

# AUTUMN TRAINING SCHOOL 2024

---



## BIOIMAGE.IO MODEL ZOO IN ACTION: NO-CODE AI TOOLS FOR BIOIMAGE PROCESSING

CARLOS JAVIER GARCIA LOPEZ DE HARO

AI4LIFE, INSITUT PASTEUR, PARIS FRANCE

# OUTLINE

- Introduction to AI in Bioimaging
- Bioimage.io
- No-code tools
- Exercise

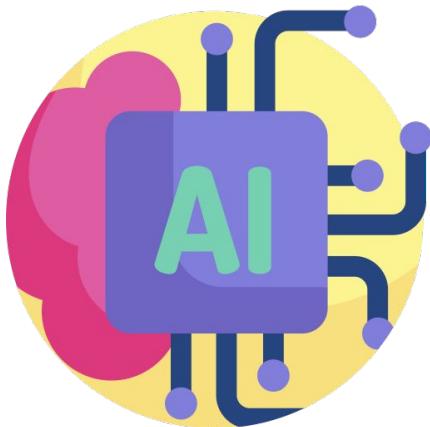
# INTRODUCTION - Artificial Intelligence in Bioimaging

---

# Artificial Intelligence (AI)

## SIMULATES HUMAN INTELLIGENCE

Artificial Intelligence (AI) is the **simulation of human intelligence** by machines.  
- ChatGPT



## DATA DRIVEN

Software used to **produce output previously thought to require human intelligence**, by extrapolating from **large collections of data**.

- Oxford dictionary

AI is here to help us

# Artificial Intelligence (AI)

AI ≈ Deep Learning

AI ≈ Algebra + Data + Compute



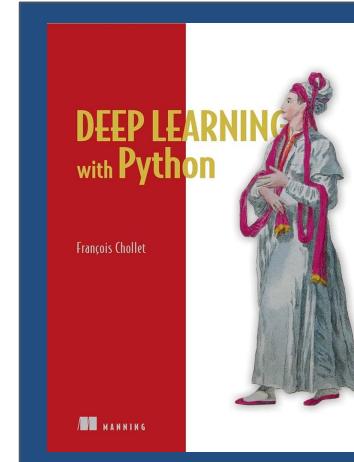
ChatGPT



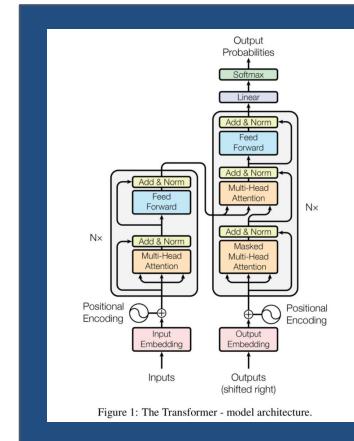
Music generation



Protein folding



François Chollet  
Deep learning  
with python.  
2021  
*free PDF online*



Vaswani et al.  
Attention is all  
you need.  
2017

# AI in Bioimage Processing

---

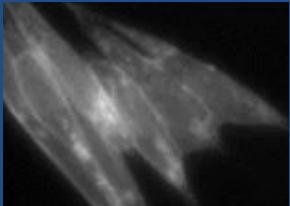
Image Reconstruction	Bioimage Analysis
Deconvolution	Image Classification
Registration	Object Classification
Super-resolution	Object detection
Tomography	Segmentation

*Belthangady and Royer, Nat. Meth., 2019  
McCann et al., IEEE 2017*

*Meijering, A bird's-eye view of Deep Learning, 2020  
Hallou et al., Development, 2021*

# AI tasks in Bioimage Processing

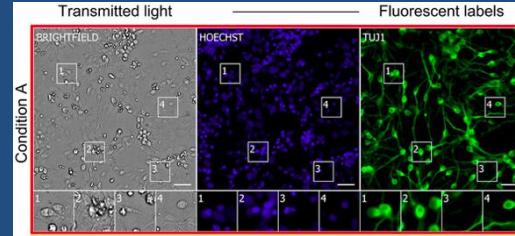
## Image Classification



*Assessing microscope image focus quality with DL*

*Yang et al., BMC Bioinformatics 2018*

## Style transfer



*In Silico Labeling: Predicting Fluorescent Labels in Unlabeled Images*

*E. M. Christiansen, Cells 2018*

## Image restoration

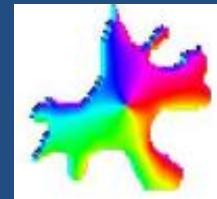
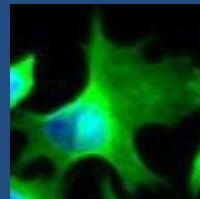


*Noise2Void*

*Krull et al., CVPR 2019*

**Runtime: ~1.3s**

## Segmentation



*Cellpose*

*Stringer et al., Nat. Methods 2021*

# When to use AI



## Recommendations

Good data available

Complex relationships

High dimensional data

## Benefits of AI

Ability to generalize

End to end learning

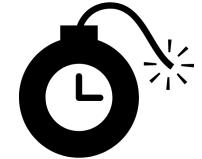
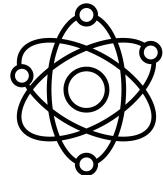
Great results

