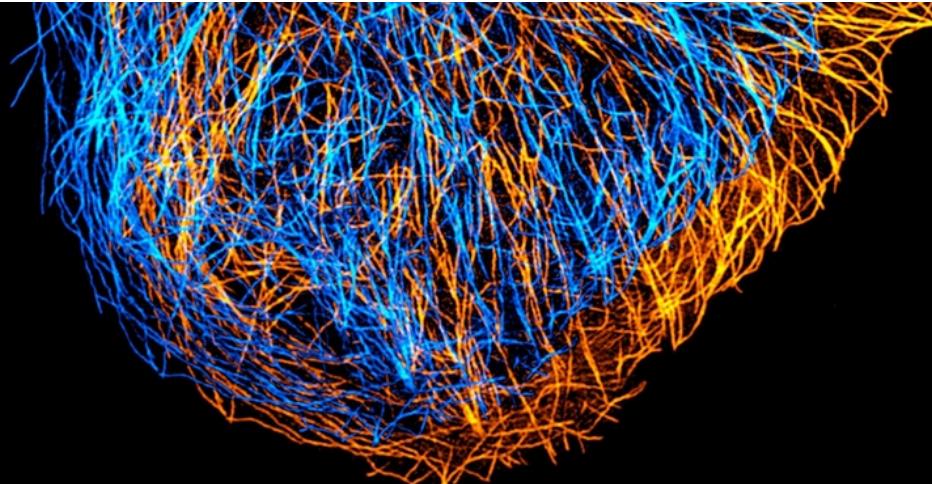


USING DEEP LEARNING FOR MICROSCOPY



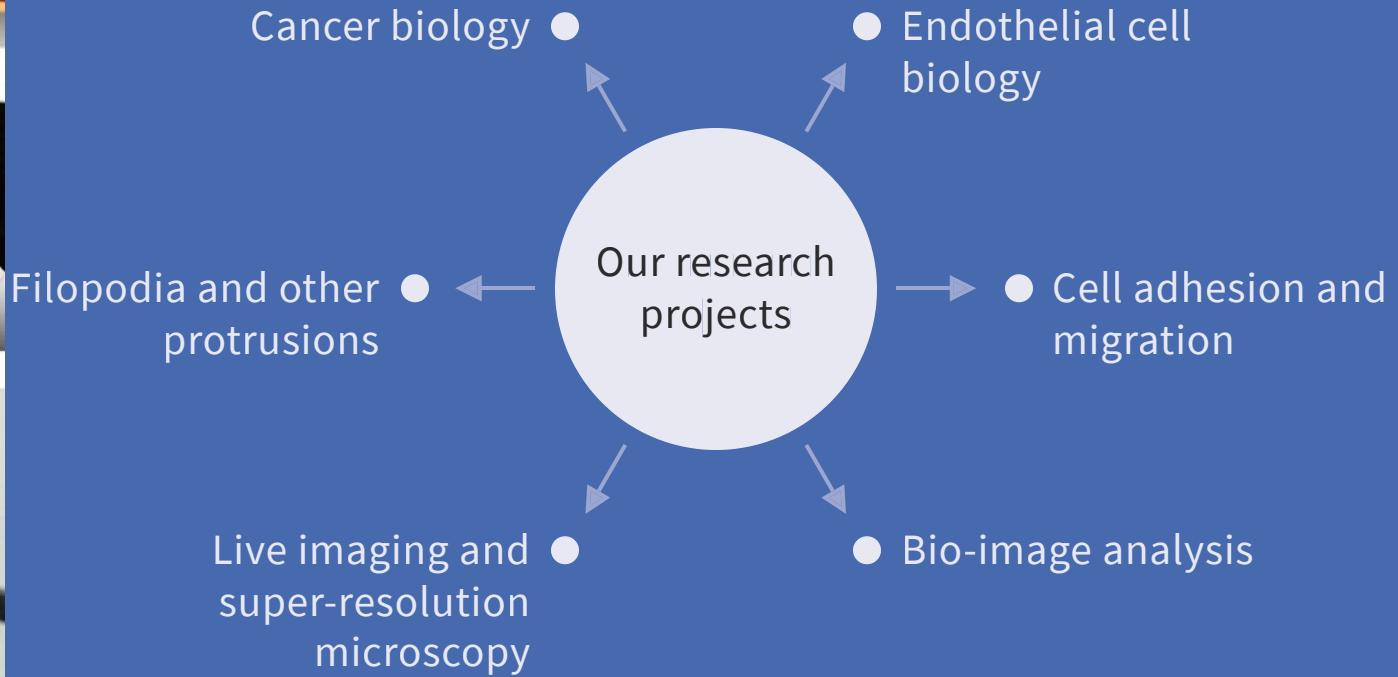
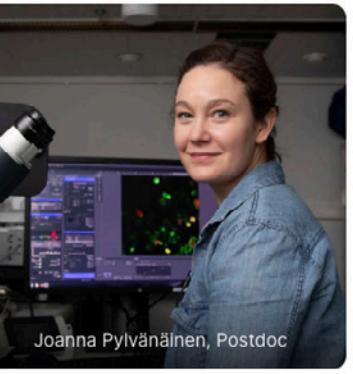
OUTLINE OF THE TALK

- 1 Deep Learning for image analysis what can you use it for?
- 2 Deep Learning how does it work?
- 3 The ZeroCostDL4Mic platform

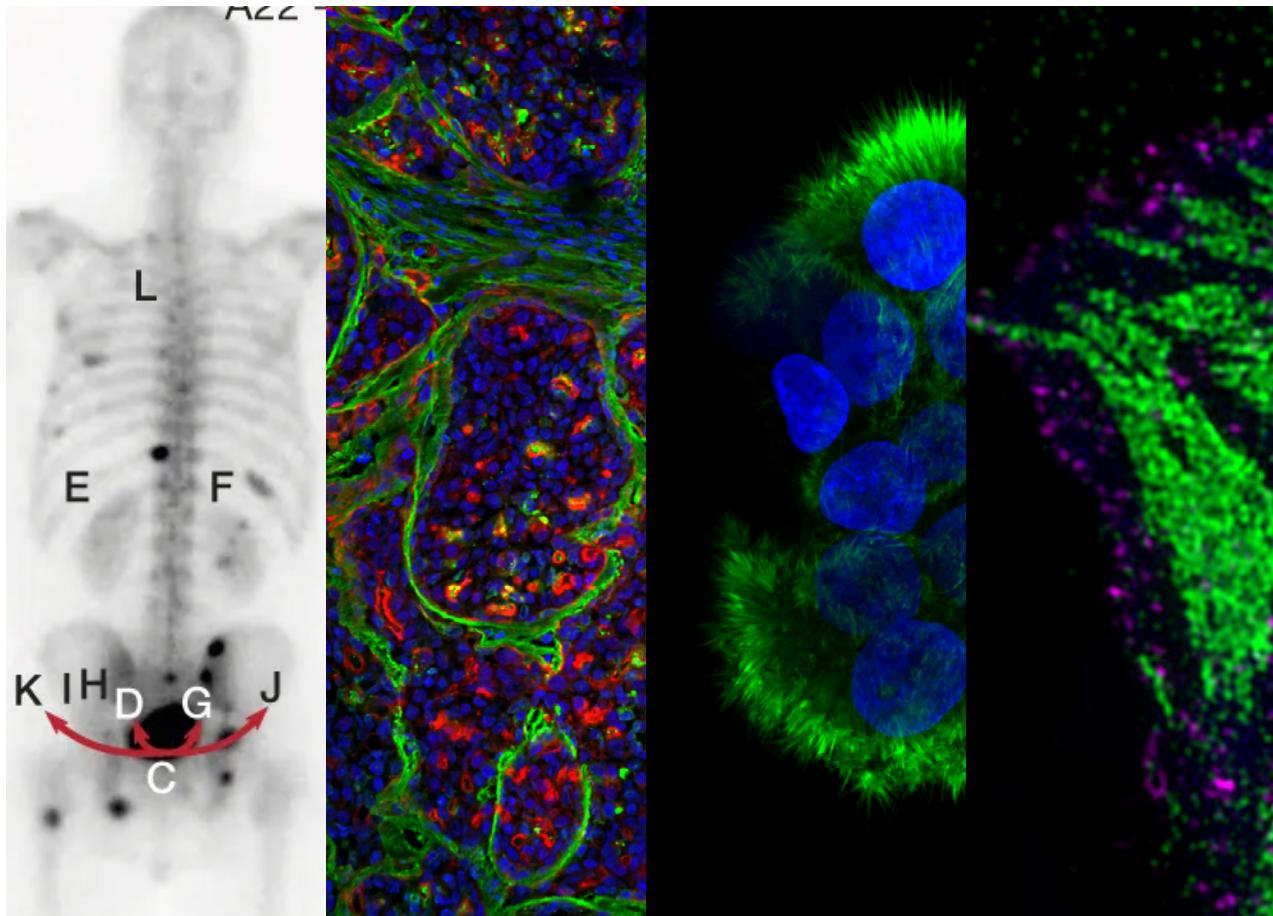
THE LAB

www.cellmig.org

Postdoc position available in my lab, Turku Finland

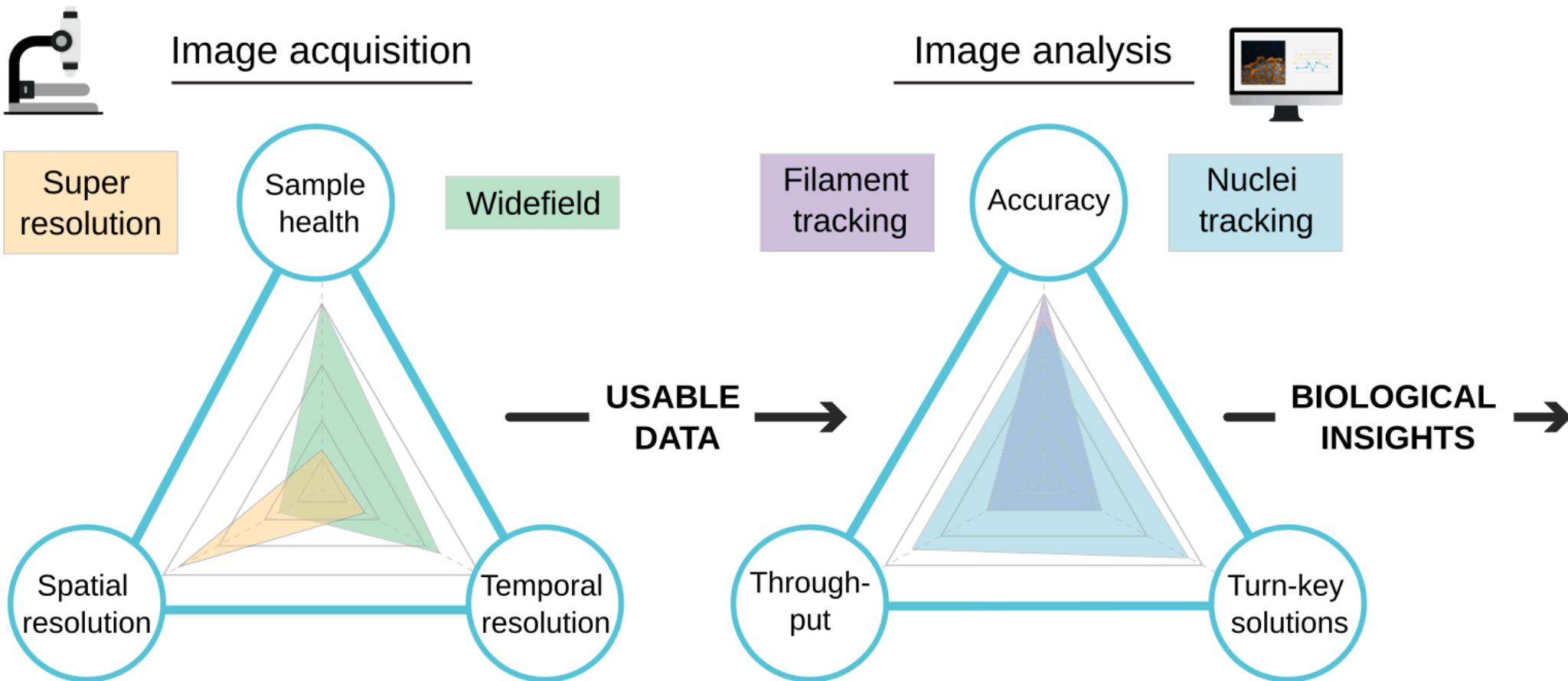


WHY DO WE NEED IMAGE ANALYSIS?

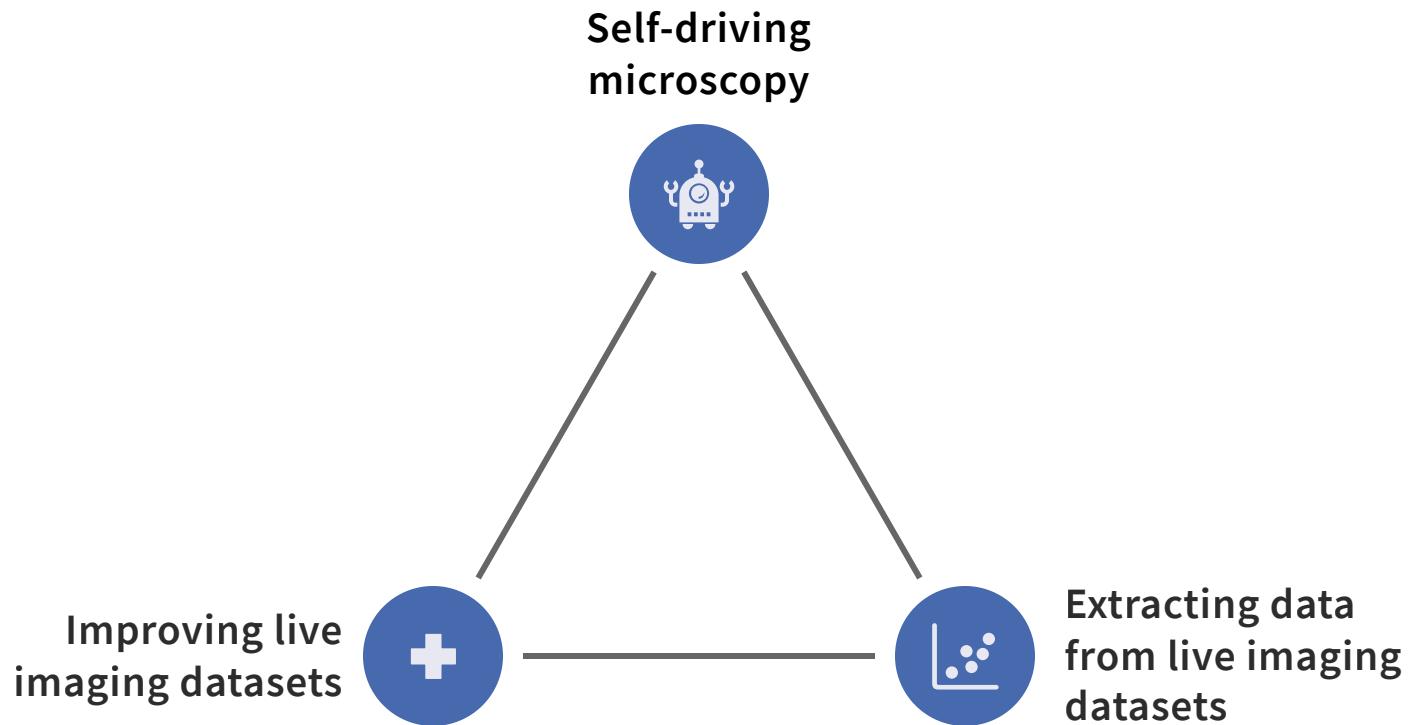


- Images are very rich in information
- The information to be extracted depends on the context
- Extracting this information is not always easy

