Bioimage Analysis for Quantitative Microscopy

30.9-4.10.2024

Trainers:
Hanna Grobe
Elnaz Fazeli
Joanna Pylvänäinen
Sujan Ghimire
Stéphane Rigaud

*open to all interested

Locations:
Auditorium Biokemi, Biocity 2nd floor
Auditorium Biologi, Biocity 3rd floor
Putous Auditorium, Joki conference center

	Mon	Tue	Wed	Thu	Fri
9-10.00	Intro + Fiji Basics Hanna Grobe	**DL lecture** Joanna Pylvänäinen	QuPath Stéphane Rigaud	TrackMate + CellTracks Colab lecture Joanna Pylvänäinen	
10.00-10:15					Presentation: TBI image data team Pasi Kankaanpää
10-15-10.30	*break*	*break*	*break*	*break*	Project presentations
10.30-11.45	Fiji Basics Hanna Grobe	**DL demo annotation and training** Sujan Ghimire	QuPath Stéphane Rigaud	Cell tracking with TrackMate Joanna Pylvänäinen	
12-13	*Lunch (at own cost)*	*Lunch (at own cost)*	*Lunch (at own cost)*	*Lunch (at own cost)*	*Lunch (at own cost)*
13.00-13.15	Presentation: Euro-BioImaging Jiri Funda, Susanne Va	**Science talk *: Deep Learning in Microscopy** Guillaume Jacquemet	Science talk *: Deep Learning in Histopathology Pekka Ruusuvuori	Keynote talk *: How to not lie with charts - better data visualisations for life sciences Helena Jambor	Science talk *: Next-generation file formats, version control, and publishing your data Junel Solis
13.15-14	Fiji Macros Elnaz Fazeli				
14.14-15.15		**DL demo quality control** Joanna Pylvänäinen	QuPath Stéphane Rigaud	Track Analysis using CellTracksColab Hanna Grobe	Work with your own data
15.15-15.30	*break*	*break*	*break*	*break*	*break*
15.30-16.45	Fiji Macros Elnaz Fazeli	**DL demo apply to own data** Joanna Pylvänäinen	QuPath Stéphane Rigaud	GPU accelerated Fiji image processing Stéphane Rigaud	Work with your own data Goodbye and farewell
17.00-21.00				Course dinner in Mauno	

#ZeroCostDL4Mic

ZEROCOSTDL4MIC

What, where, how?

Can I also use it?

Joanna Pylvänäinen, joanna.pylvanainen@abo.fi

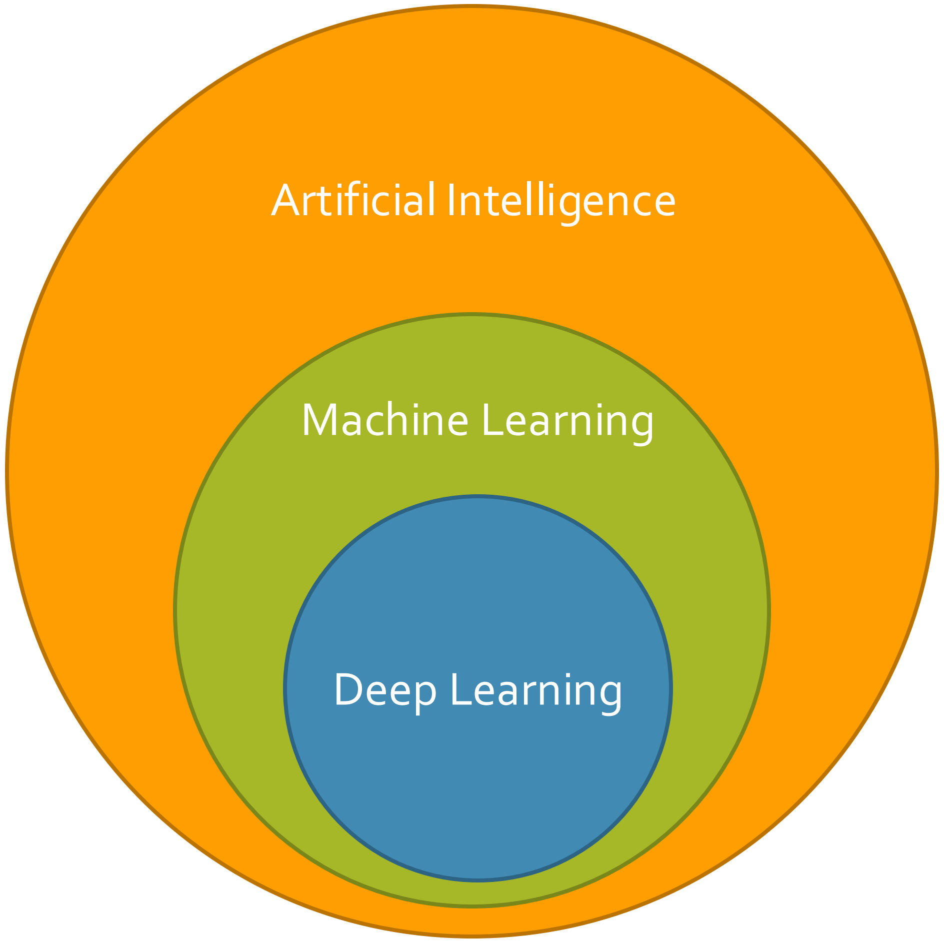
Image analysis course, Turku

October 1 2024

Materials adapted from Guillaume Jacquemet and Junel Solis

Contents

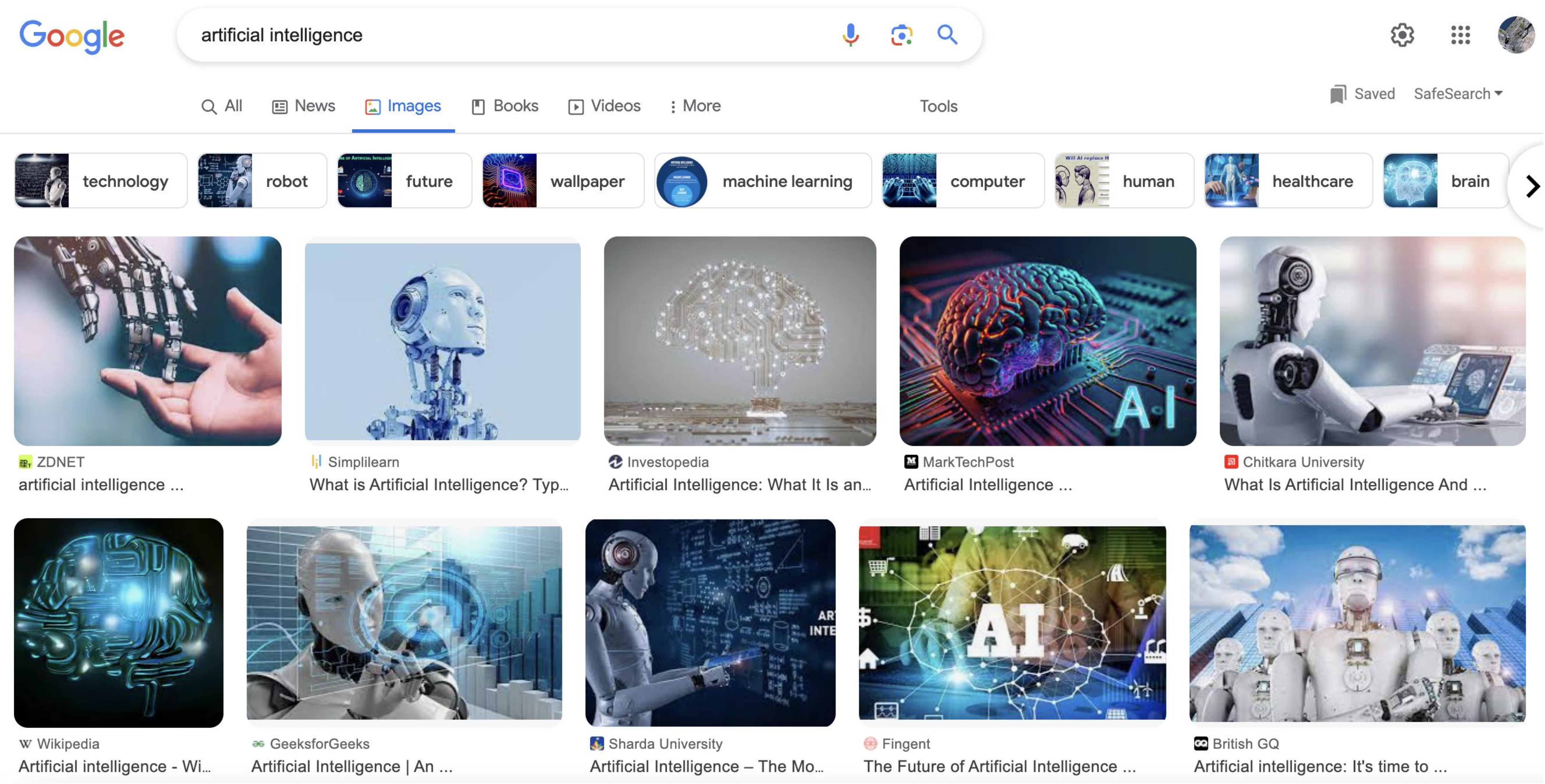
- What is Deep learning?
- What is ZerocostDL4Mic
- What can be done using ZerocostDL4Mic
- Example experiment



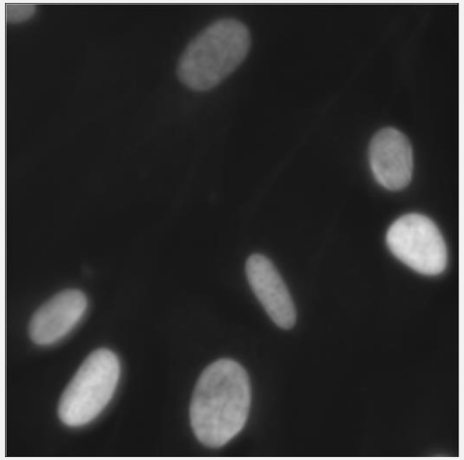
Artificial intelligence (AI) is a field of computer science that focuses on developing intelligent machines that can perform tasks that typically require human intelligence.

Machine learning (ML) is a subset of AI that focuses on the development of algorithms that can learn patterns and insights from data, and then use this knowledge to make predictions or decisions.

Deep learning (DL) is a subset of ML that focuses on the development of artificial neural networks, which are modeled after the structure and function of the human brain.


Google
artificial intelligence
All
News
Images
Books
Videos
More
Tools
Saved
SafeSearch
technology
robot
future
wallpaper
machine learning
computer
human
healthcare
brain
ZDNET
artificial intelligence ...
Simplilearn
What is Artificial Intelligence? Typ...
Investopedia
Artificial Intelligence: What It Is an...
MarkTechPost
Artificial Intelligence ...
Chitkara University
What Is Artificial Intelligence And ...
Wikipedia
Artificial intelligence - Wi...
GeeksforGeeks
Artificial Intelligence | An ...
Sharda University
Artificial Intelligence – The Mo...
Fingent
The Future of Artificial Intelligence ...
British GQ
Artificial intelligence: It's time to ...

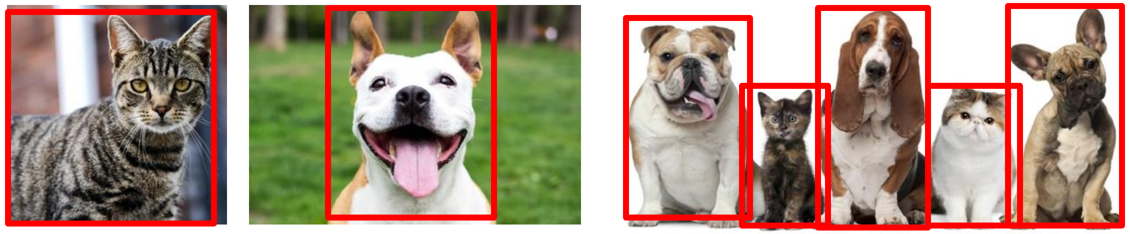
TRAINED FOR A SPESIFIC TASK



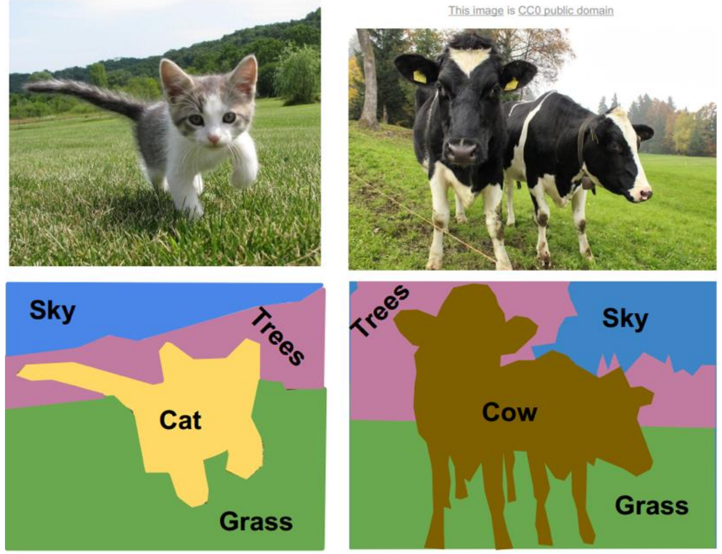
How to get from fluorescence image to label image?

