



B i a P y

DEEP LEARNING USING BIAPY

What, where, how?

Can I also use it?

Image analysis course, Institut Pasteur

May 15, 2025

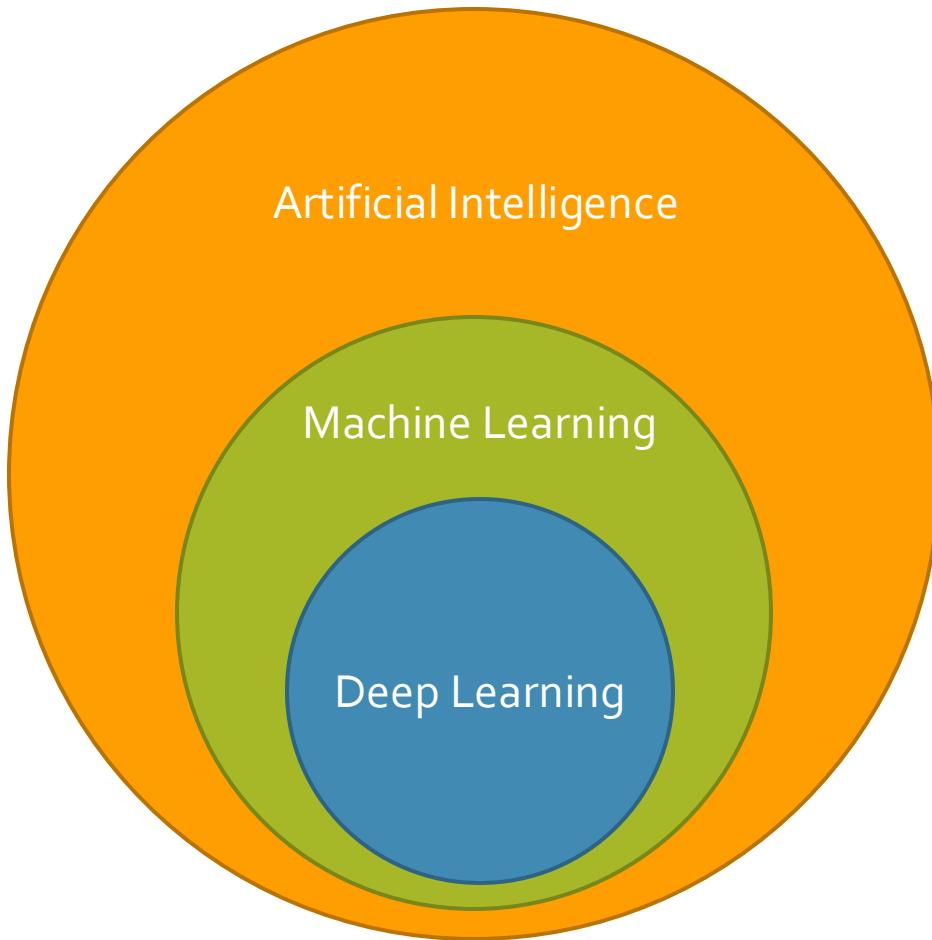
Joanna Pylvänäinen, joanna.pylvanainen@abo.fi

 @jwpylvanainen.bsky.social

Materials adapted from Guillaume Jacquemet and Ignacio Arganda-Carreras

Contents

- What is Deep learning?
- What is BiaPy
- What can be done using BiaPy
- 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.

All

News

Images

Books

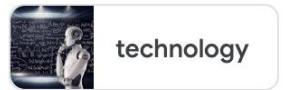
Videos

More

Tools

Saved

SafeSearch ▾



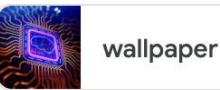
technology



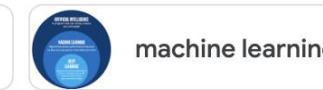
robot



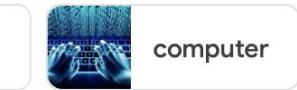
future



wallpaper



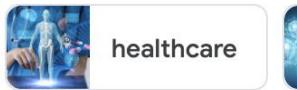
machine learning



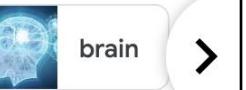
computer



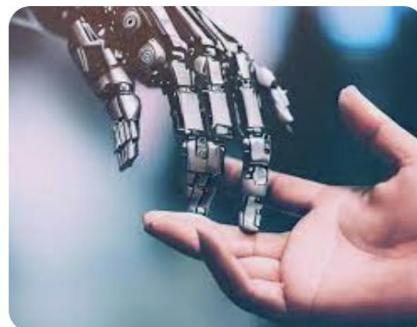
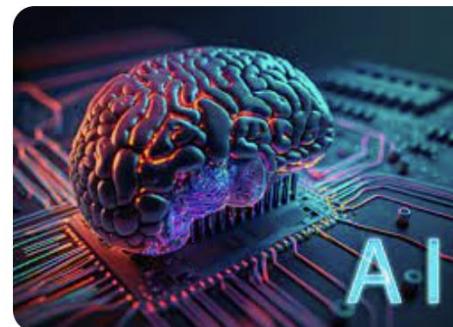
human



healthcare

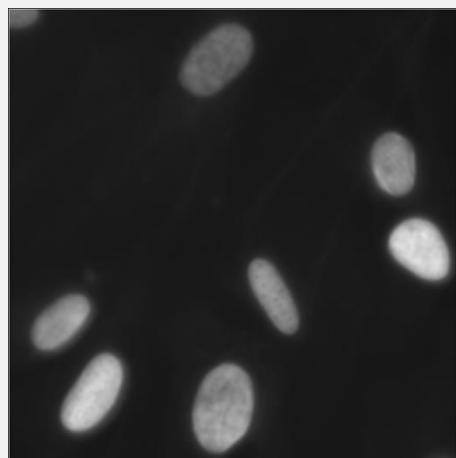


brain

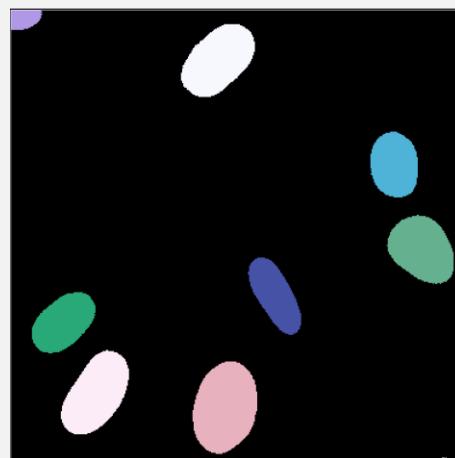
 ZDNET
artificial intelligence ... Simplilearn
What is Artificial Intelligence? Typ... Investopedia
Artificial Intelligence: What It Is an... MarkTechPost
Artificial Intelligence ... Chitkara University
What Is Artificial Intelligence And ... Wikipedia
Artificial intelligence - Wi... GeeksforGeeks
Artificial Intelligence | An ... Sharda University
Artificial Intelligence – The Mo... Fingent
The Future of Artificial Intelligence ... British GQ
Artificial intelligence: It's time to ...

[All](#)[News](#)[Images](#)[Books](#)[Videos](#)[More](#)[Tools](#)[Saved](#)[SafeSearch ▾](#)

TRAINED FOR A SPESIFIC TASK



How to get a label
image from
fluorescence image?