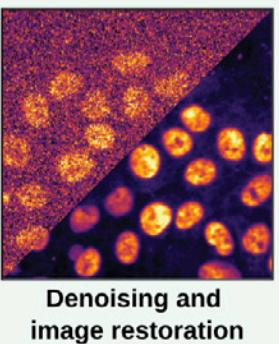
LIVE IMAGING AND IMAGE ANALYSIS



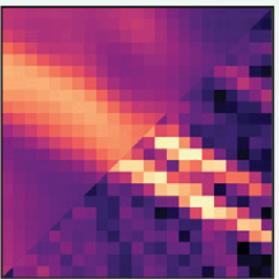
INCREASINGLY, MICROSCOPY IS ENHANCED BY POWERFUL SOFTWARE



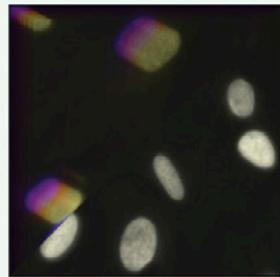
Improvement of live imaging datasets



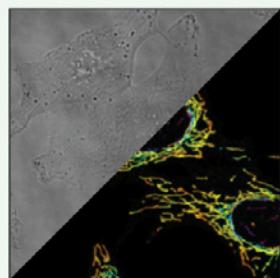
Denoising and
image restoration



Super-resolution

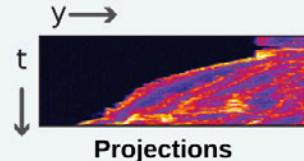


Drift correction

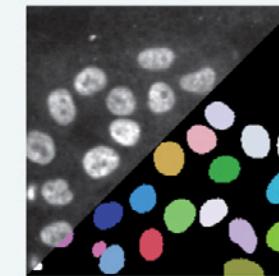


Artificial labeling

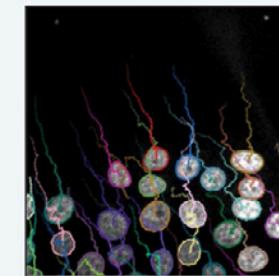
Analysis of live imaging datasets



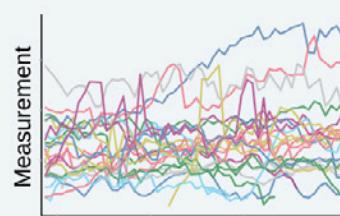
Projections



Segmentation and
object detection



Tracking



Time series analysis

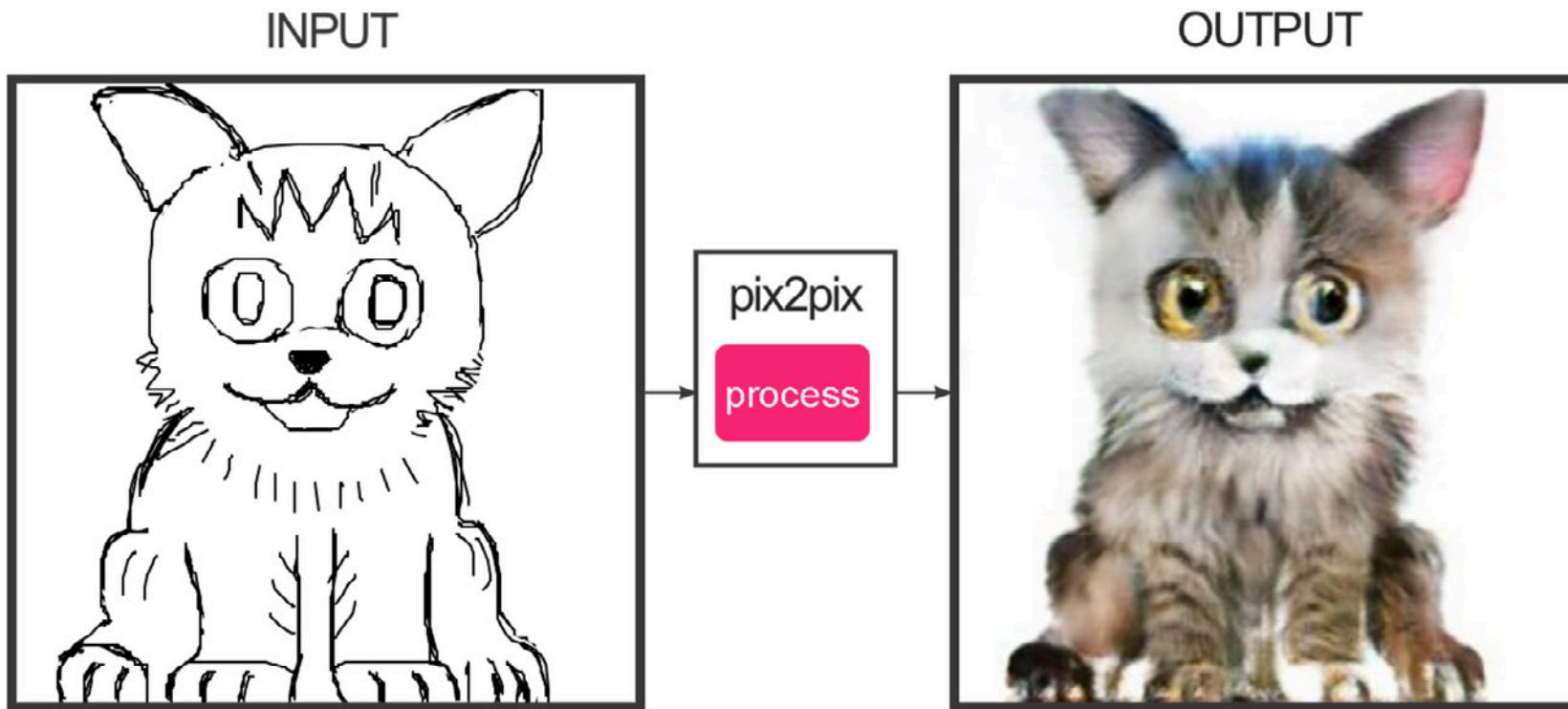
AI - DEEP LEARNING

Teaching software to do a specific task



Video credit: <https://www.tesla.com/autopilotAI>

Artificial intelligence and images



Section 1

DEEP LEARNING FOR IMAGE ANALYSIS WHAT CAN YOU USE IT FOR?

DEEP LEARNING FOR MICROSCOPY WHAT CAN YOU USE IT FOR?

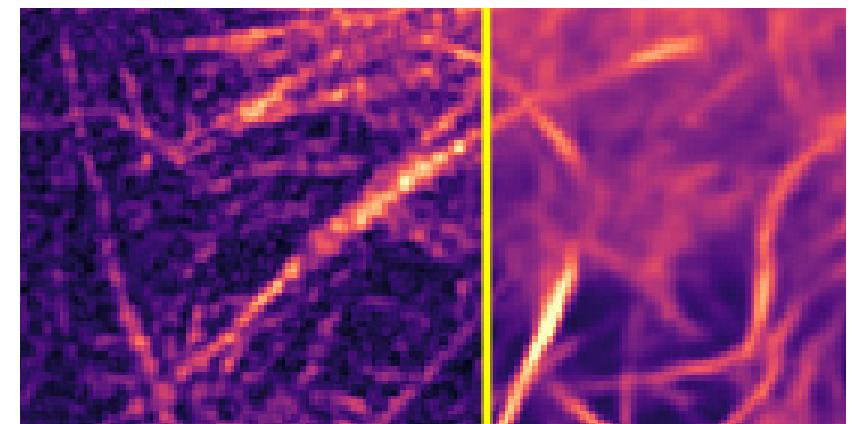
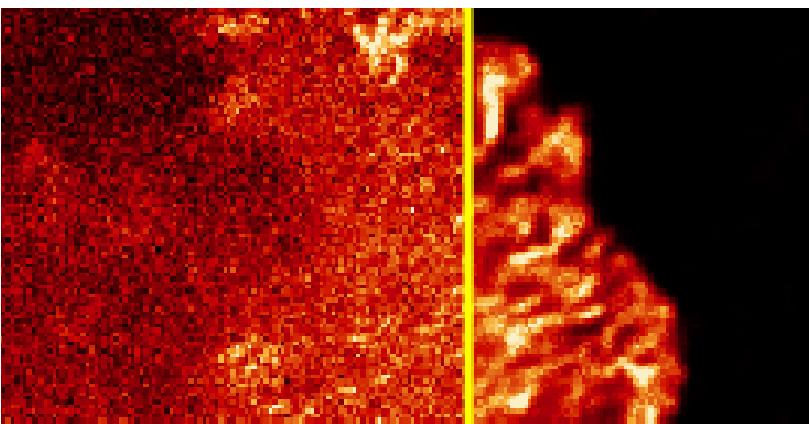
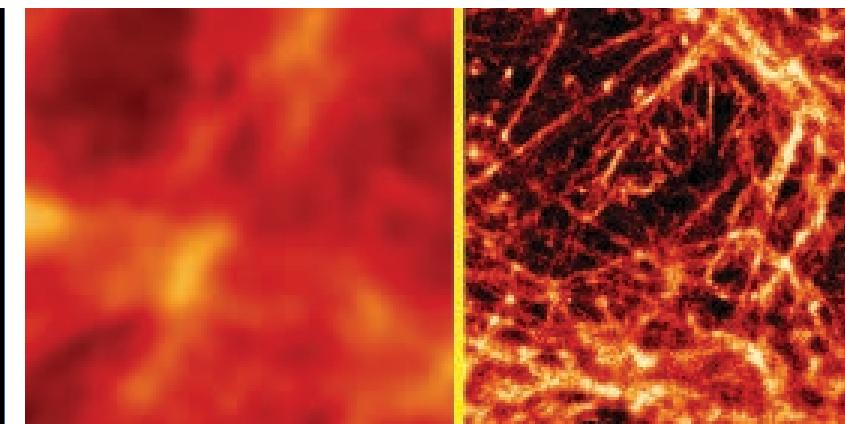


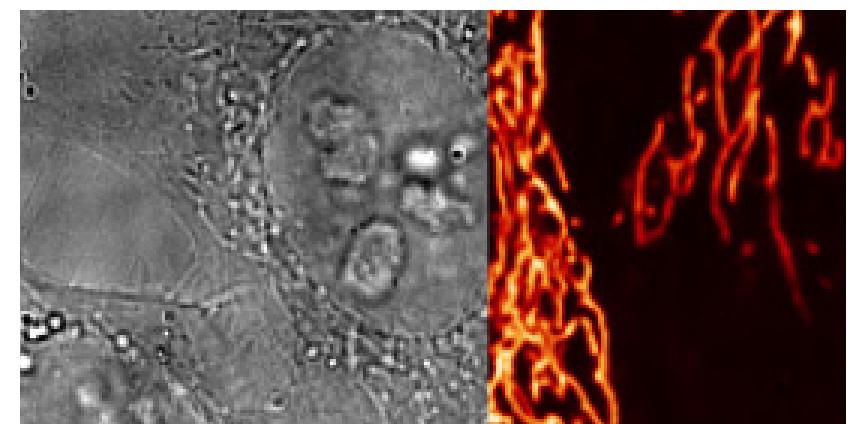
IMAGE RESTORATION



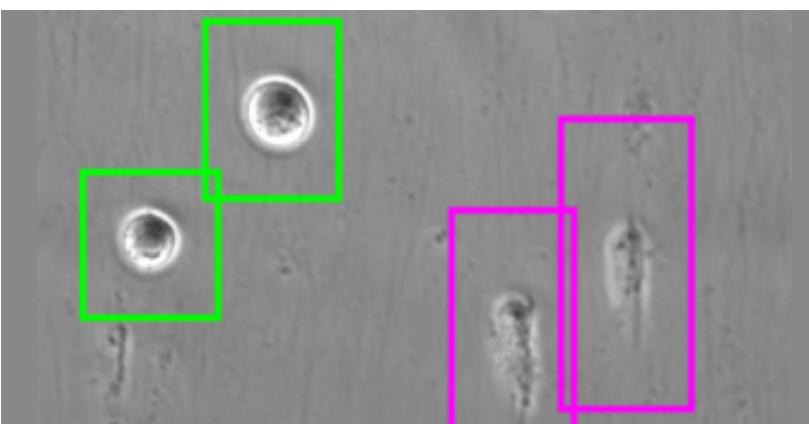
DENOISING



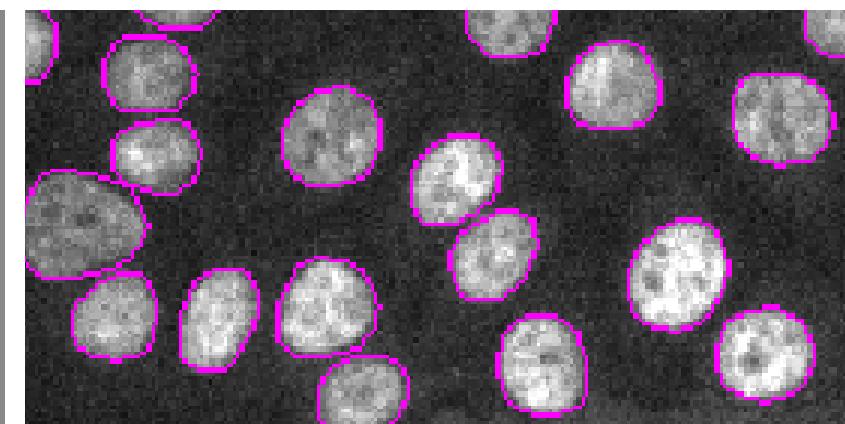
IMPROVEMENT IN RESOLUTION



ARTIFICIAL LABELING



OBJECT DETECTION

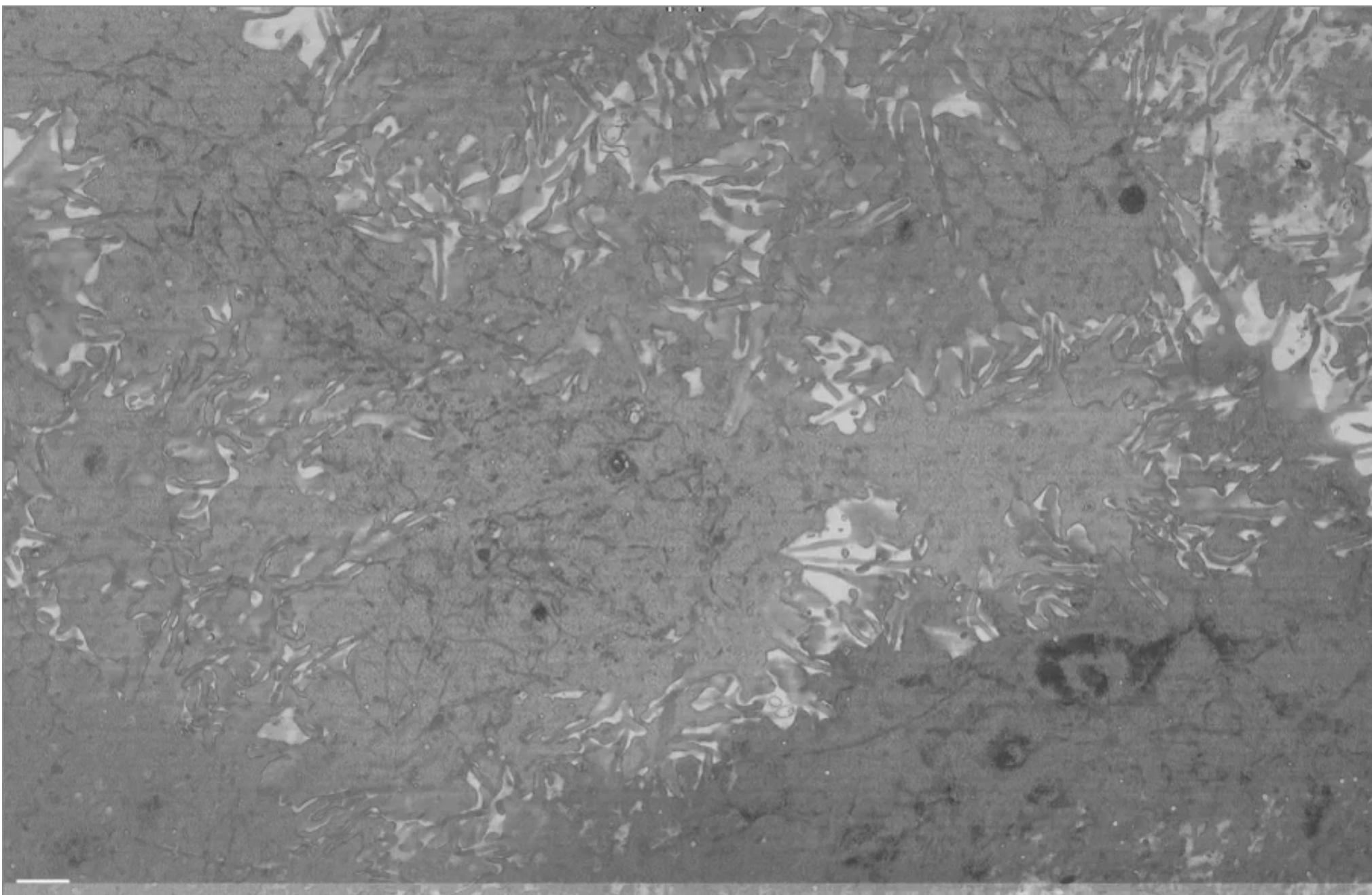


SEGMENTATION

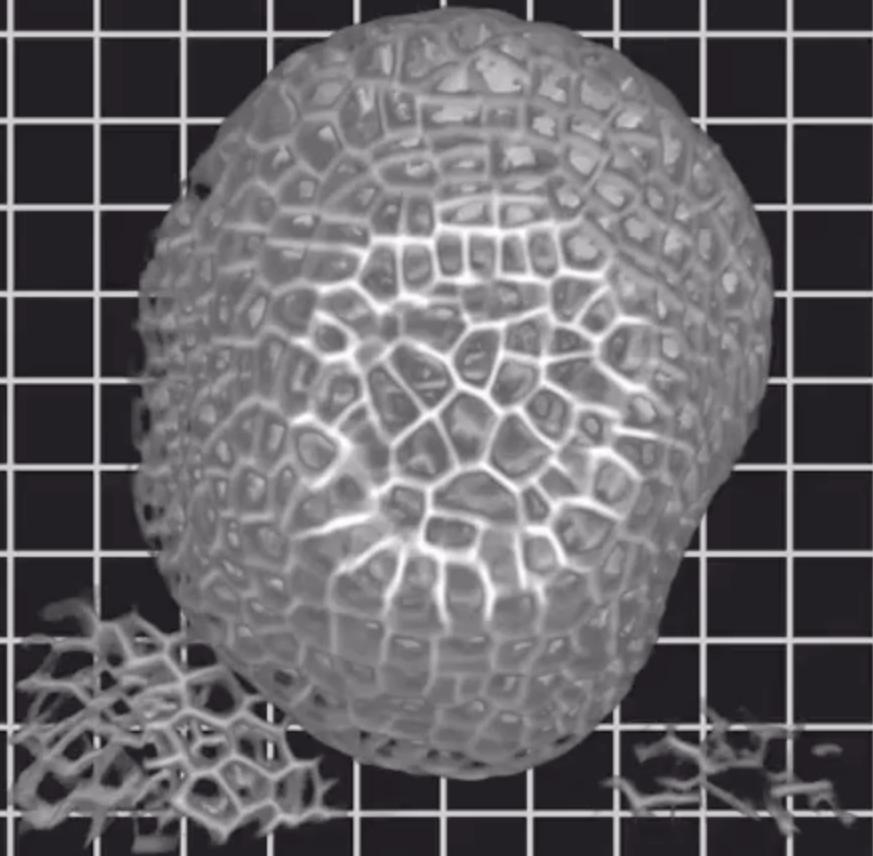
SEGMENTATION

ZeroCostDL4Mic Video #3: Using **3D U-Net** for the **segmentation** of mitochondria from electron microscopy data

ZeroCostDL4Mic: an open platform to use Deep-Learning in Microscopy
Lucas von Chamier, Romain F. Laine *et al.* (2020)



Marie-Catherine, in collaboration with Eija Jokitalo and Helena Vihinen

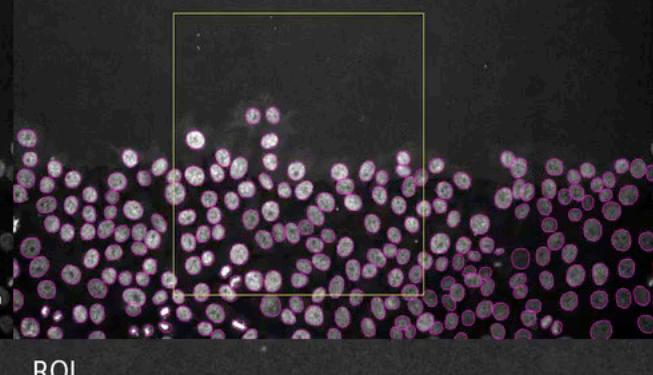
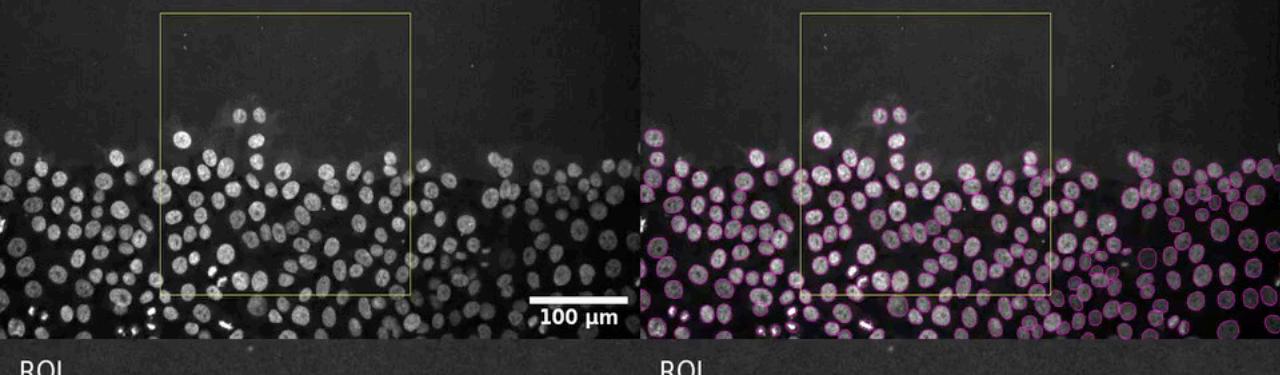