Common Deep learning tasks

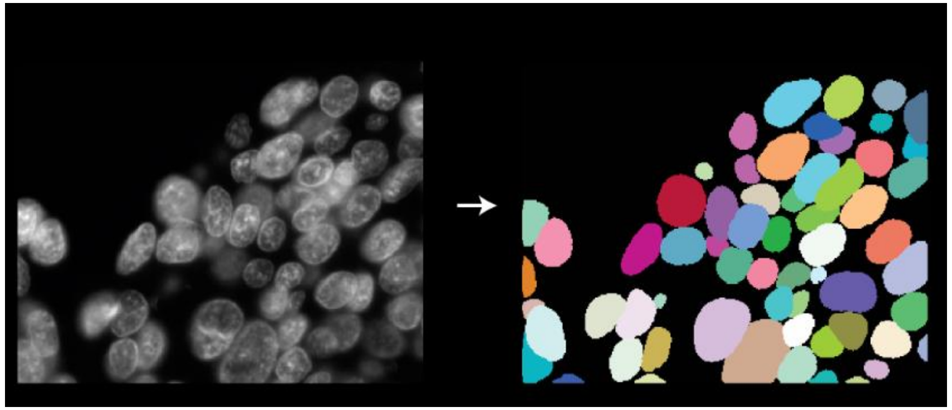
Classification



Object Detection

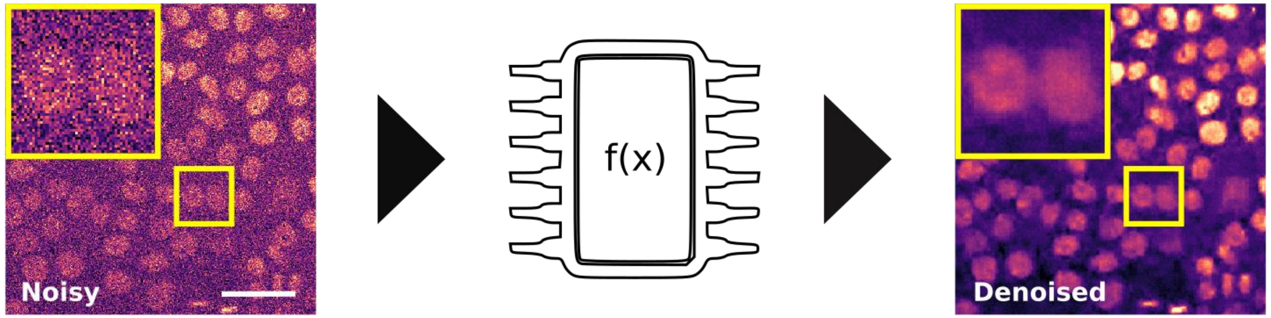


Semantic Segmentation



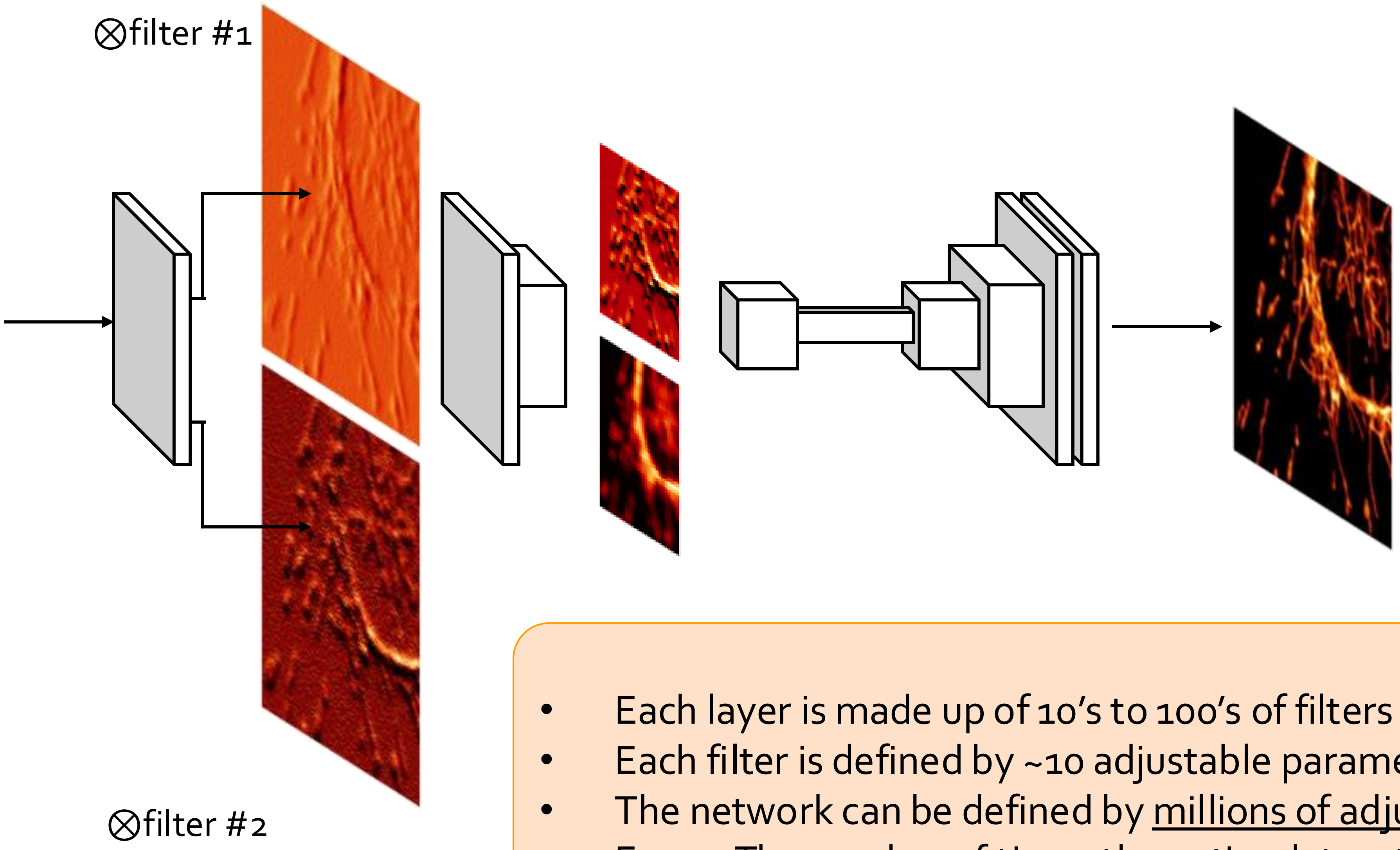
Instance segmentation

Classical algorithm



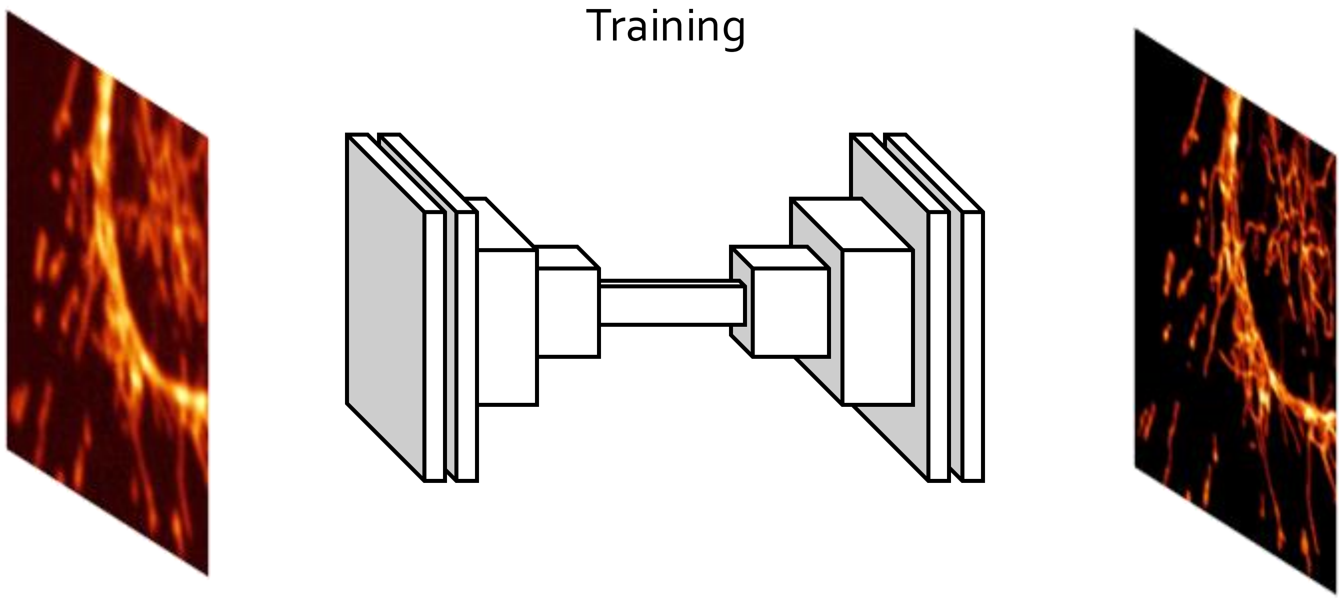
DEEP LEARNING IN MICROSCOPY ?

Laine et al. (Nature Methods, 2021)



- 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 <u>millions of adjustable weights</u>
- Epocs: The number of times the entire dataset is passed forward and backward through the neural network.

What happens during training?



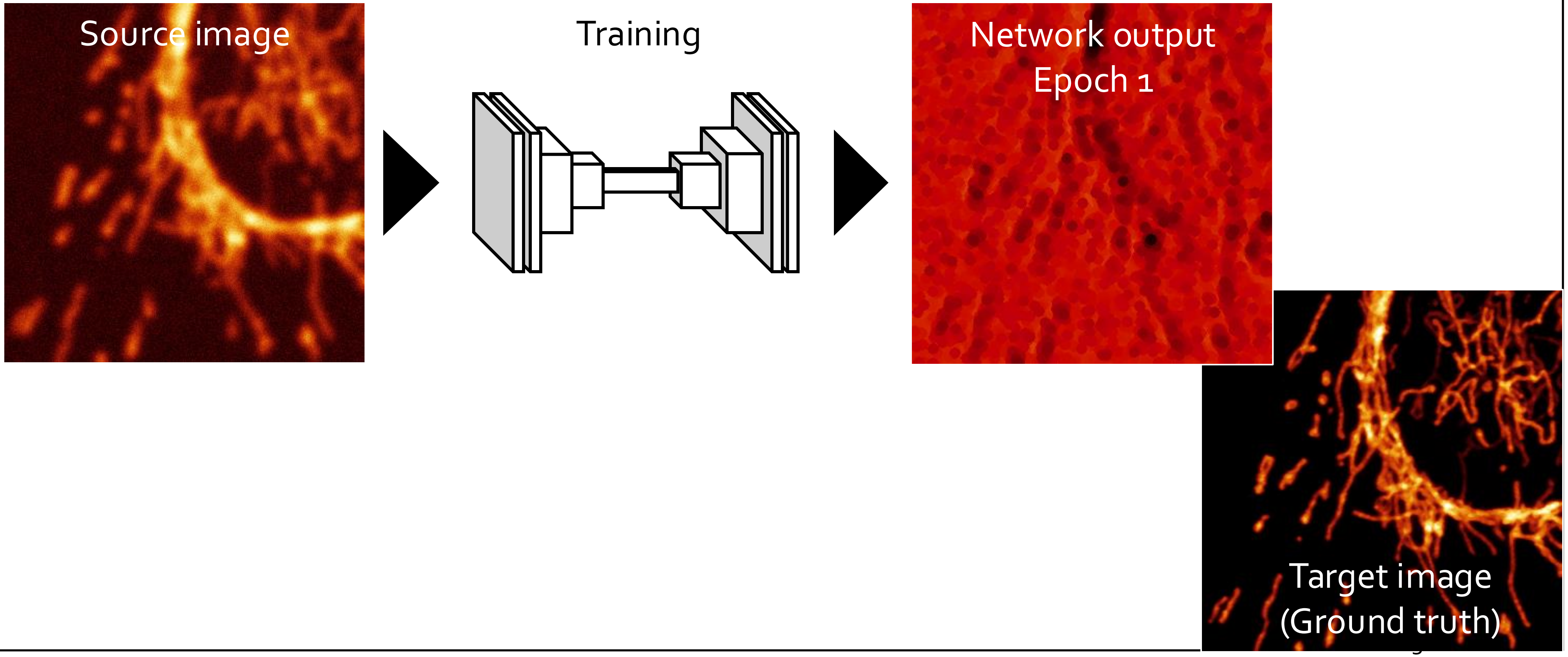
How does the neural network learn?