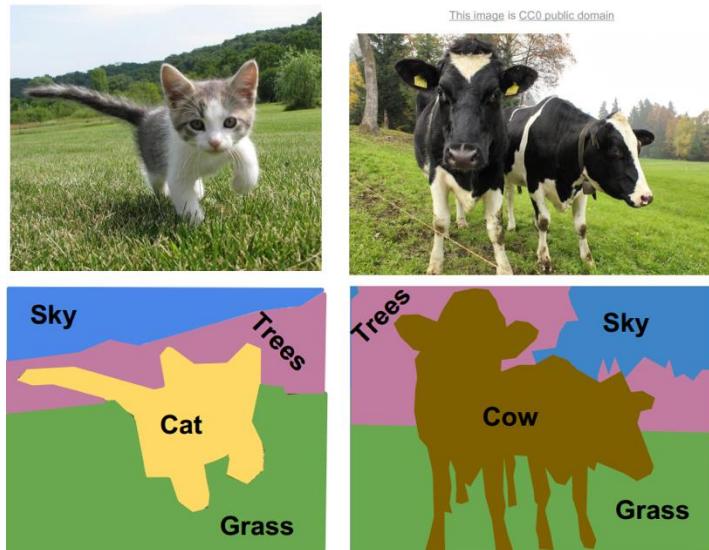
Common Deep learning tasks



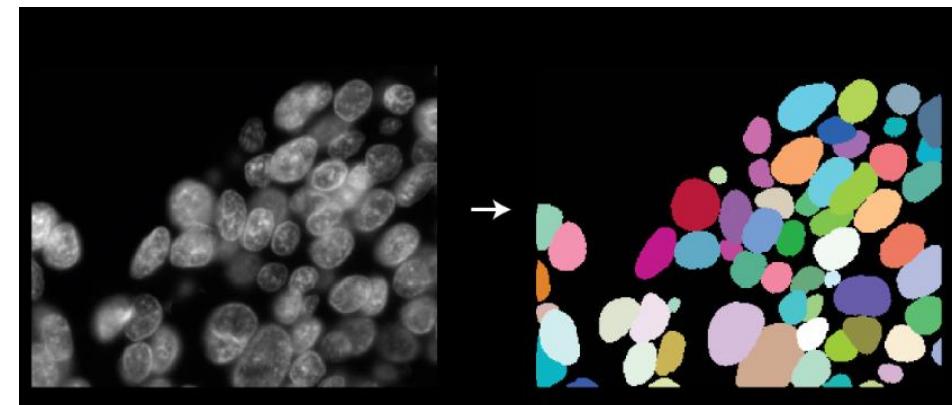
Classification



Object Detection

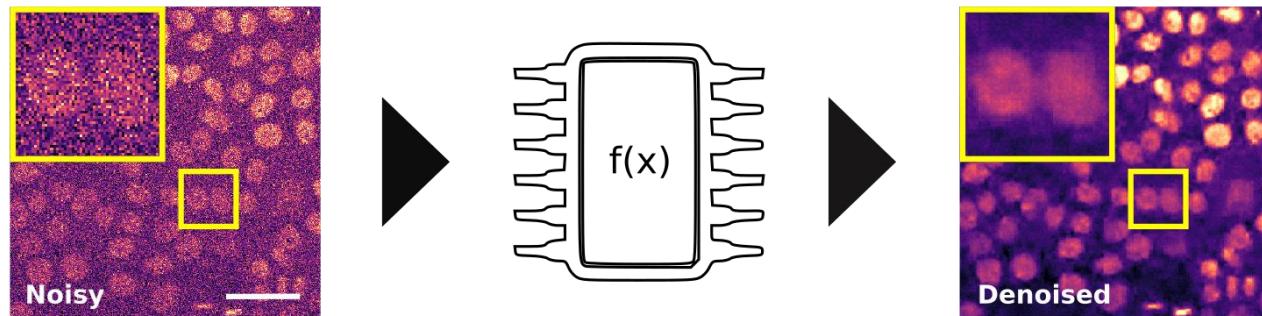


Semantic Segmentation

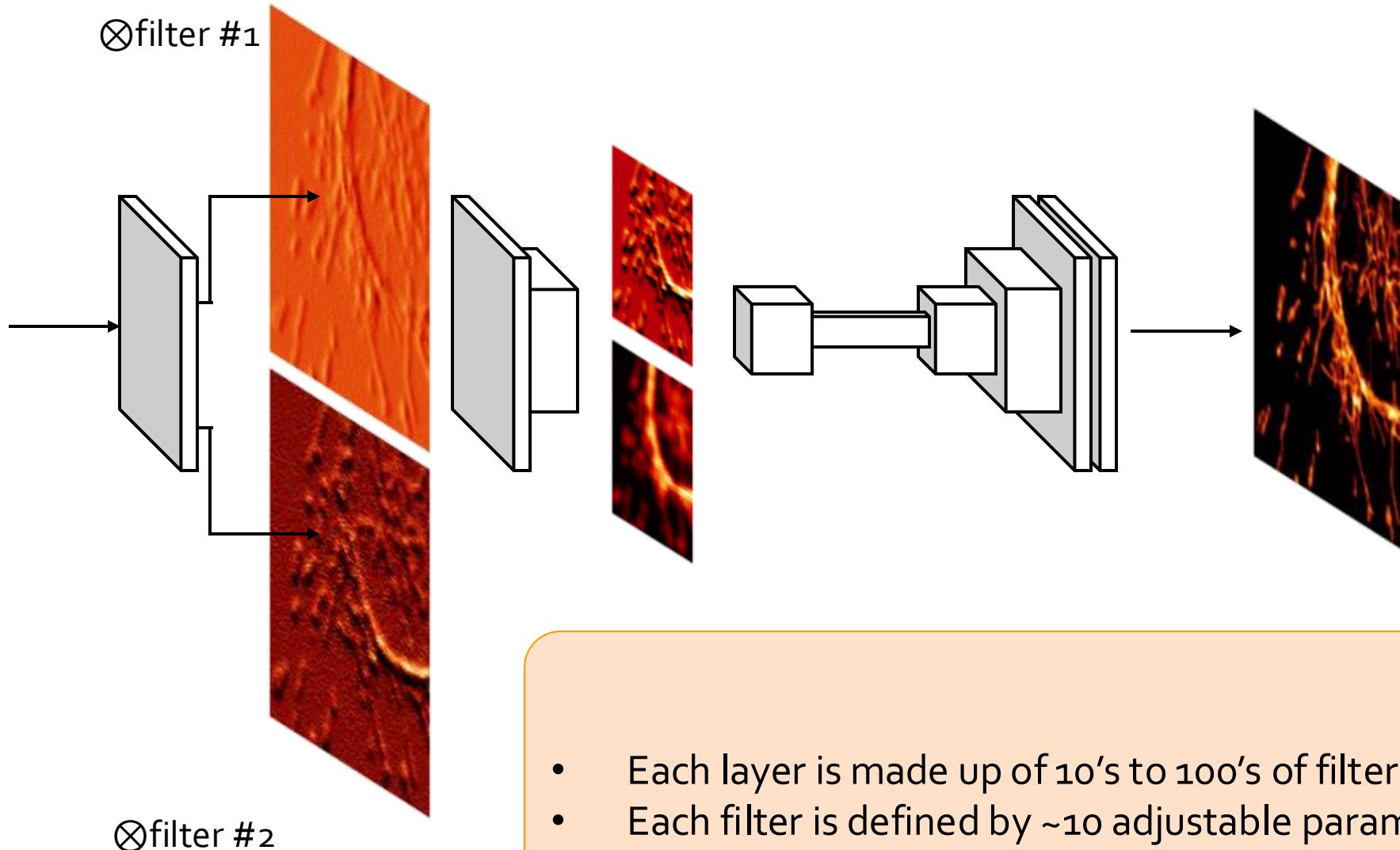


Instance segmentation

Classical algorithm

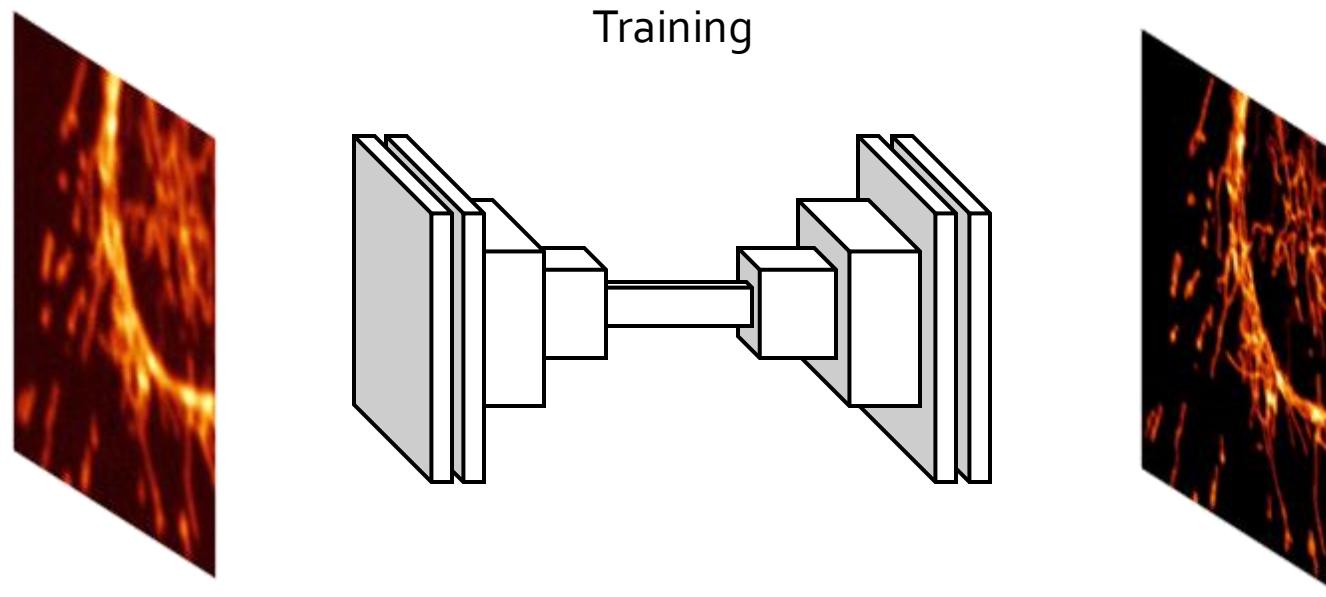


**DEEP
LEARNING IN
MICROSCOPY?**



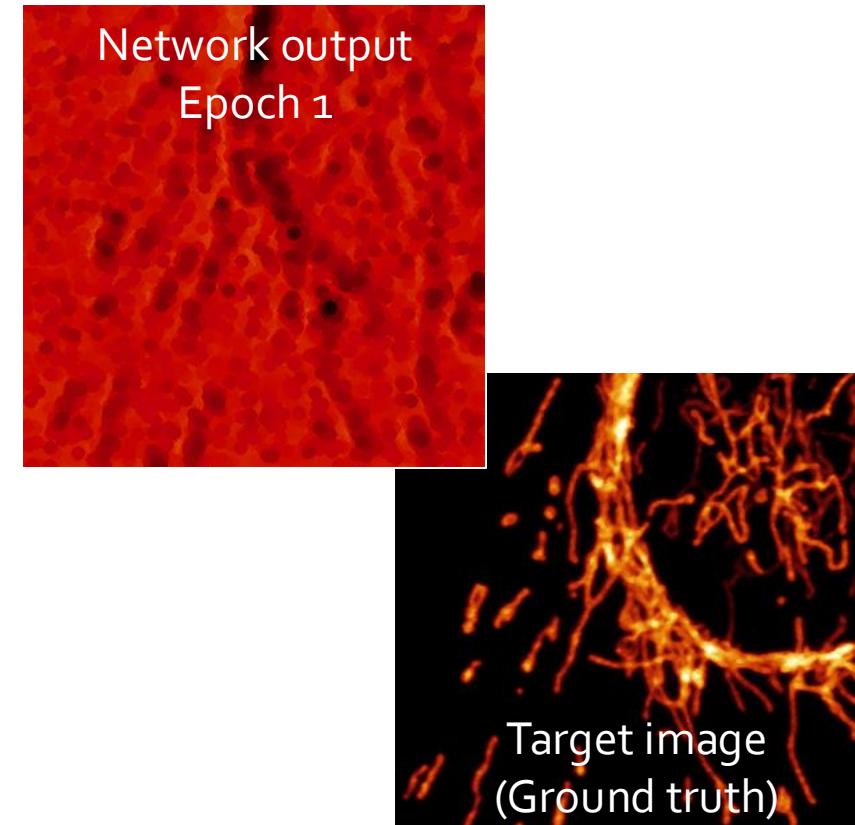
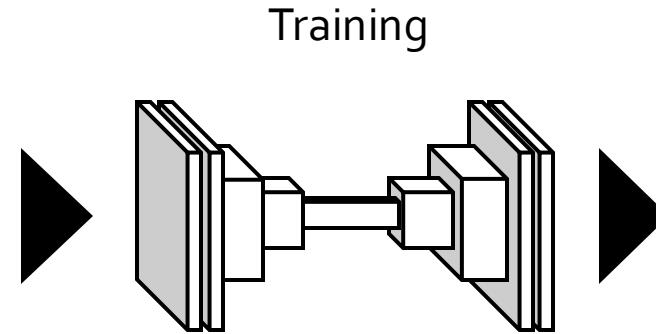
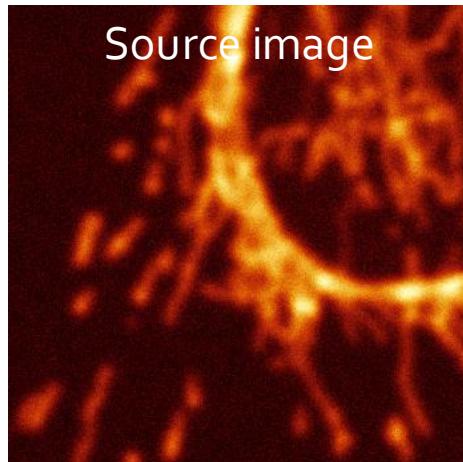
- Each layer is made up of 10's to 100's of filters
- Each filter is defined by ~10 adjustable parameters ("weights")
- The network can be defined by millions of adjustable weights

What happens during training?



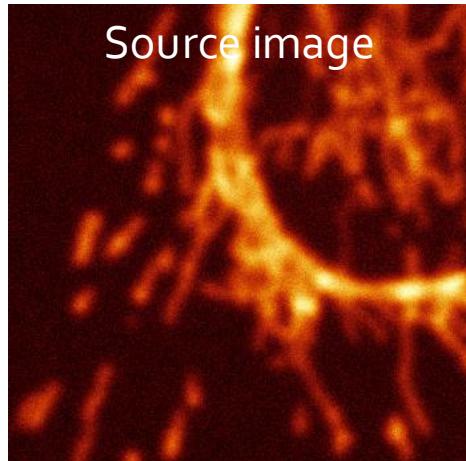
How does the neural network learn?

- Step #1: Initialize 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)

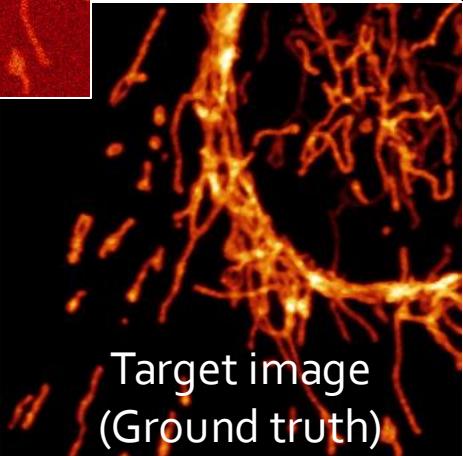
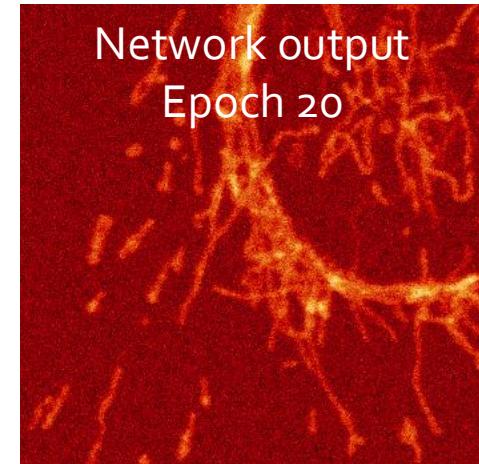
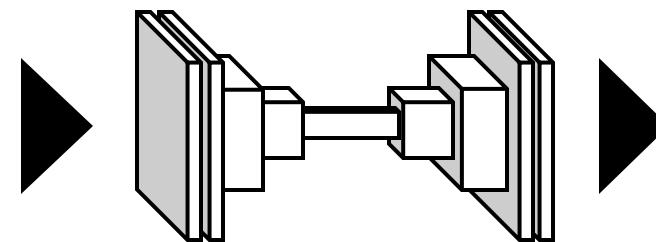


How does the neural network learn?