Sir-DNA (input)

00:00 Stardist prediction

Stardist prediction
Local tracks

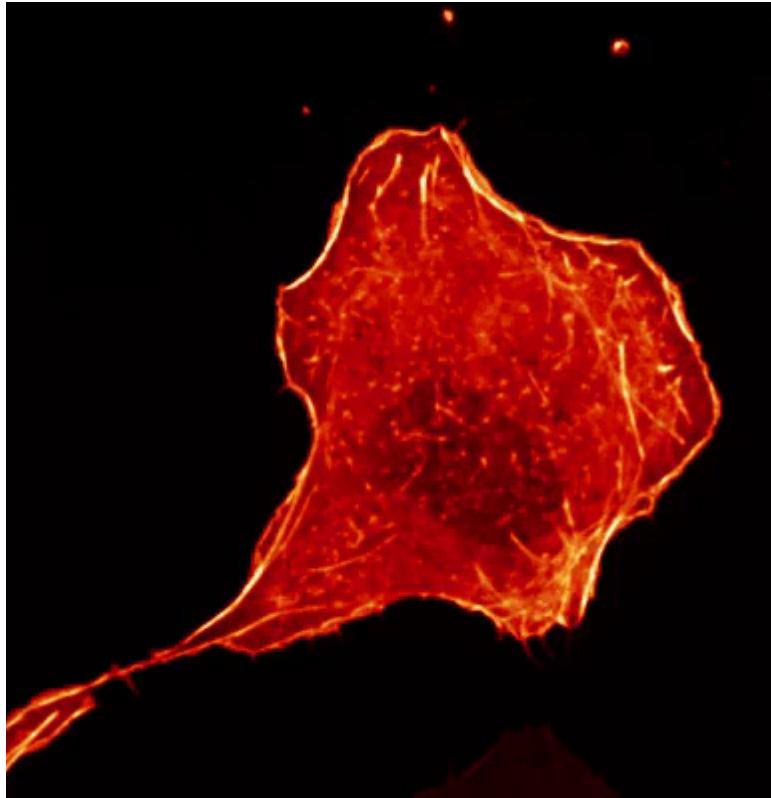
Sir-DNA (input)
Stardist prediction
Local tracks



TRACKMATE

DENOISING

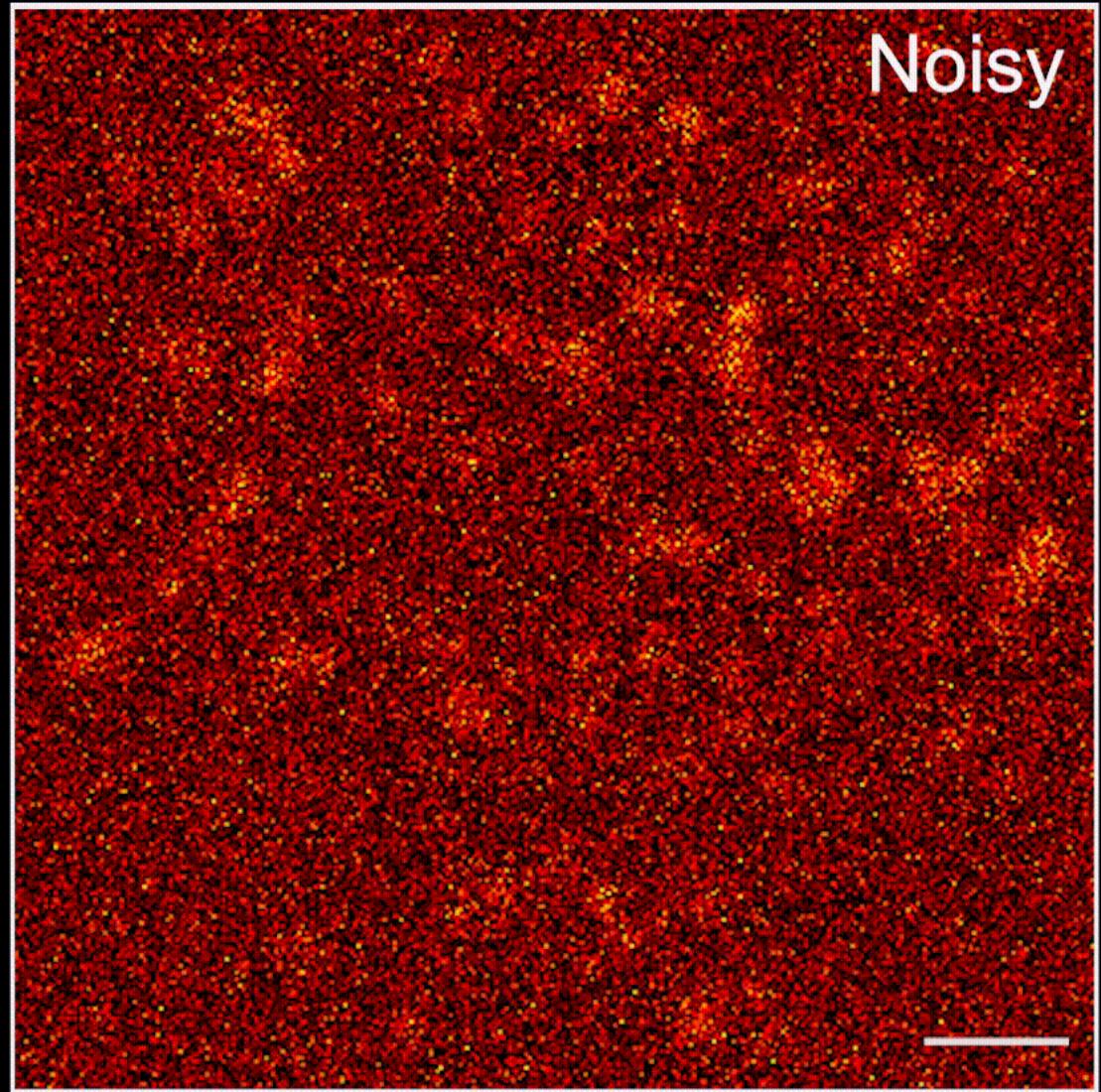
DENOISING IS ESSENTIAL FOR FLUORESCENCE LIVE-CELL IMAGING



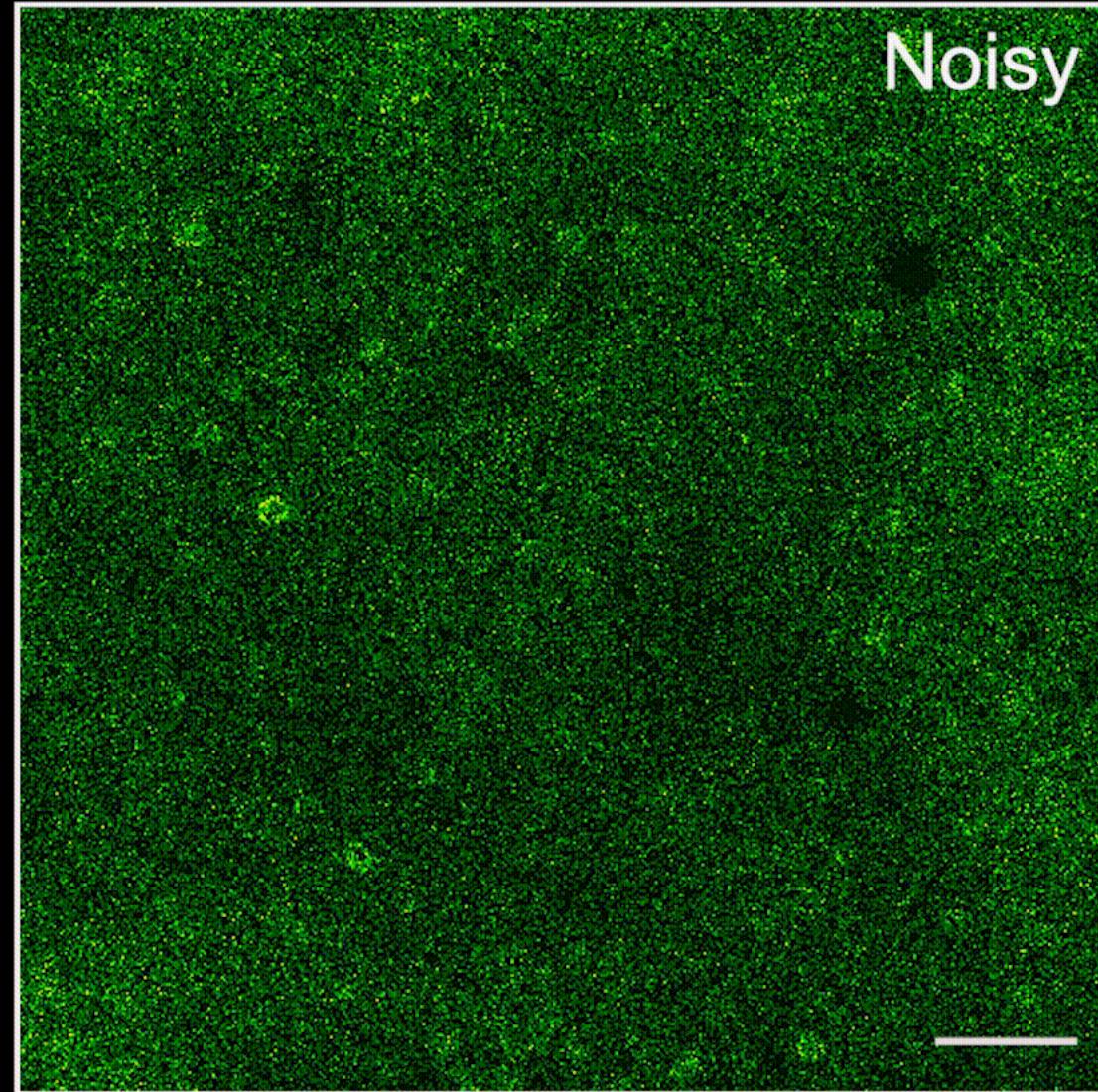
- Lasers are toxic for cells
Low laser power
Higher resolution generally means higher laser power
- Over-expression of molecules leads to artifacts
Low / endogenous expression levels

Noisy images....

STORM



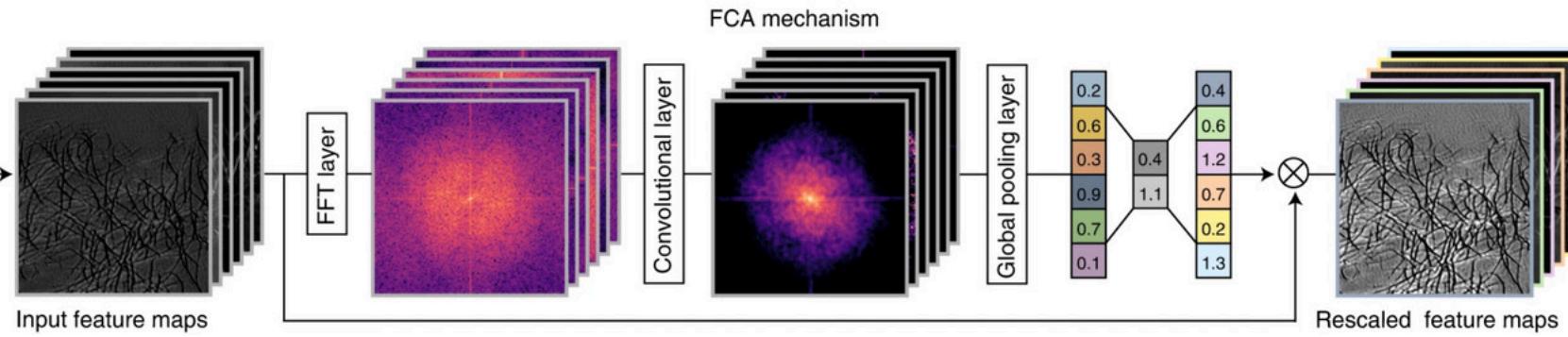
Calcium imaging



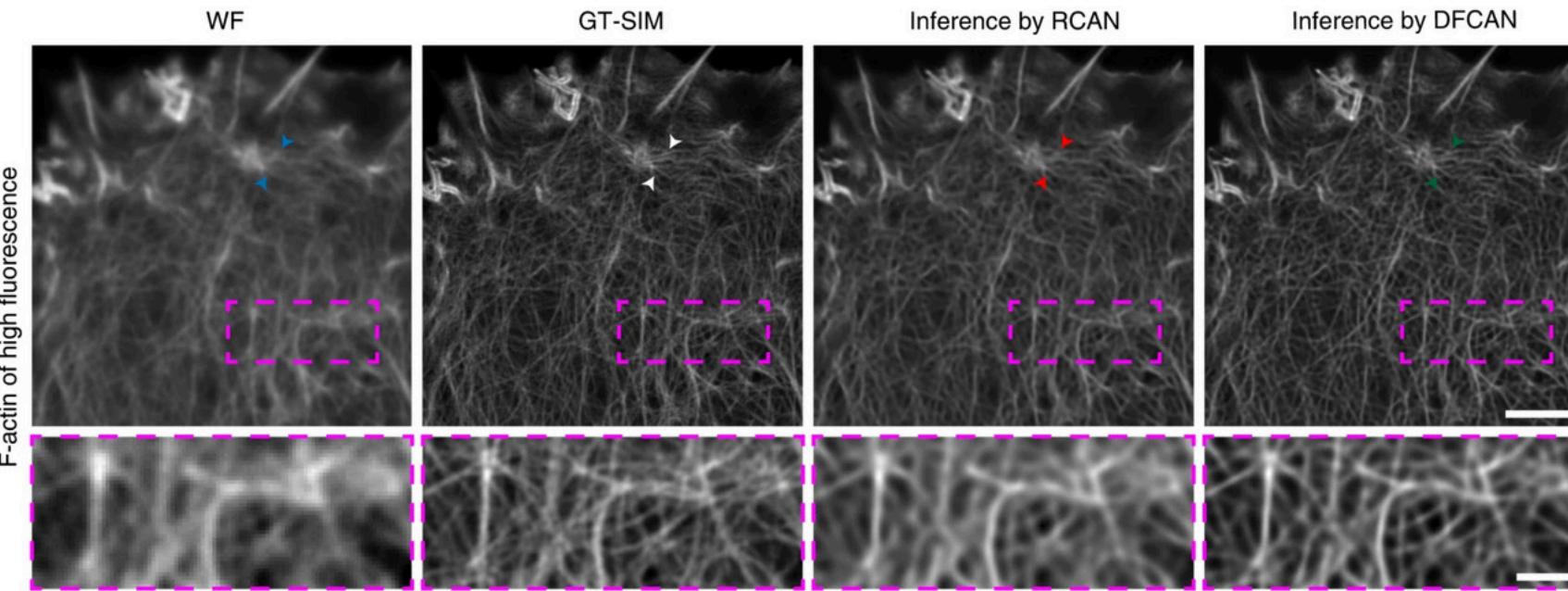
SUPER-RESOLUTION

IMPROVING THE RESOLUTION OF MICROSCOPY VIDEO

Mitochondrial dynamics revealed by SFSRM

a

- Predict SIM images from WF

b

- Predict STED images from confocal

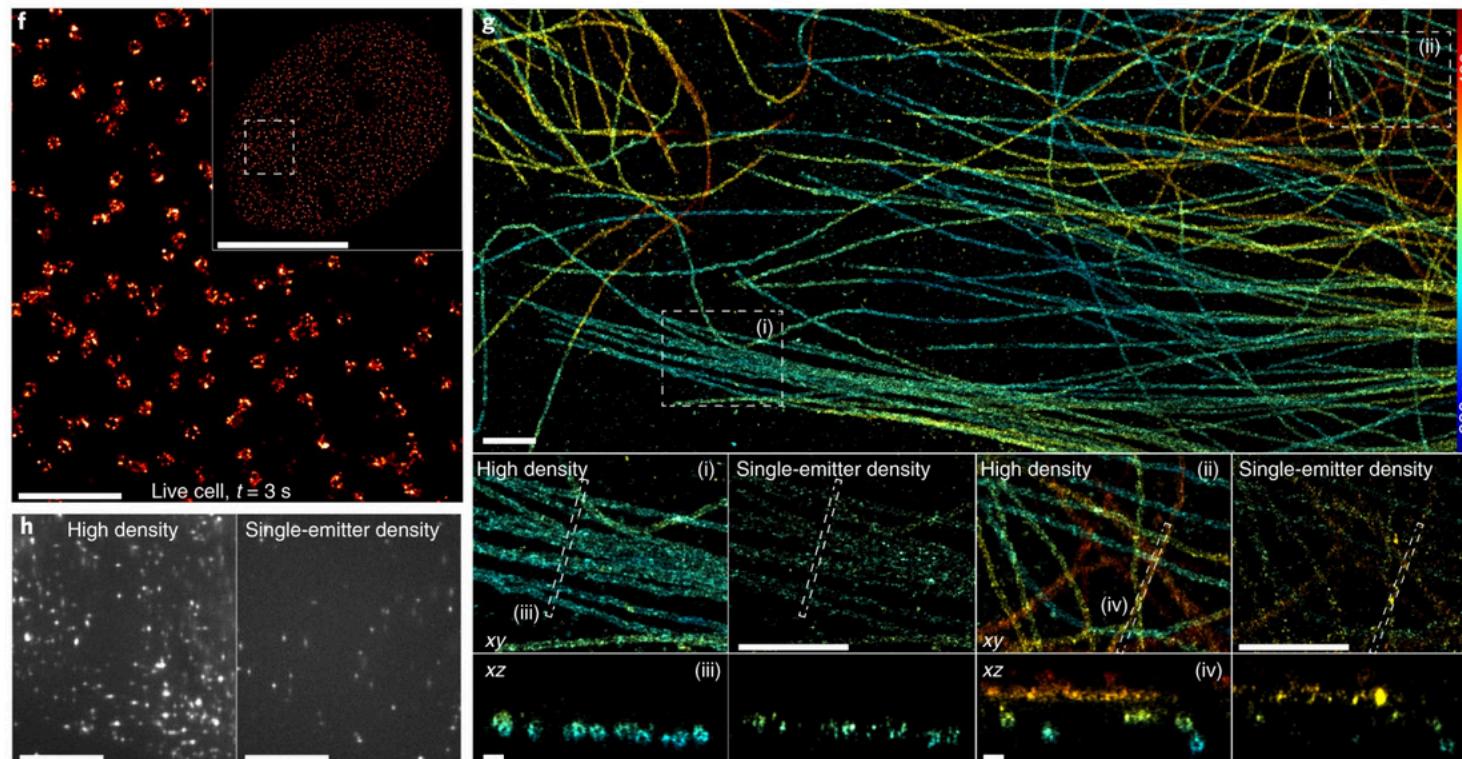
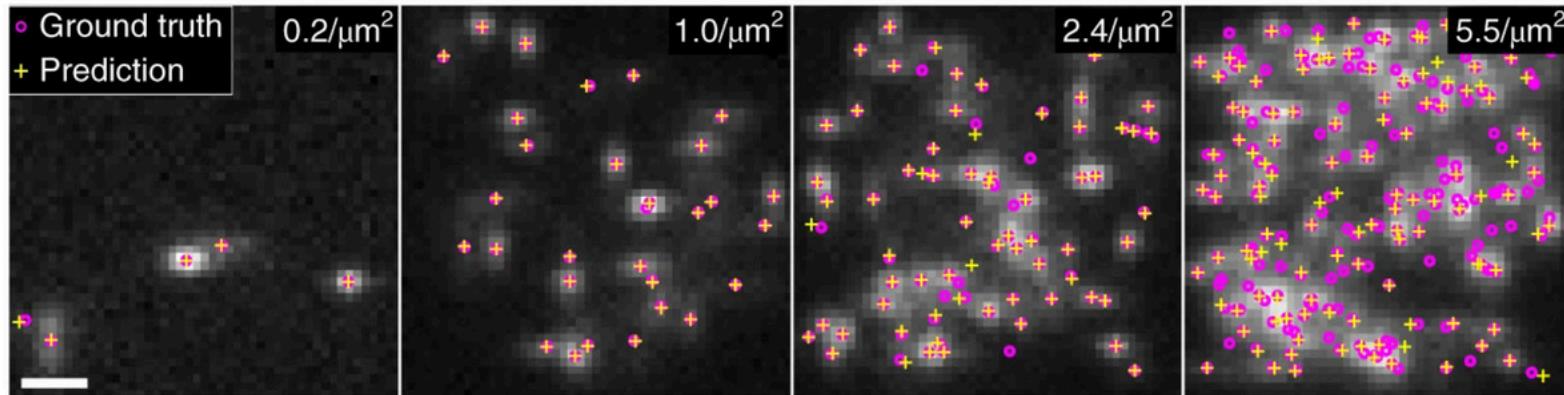
ZeroCostDL4Mic Video #10: Using Deep-STORM to obtain super-resolution microscopy images of the actin cytoskeleton