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)



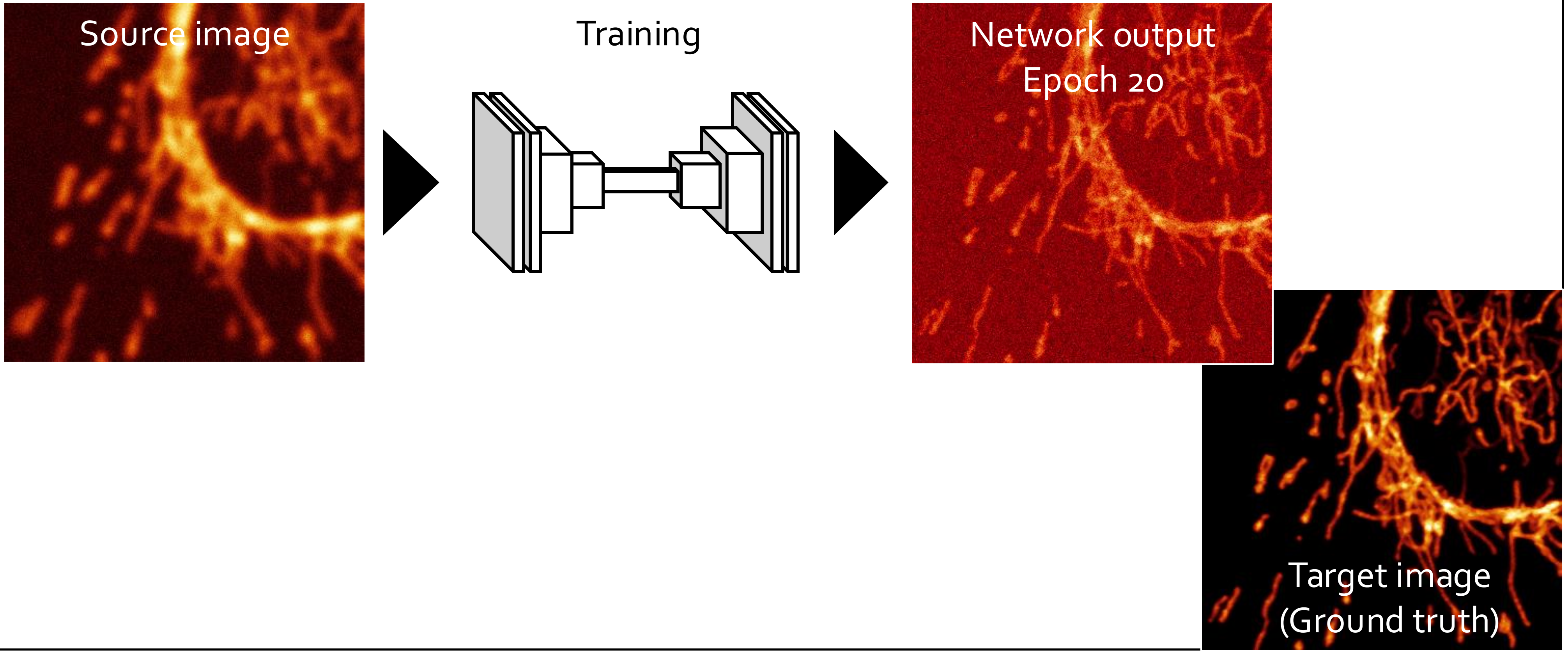
How does the neural network learn?

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)



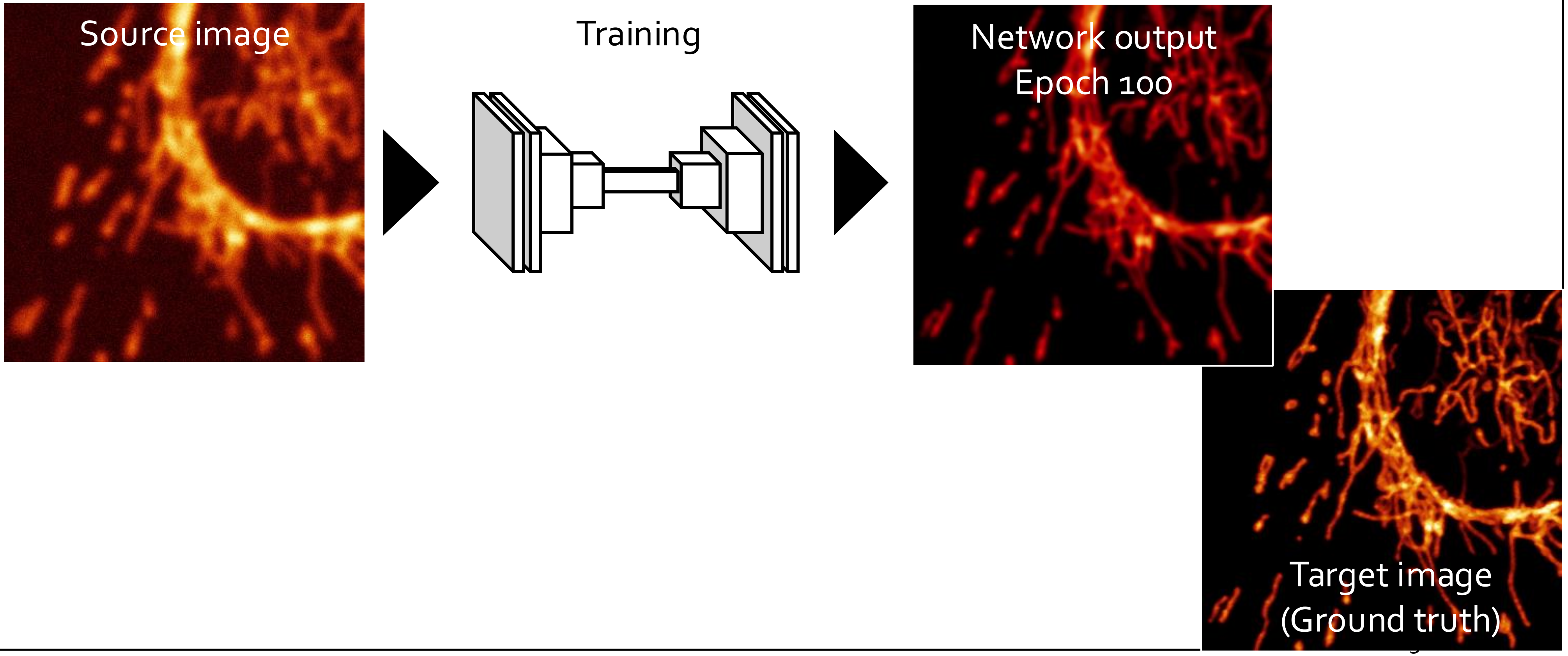
How does the neural network learn?

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)



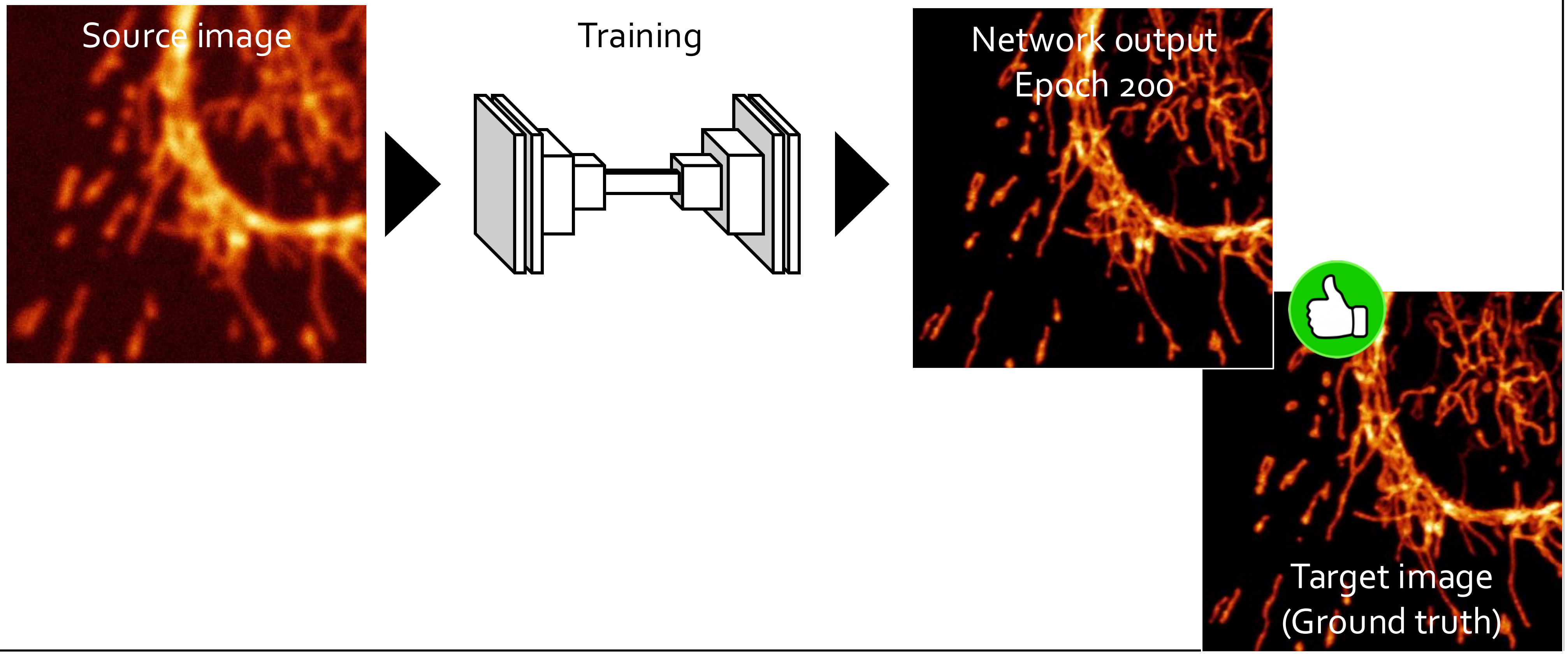
How does the neural network learn?

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)

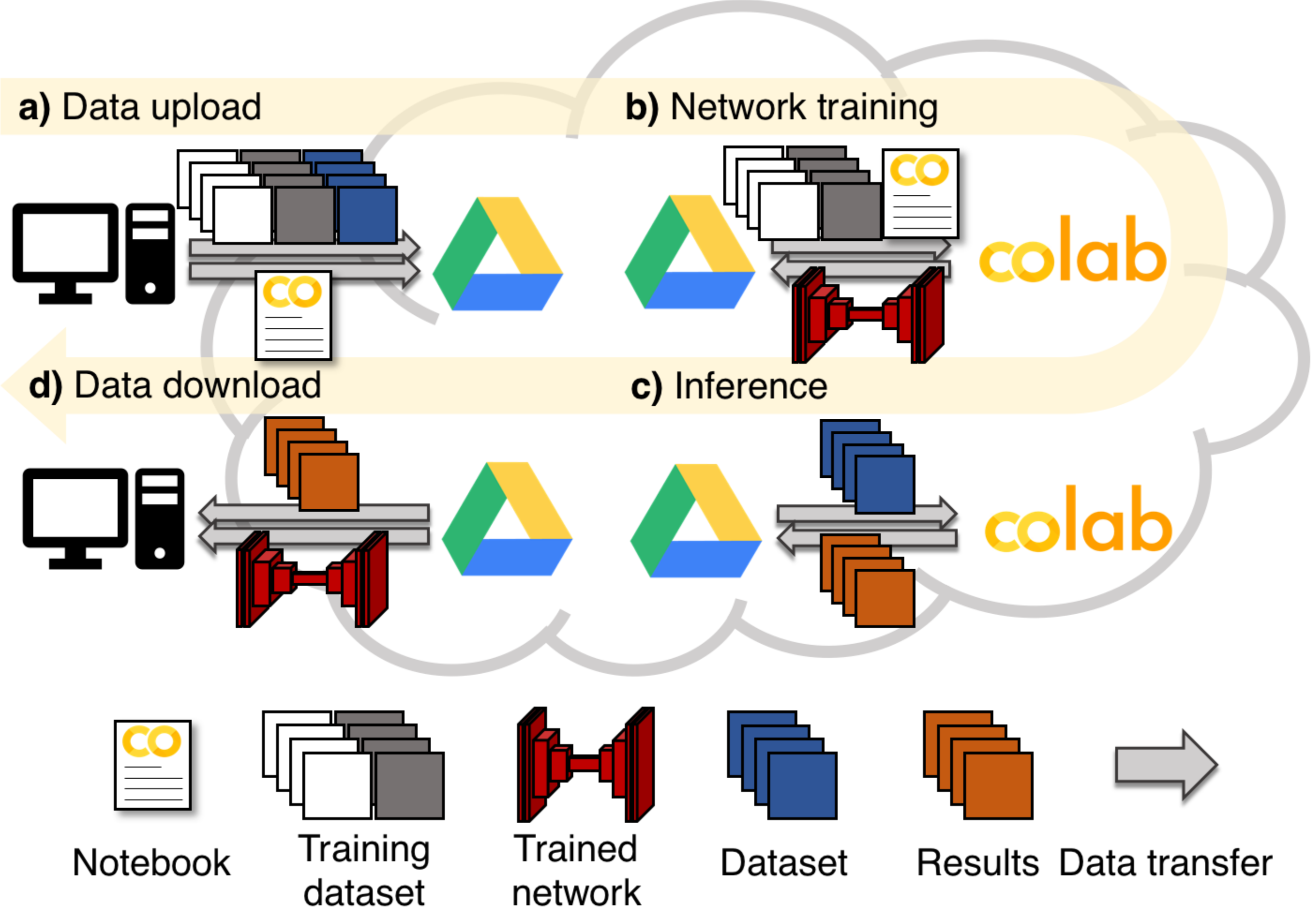


SOUNDS GREAT!
I WANT TO TRAIN A NEURONAL NETWORK

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

#ZeroCostDL4Mic

ZeroCostDL4Mic to the rescue!