## Obstacles

Time and resources

Required skills

Model evaluation



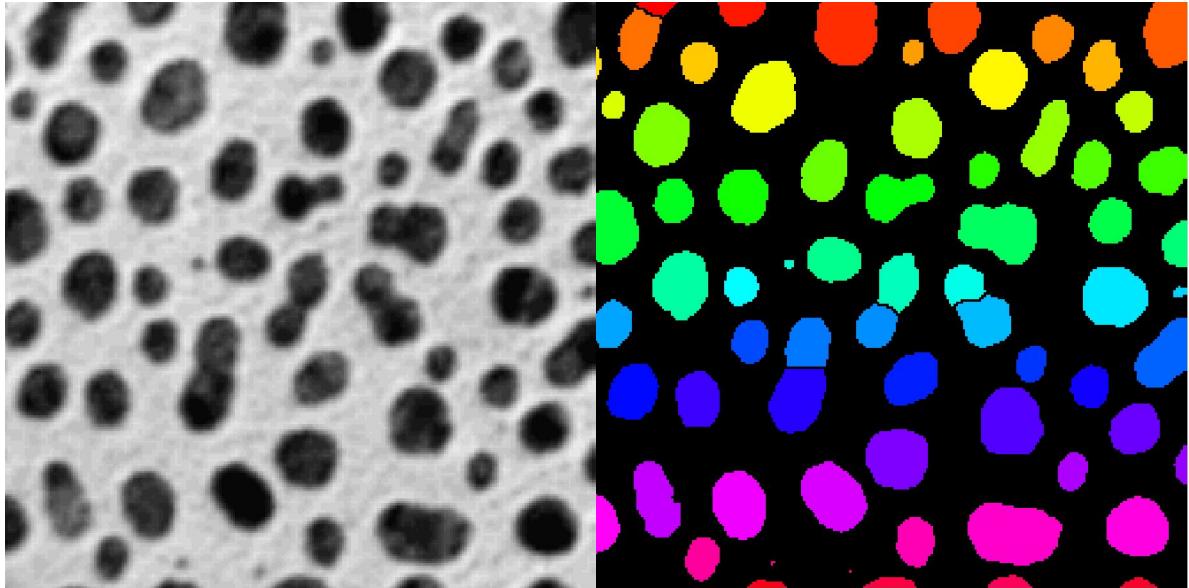
# When NOT to use AI

---

The problem is simple

There are clear patterns in the data

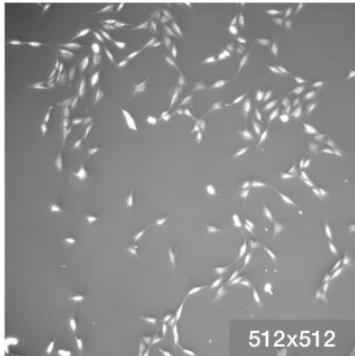
Traditional methods work well



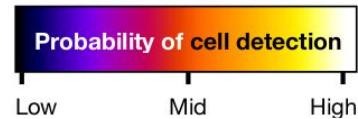
Threshold + Watershed + Connected components

# When NOT to use AI

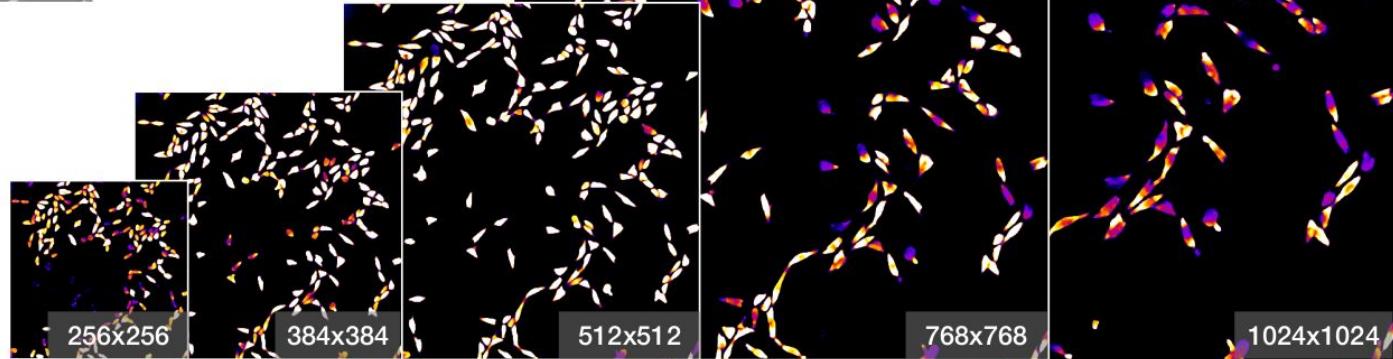
Model is not adequate for our data - **Pixel size**



Input image



Low      Mid      High



# When NOT to use AI

Model is not adequate for out data - **Hallucination**

## ANNA-PALM

Wei Ouyang et al.  
Nature Biotechnology, 2018.

Image of the dataset

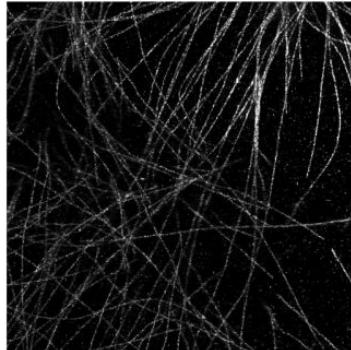
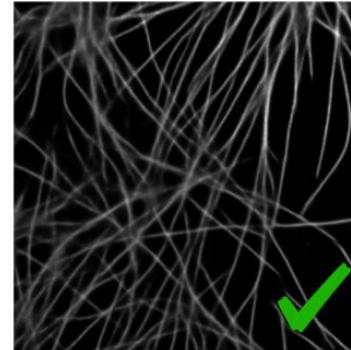


Image used to  
train the network

## Super-resolution Tubulins

Prediction NN **Tubulins**



Prediction NN **NPC**

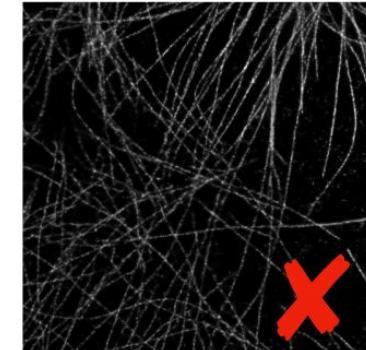
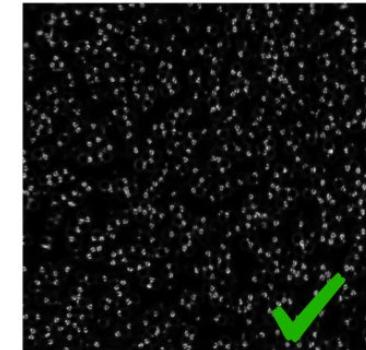
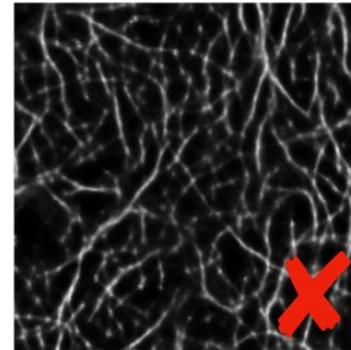
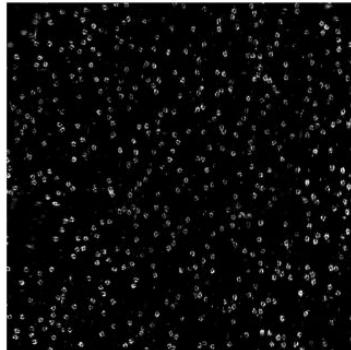


