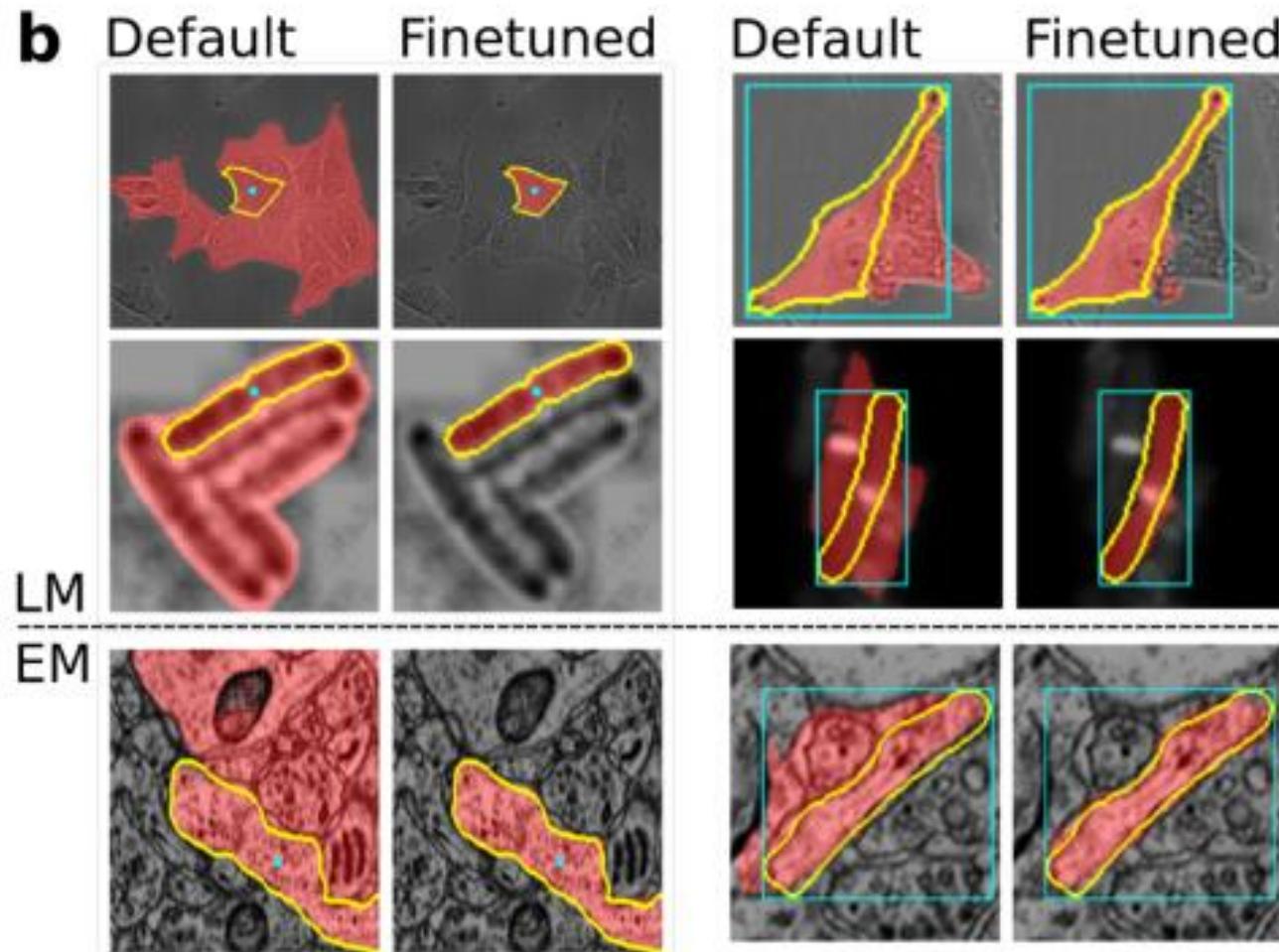


Segment Anything for Microscopy

Real-world scenarios: human-in-the-loop



Segment Anything for Microscopy



Summary: Deep Learning for Bio-image Analysis

- [Convolutional] Neural Networks is a decade old technology that enabled breakthroughs recently.
 - Image Denoising
 - Image Segmentation
- Common scheme: Smart algorithms for processing input/output of neural networks + standard NN architectures
 - ~~Image in, instance segmentation out~~
- Training these models is
 - computationally expensive,
 - needs large amounts of training data (~~single images~~),
 - requires a certain level of [python] expertise
- If Voronoi-Otsu-Labeling does the job, don't dive into deep learning!

Exercises

Robert Haase

Funded by



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.

Exercises

Make noise2void, stardist, cellpose [and micro-sam] work.
Hint: This may screw up your conda environment.

In case of weird errors:

1. Don't panic
2. Recreate you environment
3. Install the thing you want to use, preferably using mamba/conda.

Creating conda environments is not a big deal, just like grabbing a spoon before eating soup.

The screenshot shows two command prompt windows and a Jupyter Notebook cell. The left command prompt shows the uninstallation of several packages: matplotlib, xarray, google-auth-oauthlib, csbdeep, bioimageio.s, wrapt, typing-extensions, and six. The right command prompt shows the uninstallation of matplotlib-3.8.4. The Jupyter Notebook cell [1] contains the code: `[1]: import n2v
from skimage.io import imread`. It then shows a `ValueError` from `skimage._init_.py`: `ValueError: numpy.ndarray size changed, may indicate binary incompatibility. Expected 96 from C header, got 80 from PyObject`.

```
Command Prompt - pip install n2v tensorflow==2.5.1
matplotlib, xarray, google-auth-oauthlib, csbdeep, bioimageio.s
Attempting uninstall: wrapt
  Found existing installation: wrapt 1.16.0
  Uninstalling wrapt-1.16.0:
    Successfully uninstalled wrapt-1.16.0
Attempting uninstall: typing-extensions
  Found existing installation: typing_extensions 4.11.0
  Uninstalling typing_extensions-4.11.0:
    Successfully uninstalled typing_extensions-4.11.0
Attempting uninstall: six
  Found existing installation: six 1.16.0
  Uninstalling six-1.16.0:
    Successfully uninstalled six-1.16.0
Attempting uninst...
[1]: import n2v
from skimage.io import imread

-----
ValueError
Cell In[1], line 2
  1 import n2v
----> 2 from skimage.io import imread

File ~\mambaforge\envs\tea2\lib\site-packages\skimage\_init__.py:122
  118     # We are not importing the rest of the scikit during the build
  119     # process, as it may not be compiled yet
  120 else:
  121     try:
--> 122         from ._shared import geometry
  123         del geometry
  124     except ImportError as e:

File geometry.pyx:1, in init skimage._shared.geometry()

ValueError: numpy.ndarray size changed, may indicate binary incompatibility. Expected 96 from C header,
got 80 from PyObject
```