Guillaume Jacquemet

Ricardo Henriques

Lucas von Chamier

Romain Laine

- A toolbox for the training and implementation of common Deep Learning approaches to microscopy imaging.
- It uses the GPU provided by Google Colab.
- Free! (with limitations)

https://github.com/HenriquesLab/ZeroCostDL4Mic/wiki

Segmentation networks

Network	Paper(s)	Tasks	Status	Last test	Link to example training and test dataset	Direct link to the notebook in Colab
U-Net (2D)	here and here	Binary segmentation	Fully supported	23/02/23 ✅ working (EGM)	here	Open in Colab
U-Net (3D)	here	Binary segmentation	Fully supported	27/02/23 ✅ working (EGM)	EPFL dataset	Open in Colab
U-Net (2D) multilabel	here and here	Semantic segmentation	Under beta-testing	27/02/23 ✅ working (EGM)	here	Open in Colab
DenoiSeg	here	Joint denoising and binary segmentation	Fully supported	⚠ broken (no GPU) (GJ)	Available soon	Open in Colab
StarDist (2D)	here and here	Instance segmentation	Fully supported	19/05/23 ✅ working (EGM)	here	Open in Colab
StarDist (3D)	here and here	Instance segmentation	Fully supported	07/10/22 ✅ working (GJ)	from Stardist github	Open in Colab
Cellpose (2D and 3D)	here	Instance segmentation (Cells or Nuclei)	Fully supported	08/10/22 ✅ working (GJ)	Coming soon!	Open in Colab
SplineDist (2D)	here	Instance segmentation	Fully supported	07/10/22 ✅ working (GJ)	here	Open in Colab
EmbedSeg (2D)	here	Instance segmentation	Under beta-testing	01/01/23 ✅ working (AR)	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	Last test	Link to example training and test dataset	Direct link to the notebook in Colab
Noise2Void (2D)	here	Self-supervised denoising	Fully supported	06/04/23 ✅ working (GJ)	here or here	Open in Colab
Noise2Void (3D)	here	Self-supervised denoising	Fully supported	07/10/22 ✅ working (GJ)	here	Open in Colab
CARE (2D)	here	Supervised denoising	Fully supported	03/04/23 ✅ working (IH)	here or here	Open in Colab
CARE (3D)	here	Supervised denoising	Fully supported	07/10/22 ✅ working (GJ)	here	Open in Colab
3D-RCAN	here	Supervised denoising	Under beta-testing	⚠ broken (no GPU)	here	Open in Colab
DecoNoising (2D)	here	Self-supervised denoising	Under beta-testing	07/10/22 ✅ working (GJ)	here or here	Open in Colab

Super-resolution microscopy networks

Network	Paper(s)	Tasks	Status	Last test	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	08/10/22 ✅ working (GJ)	Training data simulated in the notebook or available from here	Open in Colab
DFCAN	here	image upsampling	Under beta-testing	08/10/22 ✅ working (GJ)	here	Open in Colab
WGAN	here	image upsampling	Under beta-testing	22/09/22 ✅ working (IvanHidalgo & EGM)	here	Open in Colab

Object detection networks

Network	Paper(s)	Tasks	Status	Last test	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

Tools

Network	Paper(s)	Tasks	Status	Last test	Link to example training and test dataset	Direct link to the notebook in Colab
Augmentor	here	Image augmentation	Fully supported	12/08/22 ✅ working (GJ)	None	Open in Colab
Quality Control	here	Error mapping and quality metrics estimation	Fully supported	12/08/22 ✅ working (GJ)	None	Open in Colab
Mounting DropBox or MEGA in Google Colab	None	Tutorial explaining how to mount a Dropbox or MEGA account in Google Colab using Rsync	Under beta-testing		None	Open in Colab

BioImage.io notebooks

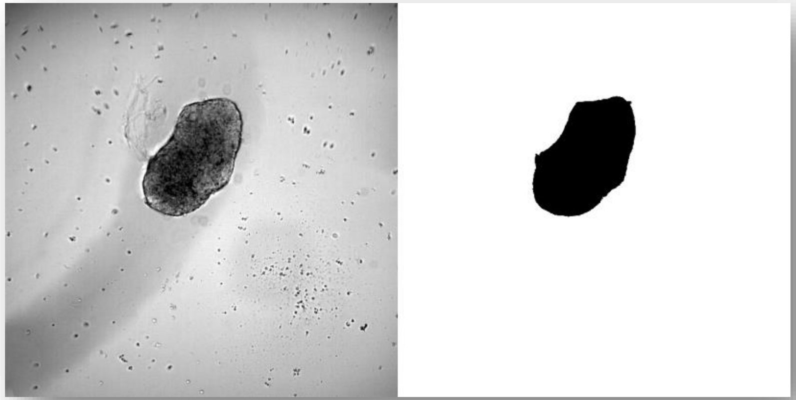
Networks that are compatible with BioImage.IO and can be used in ImageJ via deepImageJ or Ilastik. The trained models in these notebooks are also exported in the BioImage.IO format and can be uploaded to the BioImage Model Zoo. Check our user guide to learn how to use the resources in the BioImage Model Zoo with ZeroCostDL4Mic.

Network	Paper(s)	Task	Last test	Link to example training and test dataset	Direct link to the notebook in Colab
StarDist (2D)	StarDist: here and here, and DeepImageJ and BioImage Model Zoo	Nuclei segmentation	09/08/22 ✅ working (EGM)	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	10/08/22 ✅ working (EGM)	Training data simulated in the notebook or available from here	Open in Colab
U-Net (2D)	U-Net and DeepImageJ and BioImage Model Zoo	Segmentation	23/02/23 ✅ working (EGM)	ISBI challenge or here	Open in Colab
U-Net (2D) multilabel	here and here and DeepImageJ and BioImage Model Zoo	Semantic segmentation	27/02/23 ✅ working (EGM)	here	Open in Colab
U-Net (3D)	3D U-Net and DeepImageJ and BioImage Model Zoo	Segmentation	27/02/23 ✅ working (EGM)	EPFL dataset	Open in Colab

Network	Paper(s)	Tasks	Status	Last test	Link to example training and test dataset	Direct link to the notebook in Colab
U-Net (2D)	here and here	Binary segmentation	Fully supported	23/02/23 ✅ working (EGM)	here	Open in Colab

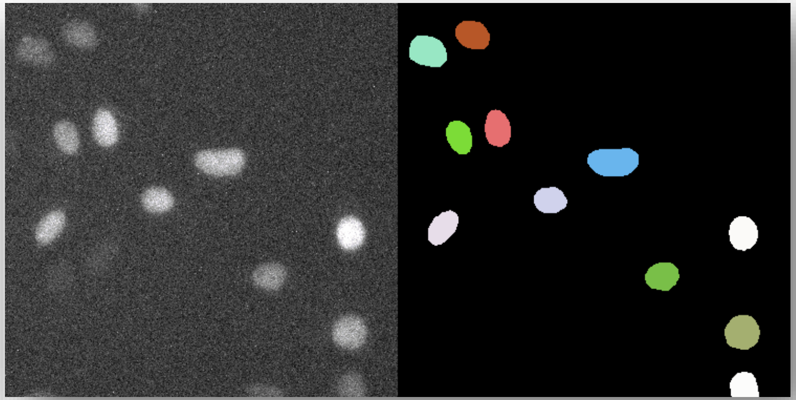
Segmentation tools available in ZeroCost

UNET (2D, 3D, 2D multi-label)



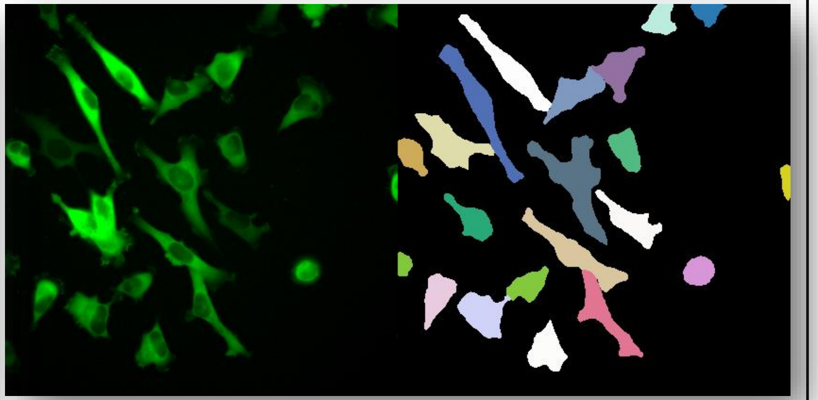
Gastruloid development
UNET 2D segmentation
Maja Solman (unpublished)

StarDist 2D and 3D



Migrating breast cancer cell nuclei
Cellpose 2D segmentation

Cellpose 2D and 3D

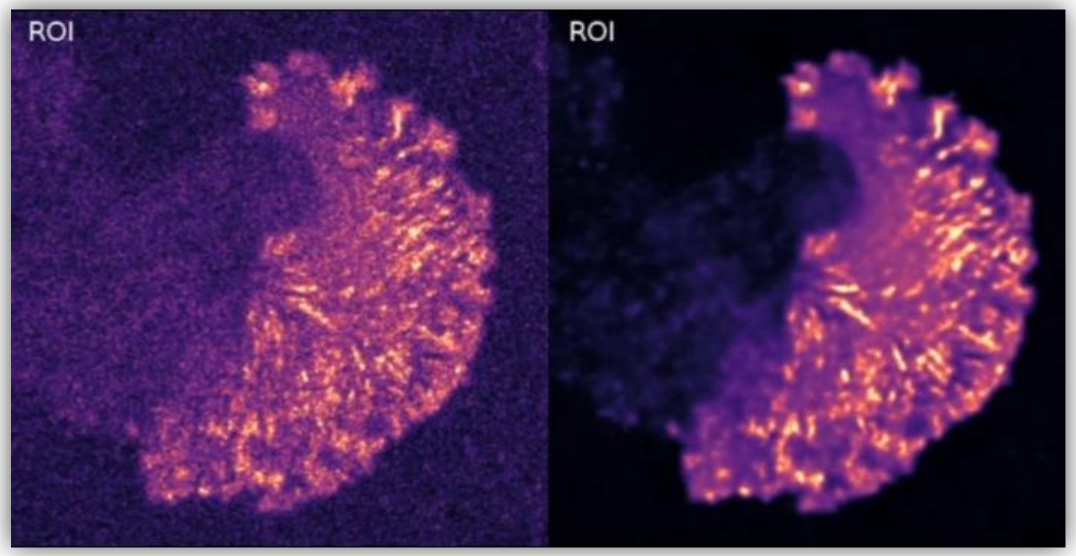


Migrating breast cancer cells
Cellpose 2D segmentation

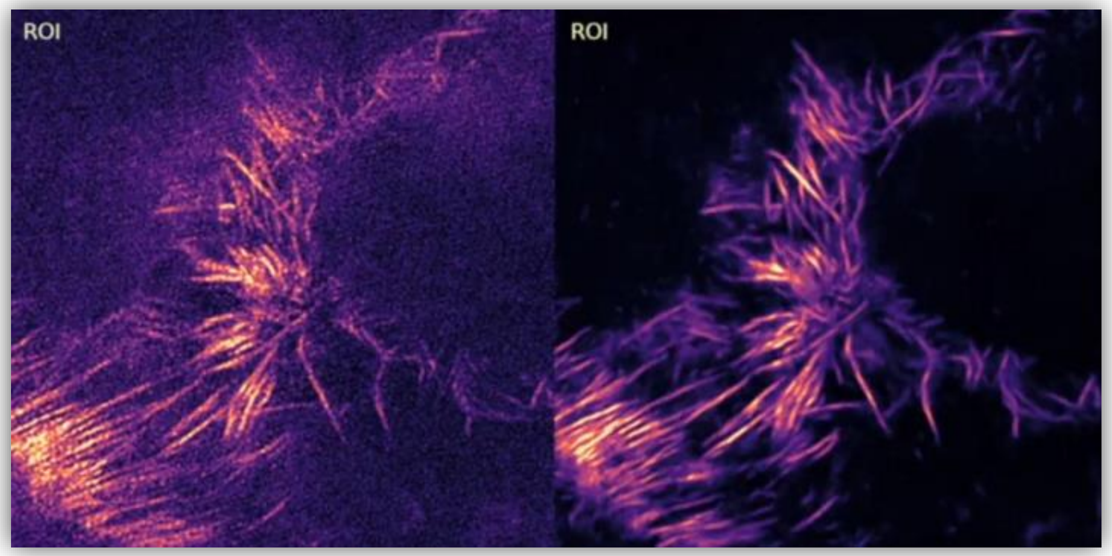
To train a UNET, StarDist or Cellpose network you need matching images of source image and the corresponding masks.