Labels: Actin (Phalloidin-AlexaFluor 647)

Microscope: N-STORM (100x)

ZeroCostDL4Mic: an open platform to use Deep-Learning in Microscopy

Lucas von Chamier, Romain F. Laine *et al.* (2020)

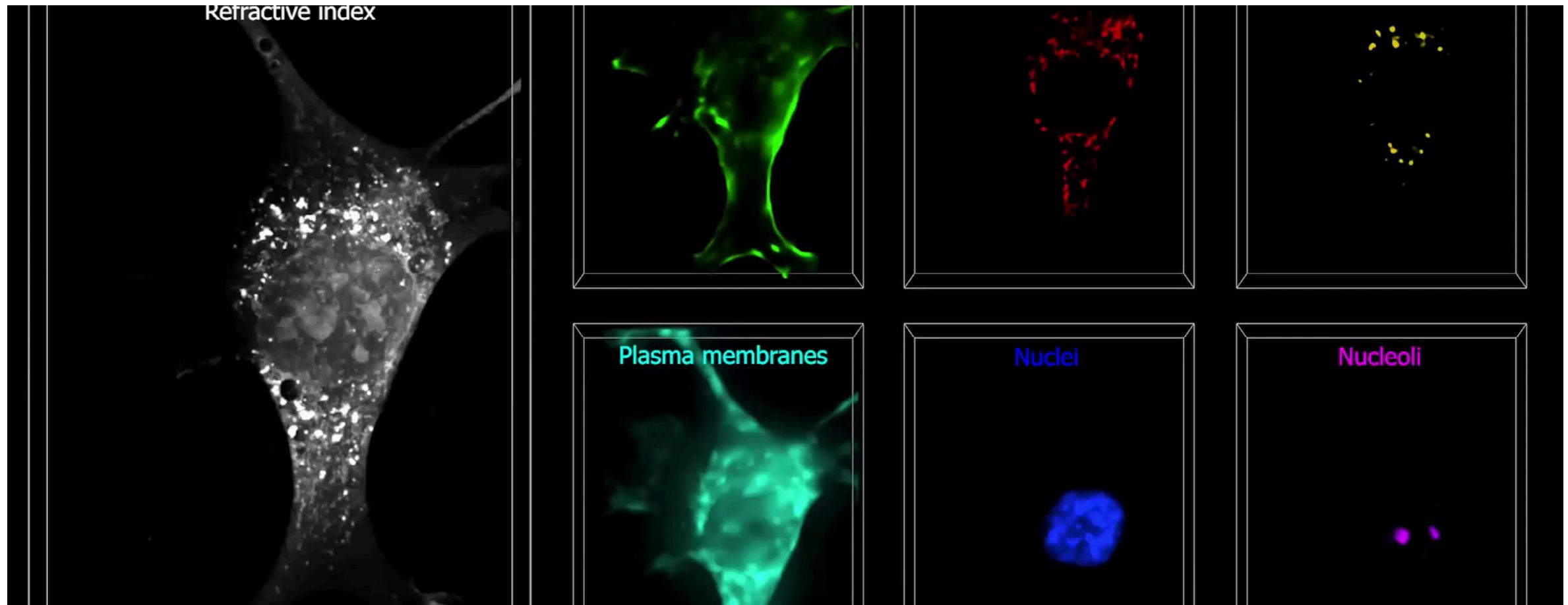


Deep learning enables fast and dense single-molecule localization with high accuracy

Artur Speiser, Lucas-Raphael Müller, Philipp Hoess, Ulf Matti, Christopher J. Obara, Wesley R. Legant, Anna Kreshuk, Jakob H. Macke, Jonas Ries & Srinivas C. Turaga

AFTIFICIAL LABELLING

ARTIFICIAL LABELING CAN PROVIDE ADDITIONAL CONTEXT



Jo et al., 2021

OTHER

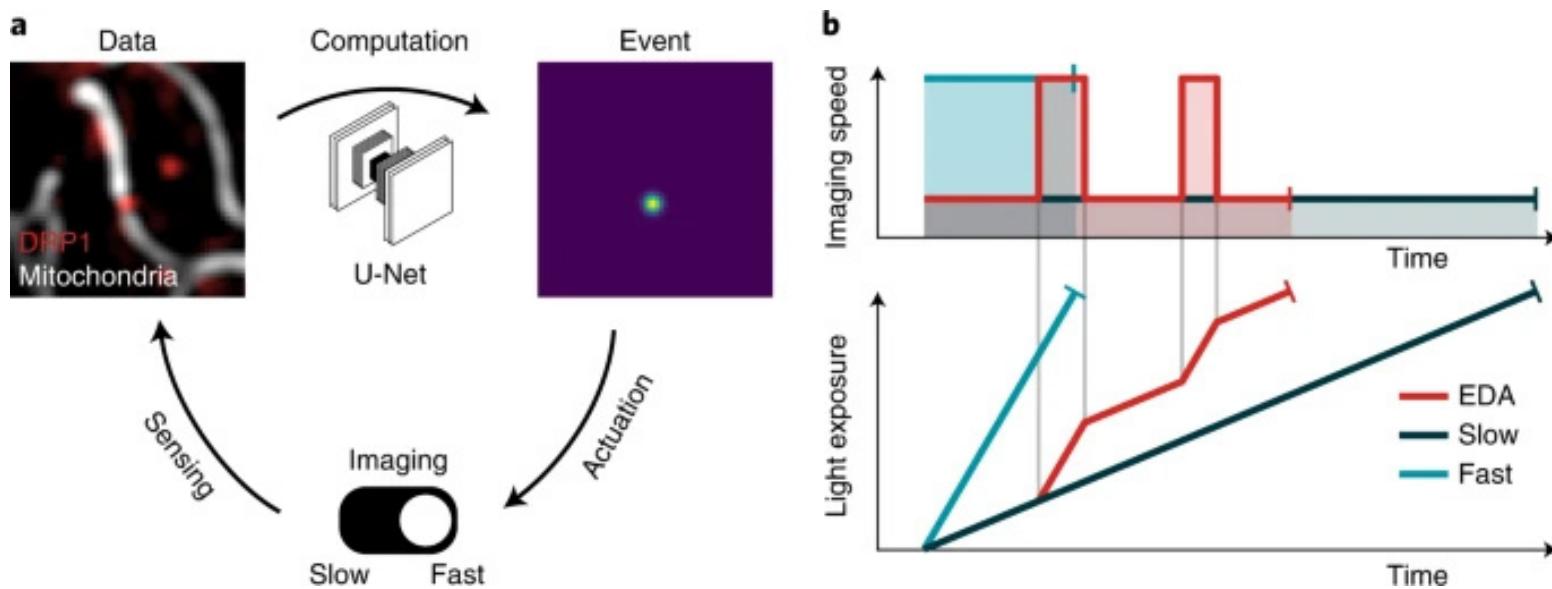
SELF-DRIVING MICROSCOPY



The thinking microscope. Credit: David Parkin

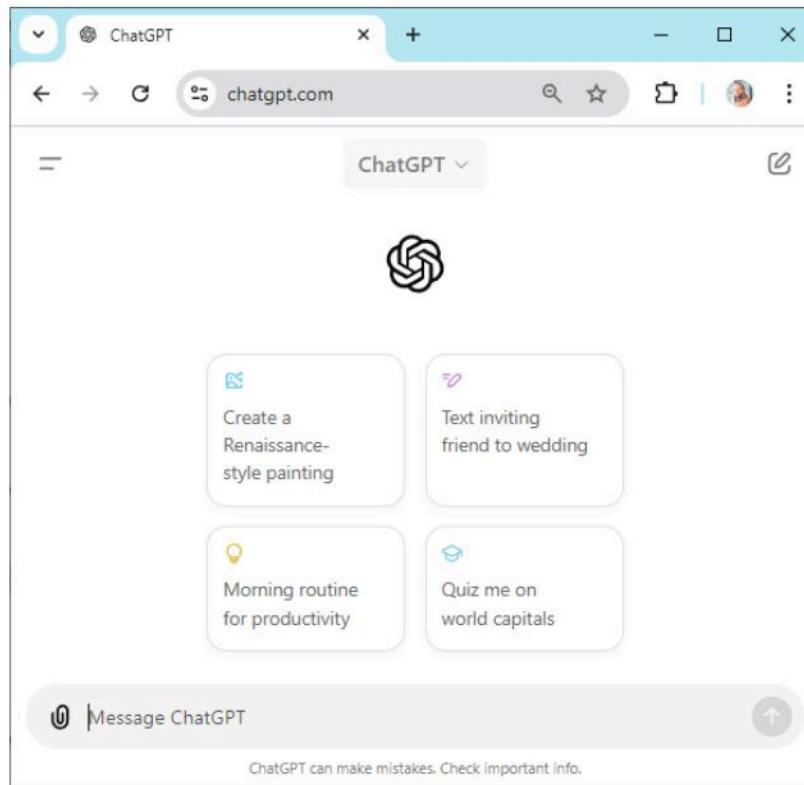
“ SELF-DRIVING MICROSCOPY REVOLUTIONIZES IMAGING BY AUTONOMOUSLY ADAPTING ACQUISITION PARAMETERS IN RESPONSE TO CHANGES WITHIN THE FIELD OF VIEW ,”

MODIFYING IMAGING RATES ON THE FLY

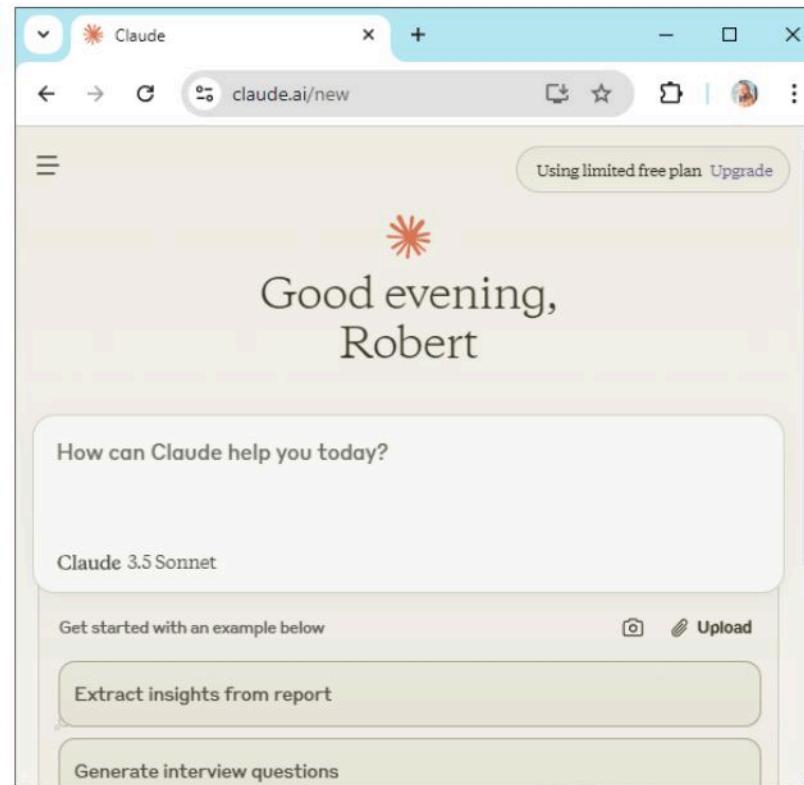


LLMs are everywhere

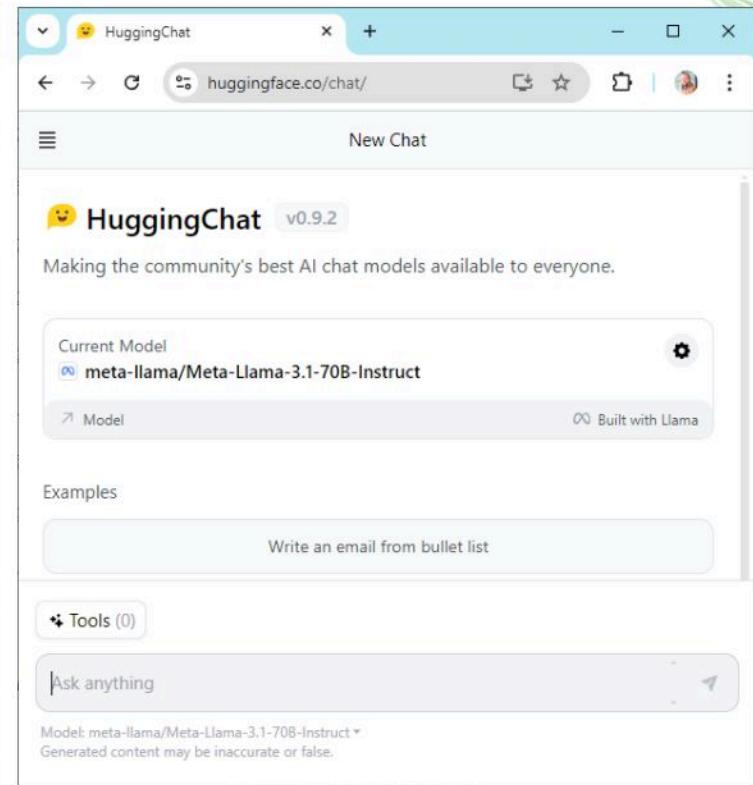
ChatGPT



Claude



Llama 3.1 405B

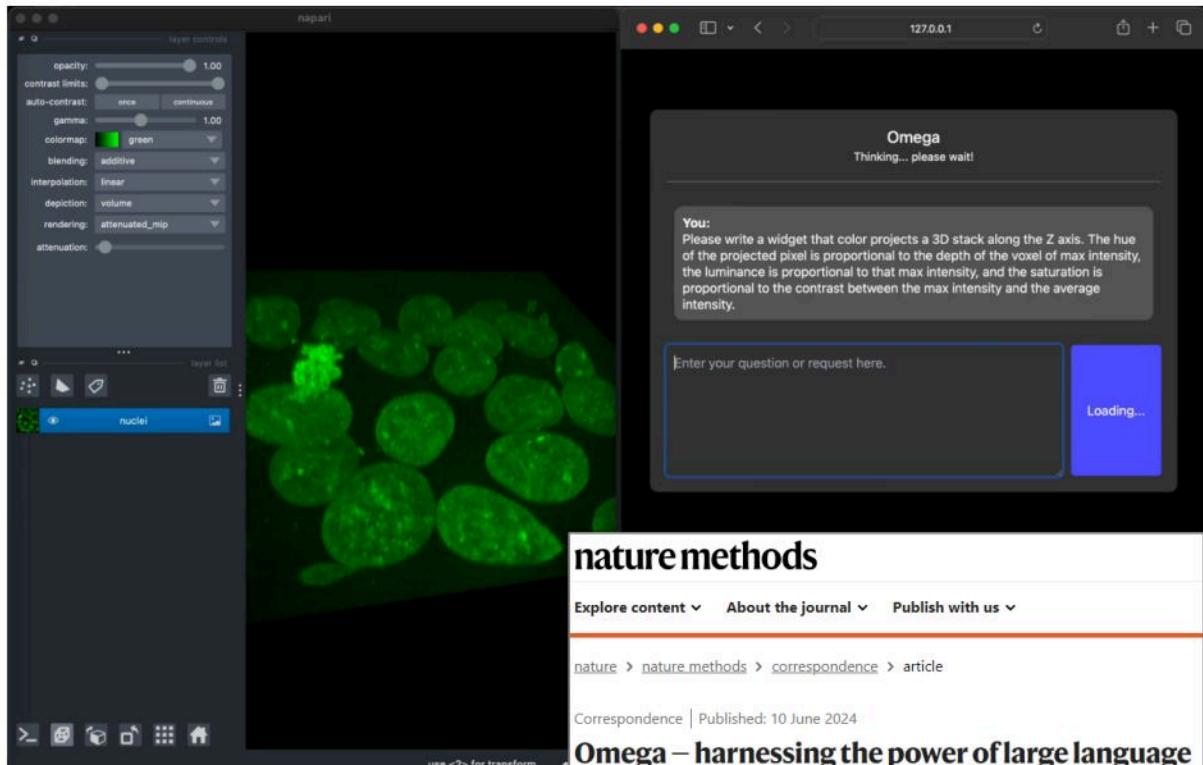


GloBIAS Seminar Series

Sept. 27th 2024
Oct. 18th 2024

LLMs are everywhere

Napari-chatGPT / Omega



<https://github.com/royerlab/napari-chatgpt>
<https://www.nature.com/articles/s41592-024-02310-w>

nature methods

Explore content ▾ About the journal ▾ Publish with us ▾

nature > nature methods > correspondence > article

Correspondence | Published: 10 June 2024

Omega – harnessing the power of large language models for bioimage analysis

Loïc A. Royer

Nature Methods (2024) | [Cite this article](#)

58 Altmetric | Metrics

Bioimage-io ChatBot

BioImage.IO Chatbot

Hi there! I'm Melman. I am help you navigate the bioimaging tools and provide information about bioimaging. How can I help you today?