Image used to  
train the network

## Super-resolution NPC



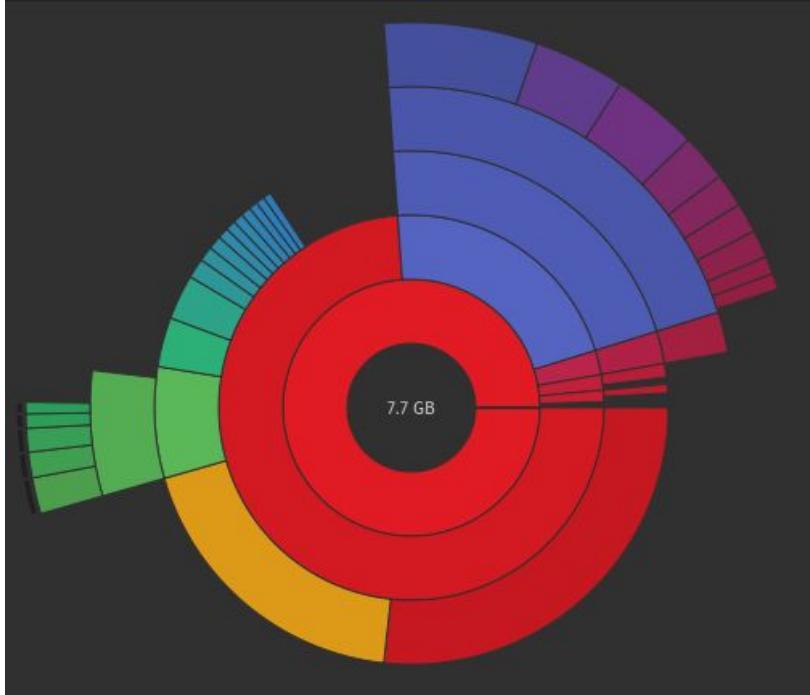
# When NOT to use AI

---

## Other considerations

Require a lot of space

Slow depending on hardware



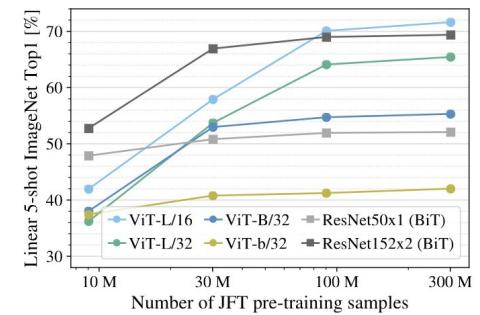
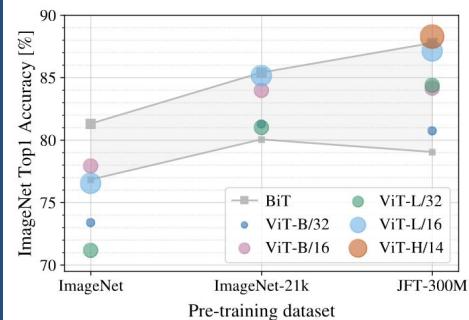
Almost 8 GB required to use SAMJ

# The importance of the DATA

Model performance depends on its **data**

Data quantity

DATA QUALITY



An image is worth 16x16 words transformers for image recognition at scale.  
Dosovitskiy et al., 2020

# The importance of the DATA

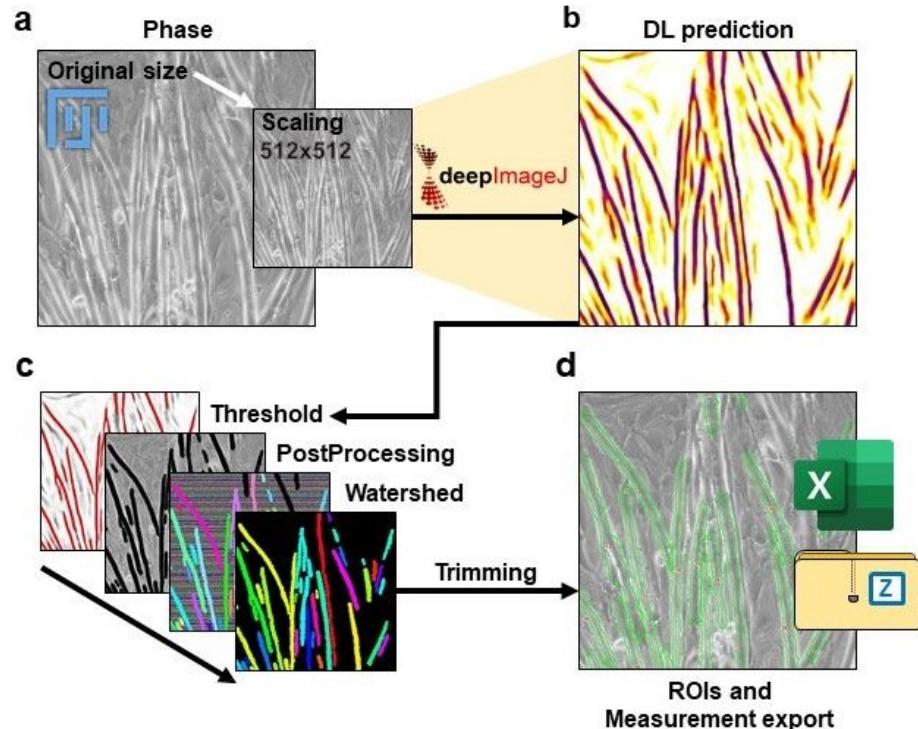
## DATA QUALITY

Small, domain-specific models with high-quality annotated data often outperform large, general-purpose models



Often

SMALL SPECIALIST > LARGE GENERALIST



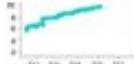
Muscle cell segmentation with **small model + post-processing**  
TRUEFAD, Brun et al., Nat. Methods 2024

# The importance of the DATA

---

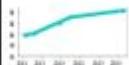
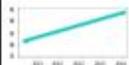
## Image classification

Natural images

Trend	Dataset	Best Model	
	ImageNet	OmniVec(ViT)	BIG
	CIFAR-10	ViT-H/14	BIG
	CIFAR-100	EffNet-L2 (SAM)	BIG

# The importance of the DATA

## Medical Image segmentation

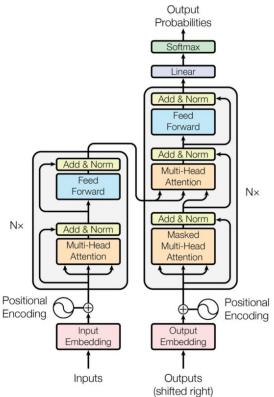
CT scans		Synapse multi-organ CT	Swin UNETR	SMALL
MRI cardiac images		Automatic Cardiac Diagnosis Challenge (ACDC)	FCT	SMALL
Tissue images		MoNuSeg	Hi-gMISnet	SMALL
Nuclei images		2018 Data Science Bowl	DuAT	BIG
Gland segmentation in Colon Histology images		GlaS	Hi-gMISnet	SMALL