Denoising tools available in ZeroCost

Noise2Void (2D, 3D)



CARE (2D, 3D)

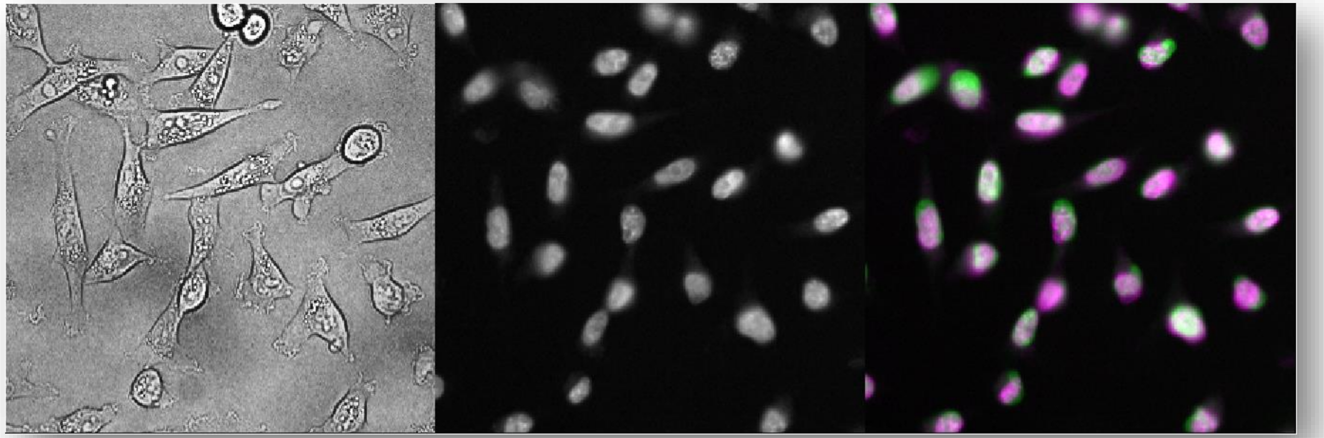


To train a Noise2Void network, all you need are your noisy images. One noisy image is even sufficient to train a network.

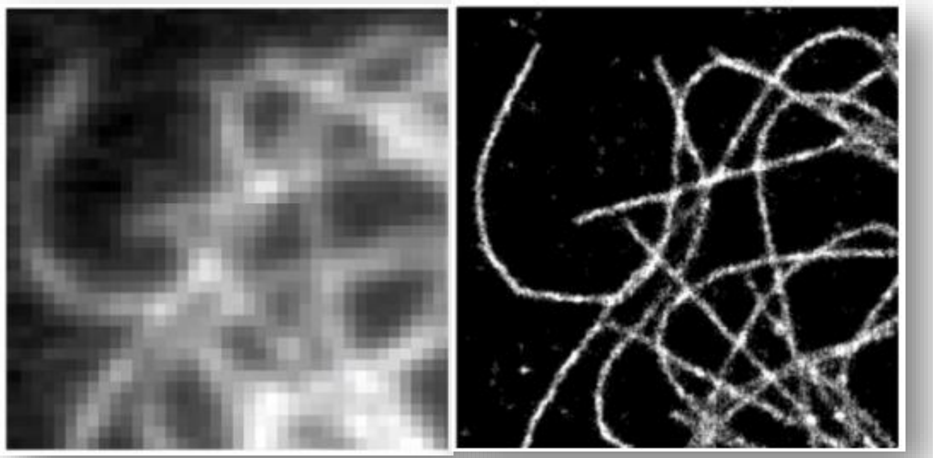
To train a CARE network you need a dataset containing matching images, for instance low signal-to-noise ratio (SNR) images and high SNR.

Other tools available in ZeroCost

Virtual labelling (pix2pix)



Super-resolution (DeepSTORM)



Colab Notebooks - Google Drive | Pasteur_of StarDist_2D_ZeroC

colab.research.google.com/drive/1m0Mz9FHh4VRUXerDga1aNuj-sO1jaKHf

My Drive - Google... | Colabs | dubli | Fast4DReg - Goo... | Lab meetings | Zerocost-protocol | Phototoxicity

Pasteur_of StarDist_2D_ZeroCostDL4Mic.ipynb

File Edit View Insert Runtime Tools Help Last edited on 26 May

+ Code + Text

StarDist (2D)

StarDist 2D is a deep-learning method that can be used to segment cell nuclei from bioimages and was first published by Schmidt *et al.* in 2018, on arXiv. It uses a shape representation based on star-convex polygons for nuclei in an image to predict the presence and the shape of these nuclei. This StarDist 2D network is based on an adapted U-Net network architecture.

This particular notebook enables nuclei segmentation of 2D dataset. If you are interested in 3D dataset, you should use the StarDist 3D notebook instead.

Disclaimer:

This notebook is part of the Zero-Cost Deep-Learning to Enhance Microscopy project (https://github.com/HenriquesLab/DeepLearning_Collab/wiki). Jointly developed by the Jacquemet (link to https://cellmig.org/) and Henriques (https://henriqueslab.github.io/) laboratories. The BioImage Model Zoo export was jointly developed by Estibaliz Gómez de Mariscal (deepImageJ team).

This notebook is largely based on the paper:

Cell Detection with Star-convex Polygons from Schmidt *et al.*, International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Granada, Spain, September 2018. (https://arxiv.org/abs/1806.03535)

and the 3D extension of the approach:

Star-convex Polyhedra for 3D Object Detection and Segmentation in Microscopy from Weigert *et al.* published on arXiv in 2019 (https://arxiv.org/abs/1908.03636)

The Original code is freely available in GitHub: https://github.com/mpicbg-csbd/stardist

The guidelines to use the trained network in ImageJ with deepImageJ are given in the following paper:

DeepImageJ: a user-friendly environment to run deep learning models in ImageJ, bioRxiv (2019) by *Estibaliz Gómez-de-Mariscal, Carlos García-López-de-Haro, Wei Ouyang, Laurène Donati, Emma Lundberg, Michael Unser, Arrate Muñoz-Barrutia and*

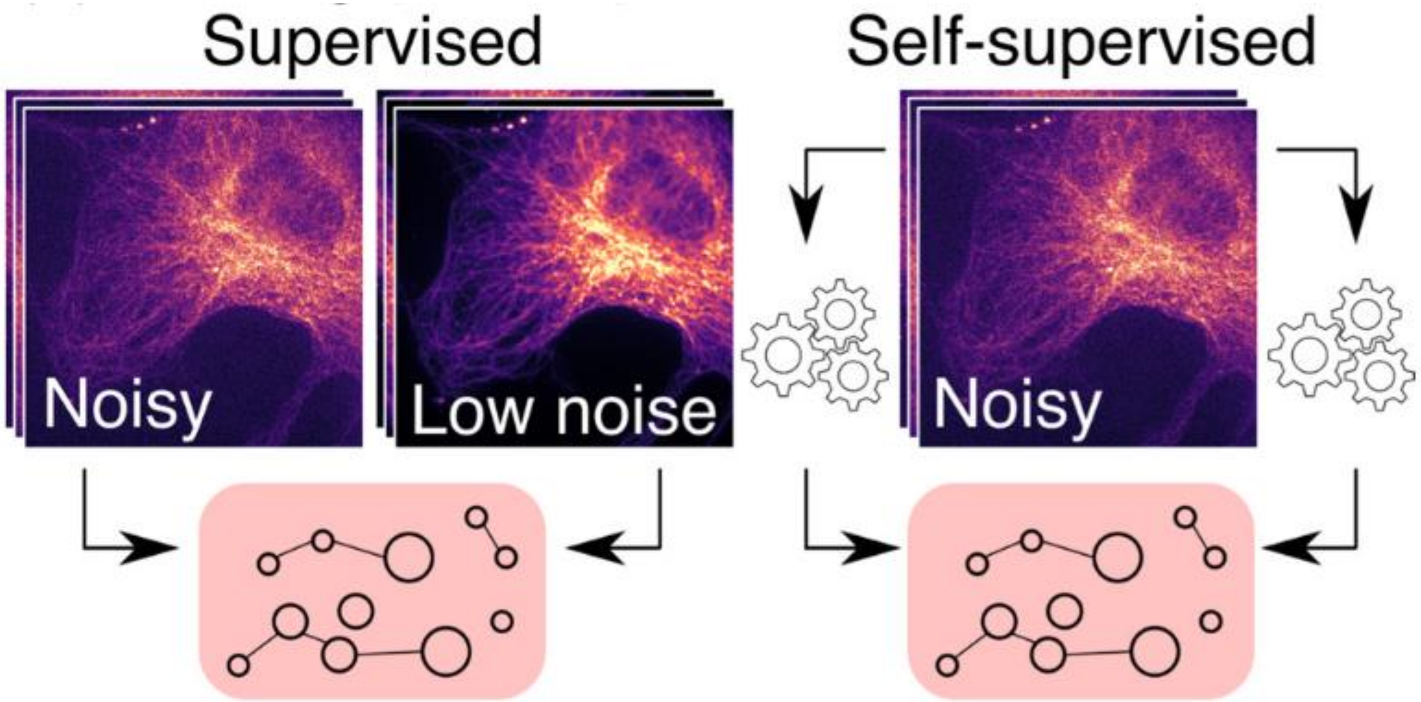
Steps for model training in ZeroCost

1. Create training dataset
2. Upload data to G-drive
3. Install dependencies + check GPU access
4. Define paths to training data
5. Train
6. Assess the quality of your model
7. Run predictions on unseen data

Or any other tool

#ZeroCostDL4Mic

All starts with training datasets



U-Net, CARE, pix2pix, fnet, StarDist, Deep-STORM, DenoiSeg, 3D-RCAN, YOLOv2

Noise2Void, CycleGAN

Each network has specific data requirement to train

- Type of training
- Image format
- Info on how to generate datasets available in ZeroCost Wiki
- Check each notebook for more information

Model from Scratch vs. Transfer Learning

- When building deep learning models, there are two primary approaches:

Model from scratch	Transfer learning
Requires large amounts of labeled data	Leverage Pre-trained Models
Longer Training Time	Faster Training
Flexibility to tailor the model for specific use cases.	Less Data Required
	Fine-tuning pre-trained models significantly reduces time and computational cost.

Data Augmentation in Deep Learning

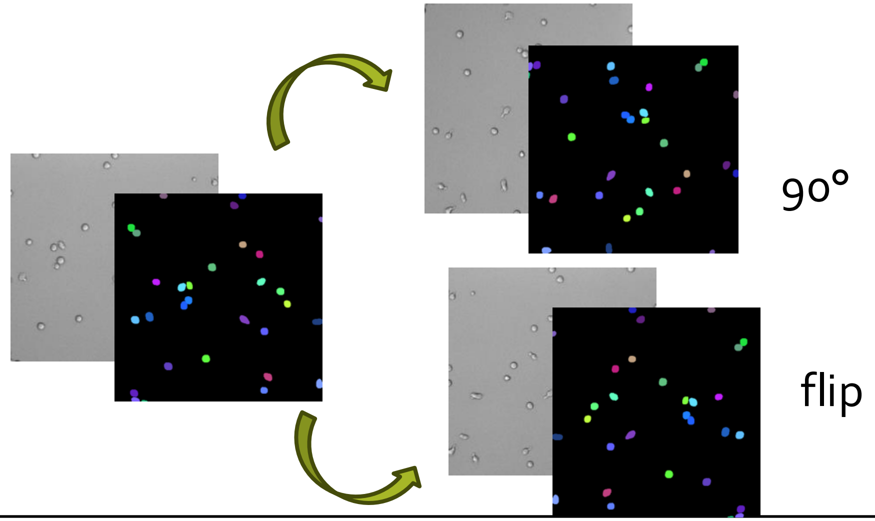
- Data augmentation is a technique used to artificially increase the diversity of a training dataset by applying various transformations

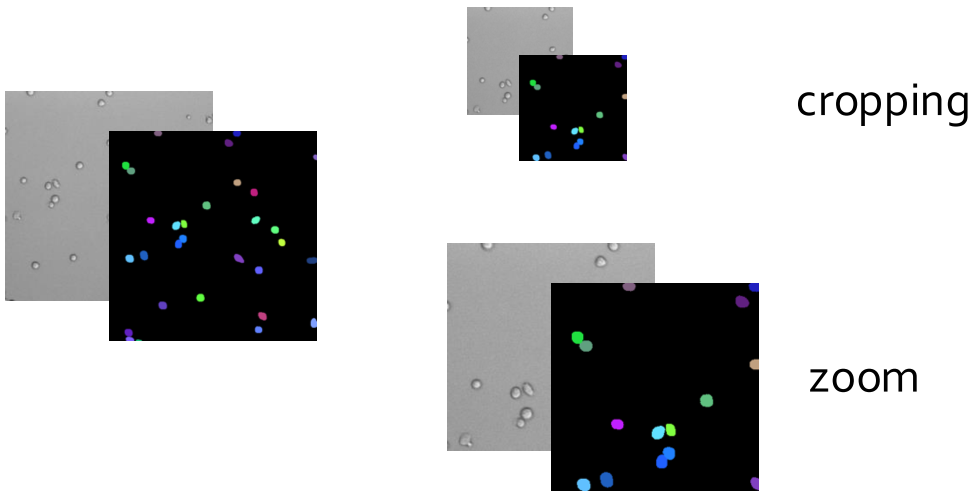
Rotation and Flipping

Rotating images by various degrees or flipping them horizontally/vertically.