- Step #1: Initialize 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)

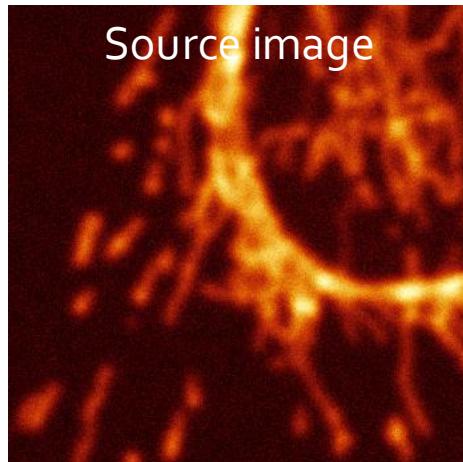


Training

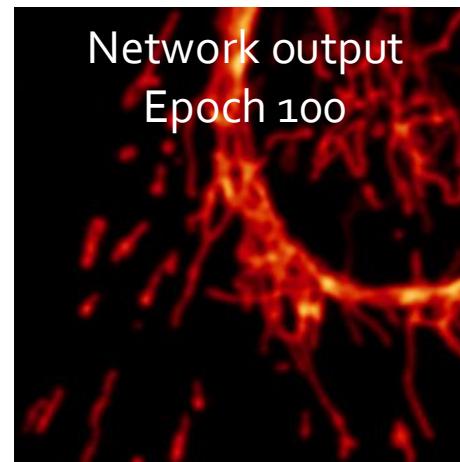
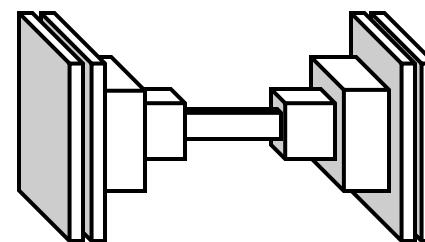


How does the neural network learn?

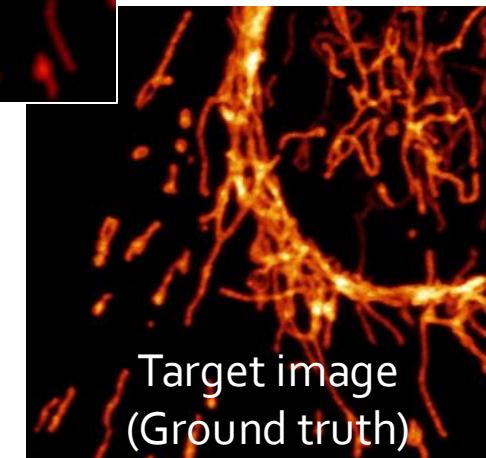
- Step #1: Initialize all the weights randomly
- Step #2: Compute network output (feed forward)
- Step #3: Compare to target image (ground truth)
- Step #4: Update weights to decrease the error (backpropagation)



Training



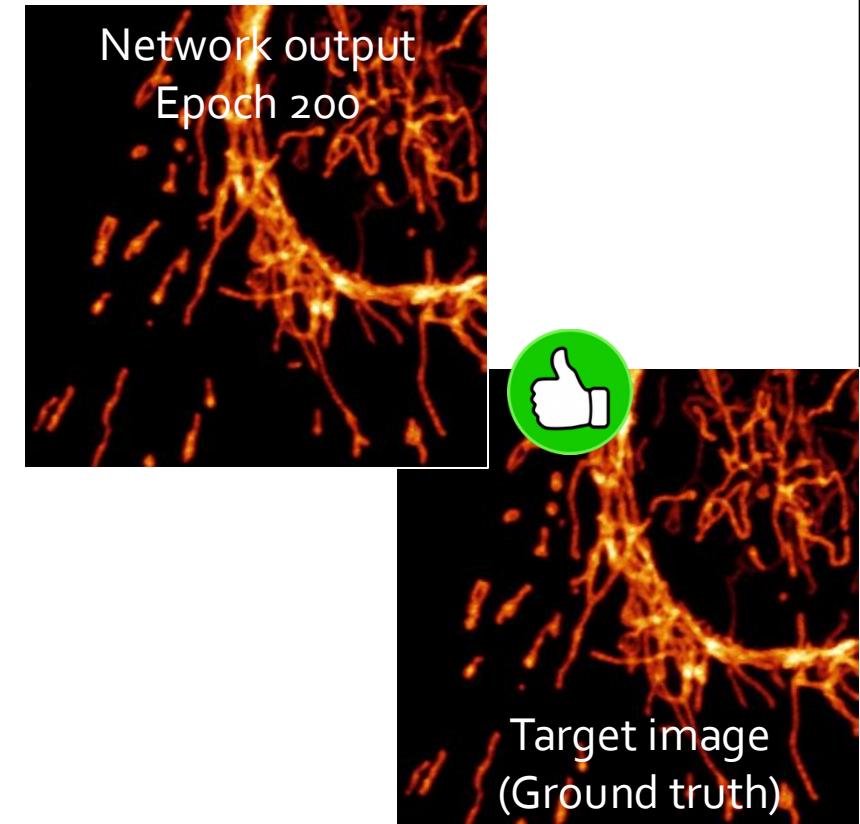
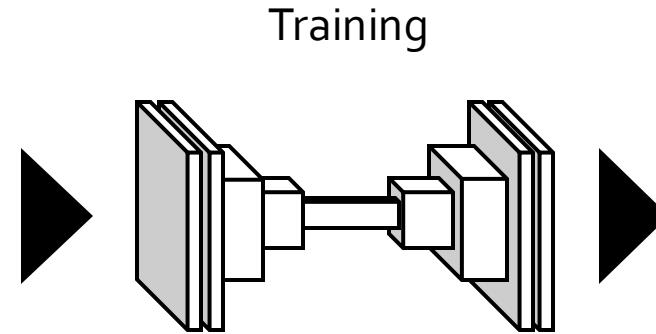
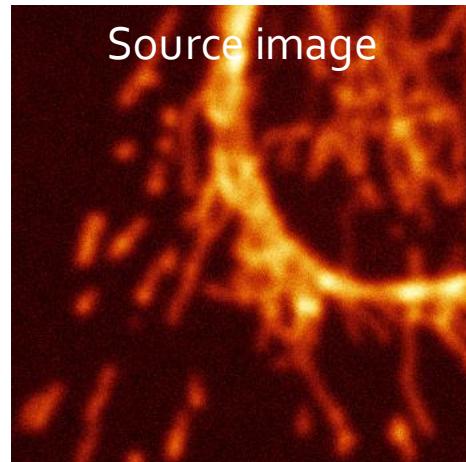
Network output
Epoch 100



Target image
(Ground truth)

How does the neural network learn?

- Step #1: Initialize 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

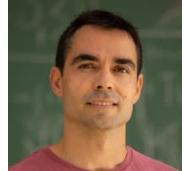




BiaPy to the rescue!



Daniel Franco-Barranco



Ignacio Arganda-Carreras

BiaPy: accessible deep learning on bioimages

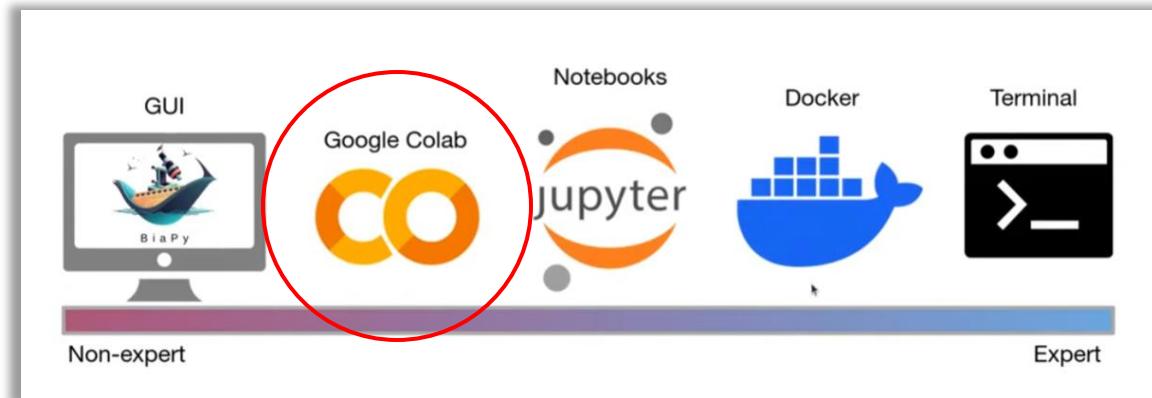
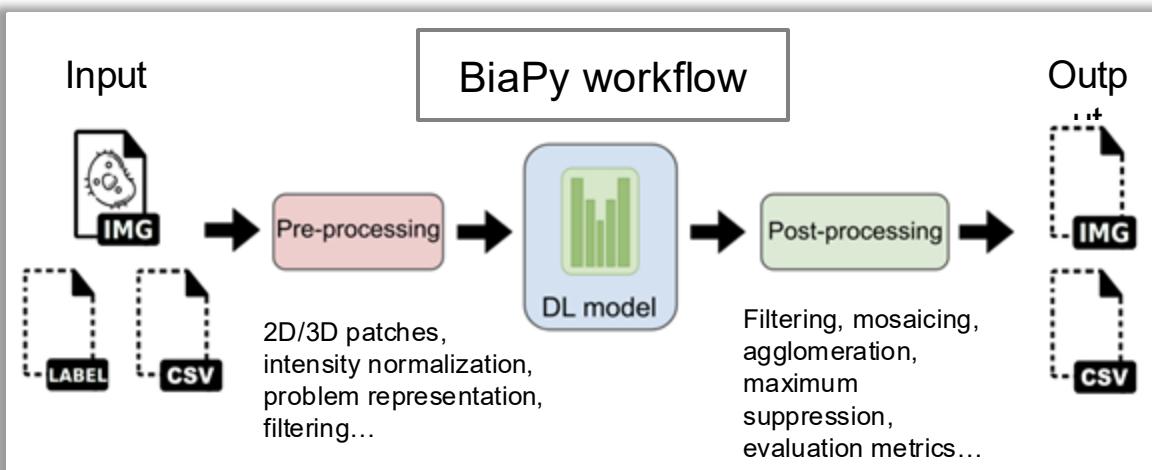
[Daniel Franco-Barranco](#), [Jesús A. Andrés-San Román](#), [Ivan Hidalgo-Cenalmor](#), [Lenka Backová](#), [Aitor González-Marfil](#), [Clément Caporal](#), [Anatole Chessel](#), [Pedro Gómez-Gálvez](#), [Luis M. Escudero](#), [Donglai Wei](#), [Arrate Muñoz-Barrutia](#) & [Ignacio Arganda-Carreras](#)

[Nature Methods](#) (2025) | [Cite this article](#)