# What is an AI model

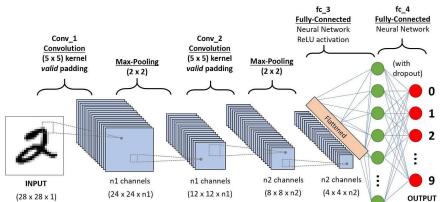
MODEL = ARCHITECTURE + WEIGHTS

## Architecture

### Transformer



## Convolutional Neural Network (CNN)



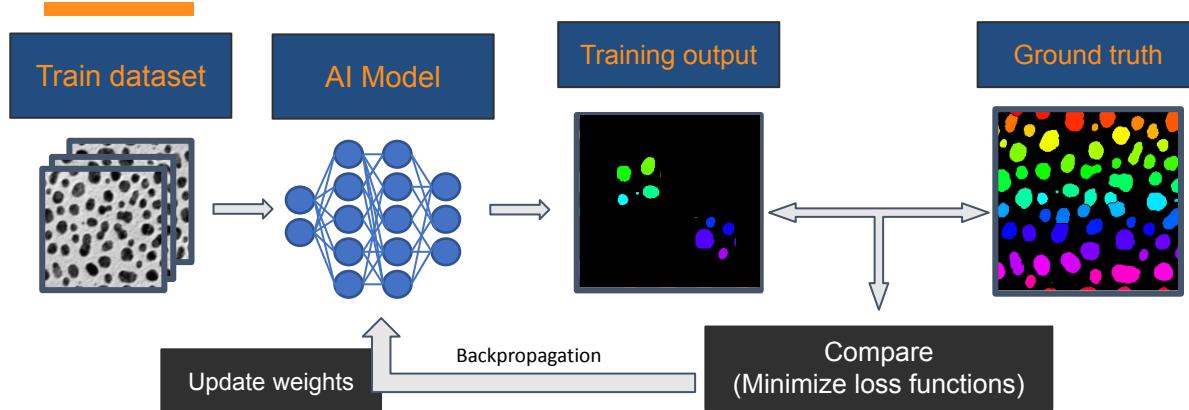
## Weights

Values that adjust connections in a neural network

Need to be **learnt** during **training**

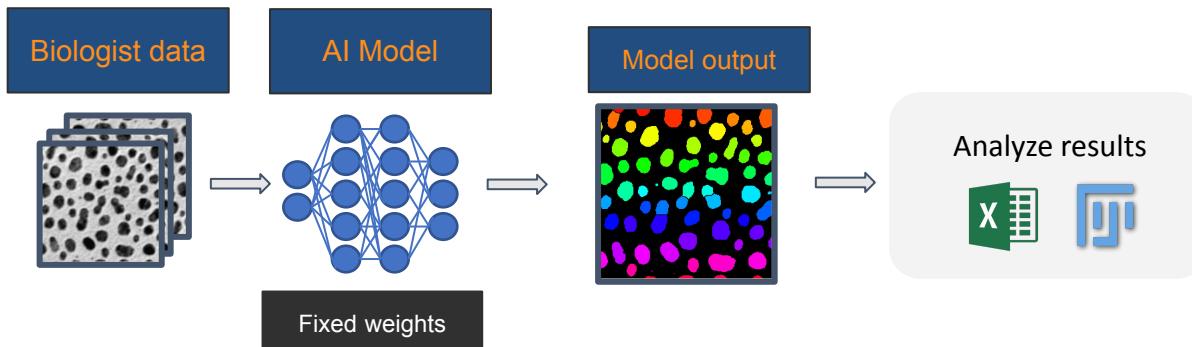
Figure 1: The Transformer - model architecture.

# Training vs Inference



## TRAINING

- Done once
- Knowledge about the problem
- Programming knowledge
- GPUs



## INFERENCE

- Done many times
- Biology knowledge
- Can be used in CPU
- End user
- Model results are not the end

# Foundational models

---

Good results across many modalities

General models

Already trained

## Detection of anything

---

Segment Anything Model (SAM):

- Plugins in Fiji, QuPath, Napari....
- 2D

## Detection of nucleus

---

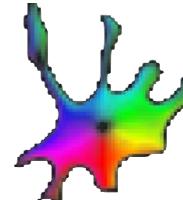


Stardist:

- Fiji/Napari plugin
- 2D/3D

## Detection of cells

---



Cellpose:

- Has its own software
- 2D

# Pre-trained models

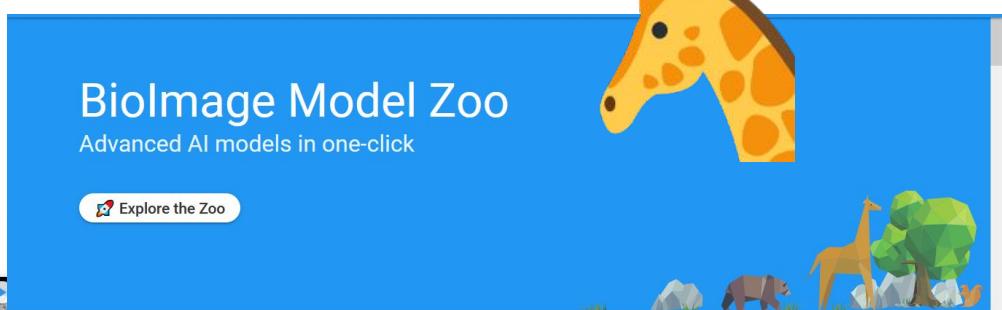
## Bioimage.io

All    **models**    applications    datasets

Type a keyword and press enter    Tags & Filters

The screenshot shows a grid of eight pre-trained AI models for segmentation, each with a thumbnail image, a title, a brief description, and download statistics.

- Neuron Segmentation in EM ...**  
Neuron segmentation in EM, trained on the CREMI challenge data.  
unet neurons instance-segmentation electron-microscopy ...  
downloads 7709 license CC-BY4.0
- MitochondriaEMSegmentati...**  
Mitochondria segmentation for electron microscopy.  
3d electron-microscopy mitochondria instance-segmentation ...  
downloads 7234 license CC-BY4.0
- PlatynereisEMcellsSegment...**  
Cell segmentation in EM of platynereis.  
unet cells instance-segmentation electron-microscopy ...  
downloads 993 license CC-BY4.0
- PlatynereisEMnucleiSegmen...**  
(organelle) segmentation in EM of platynereis.  
unet nuclei instance-segmentation electron-microscopy ...  
downloads 9495 license CC-BY4.0
- LiveCellSegmentationBound...**  
Cell segmentation for phase-contrast microscopy.  
2d transmission-light-microscopy label-free cells ...  
downloads 1000 license CC-BY4.0
- StarDist H&E Nuclei Segm...**  
StarDist - Object Detection with Star-convex Shapes  
whole-slide-imaging 2d nuclei tensorflow ...  
downloads 1000 license CC-BY4.0
- NucleiSegmentationBounda...**  
Nucleus segmentation for fluorescence microscopy  
fluorescence-light-microscopy nuclei instance-segmentation unet ...  
downloads 1000 license CC-BY4.0
- StarDist Fluorescence Nucle...**  
StarDist - Object Detection with Star-convex Shapes  
fluorescence-light-microscopy whole-slide-imaging other 2d ...  
downloads 1000 license CC-BY4.0