How can I segment dense nuclei in a fluorescence microscopy image?

To segment dense nuclei in fluorescence microscopy images, you can use several tools and methods available in the bioimaging community. Here are some recommended approaches and tools based on your needs:

1. Cellpose: This tool is designed for 2D nuclei segmentation and can handle dense and overlapping nuclei. It outputs a binary mask showing where nuclei were detected. More information can be found [here](#).
2. CellProfiler: This software offers a workflow for nuclei segmentation in 2D images. It can detect and segment nuclei, outputting a binary mask. More details are available [here](#).
3. ImageJ/Fiji: The Watershed method in ImageJ/Fiji is effective for separating touching and overlapping nuclei. A detailed tutorial on using this method for DAPI-stained nuclei can be found [here](#).
4. Illastik: This tool allows for interactive learning and segmentation. You can create label classes for different objects (e.g., background, nuclei, mitotic nuclei, cytoplasm) and use live prediction mode to refine the segmentation. More information can be found in the [illastik documentation](#).
5. Scikit-Image: This Python library provides various functions for image processing, including segmentation. An example of segmenting human cell nuclei using scikit-image can be found in the [scikit-image documentation](#).

These tools and methods should help you effectively segment dense nuclei in your fluorescence microscopy images.

[More Details](#)

nature methods

Explore content ▾ About the journal ▾ Publish with us ▾

nature > nature methods > correspondence > article

Correspondence | Published: 09 August 2024

BioImage.IO Chatbot: a community-driven AI assistant for integrative computational bioimaging

Wanlu Lei, Caterina Fuster-Barceló, Gabriel Reder, Arrate Muñoz-Barrutia & Wei Ouyang

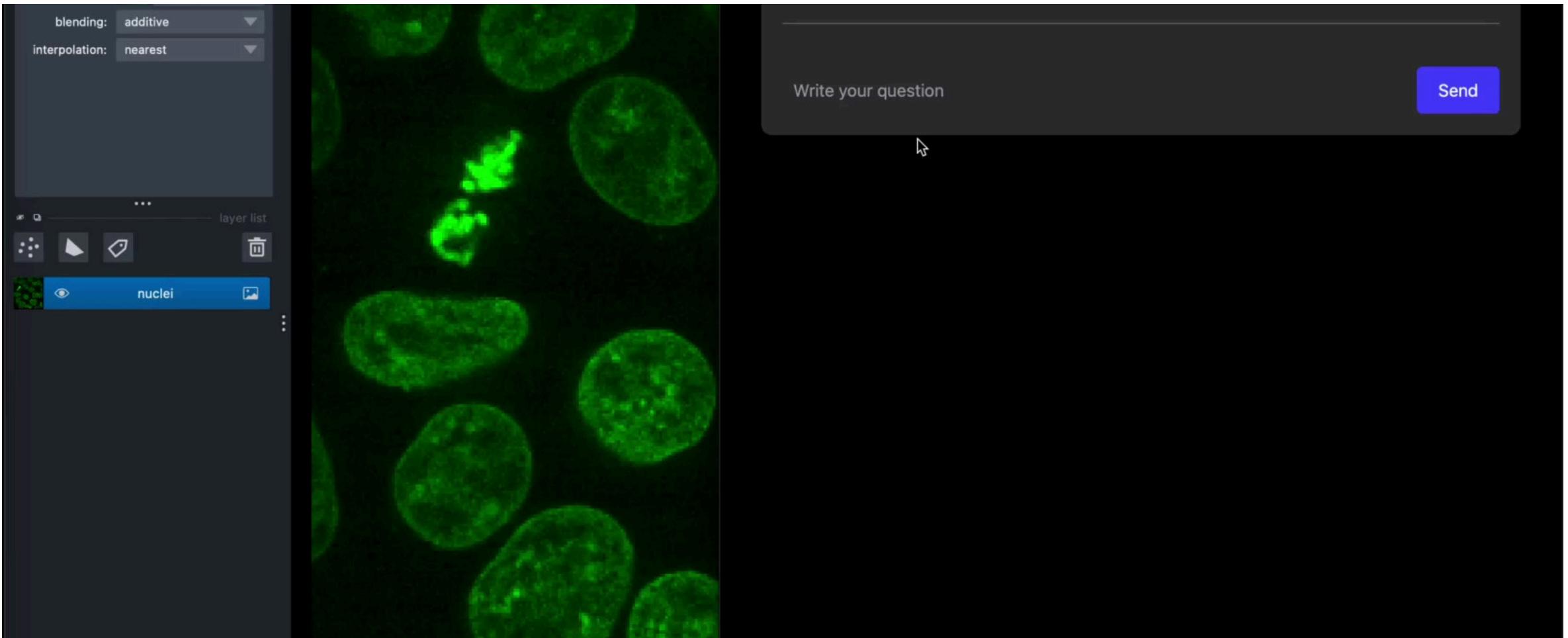
Nature Methods 21, 1368–1370 (2024) | [Cite this article](#)

865 Accesses | 1 Altmetric | Metrics

<https://bioimage.io/chat/>
<https://www.nature.com/article/s/s41592-024-02370-y>

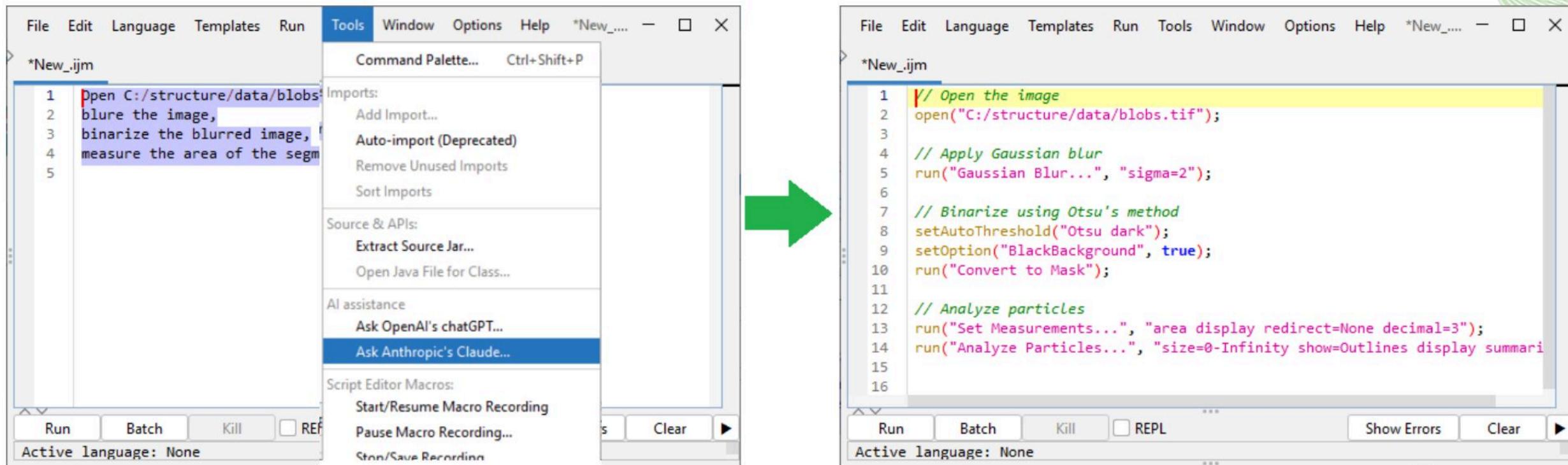
CHATGTP IN NAPARI

Using natural language for bioimage analysis



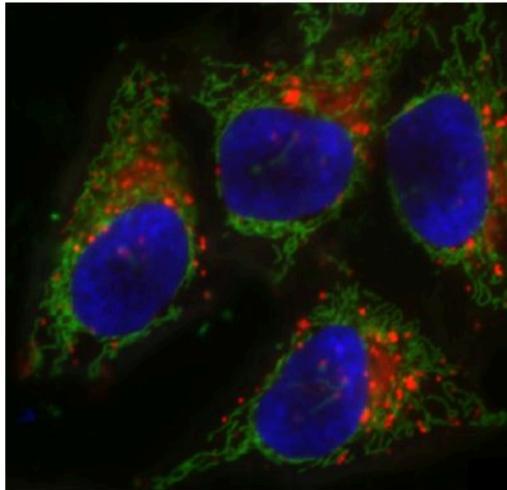
LLMs are everywhere

Fiji's Script Editor

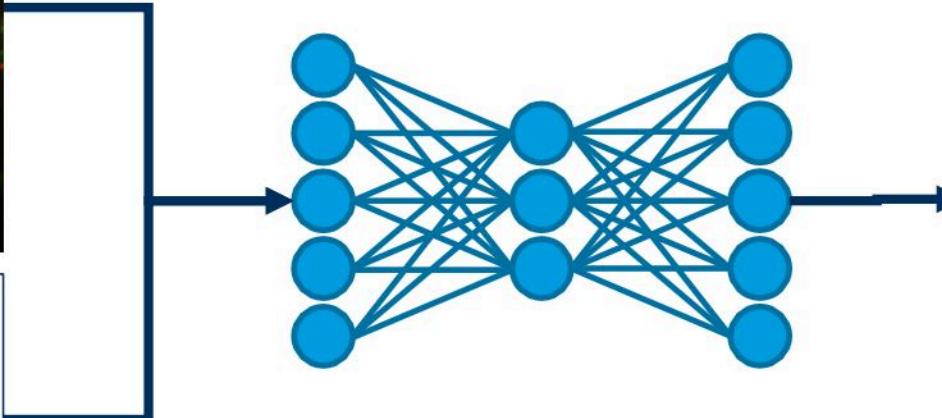


Vision Language Models VLMs

Combining image and text: Multi-Modal LMs



What's in the
blue-channel of
this image?



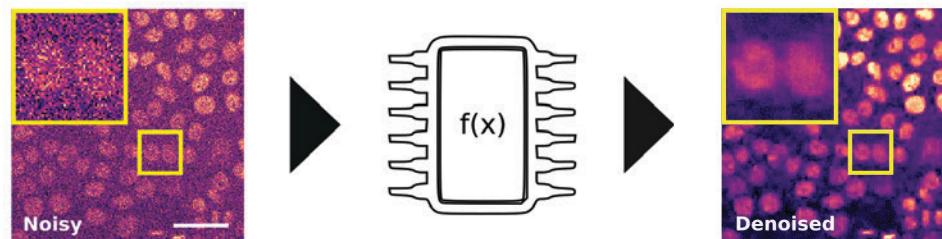
There are nuclei,
presumably stained
with DAPI and imaged
using a fluorescence
microscope.

Section 2

DEEP LEARNING HOW DOES IT WORK?

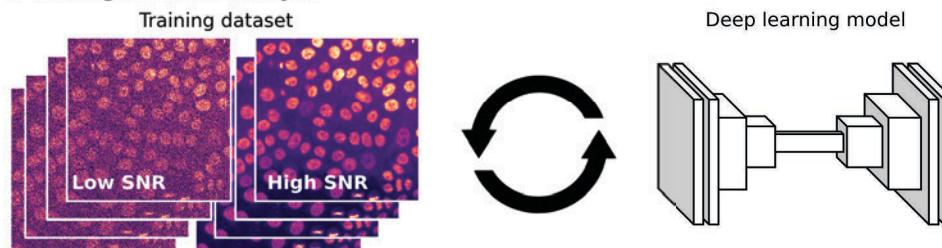
DEEP LEARNING, HOW DOES IT WORK?

Classical algorithm

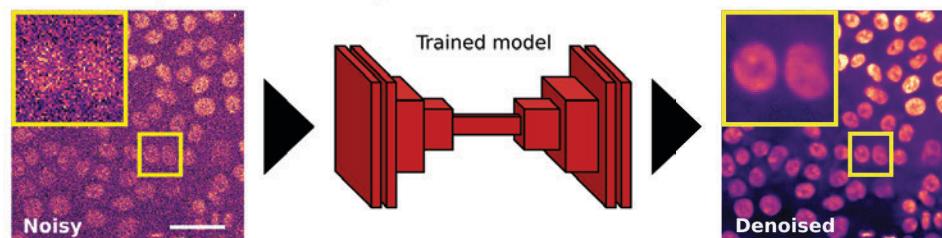


Deep Learning algorithm

1. Training (minutes to days)

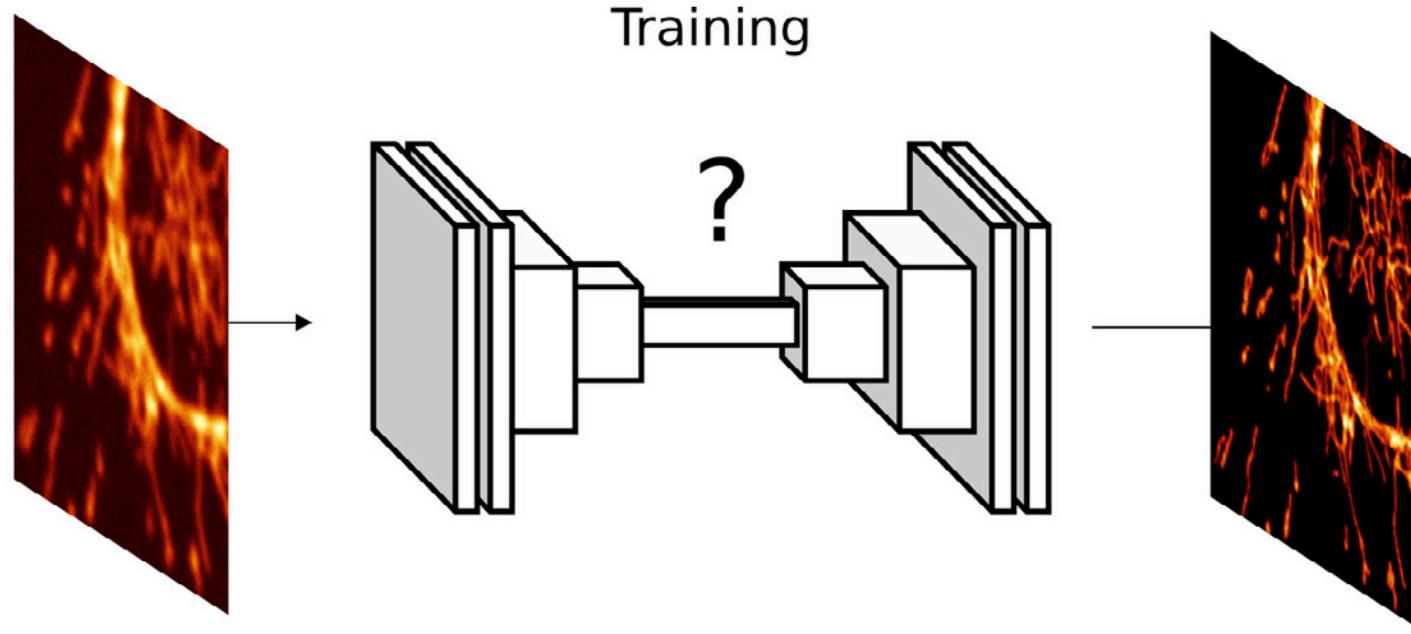


2. Predictions (seconds to minutes)

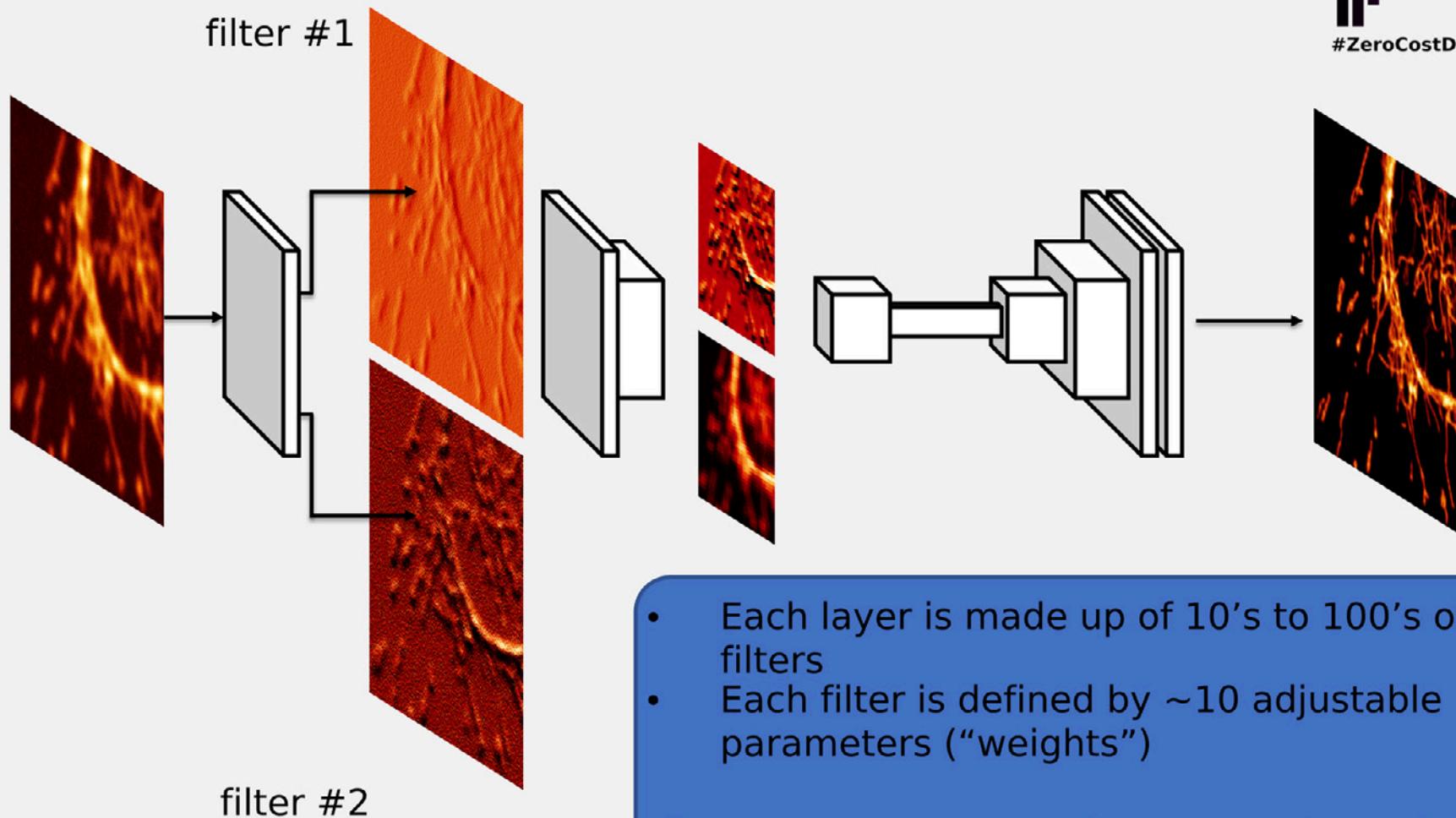


Laine et al., 2021

What does a neural network do?