Scaling and Cropping

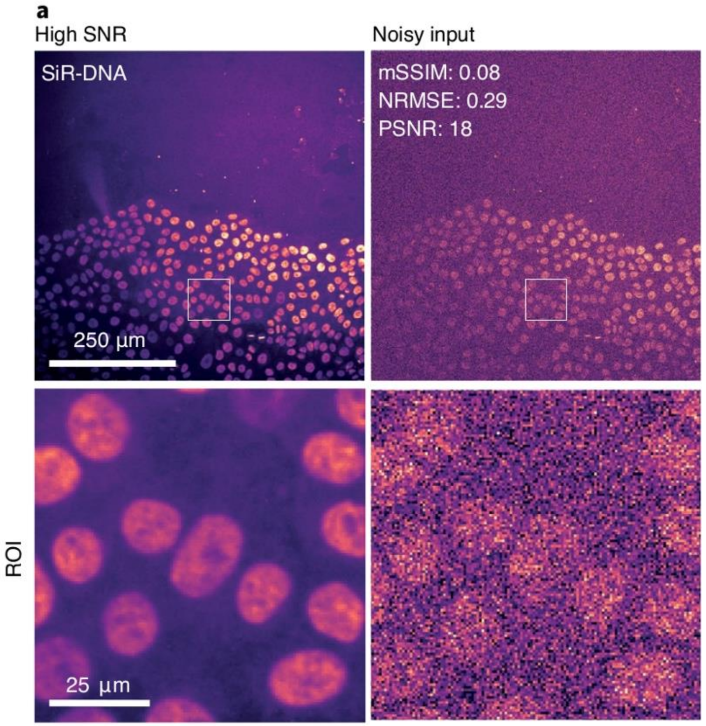
Adjusting the size or cropping parts of images to simulate zoom-in/out scenarios.





The performance is dependent on the training data

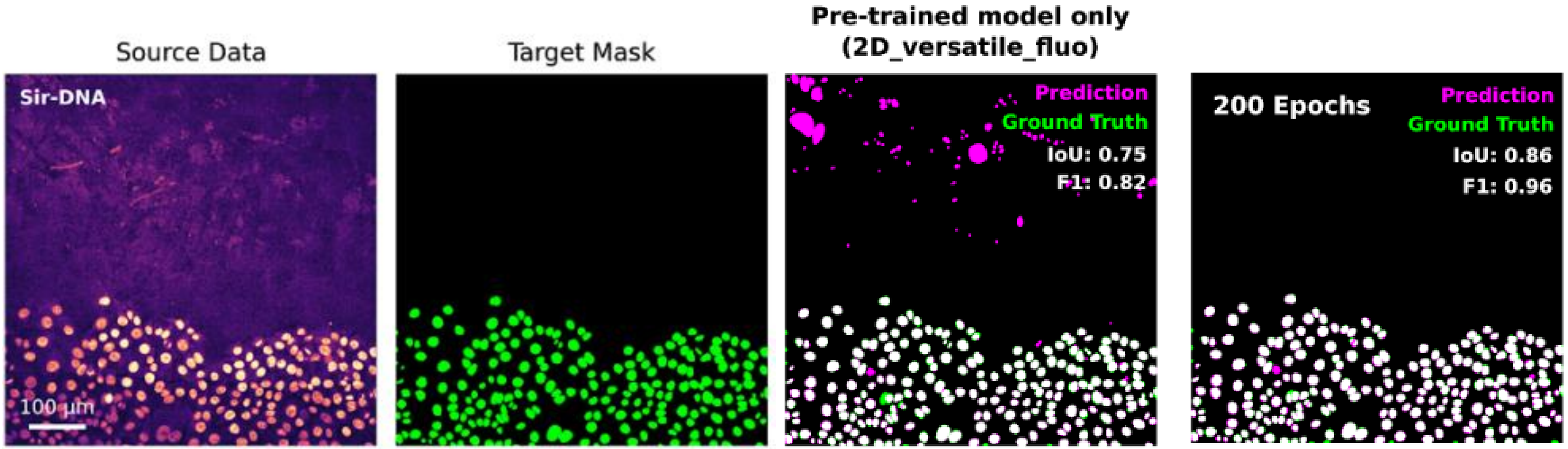
- The dataset used to train a DL network is key
- Always validate your models using ground truth data



Denoising using CARE

The performance is dependent on the training data

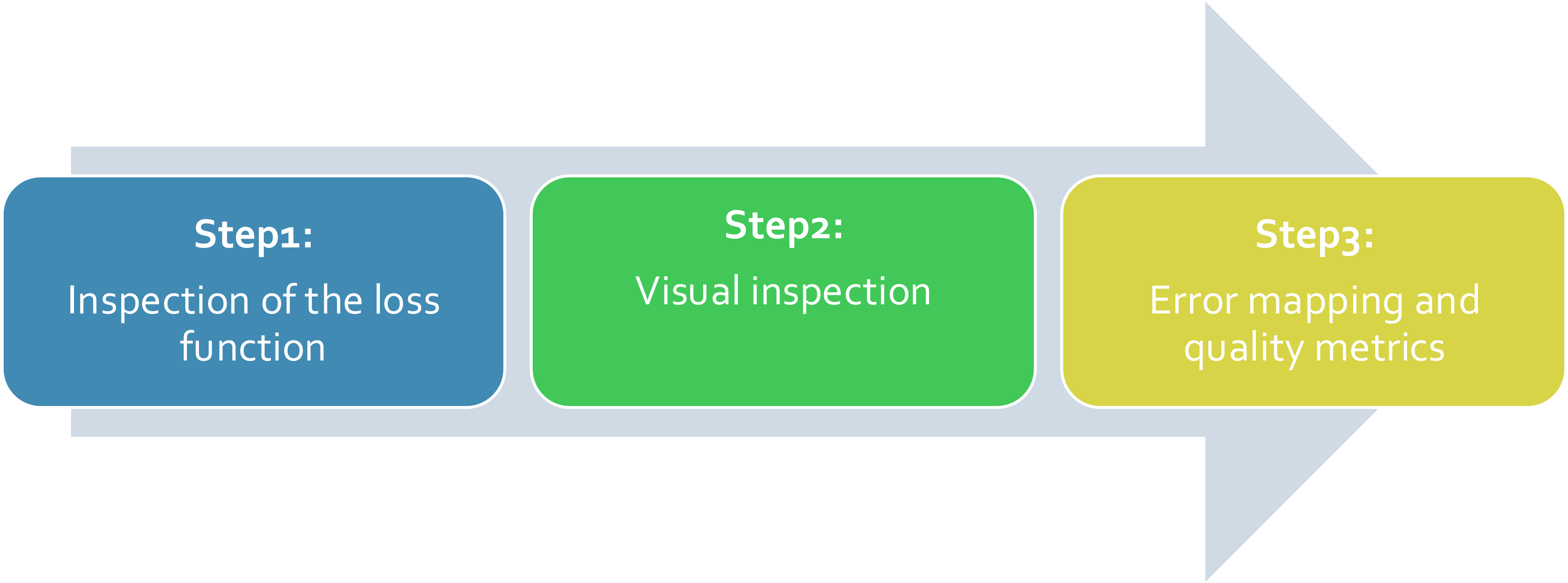
- The dataset used to train a DL network is key
- Always validate your models using ground truth data



[Chamier., *et al.* Nature Communications (2021)]

Segmentation using StarDist

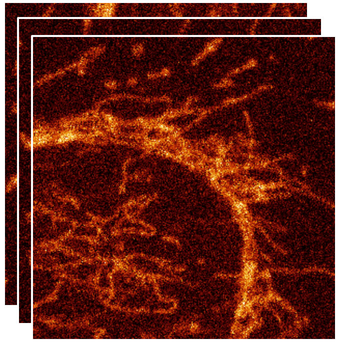
Steps in Quality control

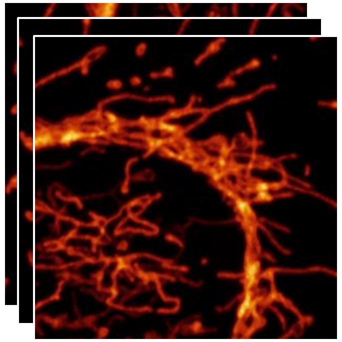


Step #1.1: Validation split

Training dataset is divided into training and validation data

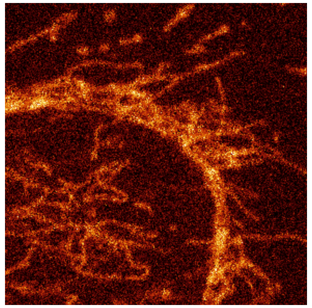
Training data

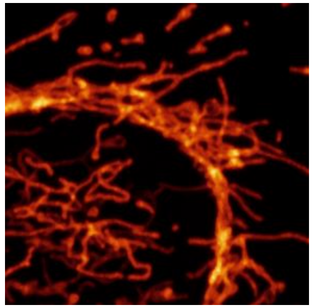




90 %

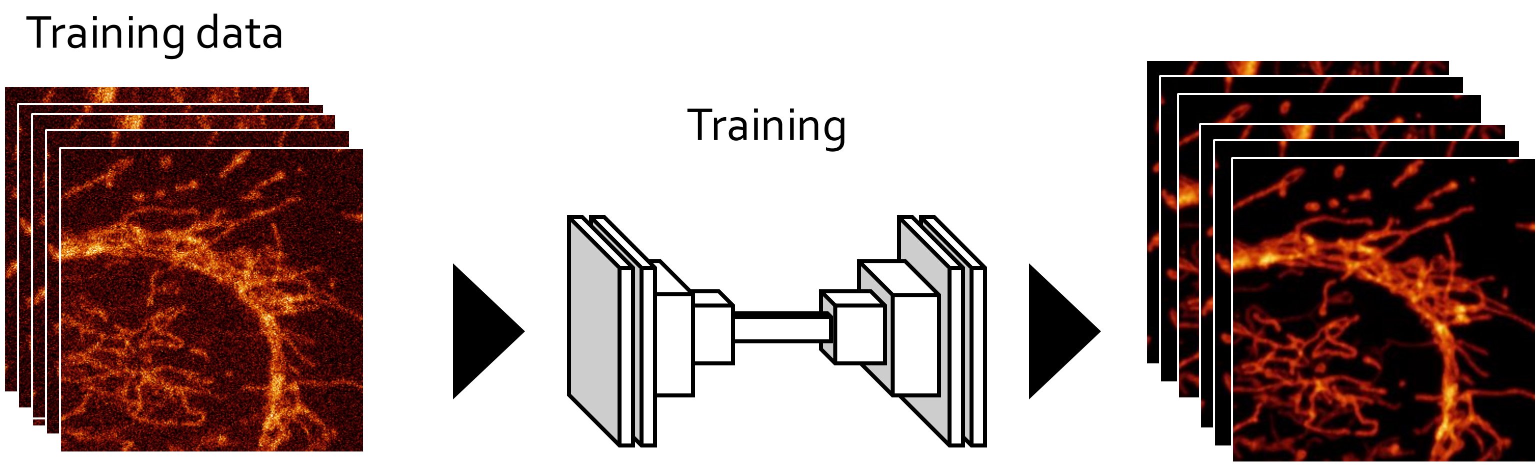
Validation data





10 %

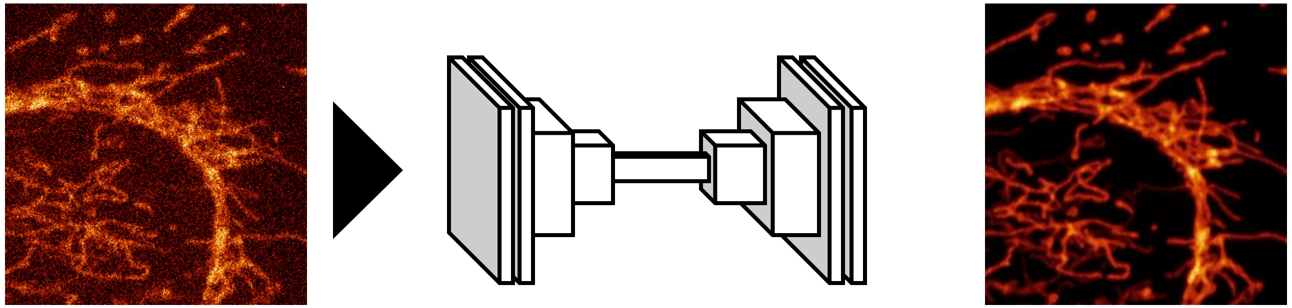
Step #1.2: Learning from the training data



EPOCH: One round of training. A model is generated at the end of each EPOCH
EPOCH are divided into multiple "steps"
Batch size defines how many images are seen by the network at each step
Score: **Training loss: How well the model performs on the training data**

Step #1.3: Testing the model on the validation data

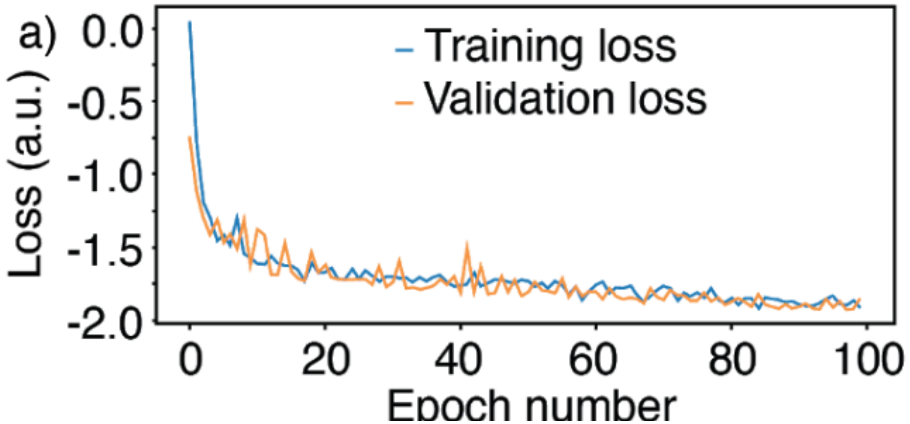
Validation data

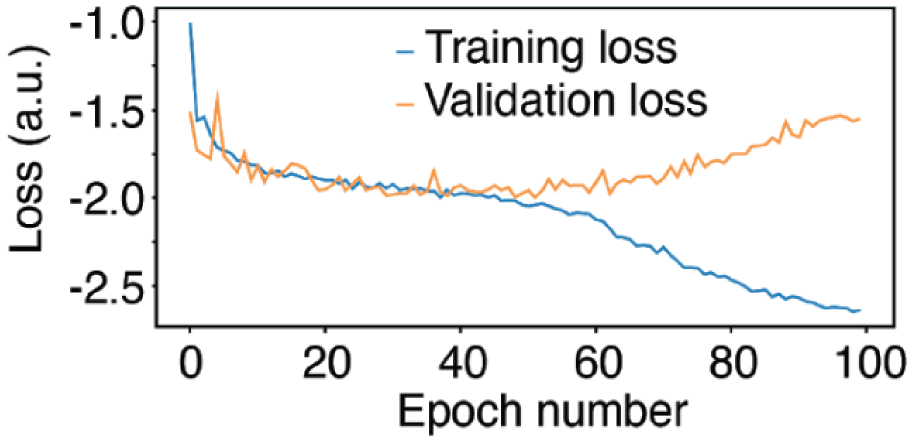


At the end of every EPOCH
Assess how well the model is performing on unseen data
Score: **Validation loss: How well the model performs on the validation data**

Step #1.4: Repeat

- Train model for hundred of EPOCH





Quality Control 1:
Inspection of the loss function

Always evaluate the training progress, compare the training loss with the validation loss.