Ready to use already trained models

# Challenges of AI in Bioimage Processing

---

## Challenges in Bioimage Processing

**Usability and interactivity**

*Simple to use plugins and programs*

**Interpretability**

*What are the models doing? (black box)*

**Accessibility and dissemination**

*How do the end users access the best newest models?*

**Computing scalability**

*Smaller models? Stronger computers? Cloud computing?*

**Data sharing**

*We need large high-quality datasets to get better models*

**Flexibility**

*Models that work for every modality*

# Challenges of AI in Bioimage Processing

In this workshop

## Challenges in Bioimage Processing

	<b>Usability and interactivity</b>	<i>Simple to use plugins and programs</i>
	<b>Interpretability</b>	<i>What are the models doing? (black box)</i>
	<b>Accessibility and dissemination</b>	<i>How do the end users access the best newest models?</i>
	<b>Computing scalability</b>	<i>Smaller models? Stronger computers? Cloud computing?</i>
	<b>Data sharing</b>	<i>We need large high-quality datasets to get better models</i>
	<b>Flexibility</b>	<i>Models that work for every modality</i>

# Bioimage.io

---

## BioImage Model Zoo

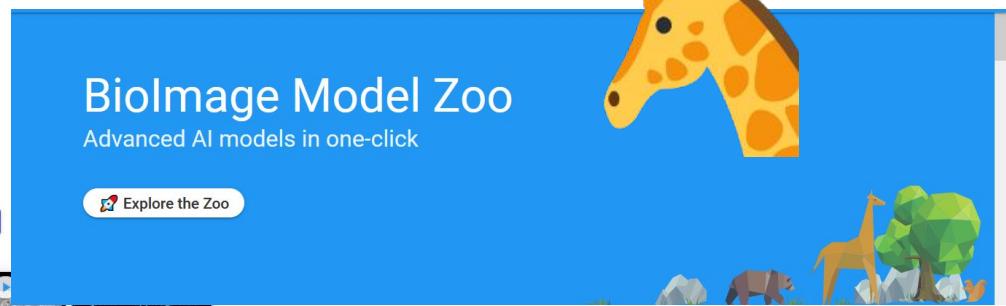
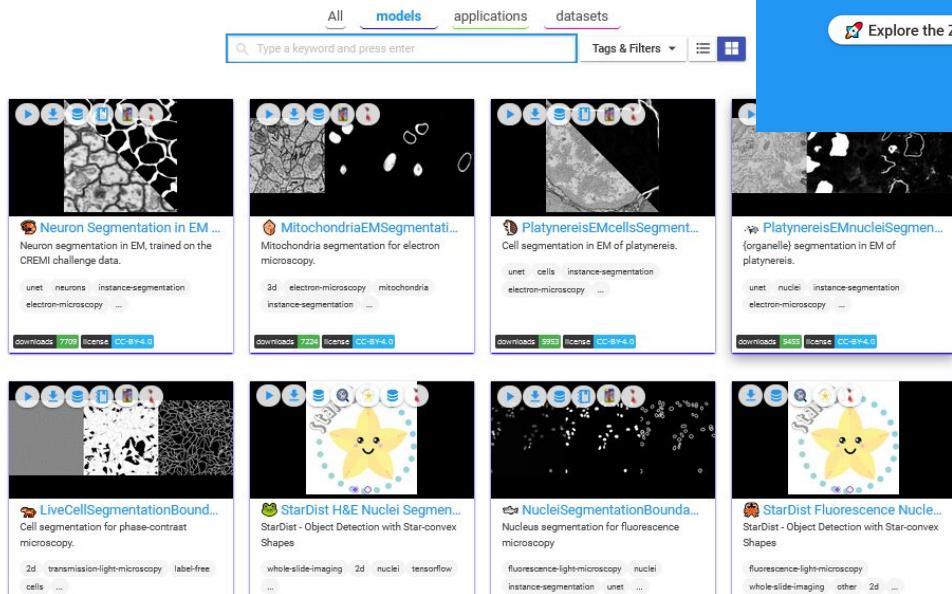
Advanced AI models in one-click

 Explore the Zoo



# Bioimage.io

<https://bioimage.io/#/>



Initiative to provide cutting edge bioimaging methods to end-users

AI models, datasets and tools

# Bioimage.io community partners

Softwares ready to use Bioimage.io models

Enhances accessibility and dissemination of models

## Biolmage Model Zoo

Advanced AI models in one-click

Integrated with Fiji, ilastik, ImJoy and more

Try model instantly with BioEngine

Link models to datasets and applications

 Explore the Zoo



### Community Partners



# Bioimage.io community partners

Softwares ready to use Bioimage.io models

Enhances accessibility and dissemination of models

## Biolmage Model Zoo

Advanced AI models in one-click

Integrated with Fiji, ilastik, ImJoy and more

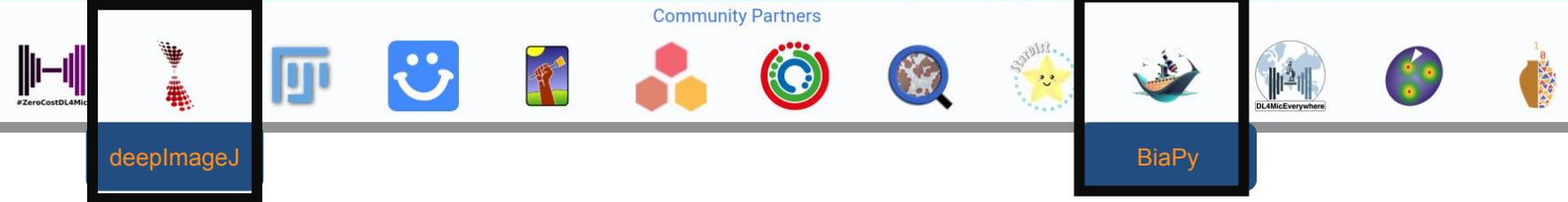
Try model instantly with BioEngine

Link models to datasets and applications

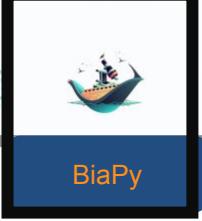
 Explore the Zoo



Community Partners



Community Partners

-  
- 
- 

# Bioimage.io community partners

---

Model creation (training)