What does a neural network do?



- Each layer is made up of 10's to 100's of filters
- Each filter is defined by ~10 adjustable parameters ("weights")

The network can be defined by millions of adjustable weights



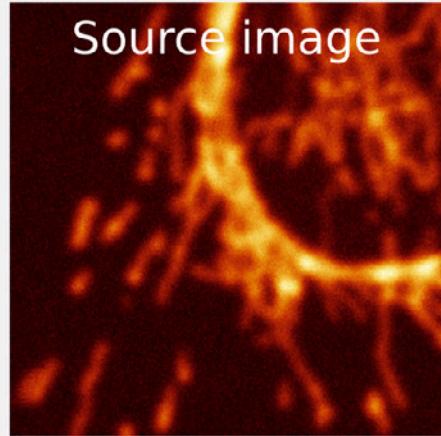
How does the network learn?

Step #1: Initialise all the weights randomly

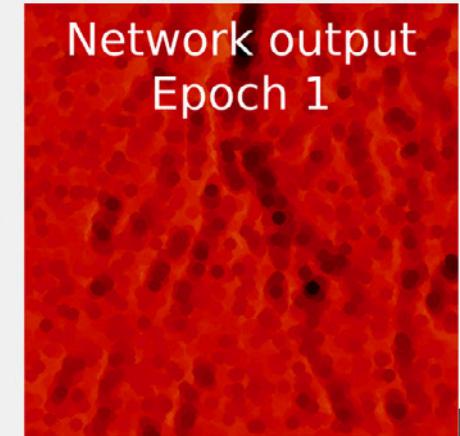
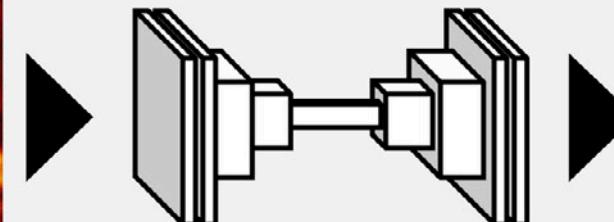
Step #2: Compute network output (feed forward)

Step #3: Compare to target image (ground truth)

Step #4: Update weights to decrease the error (backpropagation)



Training



Target image
(Ground truth)

How does the neural network learn?

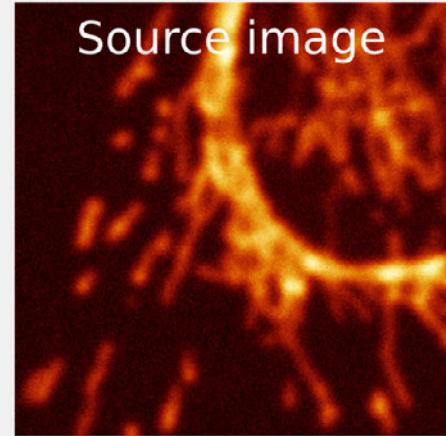


Step #1: Initialise all the weights randomly

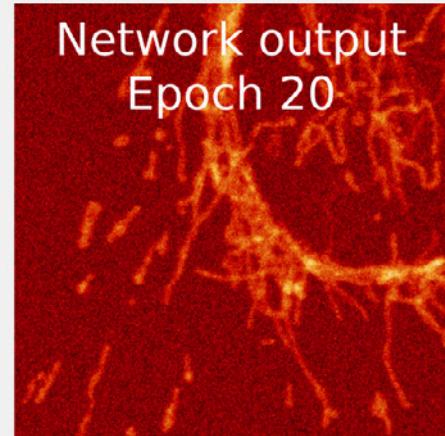
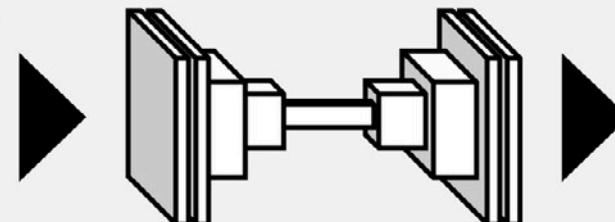
Step #2: Compute network output (feed forward)

Step #3: Compare to target image (ground truth)

Step #4: Update weights to decrease the error (backpropagation)



Training



Target image
(Ground truth)

How does the neural network learn?

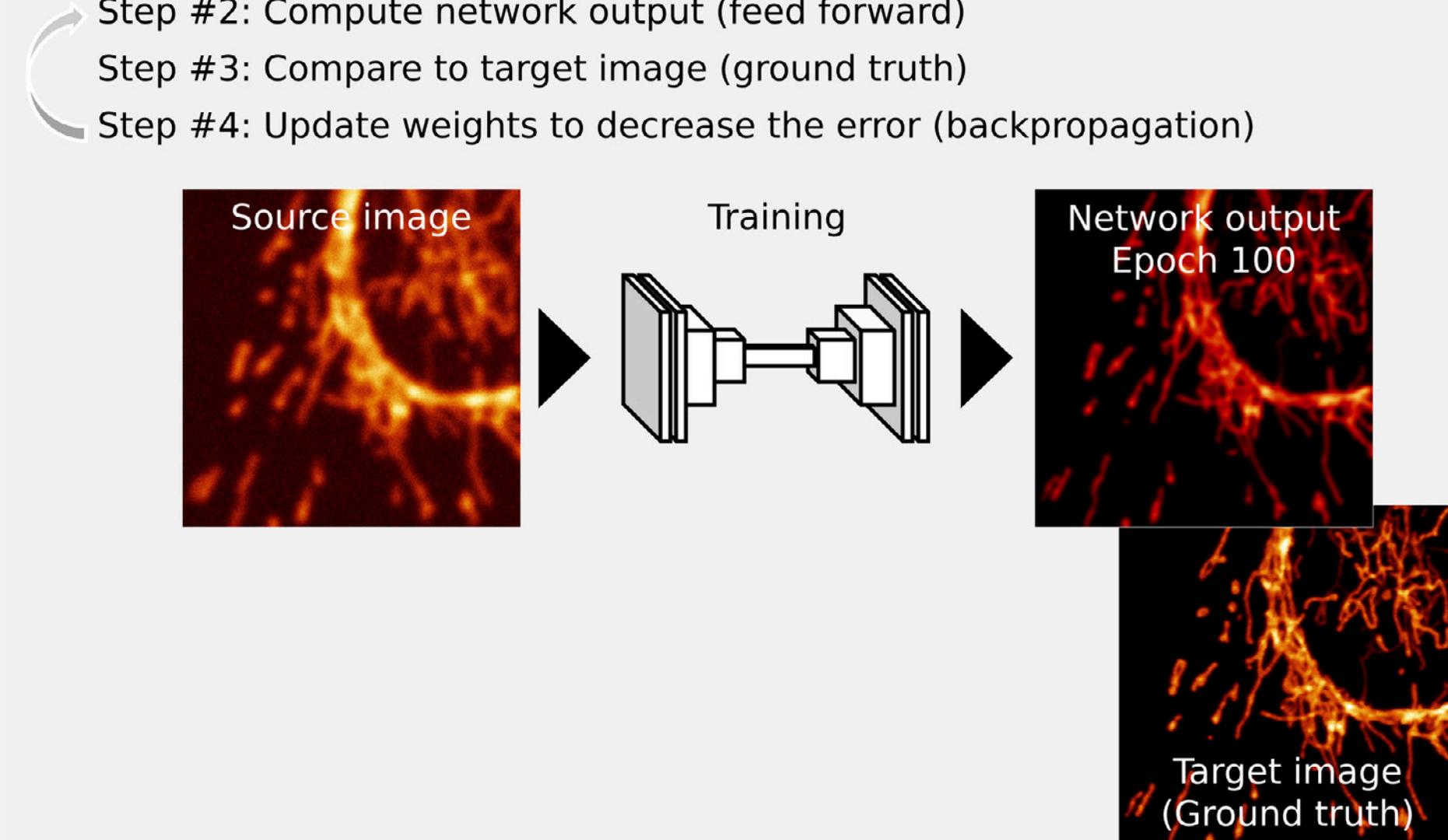


Step #1: Initialise all the weights randomly

Step #2: Compute network output (feed forward)

Step #3: Compare to target image (ground truth)

Step #4: Update weights to decrease the error (backpropagation)



How does the neural network learn?

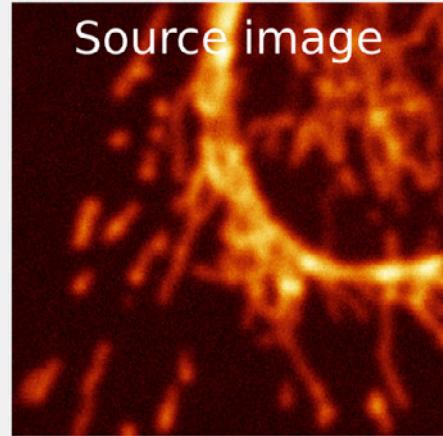


Step #1: Initialise all the weights randomly

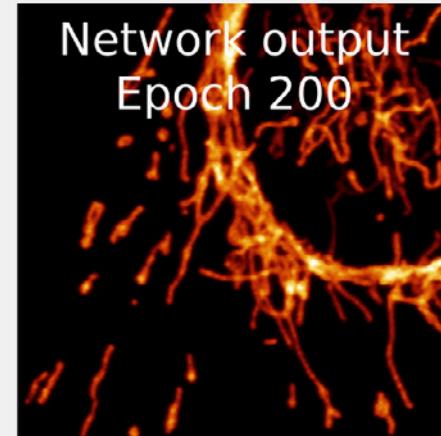
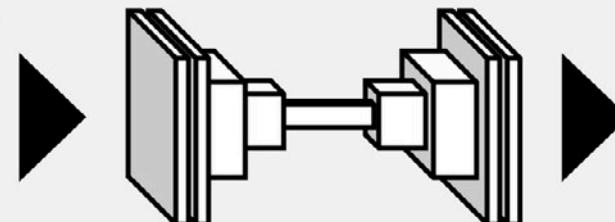
Step #2: Compute network output (feed forward)

Step #3: Compare to target image (ground truth)

Step #4: Update weights to decrease the error (backpropagation)



Training

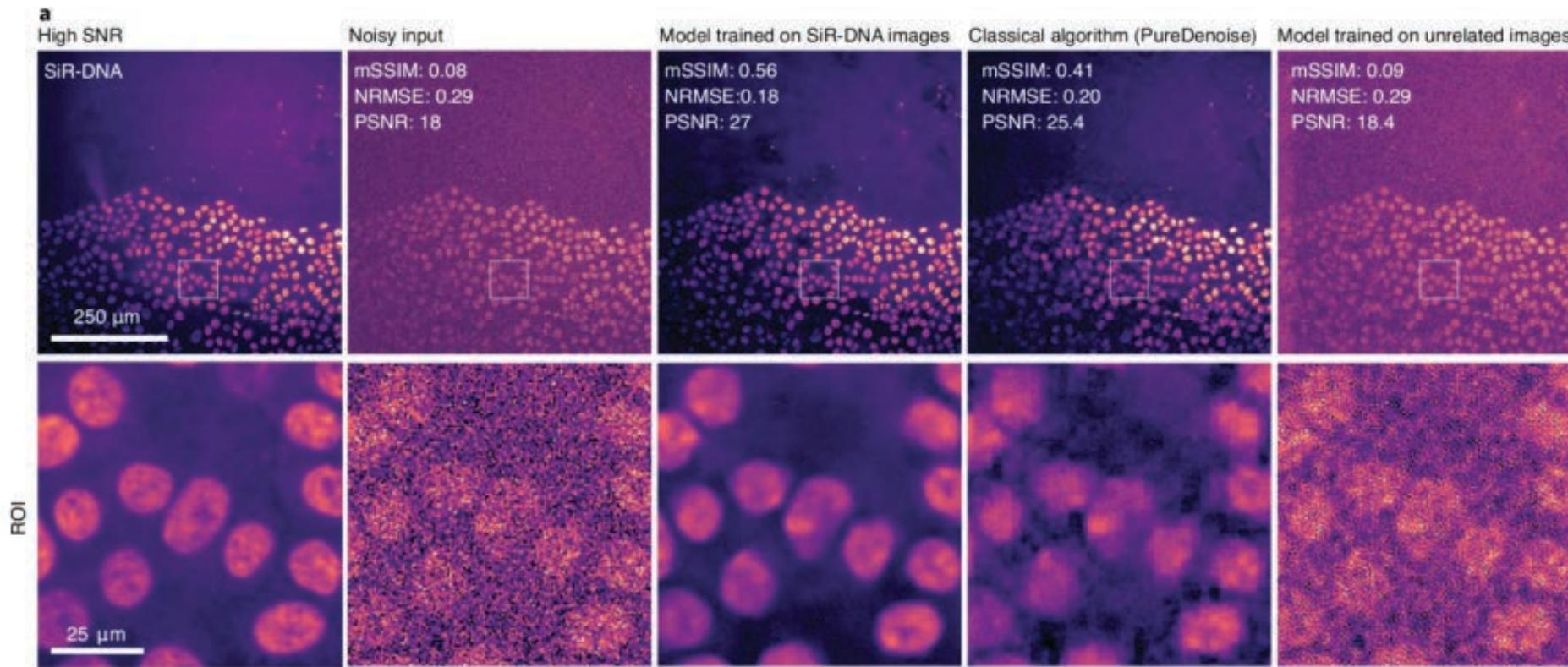


Network output
Epoch 200



Target image
(Ground truth)

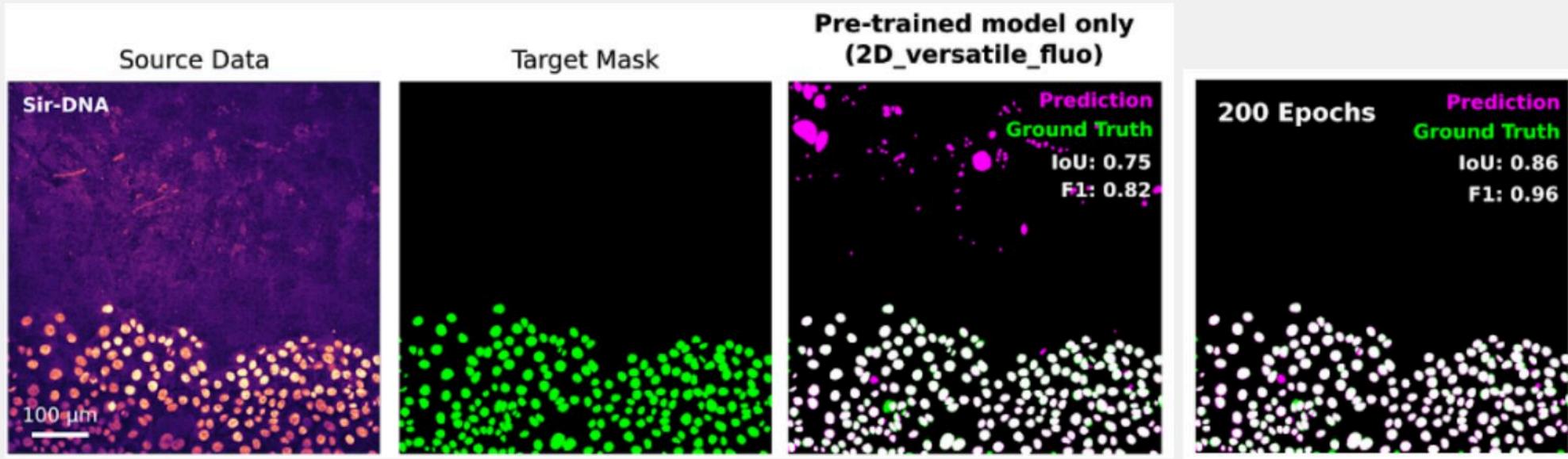
DEEP LEARNING MODELS WORK WELL ON DATA SIMILAR TO THE ONE USED DURING TRAINING



Laine et al., 2021

The performance is dependant on the training data

- The dataset used to train a DL network is key
- Always validate your models using ground truth data



CAN YOU TRUST DEEP LEARNING?

✓ Validate

Quality control strategies exist (same as any other image analysis methods)

✓ Reproduce

Model = dataset + software (code + underlying libraries)

Both dataset and software can evolve overtime

Models are typically not stable over time

✗ Explain

Models are often black boxes



Avoiding a replication crisis in deep-learning-based bioimage analysis

Deep learning algorithms are powerful tools for analyzing, restoring and transforming bioimaging data. One promise of deep learning is parameter-free one-click image analysis with expert-level performance in a fraction of the time previously required. However, as with most emerging technologies, the potential for inappropriate use is raising concerns among the research community. In this Comment, we discuss key concepts that we believe are important for researchers to consider when using deep learning for their microscopy studies. We describe how results obtained using deep learning can be validated and propose what should, in our view, be considered when choosing a suitable tool. We also suggest what aspects of a deep learning analysis should be reported in publications to ensure reproducibility. We hope this perspective will foster further discussion among developers, image analysis specialists, users and journal editors to define adequate guidelines and ensure the appropriate use of this transformative technology.