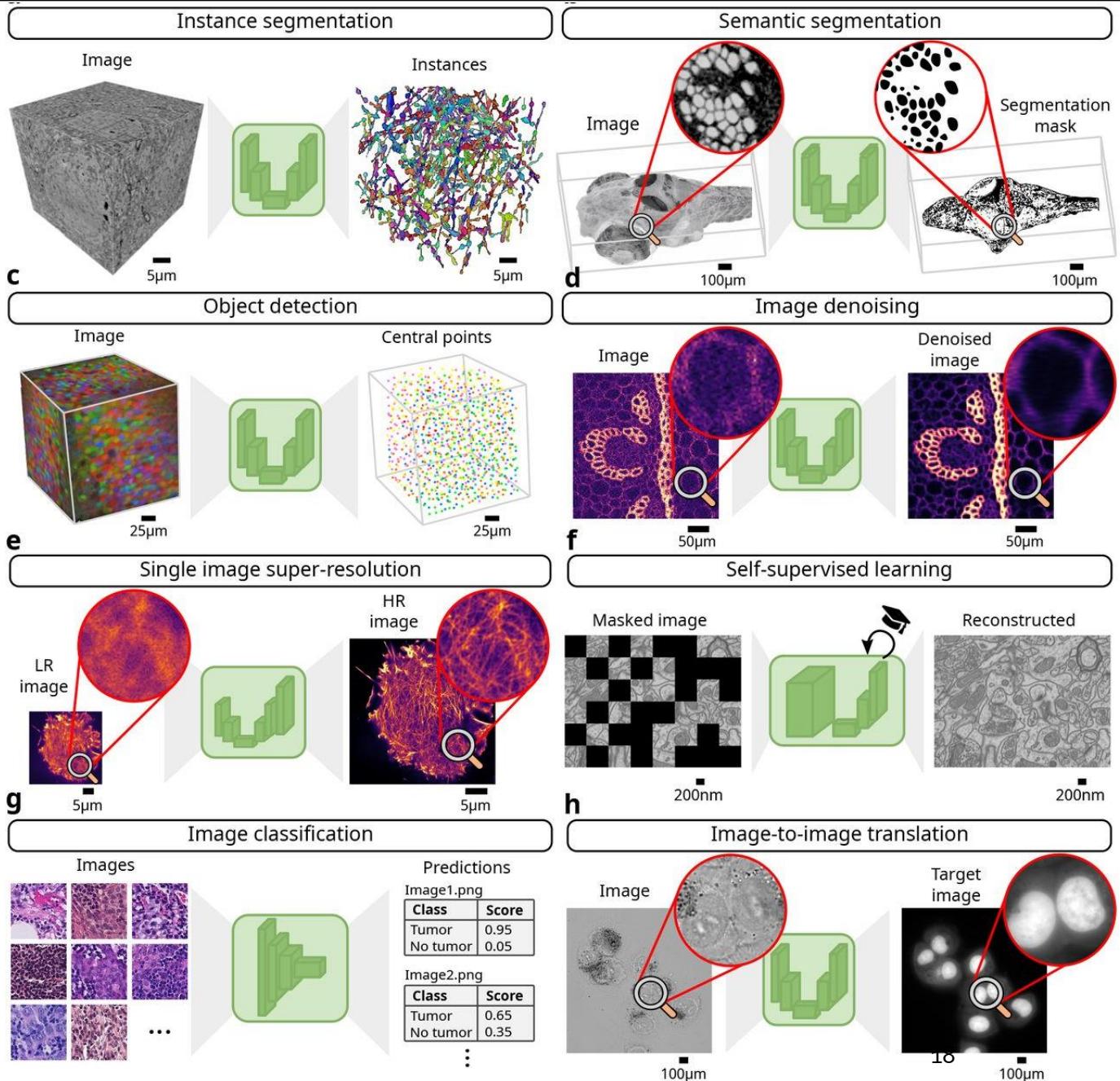
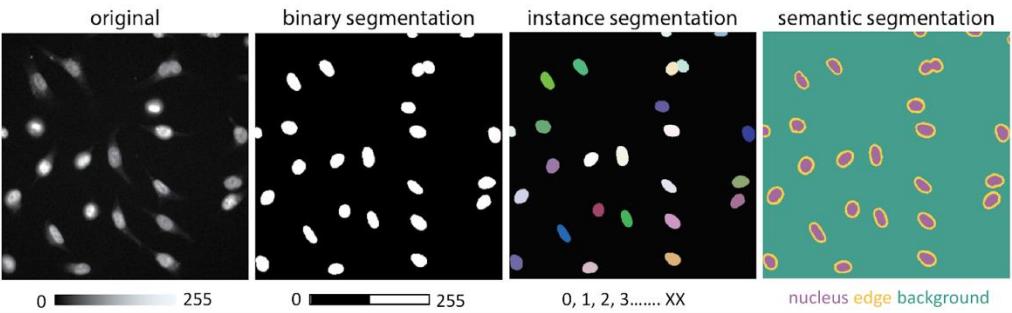
Open-source software platform designed to make deep learning-based analysis of biological images accessible, even for researchers without coding experience.

- **User-Friendly Interface:** BiaPy offers an intuitive graphical user interface, allowing complex image analysis tasks without writing any code.
- **Versatile Image Analysis:** It supports various tasks such as image classification, object detection, and image enhancement, applicable to both 2D and 3D microscopy images.
- **Pretrained Models Available:** BiaPy integrates with the BioImage Model Zoo, providing access to pretrained models that can be readily applied to new datasets.
- **Open-Source and Reproducible:** BiaPy promotes transparency and reproducibility in scientific research, allowing users to share and replicate analyses easily.
- It uses the GPU provided by Google Colab.
- Free! (with limitations)

- For **users**: easy to combine pre-processing, post-processing and models.
- For **developers**: freedom to contribute to any module.



Tools available in BiaPy



Deep learning tools in BiaPy

| | | | | | | | |
|------------------------------|-------------------------|------------------------|-------------------------------|--------------------------|--------------------------|-------------------------------|--------------------------|
| Image classification (2D) | Image denoising (2D) | Image to image (2D) | Instance segmentation (2D) | Object detection (2D) | Self-supervision (2D) | Semantic segmentation (2D) | Super-resolution (2D) |
| Image classification (3D) | Image denoising (3D) | Image to image (3D) | Instance segmentation (3D) | Object detection (3D) | Self supervision (3D) | Semantic segmentation (3D) | Super-resolution (3D) |

Step1: Model training

Step2: Apply model to own data

Inference

(2D/3D)



BiaPy_2D_Instance_Segmentation.ipynb
Tiedosto Muokkaa Näytä Lisää Suorituspalvelu Työkalut Ohje
Komennot + Koodi + Teksti Kopioi Driveen

2D Instance Segmentation pipeline

In this notebook, we demonstrate the use of the [BiaPy](#) pipeline for **2D instance segmentation** of microscopy data.

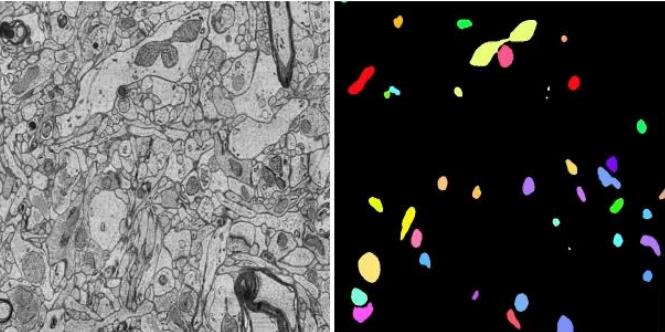


Figure 1: Example of a 2D instance segmentation problem. From left to right: 2D electron microscopy image and its corresponding mitochondria instance labels.

Without any coding, we'll guide you step-by-step through the process to:

1. Upload a set of training and test images along with their corresponding instance label images.
2. Train a Deep Neural Network (DNN) model using the training set.
3. Apply the model to the test images.
4. Download the segmentation results to your local machine.

Disclaimer: The structure of the notebook is heavily inspired by the fantastic [ZeroCostDL4Mic notebooks](#).

Contact: This notebook was created by [Ignacio Arganda-Carreras](#), [Lenka Backová](#), [Daniel Franco-Barranco](#) and [Ane Paniagua](#). For suggestions, comments, or issues, please reach out to us via email or [create an issue in BiaPy's repository](#). Thank you!

Expected inputs and outputs

Inputs

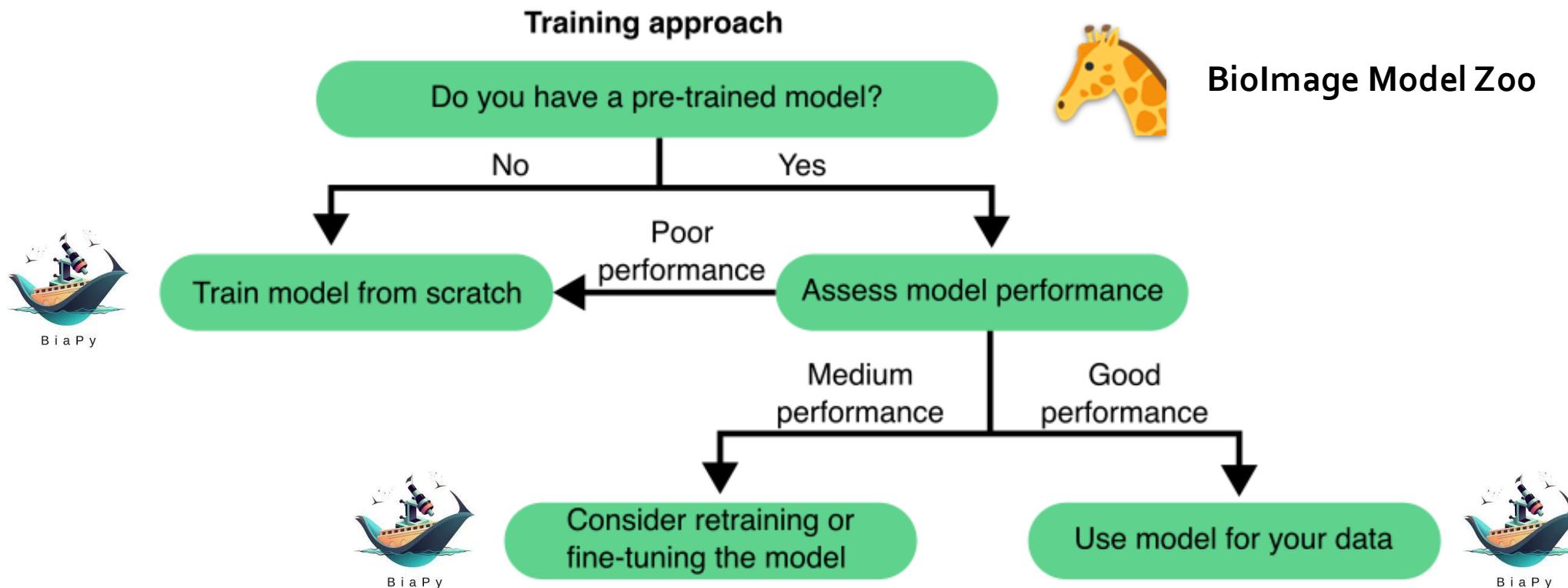
This notebook expects five folders as input:

- **Training raw images:** with the raw 2D images to train the model.
- **Training label images:** with the instance-label 2D images to train the model. Their number and sizes must match those of the training raw images.
- **Test raw images:** with the raw 2D images to test the model.



BIAPY NOTEBOOKS

Choosing a training approach



Importing pre-trained models from BiolImage Model Zoo

OPTIONAL: Check BiolImage Model Zoo (BMZ) models compatible with BiaPy

Use this option if you want to load a pretrained model from BMZ and want to generate a full list of the compatible models with BiaPy.

Important: First you will need to run the next cell and select 'BiolImage Model Zoo' as the source of the model. Then, paste the DOI into the created field.