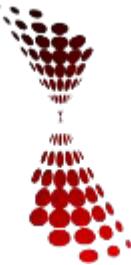


ilastik



BiaPy

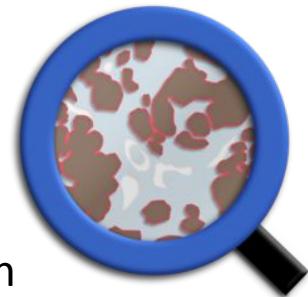
Model consumption (inference)



deepImageJ



ilastik



QuPath

# BioImage Model Zoo resources

<https://bioimage.io/#/>  
/

- ◆ **BioImage.io:** BioImage Model Zoo: A Community-Driven Resource for Accessible Deep Learning in BioImage Analysis, Ouyang et al., bioRxiv 2022 (<https://bioimage.io>)
- ◆ **DeepImageJ:** DeepImageJ: A user-friendly environment to run deep learning models in ImageJ, Gómez-de-Mariscal et al. Nature Methods 2021 (<https://deepimagej.github.io/deepimagej/>)
- ◆ **ZeroCostDL4Mic:** Democratising deep learning for microscopy with ZeroCostDL4Mic, von Charmier et al. Nature Methods 2021 (<https://github.com/HenriquesLab/ZeroCostDL4Mic/wiki>)
- ◆ **CSBDeep:** Content-aware image restoration: pushing the limits of fluorescence microscopy, Weigert et al., Nature Methods 2018 (<https://csbdeep.bioimagecomputing.com/>)
- ◆ **Ilastik:** ilastik: interactive machine learning for (bio)image analysis, Berg et al., Nature Methods 2019 (<https://www.ilastik.org/>)
- ◆ **ImJoy:** ImJoy: an open-source computational platform for the deep learning era, Ouyang et al., Nature Methods 2019 (<https://imjoy.io/#/>)
- ◆ **QuPath:** QuPath: Open source software for digital pathology image analysis, Bankhead et al, Scientific Reports 2017 (<https://qupath.github.io/>)
- ◆ **Fiji:** Fiji: an open-source platform for biological-image analysis, Schindelin et al, Nature Methods 2012 (<https://imagej.net/software/fiji/>)
- ◆ **Stardist:** Cell Detection with Star-Convex Polygons, Schmidt et al., arXiv 2018 (<https://github.com/stardist/stardist>)
- ◆ **Icy:** Icy: an open bioimage informatics platform for extended reproducible research, Chaumont et al., Nature Methods 2012 (<https://icy.bioimageanalysis.org/>)

# BioImage Model Zoo – the rdf.yaml

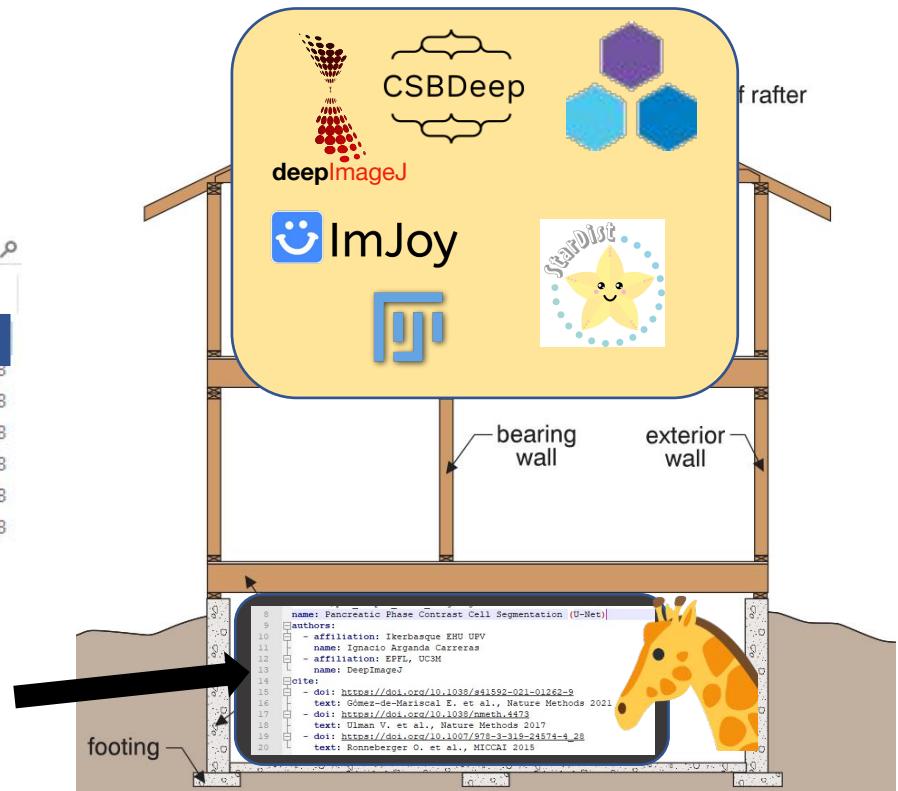
Example:  
<https://bioimage-io.github.io/collection-bioimage-io/rdf/s/10.5281/zenodo.5874741/5874742/rdf.yaml>

More than just a library of models

A **framework** to build Deep Learning applications

Name	Date modified	Type	Size
rdf.yaml	03/06/2022 15:26	YAML File	4 K
sample_input_0.tif	03/06/2022 15:26	TIF File	2,090 KB
sample_output_0.tif	03/06/2022 15:21	TIF File	27,777 KB
test_input_0.npy	03/06/2022 15:21	NPY File	2,049 KB
test_output_0.npy	03/06/2022 15:22	NPY File	16,385 KB
weights.pt	03/06/2022 15:24	PT File	333,684 KB
weights-torchscript.pt	03/06/2022 15:26	PT File	333,805 KB