Romain F. Laine, Ignacio Arganda-Carreras, Ricardo Henriques and Guillaume Jacquemet

Section 3

THE ZEROCOSTDL4MIC PLATFORM

TRAINING A DEEP LEARNING MODEL:

- Install the required software and dependencies (surprisingly hard)
- Adapt the code (Python)
- Need powerful workstations



Photo credit: <https://www.nasa.gov/sites/default/files/iss037e013962.jpg?itok=5Q93H7Qn>

The ZeroCostDL4Mic team



Lucas



Romain



Ricardo

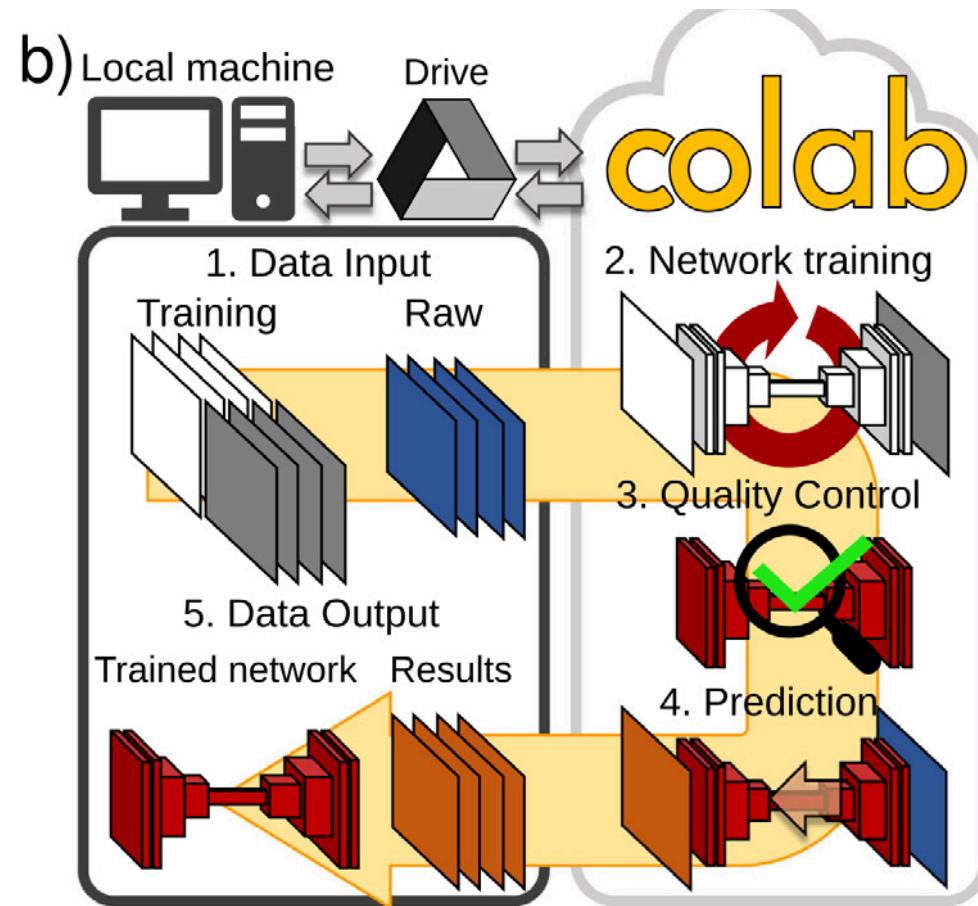
Founding

- [Lucas von Chamier](#)
- [Johanna Jukkala](#)
- [Christoph Spahn](#)
- [Martina Lerche](#)
- [Sara Hernández-Pérez](#)
- [Pieta K. Mattila](#)
- [Eleni Karinou](#)
- [Seamus Holden](#)
- [Ahmet Can Solak](#)
- [Alexander Krull](#)
- [Tim-Oliver Buchholz](#)
- [Florian Jug](#)
- [Loïc A Royer](#)
- [Mike Heilemann](#)
- [Romain F. Laine](#)
- [Guillaume Jacquemet](#)
- [Ricardo Henriques](#)

Newcomers

- [Elias Nehme](#)
- [Lucien Weiss](#)
- [Yoav Shechtman](#)
- [Christophe Leterrier](#)
- [Daniel Krentzel](#)
- [Martin Jones](#)
- [Wei Ouyang](#)
- [Estibaliz Gómez de Mariscal](#)
- [Erlantz Calvo](#)
- [Ignacio Arganda-Carreras](#)
- [Amin Rezaei](#)
- [Ainhoa Serrano](#)

ZEROCOSTDL4MIC



Chamier et al. (Nature Communications, 2021)

ZEROCOSTDL4MIC

Tasks and networks
implemented by ZeroCostDL4Mic

<https://github.com/HenriquesLab/ZeroCostDL4Mic>

von Chamier et al. 2021; Hidalgo-Cenalmor et al. 2024

The ZeroCostDL4Mic interface

The screenshot shows a Google Colab notebook titled "StarDist_2D_ZeroCostDL4Mic.ipynb". The left sidebar contains a "Table of contents" with the following structure:

- StarDist (2D)
 - How to use this notebook?
 - 0. Before getting started
 - 1. Initialise the Colab session
 - 1.1. Check for GPU access
 - 1.2. Mount your Google Drive
 - 2. Install StarDist and dependencies
 - 3. Select your parameters and paths
 - 3.1. Setting main training parameters
 - 3.2. Data augmentation
 - 3.3. Using weights from a pre-trained model as initial weights
 - 4. Train the network
 - 4.1. Prepare the training data and model for training
 - 4.2. Start Training
 - 4.3. Download your model(s) from Google Drive
 - 5. Evaluate your model
 - 5.1. Inspection of the loss function
 - 5.2. Error mapping and quality metrics estimation
 - 6. Using the trained model
 - 6.1 Generate prediction(s) from unseen dataset
 - 6.2. Download your predictions
 - Thank you for using StarDist 2D!
 - Section

The main content area is titled "4. Train the network" and includes the following sections:

 - 4.1. Prepare the training data and model for training**

Here, we use the information from 3. to build the model and convert the training data into a suitable format for training.

[] Create the model and dataset objects
 - 4.2. Start Training**

When playing the cell below you should see updates after each epoch (round). Network training can take some time.

* **CRITICAL NOTE:** Google Colab has a time limit for processing (to prevent using GPU power for datamining). Training time must be less than 12 hours! If training takes longer than 12 hours, please decrease the number of epochs or number of patches. Another way circumvent this is to save the parameters of the model after training and start training again from this point.

Of Note: At the end of the training, your model will be automatically exported so it can be used in the Stardist Fiji plugin. You can find it in your model folder (TF_SavedModel.zip). In Fiji, Make sure to choose the right version of tensorflow. You can check at: Edit-- Options-- Tensorflow. Choose the version 1.4 (CPU or GPU depending on your system).

[] Start training
 - 4.3. Download your model(s) from Google Drive**

Once training is complete, the trained model is automatically saved on your Google Drive, in the **model_path** folder that was selected in Section 3. It is however wise to download the folder as all data can be erased at the next training if using the same folder.

ZeroCostDL4Mic

Segmentation networks

Network	Paper(s)	Tasks	Status	Link to example training and test dataset	Direct link to the notebook in Colab
U-Net (2D)	here and here	Binary segmentation	Fully supported	here	Open in Colab
U-Net (3D)	here	Binary segmentation	Fully supported	EPFL dataset	Open in Colab
U-Net (2D) multilabel	here and here	Semantic segmentation	Under beta-testing	here	Open in Colab
DenolSeg	here	Joint denoising and binary segmentation	Fully supported	Available soon	Open in Colab
StarDist (2D)	here and here	Instance segmentation	Fully supported	here	Open in Colab
StarDist (3D)	here and here	Instance segmentation	Fully supported	from Stardist github	Open in Colab
Celpose (2D and 3D)	here	Instance segmentation (Cells or Nuclei)	Fully supported	Coming soon!	Open in Colab
SplineDist (2D)	here	Instance segmentation	Fully supported	here	Open in Colab
EmbedSeg (2D)	here	Instance segmentation	Under beta-testing	here	Open in Colab
MaskRCNN (2D)	here	Instance segmentation	Under beta-testing	Coming soon!	Open in Colab
Interactive Segmentation - Kaibu (2D)	here	Interactive instance segmentation	Under beta-testing	Coming soon!	Open in Colab

Denoising and image restoration networks

Network	Paper(s)	Tasks	Status	Link to example training and test dataset	Direct link to the notebook in Colab
Noise2Void (2D)	here	Self-supervised denoising	Fully supported	here	Open in Colab
Noise2Void (3D)	here	Self-supervised denoising	Fully supported	here	Open in Colab
CARE (2D)	here	Supervised denoising	Fully supported	here	Open in Colab
CARE (3D)	here	Supervised denoising	Fully supported	here	Open in Colab
3D-RCAN	here	Supervised denoising	Under beta-testing	here	Open in Colab
DecoNoising (2D)	here	Self-supervised denoising	Under beta-testing	here	Open in Colab

Super-resolution microscopy networks

Network	Paper(s)	Tasks	Status	Link to example training and test dataset	Direct link to the notebook in Colab
Deep-STORM	here	Single Molecule Localization Microscopy (SMLM) image reconstruction from high-density emitter data	Fully supported	Training data simulated in the notebook or available from here	Open in Colab
DFCAN	here	image upsampling	Under beta-testing	here	Open in Colab

- We currently provide 33 notebooks

ZeroCostDL4Mic

Object detection networks

Network	Paper(s)	Tasks	Status	Link to example training and test dataset	Direct link to the notebook in Colab
YOLOv2	here	Object detection (bounding boxes)	Fully supported	here	 Open in Colab
Detectron2	here	Object detection (bounding boxes)	Under beta-testing	here	 Open in Colab
RetinaNet	here	Object detection (bounding boxes)	Under beta-testing	here	 Open in Colab

Image-to-image translation networks

Network	Paper(s)	Tasks	Status	Link to example training and test dataset	Direct link to the notebook in Colab
Label-free prediction (fnet) 2D	here	Artificial labelling	Under beta-testing	Coming soon	 Open in Colab
Label-free prediction (fnet) 3D	here	Artificial labelling	Fully supported	here	 Open in Colab
CycleGAN	here	Unpaired Image-to-Image Translation	Fully supported	here	 Open in Colab
pix2pix	here	Paired Image-to-Image Translation	Fully supported	here	 Open in Colab

Registration networks

Network	Paper(s)	Tasks	Status	Link to example training and test dataset	Direct link to the notebook in Colab
DRMIME	here	Affine or perspective image registration	Under beta-testing	Coming soon!	 Open in Colab

BiolImage.io notebooks

Networks that are compatible with [BiolImage.IO](#) and can be used in [ImageJ](#) via [deeplImageJ](#).

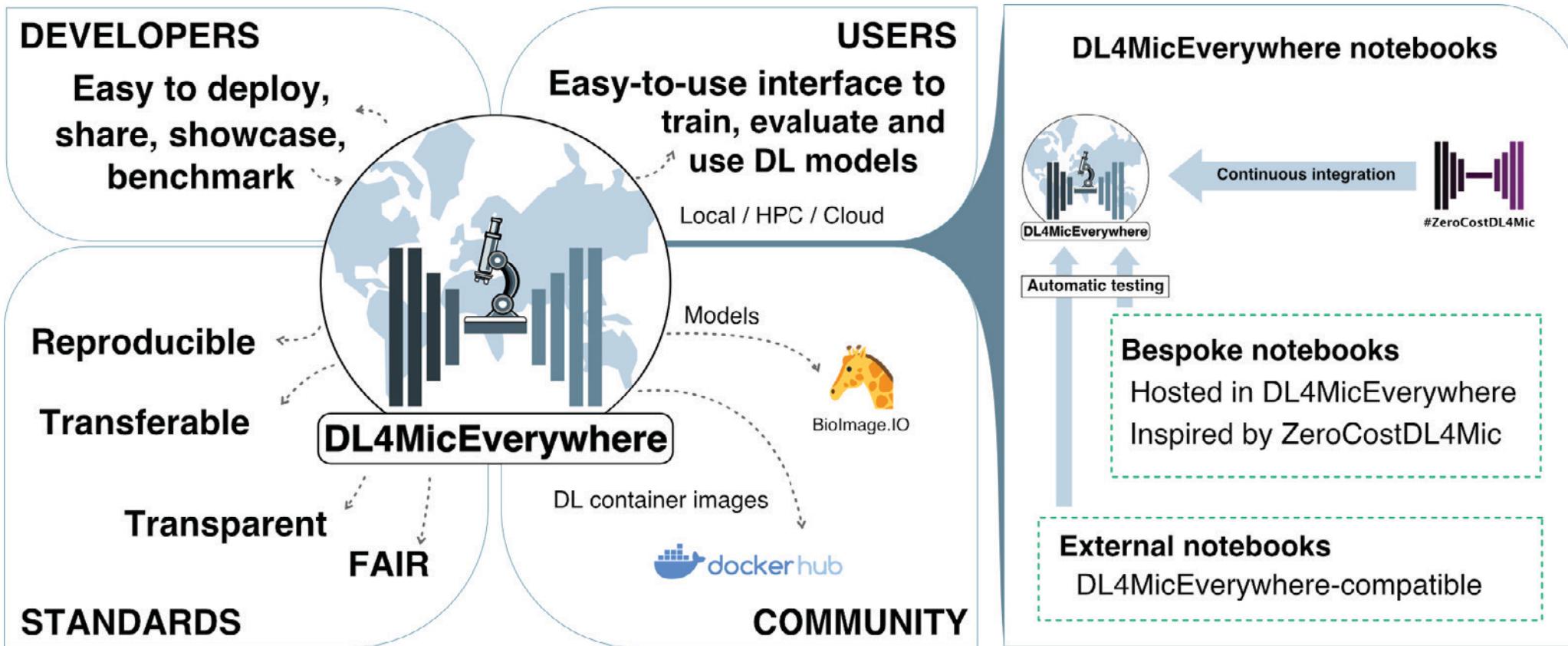
Network	Paper(s)	Task	Link to example training and test dataset	Direct link to the notebook in Colab
StarDist (2D) with DeepImageJ export	StarDist: here and here , and DeepImageJ	Nuclei segmentation	here	 Open in Colab
Deep-STORM with DeepImageJ export	Deep-STORM and DeepImageJ	Single Molecule Localization Microscopy (SMLM) image reconstruction from high-density emitter data	Training data simulated in the notebook or available from here	 Open in Colab
U-Net (2D) with DeepImageJ export	U-Net and DeepImageJ	Segmentation	ISBI challenge or here	 Open in Colab
U-Net (3D) with DeepImageJ export	3D U-Net and DeepImageJ	Segmentation	EPFL dataset	 Open in Colab

- We currently provide 33 notebooks

THE ZEROCOSTDL4MIC PLATFORM

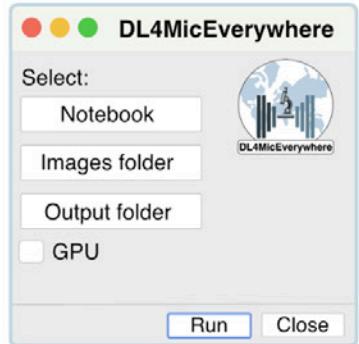
- ✓ Is it accessible?
- ✓ What about quality control?
- ✓ Is it reproducible?
It depends, libraries changes overtime. No control over Google Colab.
- ✗ Is it sustainable?
No
- ✗ Is it private?
No

DL4MICEVERYWHERE

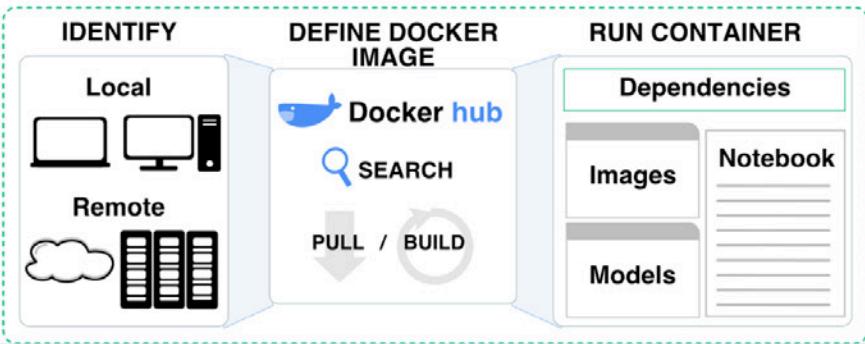


DL4MICEVERYWHERE

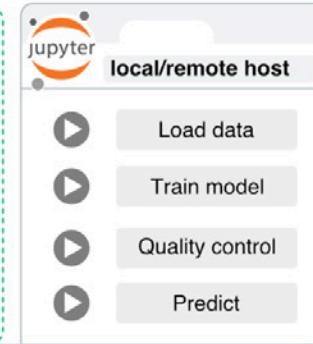
c) Interface



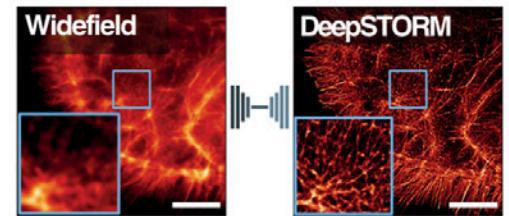
d) Automated containerisation



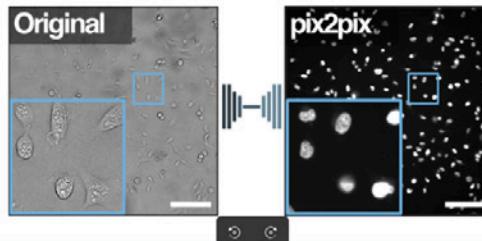
e) Run



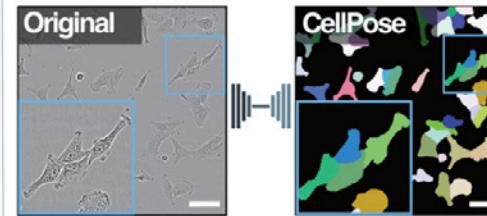
f) Super-resolution



g) Artificial labeling



h) Segmentation



How to use
DL4MicEverywhere?



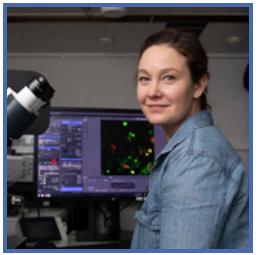
DL4MICEEVERYWHERE



Iván Hidalgo-Cenalmor



Estibaliz Gómez-de-Mariscal



Joanna Pylvänäinen



Ricardo Henriques

Iván Hidalgo-Cenalmor

Joanna W Pylvänäinen

Mariana G Ferreira

Craig T Russell

Alon Saguy

Ignacio Arganda-Carreras

Yoav Shechtman

AI4Life Consortium

Guillaume Jacquemet

Ricardo Henriques

Estibaliz Gómez-de-Mariscal

I STOLE SLIDES FROM ROBERT HAASE



Robert Haase, Lecturer and Training Coordinator, Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI), Leipzig University

FUNDING



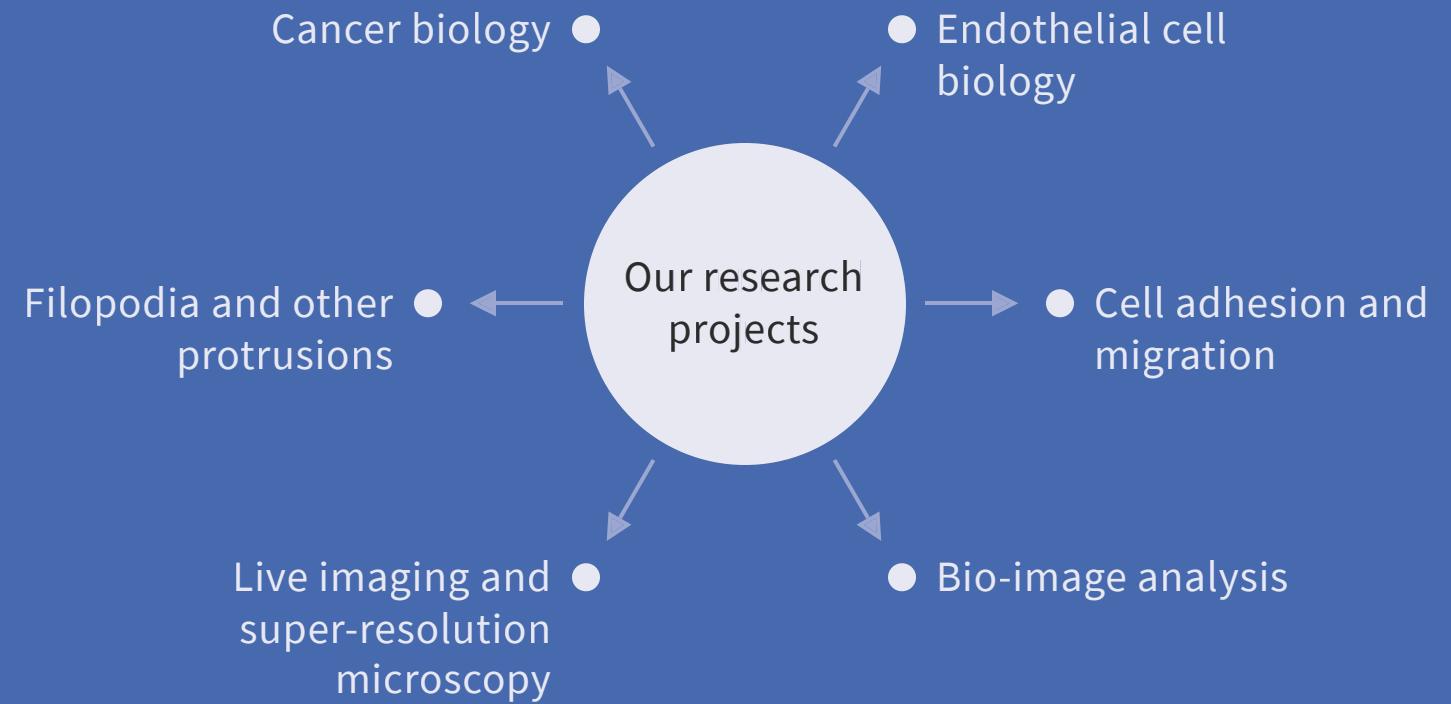
Syöpäsäätiö
Cancer Foundation Finland



JOIN US!