BiolImage.IO

Mostrar código

List of models that can be used in BiaPy:

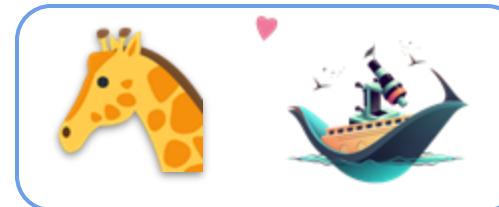
| | | |
|--|--|---|
| NucleoSegmentationBoundaryModel affable-shark (⌚) DOI: 10.5281/zenodo.5764892/6647674 | LiveCellSegmentationBoundaryModel hiding-tiger (⌚) DOI: 10.5281/zenodo.5869899/6647688 | EnhanceMitochondria2D hiding-blowfish (⌚) DOI: 10.5281/zenodo.6406756/6811922 |
| | | |
| Mitochondria2DSegmentation2D shivering-raccoon (⌚) DOI: 10.5281/zenodo.6406803/6406804 | EnhanceBoundary2D amiable-crocodile (⌚) DOI: 10.5281/zenodo.8142283/8171247 | |
| | | |

Play to select the source to build the model (BiaPy, Torchvision or BiolImage Model Zoo)

Mostrar código

Source: BiaPy Torchvision BiolImage Model Zoo

DOI: 10.5281/zenodo.5874741

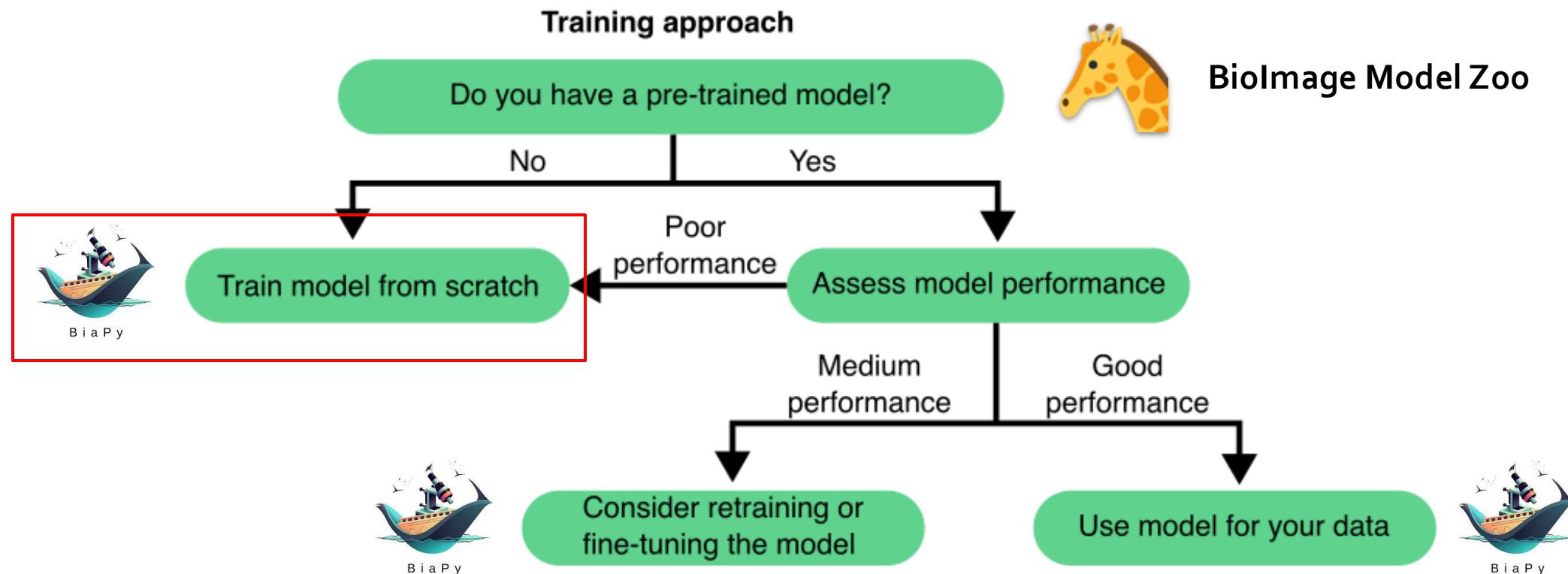


Model from Scratch vs. Transfer Learning

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

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

Choosing a training approach



Steps for model training in BiaPy

Create
training
dataset

Upload
data to G-
drive

Install
dependencies

Define
paths to
training
and QC
data

Set
training
parameters
and train

Assess the
quality of
your model

Run
predictions
on unseen
data



Orany
other tool

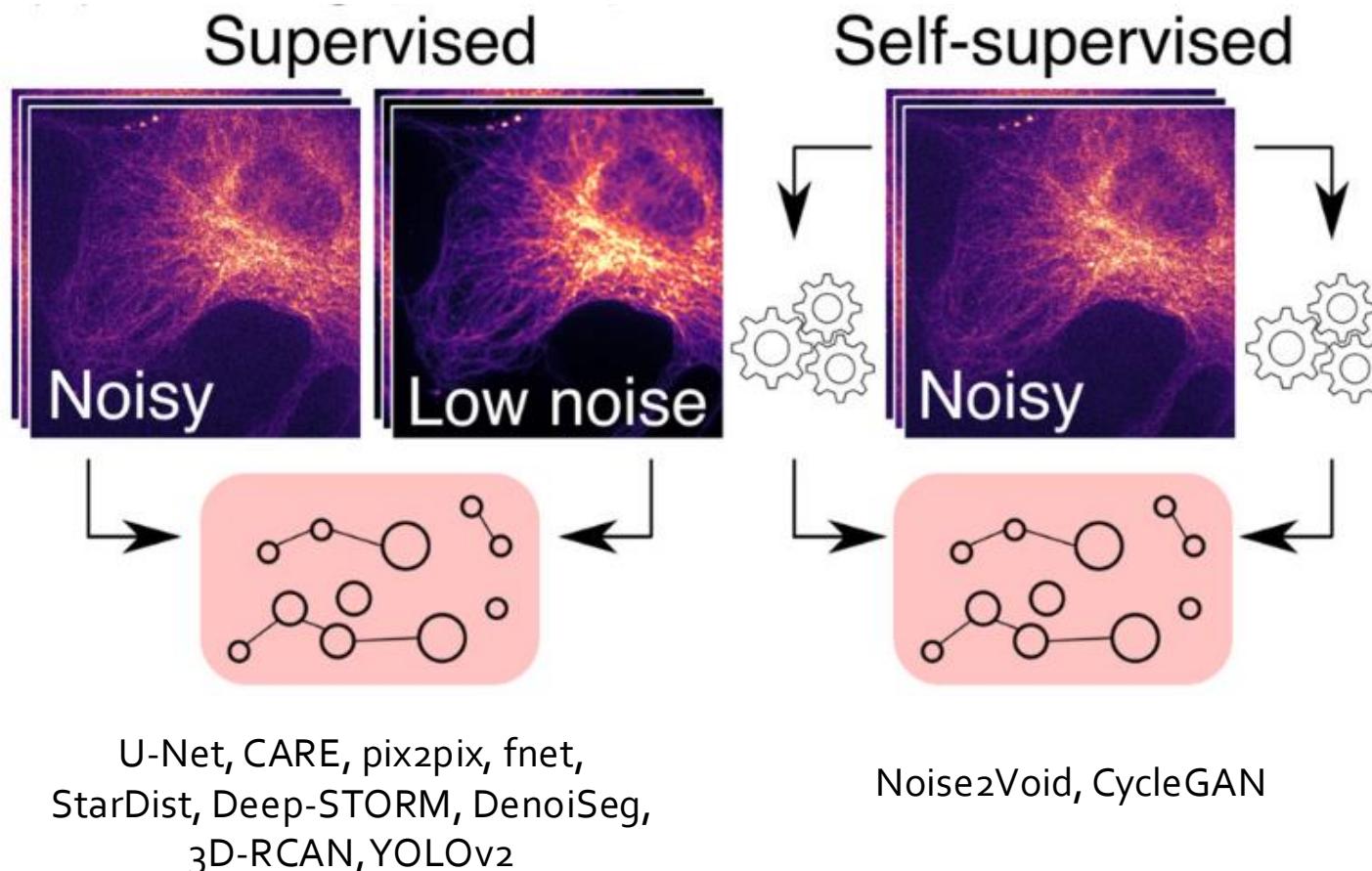


BiaPy



BiaPy

Step #0: All starts with training datasets



Each network has specific data requirement to train

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

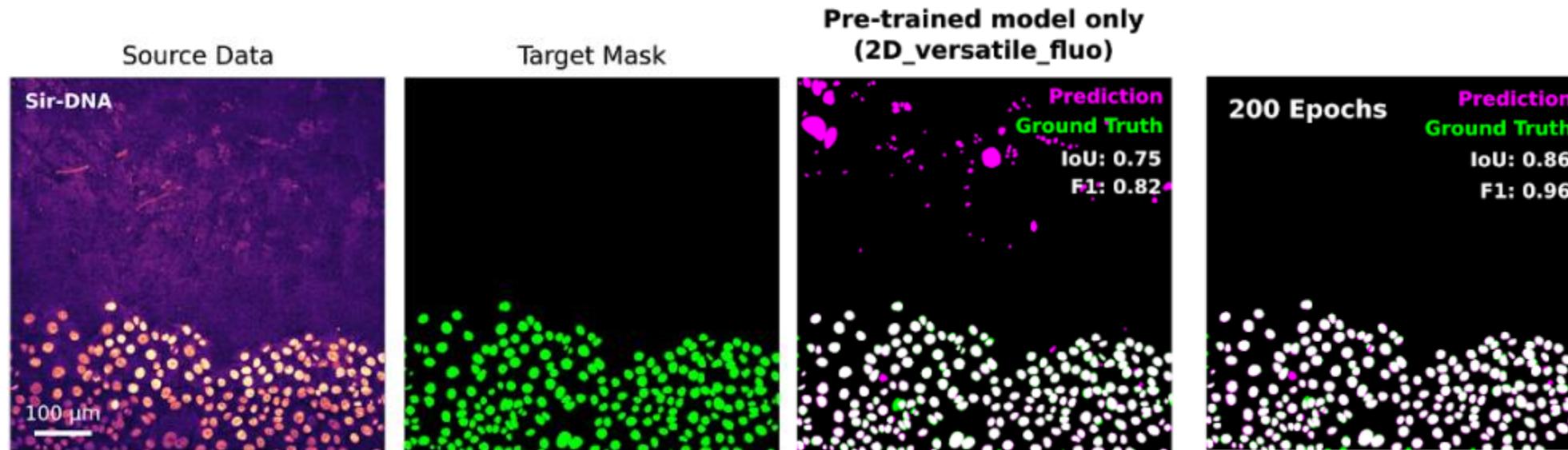
The performance is dependent on the training data

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



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



Segmentation using StarDist

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

Steps in Quality control

Step1:

Inspection of the loss
function

Step2:

Visual inspection

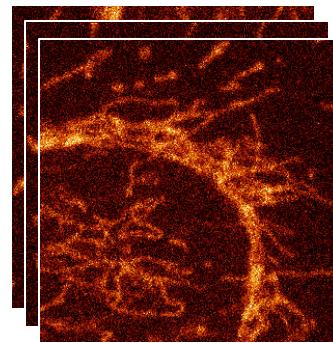
Step3:

Error mapping and
quality metrics

Step #1: Validation split

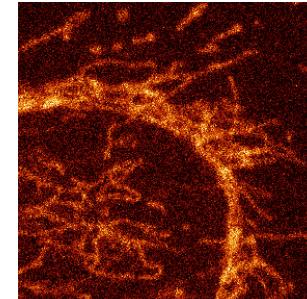
Training dataset is divided into training and validation data