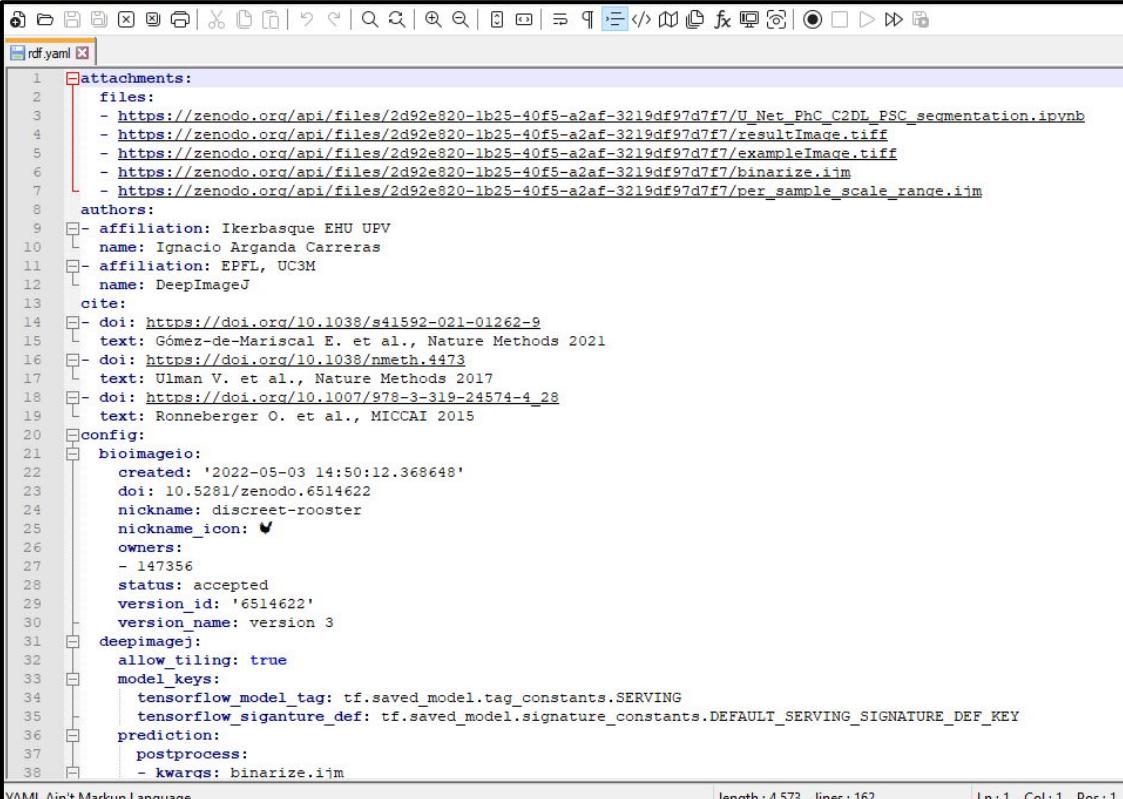
rdf.yaml



# Biolimage Model Zoo – the rdf.yaml

Parametrizes the model

Makes models compatible among softwares



The screenshot shows a code editor window with the file 'rdf.yaml' open. The code is a YAML configuration file for a machine learning model. It includes sections for attachments, authors, cite, config, and deepimagej. The attachments section lists several URLs for files like 'U\_Net\_PhC\_C2DL\_PSC\_segmentation.ipynb' and 'resultImage.tif'. The authors section lists Ignacio Arganda Carreras and DeepImageJ. The cite section references papers from Nature Methods and MICCAI conferences. The config section contains details about the model's creation and versioning, including a 'deepimagej' key that specifies 'allow\_tiling: true' and 'model\_keys' for TensorFlow saved models.

```
1 attachments:
2   files:
3     - https://zenodo.org/api/files/2d92e820-1b25-40f5-a2af-3219df97d7f7/U_Net_PhC_C2DL_PSC_segmentation.ipynb
4     - https://zenodo.org/api/files/2d92e820-1b25-40f5-a2af-3219df97d7f7/resultImage.tif
5     - https://zenodo.org/api/files/2d92e820-1b25-40f5-a2af-3219df97d7f7/exampleImage.tif
6     - https://zenodo.org/api/files/2d92e820-1b25-40f5-a2af-3219df97d7f7/binarize.iim
7     - https://zenodo.org/api/files/2d92e820-1b25-40f5-a2af-3219df97d7f7/per_sample_scale_range.iim
8   authors:
9     - affiliation: Ikerbasque EHU UPV
10       name: Ignacio Arganda Carreras
11     - affiliation: EFLL, UC3M
12       name: DeepImageJ
13   cite:
14     - doi: https://doi.org/10.1038/s41592-021-01262-9
15       text: Gómez-de-Mariscal E. et al., Nature Methods 2021
16     - doi: https://doi.org/10.1038/nmeth.4473
17       text: Ulman V. et al., Nature Methods 2017
18     - doi: https://doi.org/10.1007/978-3-319-24574-4\_28
19       text: Ronneberger O. et al., MICCAI 2015
20   config:
21     - bioimageio:
22       created: '2022-05-03 14:50:12.368648'
23       doi: 10.5281/zenodo.6514622
24       nickname: discreet-rooster
25       nickname_icon: 🐦
26       owners:
27         - 147356
28       status: accepted
29       version_id: '6514622'
30       version_name: version 3
31     - deepimagej:
32       allow_tiling: true
33       model_keys:
34         - tensorflow_model_tag: tf.saved_model.tag_constants.SERVING
35         - tensorflow_signature_def: tf.saved_model.signature_constants.DEFAULT_SERVING_SIGNATURE_DEF_KEY
36       prediction:
37         - postprocess:
38           - kwargs: binarize.iim
```