PhD and Postdoc positions available in my lab, Turku Finland

www.cellmig.org

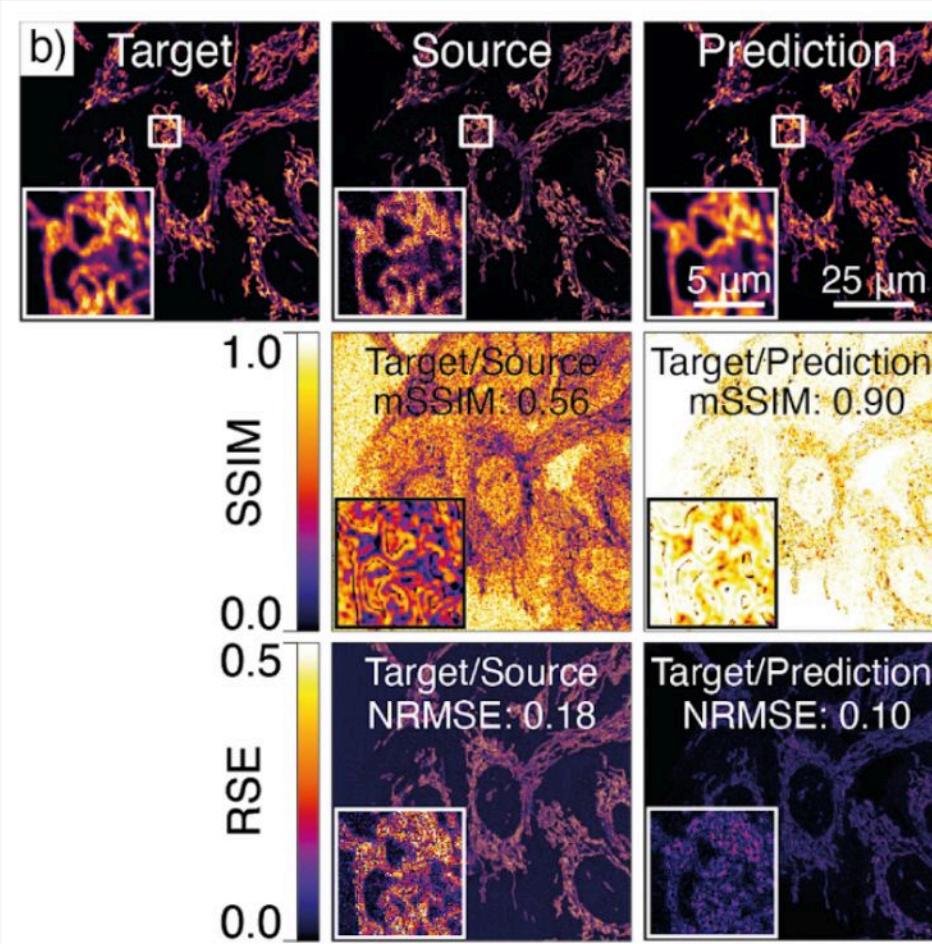


EXTRA SLIDES

The Quality control step



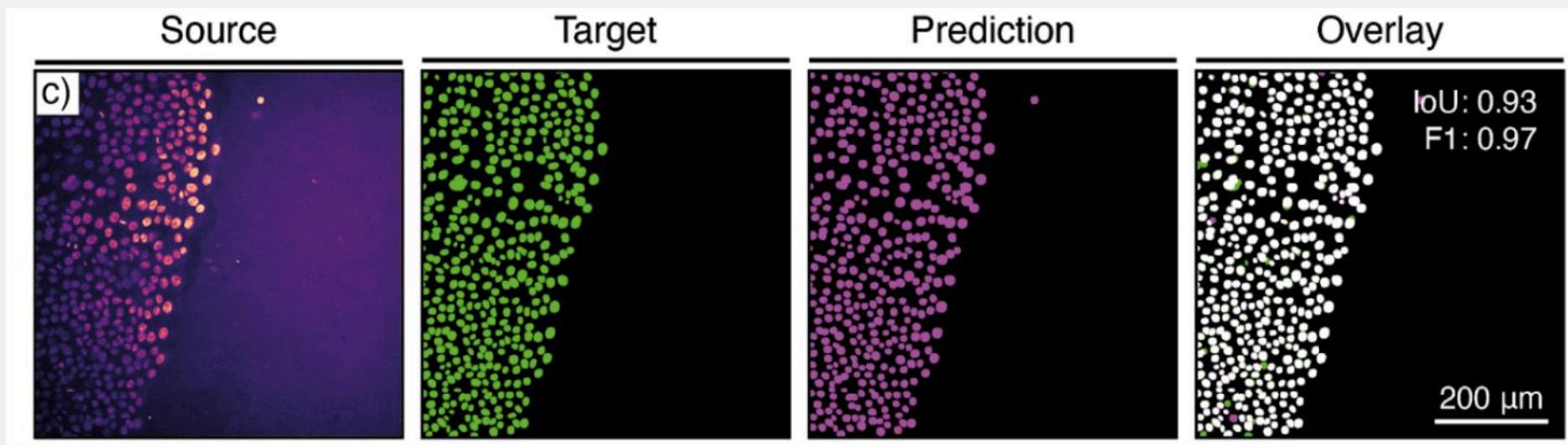
- Compare model predictions to ground truth data.
- Very important to use images that have not been used to train the network.



The Quality control step

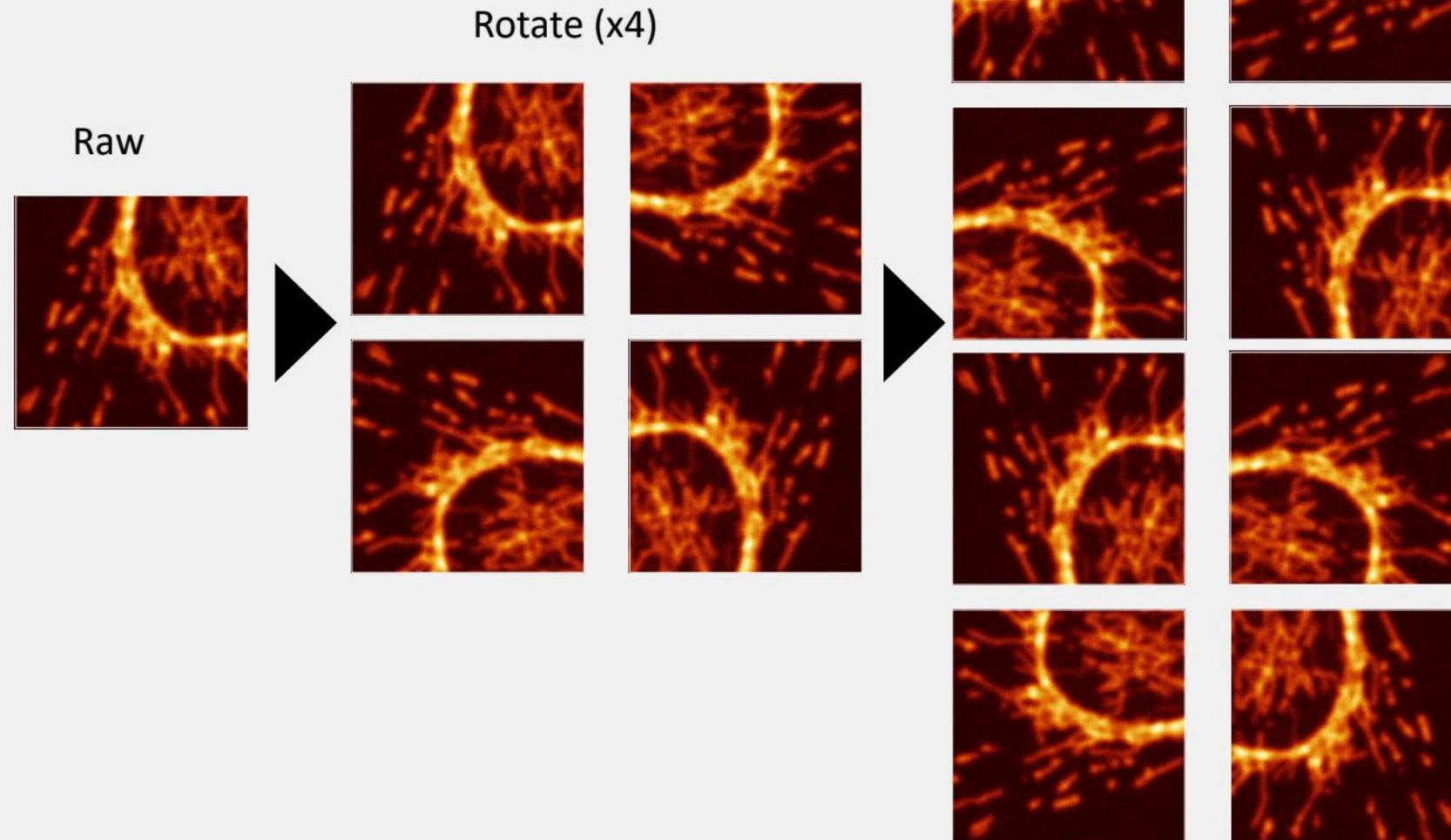


- Compare model predictions to ground truth data.
- Very important to use images that have not been used to train the network.



Data augmentation

- Artificially increase the training dataset size

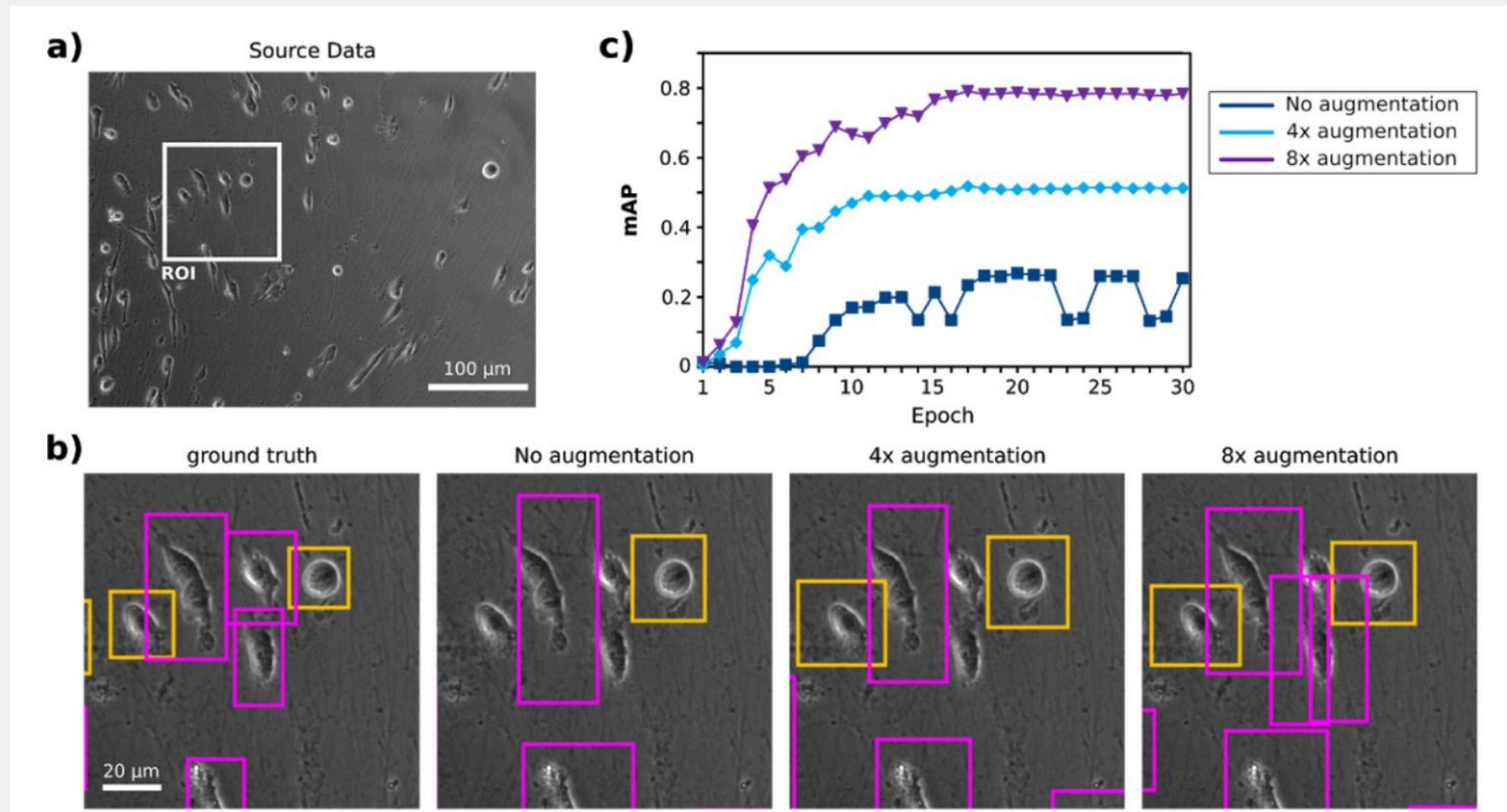


Rotate and Flip (x8)



#ZeroCostDL4Mic

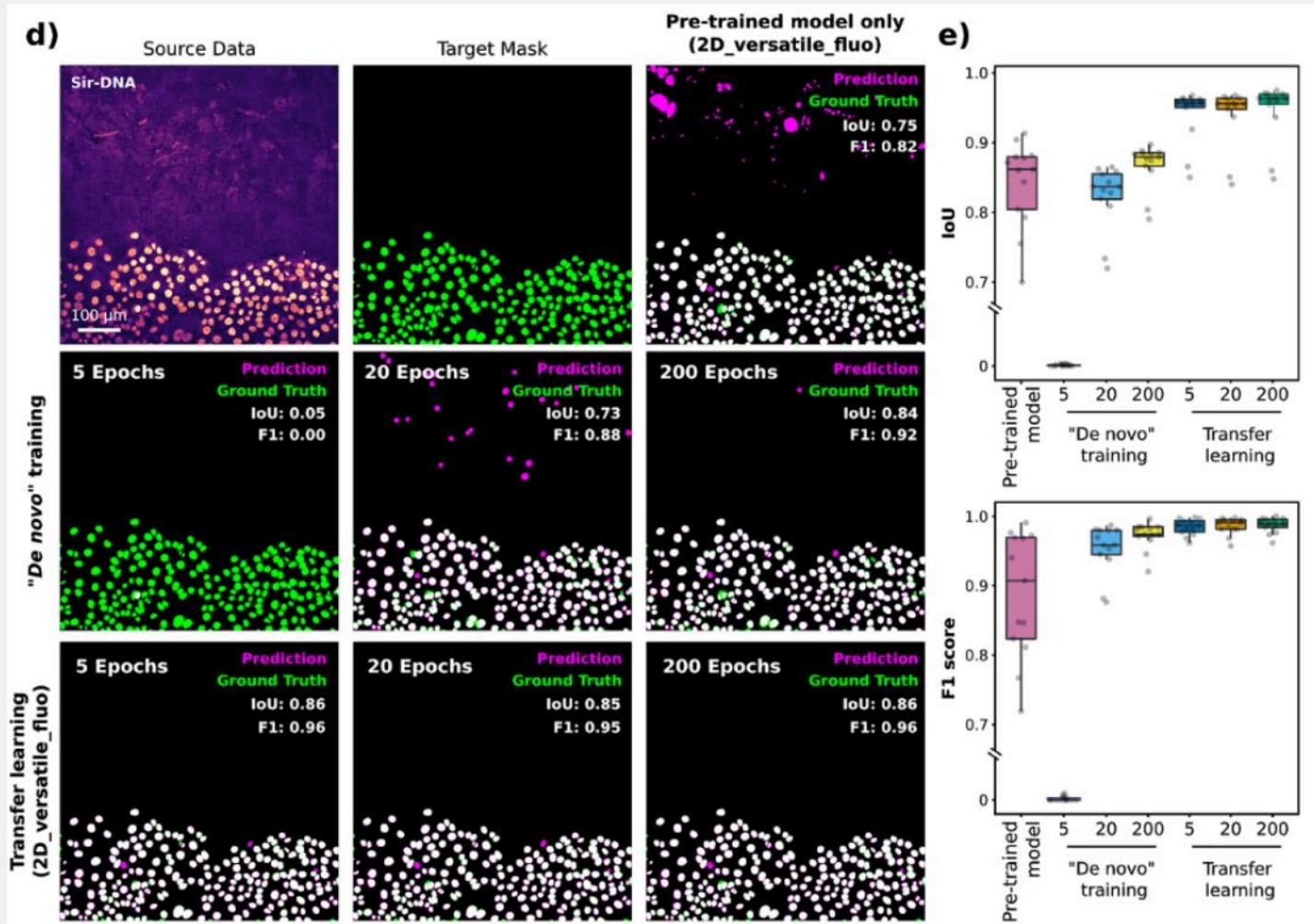
Data augmentation



Transfer learning



- Training starts from a trained model instead of using random weights



Contribute to ZeroCostDL4Mic



We welcome network contributions from the research community. If you wish to contribute, please read our guidelines

<https://github.com/HenriquesLab/ZeroCostDL4Mic/wiki/How-to-contribute>