Training data



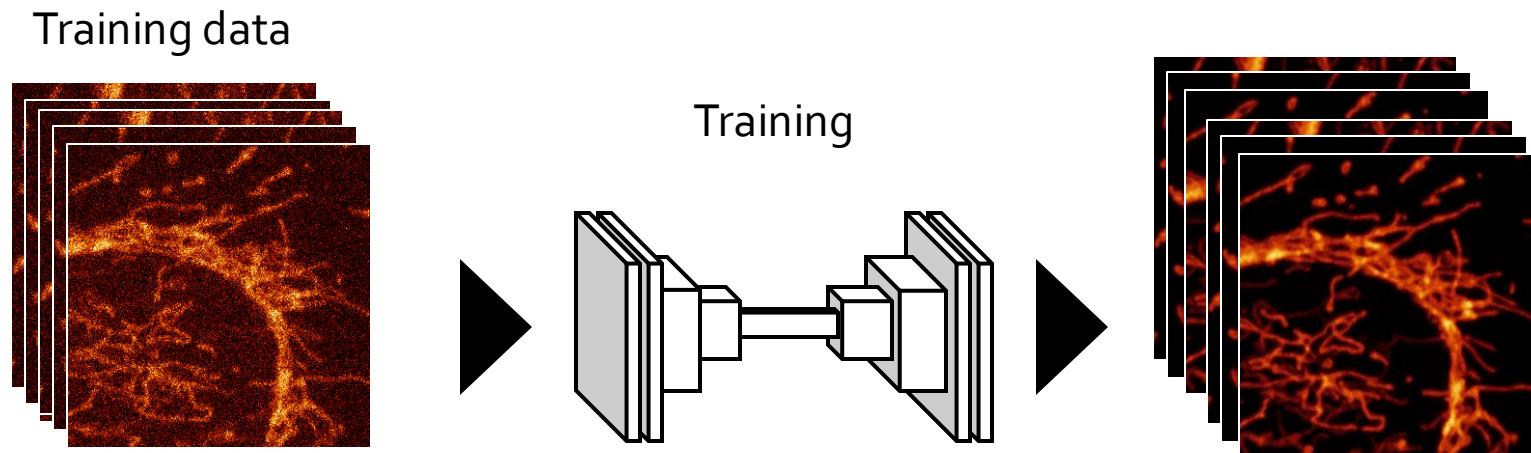
90 %

Validation data



10 %

Step #2: Learning from the training data

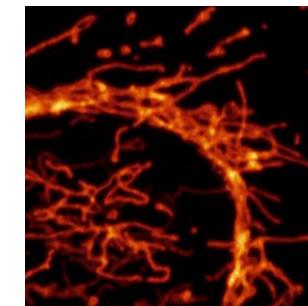
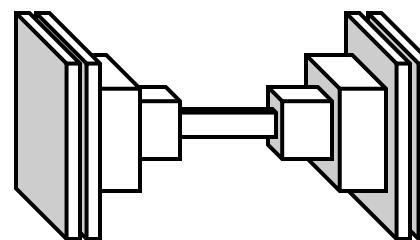
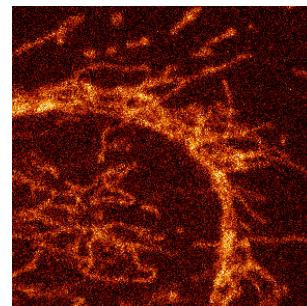


- EPOCH: One round of training. A model is generated at the end of each EPOCH
- EPOCH are divided into multiple “steps”
- Batch size define how many images are seen by the network at each steps

Score: Training loss: How well the model perform on the training data

Step #3: Testing the model on the validation data

Validation data

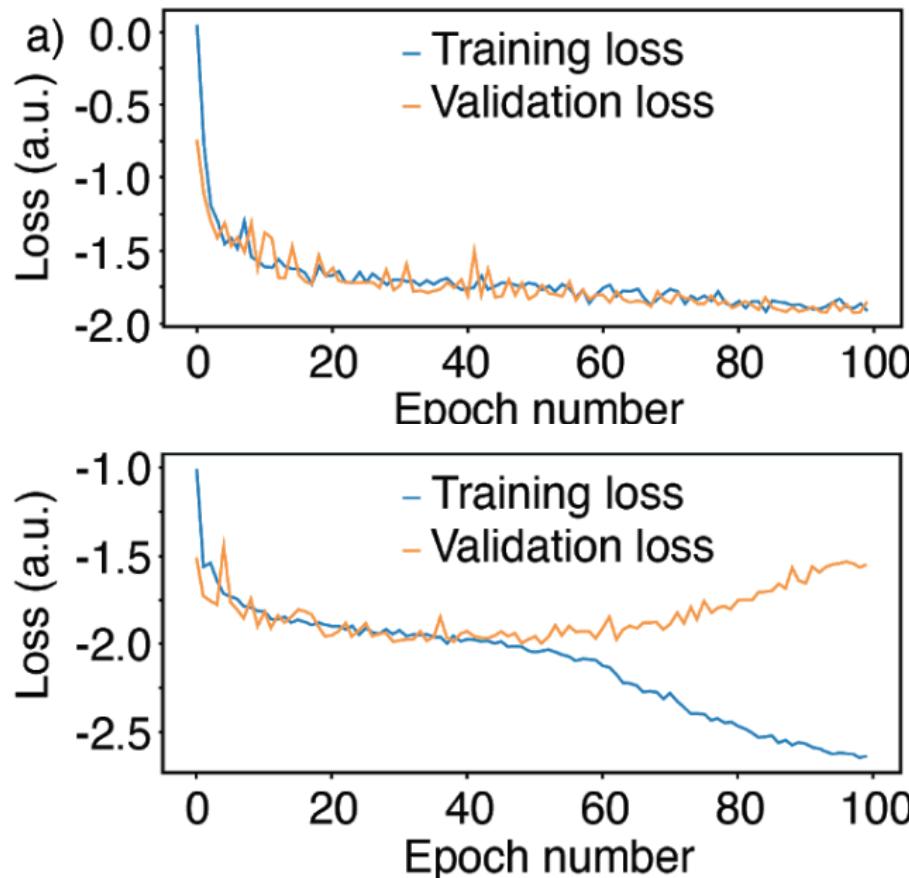


- At the end of every EPOCH
- Assesses how well the model is performing on validation data

Score: Validation loss: How well the model perform on the validation data

Step #4: Repeat

- Train model for hundreds of EPOCH



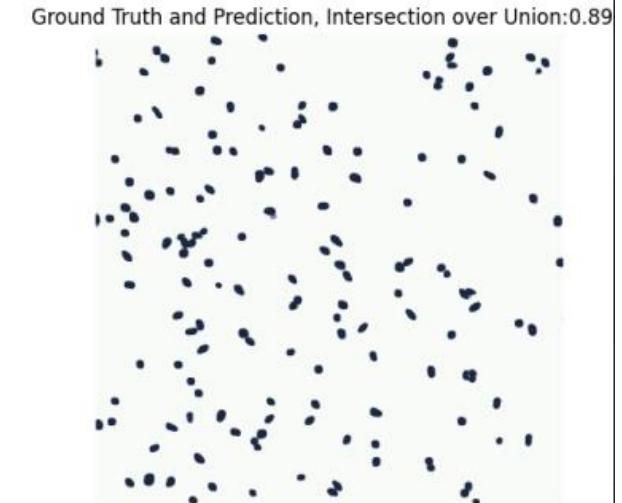
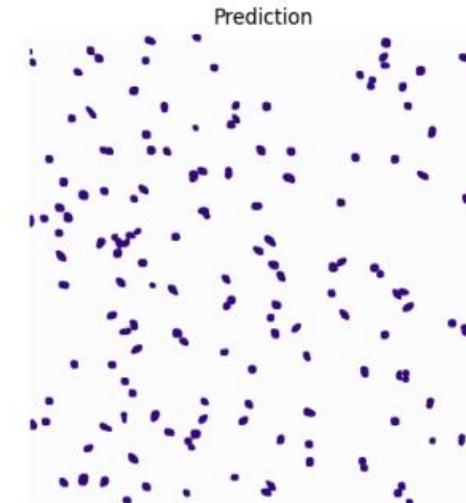
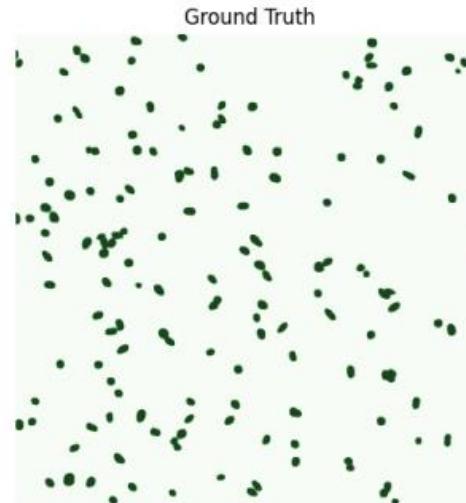
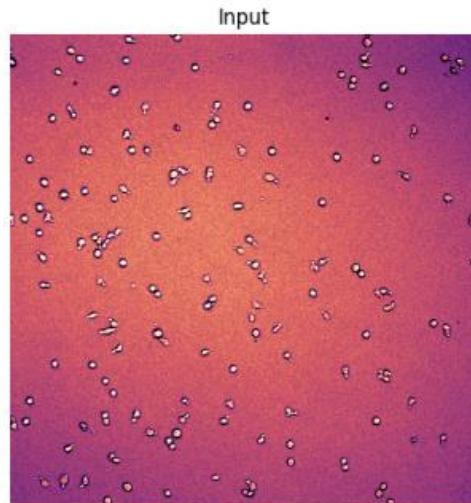
Quality Control 1: Inspection of the loss function

- 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.
- **Actions:**
 - Decreasing Training loss and Validation loss -> train for more epochs
 - Curves are flattened out -> no more training is required
 - Validation loss increases while Training loss simultaneously decreases:
 - > network is overfitting = remembering the exact patterns from the training data and no longer generalizes well to unseen data.
 - > training dataset should be increased.

Quality Control2: QC dataset

Visual inspection

Example Quality Control Visualisation

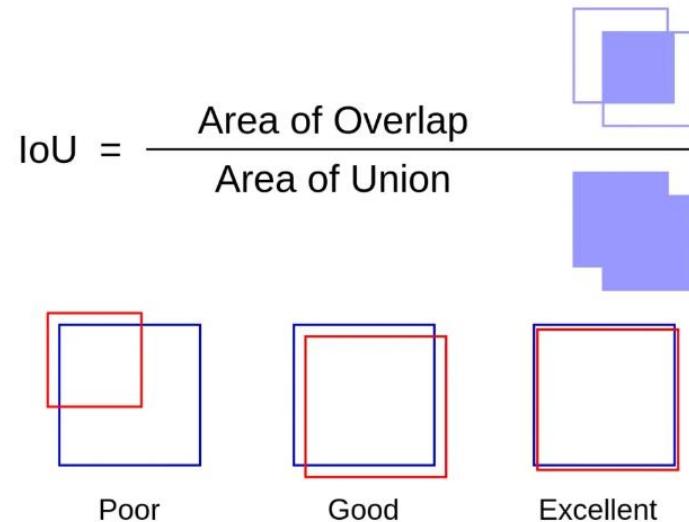


Quality Control 3:

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_d_score | panoptic_quality |
|----------------------------------|---|----------------|---------------|----------------|-----------|----------|----------|----------|--------|--------|-----------------|----------------------|------------------|
| Training_source _ICAM 1-1.tif | 0.89414007 | 2 | 153 | 2 | 0.987096 | 0.987099 | 0.974522 | 0.987096 | 155 | 155 | 0.873075 | 0.884487 | 0.873075 |

- 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 objects. (whole image)