Training loss describes an error value after each epoch for the difference between the model's prediction and its ground-truth target.

Validation loss shows how well the network performs on the validation data.

Decreasing Training Loss and Validation Loss:

- **Action**: Continue training for more epochs, train for more epochs until the validation loss plateaus.

Curves Are Flattening Out:

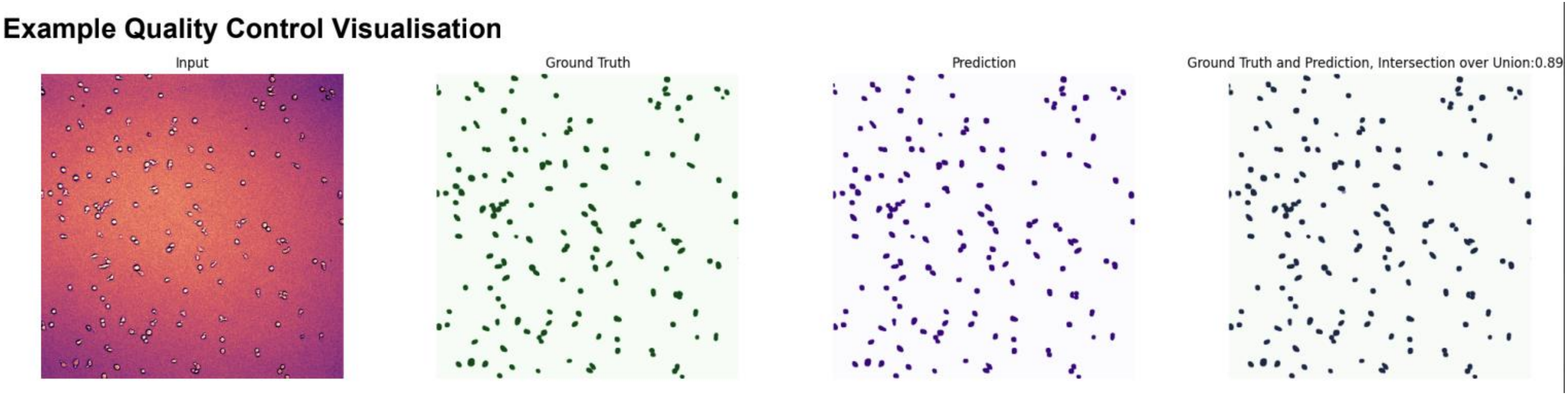
- **Action**: No more training may be needed.

Validation Loss Increases While Training Loss Decreases:

- **Diagnosis**: The network is overfitting.
  - Overfitting occurs when the model is memorizing the training data rather than generalizing well to unseen data.

- **Solutions to Overfitting:**
  1. Increase the size of the training dataset
  2. Augment your data

Quality Control Step #2: Visual inspection



Quality Control Step #3: Error mapping and quality metrics estimation

image name	Prediction v. GT Intersection over Union	false positive	true positive	false negative	precision	recall	accuracy	f1 score	n_true	n_pred	mean_true_score	mean_matched_score	panoptic_quality
Training_source_ICAM 1-1.tif	0.89414007	2	153	2	0.987096	0.98709	0.974522	0.987096	155	155	0.873075	0.884487	0.873075

IoU: percent overlap between the target mask and your prediction output. **The closer to 1, the better the performance.** This metric can be used to assess the quality of your model to accurately predict nuclei. (whole image)

- “true positive” = When a segmented object has an IuO above 0.5 (compared to the corresponding ground truth)
- “**false positive**” = “n_pred” - “true positive”
- “false negative” = “n_true” - “true positive”

The mean_matched_score is the mean IoUs of matched true positives.

f1 score = 2 * the Area of Overlap divided by the total number of pixels in both images.

Export in BioImage Model Zoo format

Estibaliz Gómez de Mariscal

5.3. Export your model into the BioImage Model Zoo format

This section exports the model into the BioImage Model Zoo format so it can be used directly with deepImageJ or Ilastik. The new files will be stored in the model folder specified at the beginning of Section 5.

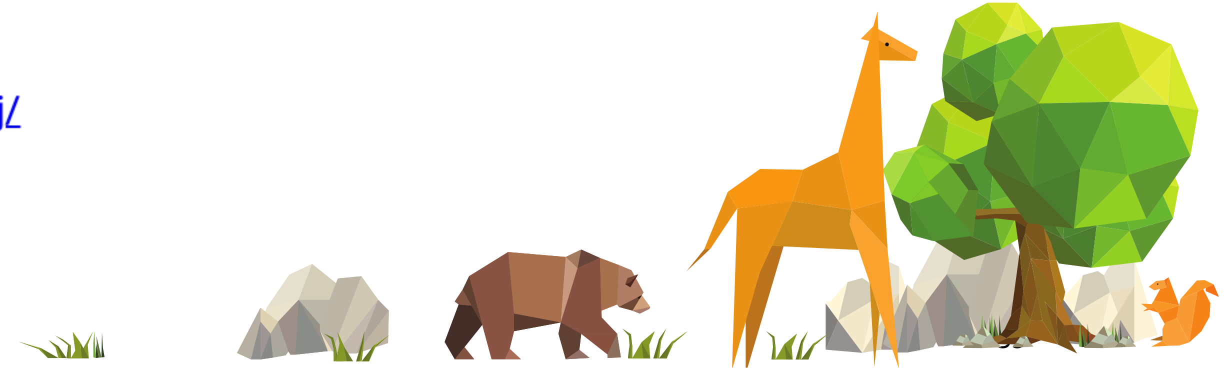
Once the cell is executed, you will find a new zip file with the name specified in `trained_model_name.bioimage.io.model`.

To use it with deepImageJ, download it and install it suing DeepImageJ Install Model > Install from a local file.

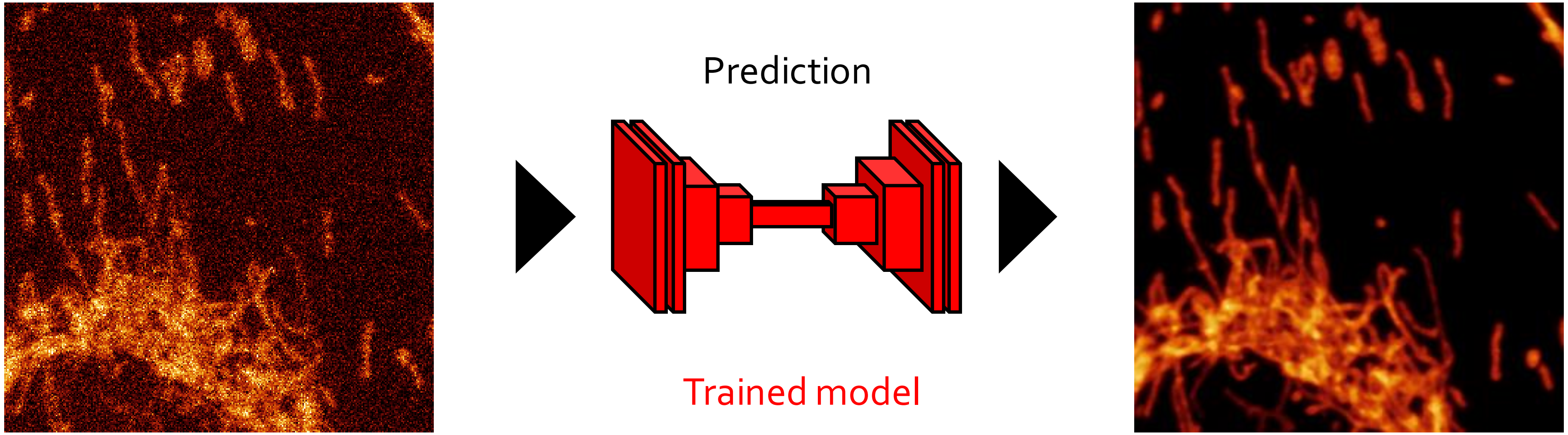
To try the model in ImageJ, go to Plugins > DeepImageJ > DeepImageJ Run, choose this model from the list and click on Test Model.

The exported model contains an additional ImageJ macro (`StarDist2D_Post-processing.ijm`) to run the StarDist postprocessing in Fiji.

More information at https://deepimagej.github.io/deepimagej/



Ready model can be used on related data



- Training takes hours to days
- Prediction takes seconds to minutes

DEEP LEARNING IS NOT A QUICK FIX FOR IMAGE PROCESSING

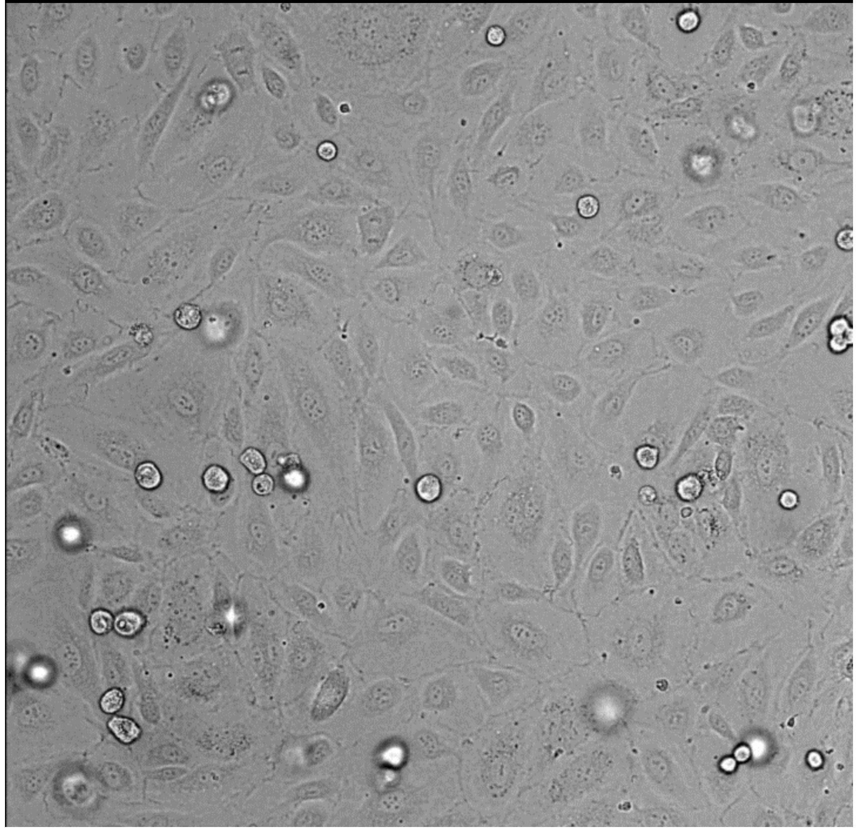
- Training dataset generation takes hours to days
- Training takes hours to days
- Prediction takes seconds to minutes
- Repeat?