True and false positives

| 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_matche_d_score | panoptic_quality |
|----------------------------------|---|----------------|---------------|----------------|-----------|----------|----------|----------|--------|--------|-----------------|---------------------|------------------|
| Training_source _ICAM 1-1.tif | 0.89414007 | 2 | 153 | 2 | 0.987096 | 0.987096 | 0.974522 | 0.987096 | 155 | 155 | 0.873075 | 0.884487 | 0.873075 |

“false positive”

- “n_pred” - “true positive”

“true positive”

- When a segmented object has an IoU above 0.5 (compared to the corresponding ground truth)

“false negative”

- “n_true” - “true positive”

n_true

- number of objects in ground truth

n_pred

- number of objects in prediction

f1 score (Dice Coefficient) and mean matched score

| 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_d_score | panoptic_quality |
|----------------------------------|---|----------------|---------------|----------------|-----------|----------|----------|----------|--------|--------|-----------------|----------------------|------------------|
| Training_source _ICAM 1-1.tif | 0.89414007 | 2 | 153 | 2 | 0.987096 | 0.987096 | 0.974522 | 0.987096 | 155 | 155 | 0.873075 | 0.884487 | 0.873075 |

$$\text{Dice} = \frac{2 \times \text{Area of overlap}}{\text{Total area}} = \frac{2 \times \text{Prediction} \cap \text{Ground truth}}{\text{Prediction} + \text{Ground truth}}$$

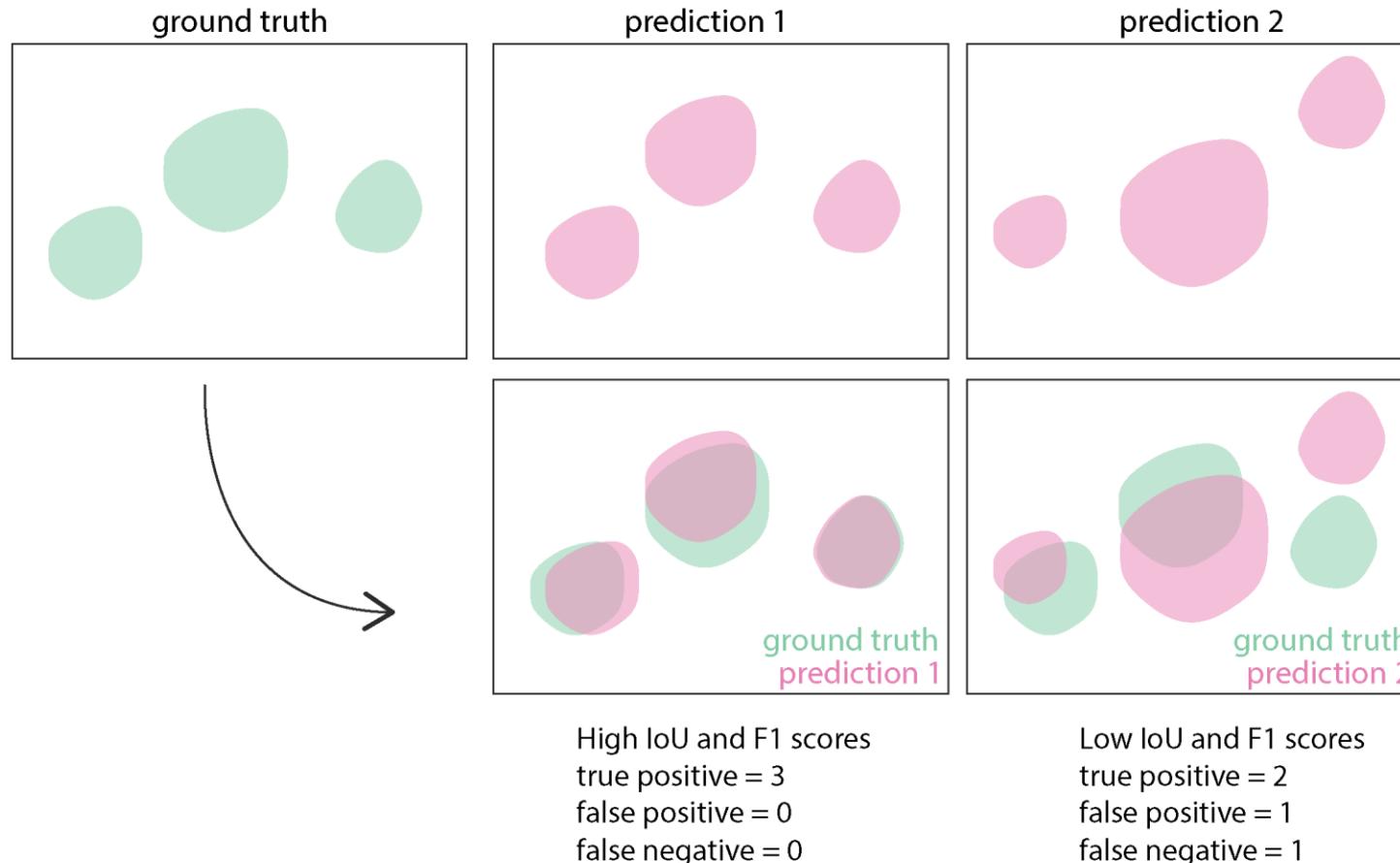
f1 score / dice score

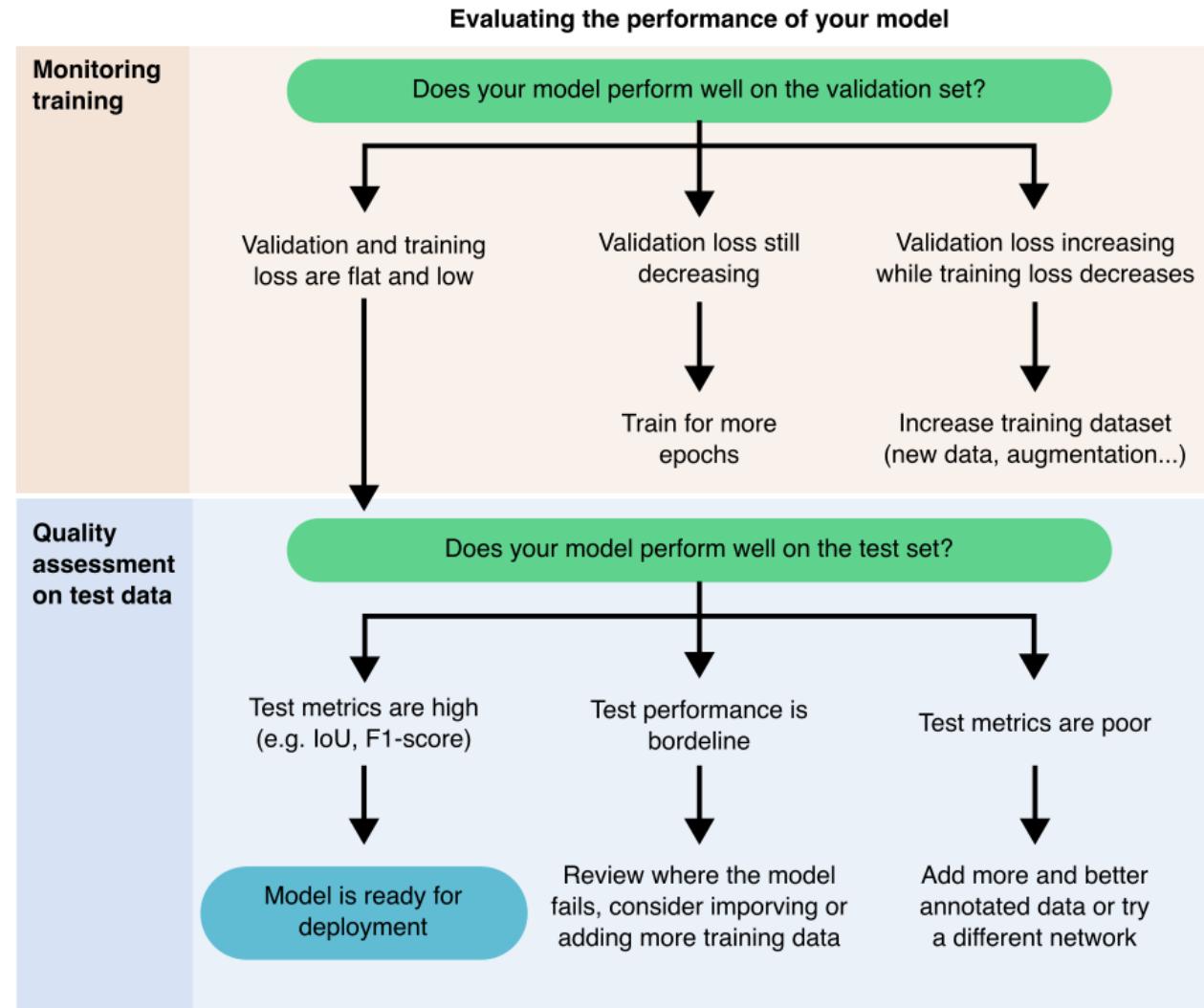
- 2 * the Area of Overlap divided by the total number of pixels in both images.

The mean_matched_score

- is the mean IoUs of matched true positives.

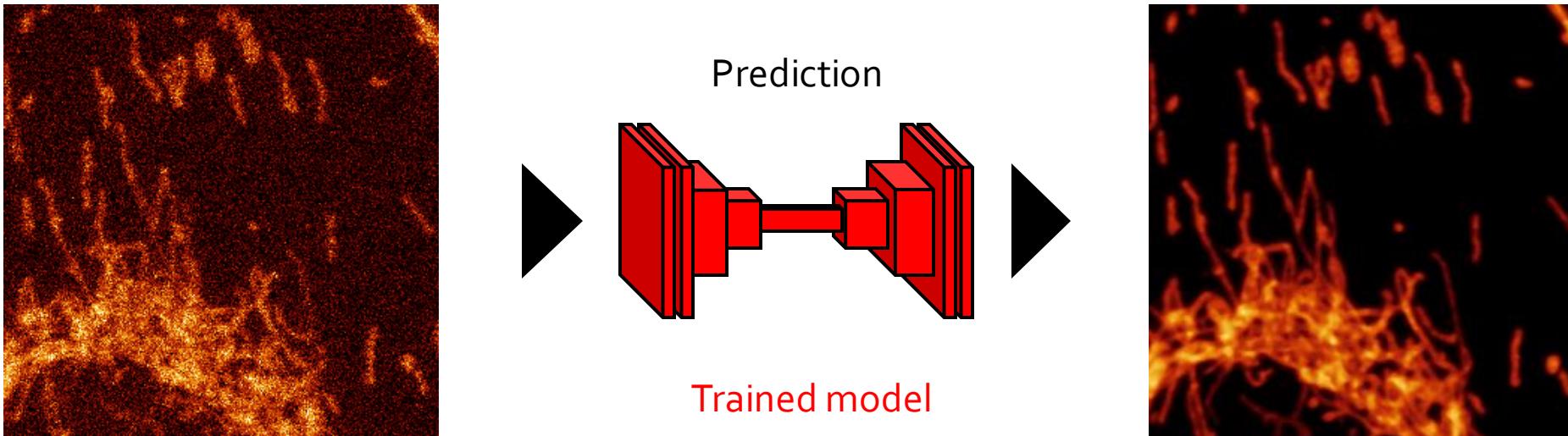
Quality control overview





MODEL EVALUATION

Step #5: Ready model can be used on related data



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

Export in Biolimage Model Zoo format

Export your model to Biolimage Model Zoo format

If you want to export the model into the [Biolimage Model Zoo](#) format, fill the metadata and run the following cell. After the cell is run a `trained_model_name.bmz.zip` file will be downloaded.

- Construct model's metadata to export it to the Biolimage Model Zoo format. Choose just one option:

Option 1: Reuse previous Biolimage Model Zoo model configuration

With this option, if you were using a model from Biolimage Model Zoo you can select this option to reuse its configuration instead of provide all fields manually. If that's not the case and you try to use this option an error will be thrown.