EXAMPLE PIPELINE

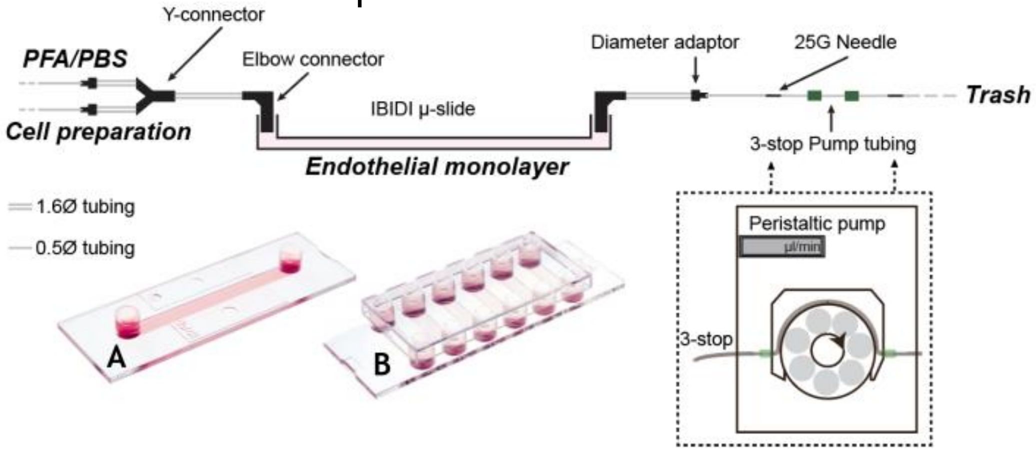
Study on how pancreatic cancer cell adhere and travel on HUVEC monolayer

Cancer cell microfluidics

Huvec monolayer
Pancreatic cancer cells on top



Microfluidics setup to mimic blood circulation



How do cancer cells adhere and migrate on the monolayer?

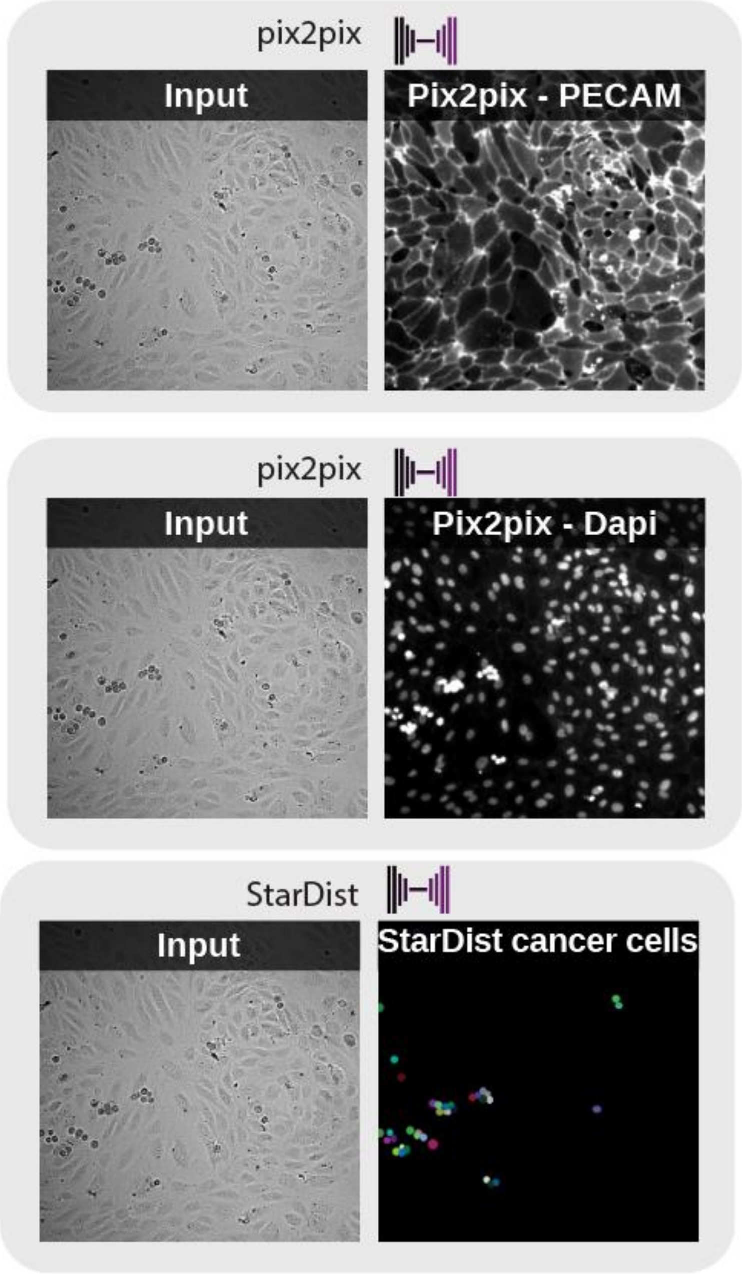
Issues:

- No suitable live cell dyes
- No microscope that could support the microfluidics setup and 3 channels
- Phototoxicity…

Guillaume Jacquemet

Gautier Follain

Sujan Ghimire

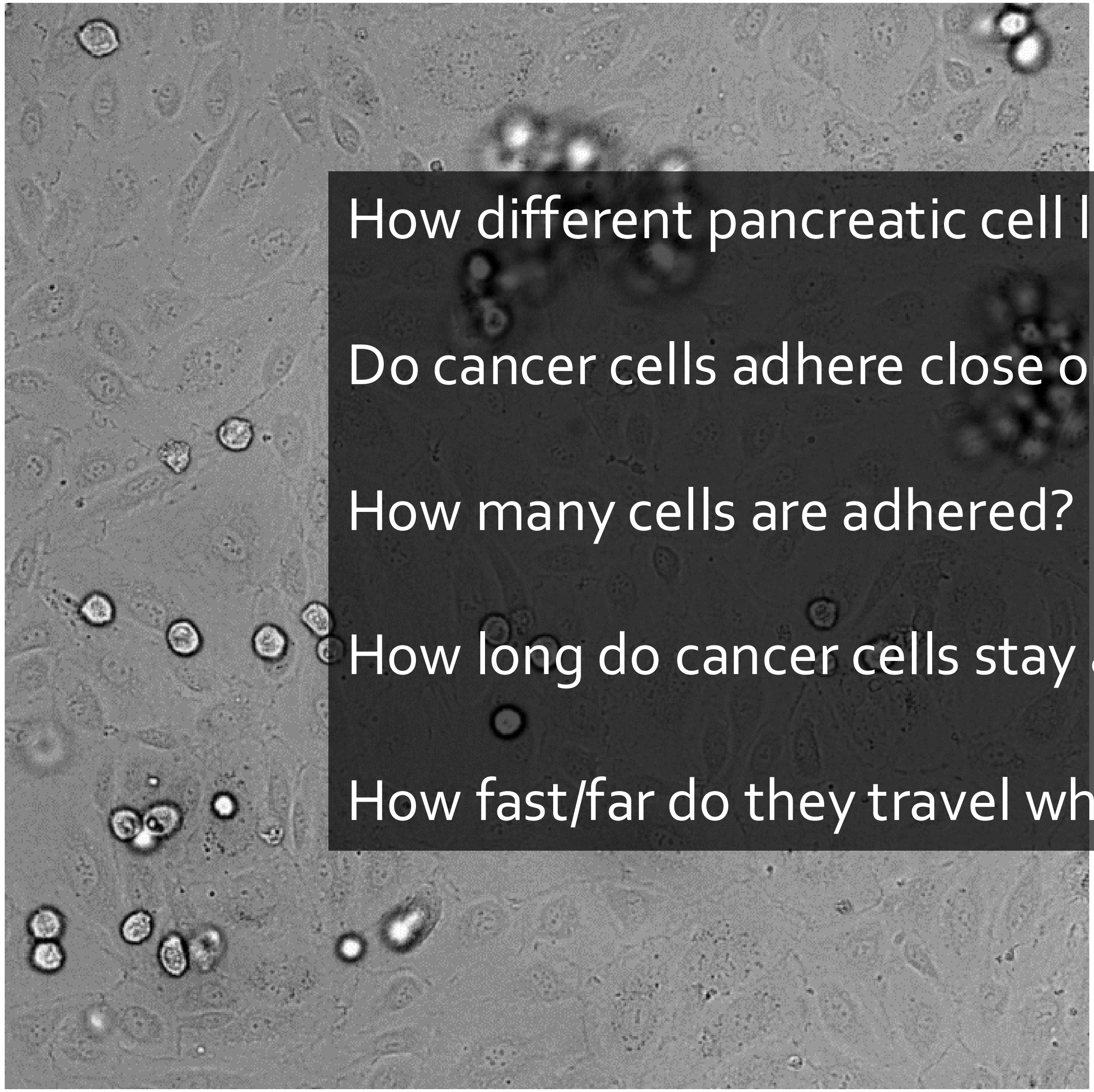


456 image pairs x aug4
fixed

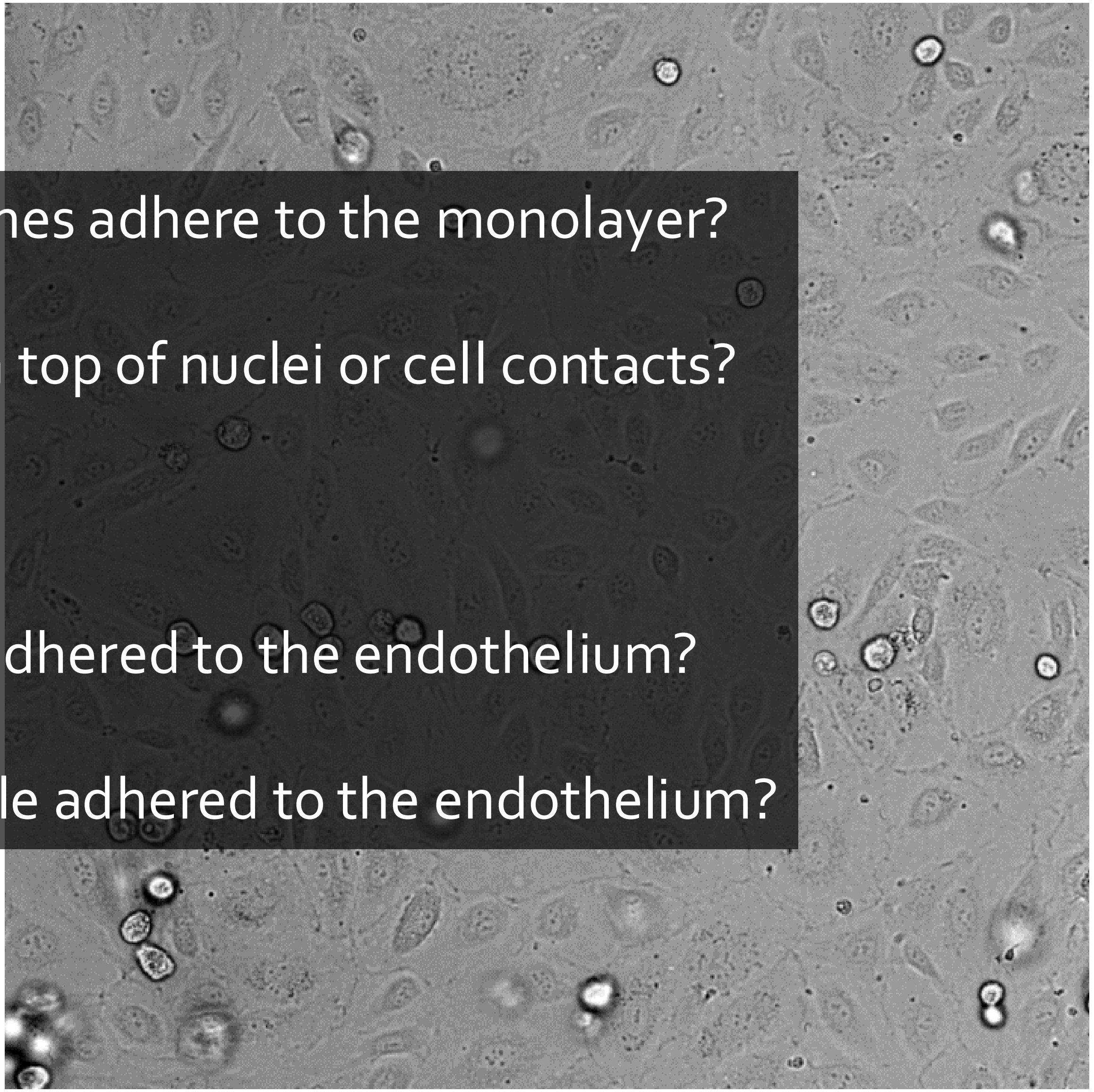
258 image pairs x aug8
fixed

20 image pairs x aug8
fixed

StarDist + TrackMate



All DL pipeline visualized



How different pancreatic cell lines adhere to the monolayer?

Do cancer cells adhere close on top of nuclei or cell contacts?

How many cells are adhered?

How long do cancer cells stay adhered to the endothelium?

How fast/far do they travel while adhered to the endothelium?

Contribute to ZeroCostDL4Mic

- We welcome network contributions from the research community.
- Guidelines available here: https://github.com/HenriquesLab/ZeroCostDL4Mic/wiki/How-to-contribute

Thank you!

Lab crew

- **Guillaume Jacquemet**
- **Sujan Ghimire**
- **Gautier Follain**
- Hanna Grobe
- Monika Vaitkevičiūtė
- Ana Gračanin
- Sarah Massaad
- Marcela Rivera

Collaborators

- Ricardo Henriques
- Lucas von Chamier
- Romain F Laine
- Estibaliz Gómez de Mariscal

Åbo Akademi
ACADEMY OF FINLAND
TURKU BIOSCIENCE
LOUNAIS-SUOMEN SYÖPÄYHDISTYS
SIGRID JUSELIUS STIFTELSE
Syöpäsäätiö Cancer Foundation Finland
InFLAMES Solution is in Immunity
TURKU BIOIMAGING { More than you can imagine }
Syöpäsäätiö Cancer Foundation Finland
InFLAMES Solution is in Immunity