`reuse_previous_BMZ_model_config:`

Option 2: Manual export fields

With this option you need to introduce manually the metadata of the model.

Note: use semicolons to separate different text elements.

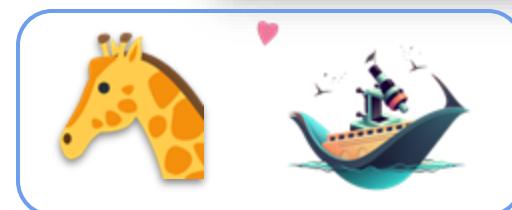
Information about the model:

| | |
|---|--|
| <code>trained_model_name:</code> | <input type="text" value="Insert text here"/> |
| <code>trained_model_authors:</code> | <input type="text" value="First Author Name; Second Author Name; Third Author Name"/> |
| <code>trained_model_authors_github_user:</code> | <input type="text" value="First Author GitHub Username; Second Author GitHub Username; Third Author GitHub Username"/> |
| <code>trained_model_description:</code> | <input type="text" value="Insert text here"/> |
| <code>trained_model_license:</code> | <input type="text" value="CC-BY-4.0"/> |
| <code>trained_model_references:</code> | <input type="text" value="Ronneberger et al. arXiv in 2015; Franco-Barranco, Daniel, et al. ISBI in 2023"/> |
| <code>trained_model_references_DOI:</code> | <input type="text" value="10.1007/978-3-319-24574-4_28; 10.1109/ISBI53787.2023.10230593"/> |
| <code>trained_model_tags:</code> | <input type="text" value="tag-1; tag-2"/> |



The grid contains the following cards:

- Input: Noisy image; Output: Clean image. Description: Denoising (2D) - BiPy. Workflow: 2D Denoising workflow where the goal is to remove the noise from...
- Input: Images; Output: Image classes. Description: Classification (2D) - BiPy. Workflow: 2D Classification workflow where full input images are labeled a...
- Input: Image + instance mask; Output: Instance mask. Description: Instance Segmentation (2D) Workflow: 2D Instance Segmentation workflow where the goal is to detect, s...
- Input: Image + mask; Output: Mask. Description: Semantic Segmentation (3D...) . Workflow: 3D Semantic Segmentation workflow where the goal is to associate...
- Input: Image + coord file; Output: Coord file. Description: Detection (3D) - BiPy. Workflow: 3D Detection workflow where the goal is to localize objects in the ...
- Input: Low res. image; Output: High res. image. Description: Super_Resolution (2D) - BiPy. Workflow: 2D Super Resolution workflow where the goal is to reconstruct hi...
- Input: Masked image; Output: Pretrained model. Description: Self_Supervision (3D) - BiPy. Workflow: 3D Self Supervision workflow where the idea is to pretrain the b...
- Input: Image + mask; Output: Mask. Description: Semantic Segmentation (2D...) . Workflow: 2D Semantic Segmentation workflow where the goal is to associate...



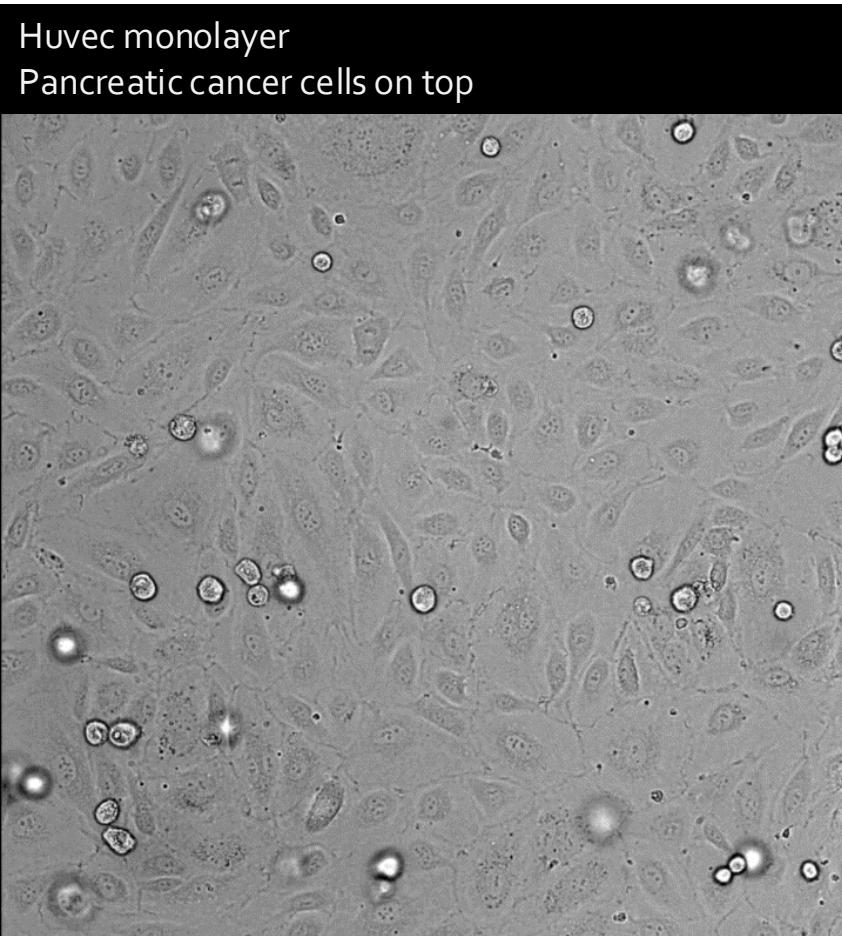
DEEP LEARNING IS NOT A QUICK FIX FOR IMAGE PROCESSING

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

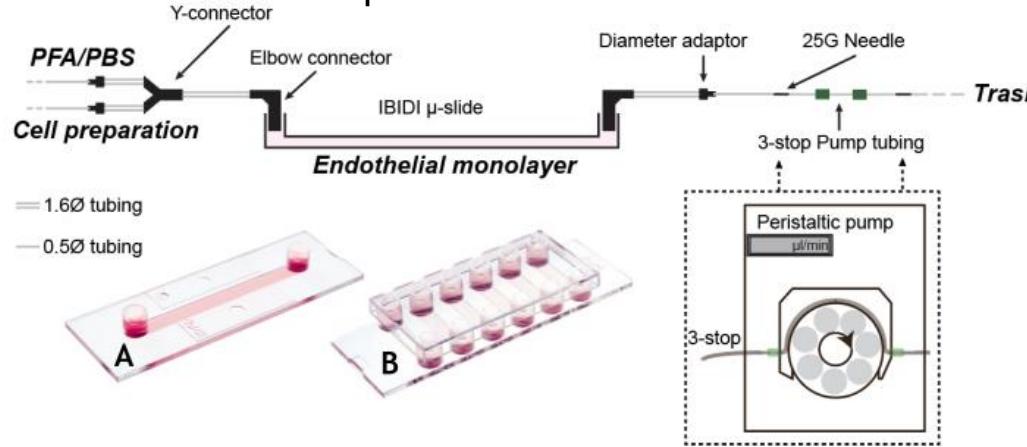
EXAMPLE PIPELINE

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

Cancer cell microfluidics



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

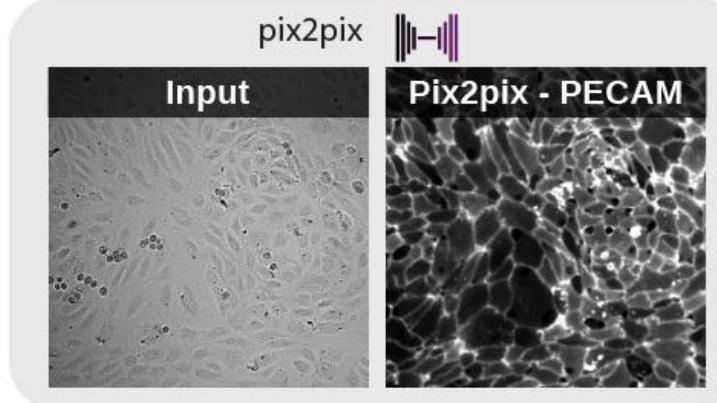


Gautier Follain

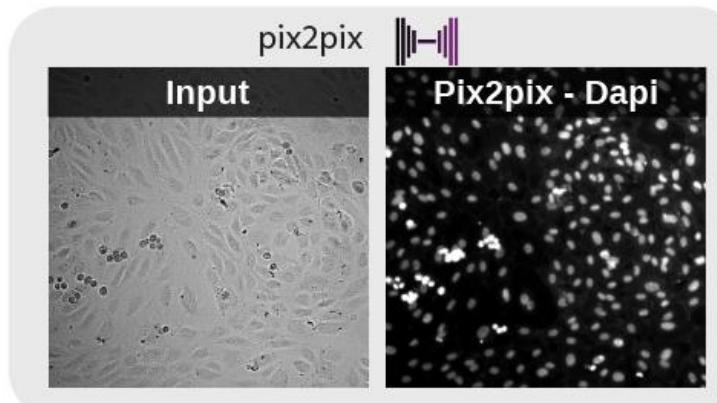


Sujan Ghimire

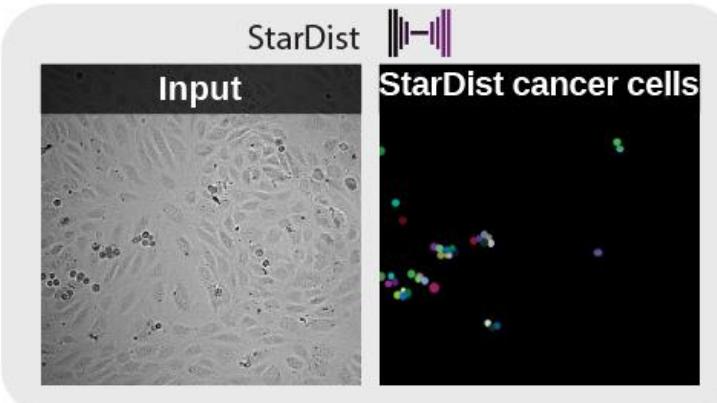




456 image pairs x aug4
fixed



258 image pairs x aug8
fixed



20 image pairs x aug8
fixed

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?

Fast label-free live imaging reveals key roles of flow dynamics and CD44-HA interaction in cancer cell arrest on endothelial monolayers

Gautier Follain, Sujan Ghimire, Joanna W. Pylyväänen, Monika Vaitkevičiūtė, Diana Wurzinger, Camilo Guzmán, James RW Conway, Michal Dibus, Sanna Oikari, Kirsi Rilla, Marko Salmi, Johanna Iivaska, Guillaume Jacquemet

doi: <https://doi.org/10.1101/2024.09.30.615654>

Contribute to BioPy



- We welcome network contributions from the research community.
- Guidelines available here:
https://biapy.readthedocs.io/en/latest/for_developers/contribute.html

Thank you!

Lab crew

- Guillaume Jacquemet
- Sujan Ghimire
- Gautier Follain
- Monika Vaitkevičiūtė
- Iván Hidalgo-Cenalmor
- Jaakko Lehtimäki
- Hanna Grobe
- Ana Gracanin
- Sarah Massaad

Collaborators

- Daniel Franco-Barranco
- Ignacio Arganda-Carreras



Syöpäsäätiö
Cancer Foundation Finland



InFLAMES
Solution is in Immunity



TURKU
BIOIMAGING
{ More than you can imagine }



LOUNAIS-SUOMEN
SYÖPÄYHDISTYS



Suomalainen
Konkordia-liitto

Interested in developing image analysis tools? Look no more, **Cell Migration lab is hiring!**

- Post-doc with coding skills (python)
- www.cellmig.org
- guillaume.jacquemet@abo.fi