YAML Ain't Markup Language

length: 4,573 lines: 162

Ln:1 Col:1 Pos:1

# No-code tools

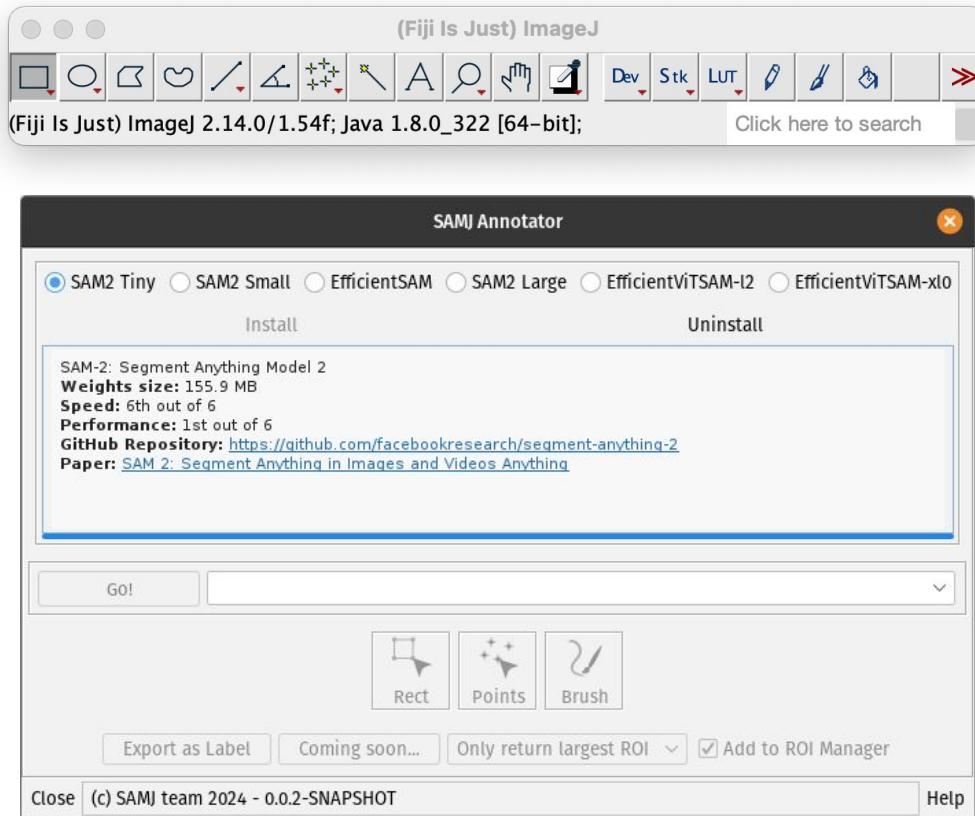
---



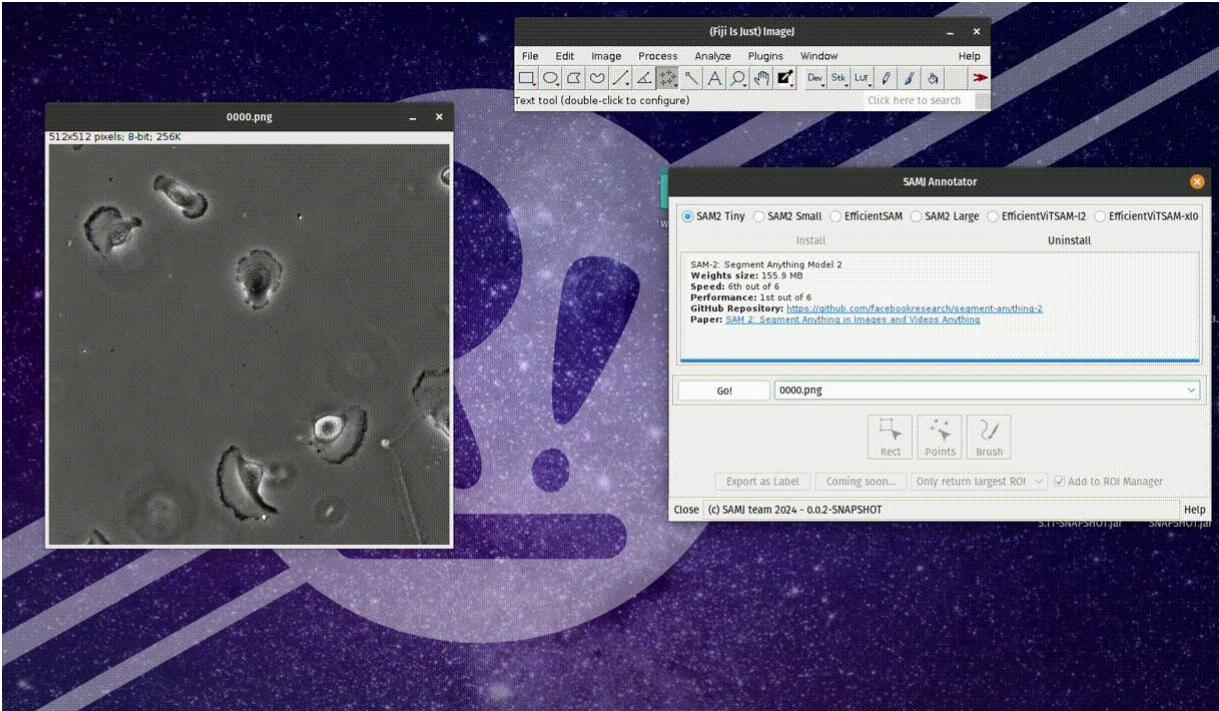
Fiji plugin using AI for  
semiautomatic annotation

Click or draw a bounding box  
on the area of interest

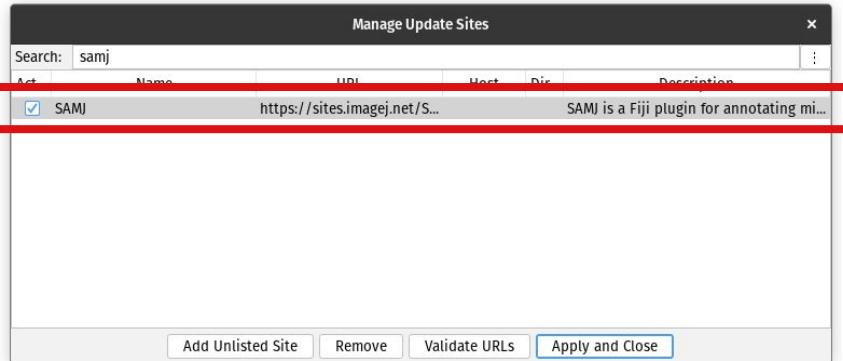
AI model draws a ROI around  
the object interest



## Example video



# SAMJ -Installation



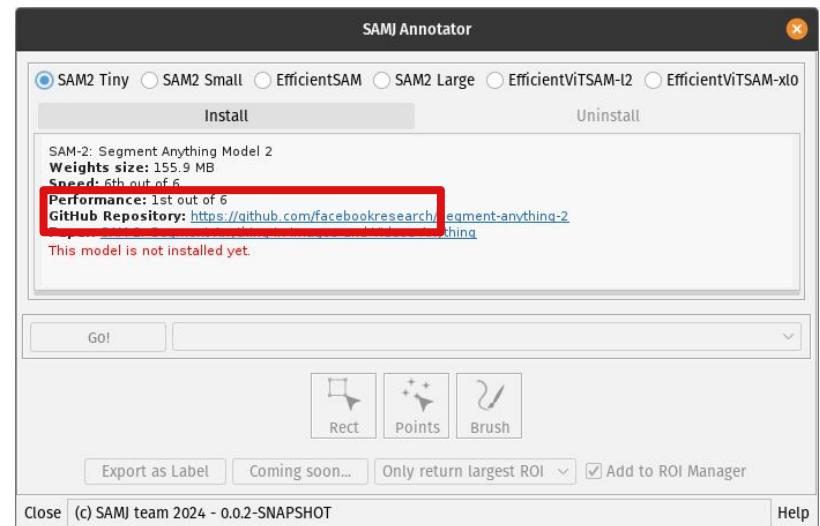
Act.	Name	URL	Host	Dir	Description
<input checked="" type="checkbox"/>	SAMJ	https://sites.imagej.net/S...			SAMJ is a Fiji plugin for annotating mi...

Add Unlisted Site Remove Validate URLs Apply and Close



Depending on hardware: 2~20 mins

## Add SAMJ in Fiji Update Site



SAMJ Annotator

SAM2 Tiny  SAM2 Small  EfficientSAM  SAM2 Large  EfficientViTSAM-L2  EfficientViTSAM-xLO

Install Uninstall

SAM-2: Segment Anything Model 2  
Weights size: 155.9 MB  
Speed: 6th out of 6

Performance: 1st out of 6  
GitHub Repository: <https://github.com/facebookresearch/segment-anything-2>

This model is not installed yet.

Go! ▾

Rect Points Brush

Export as Label Coming soon... Only return largest ROI Add to ROI Manager

Close (c) SAMJ team 2024 - 0.0.2-SNAPSHOT Help

# SAMJ -Segment Anything Model (SAM)

Promptable Segmentation  
(bounding box and points)

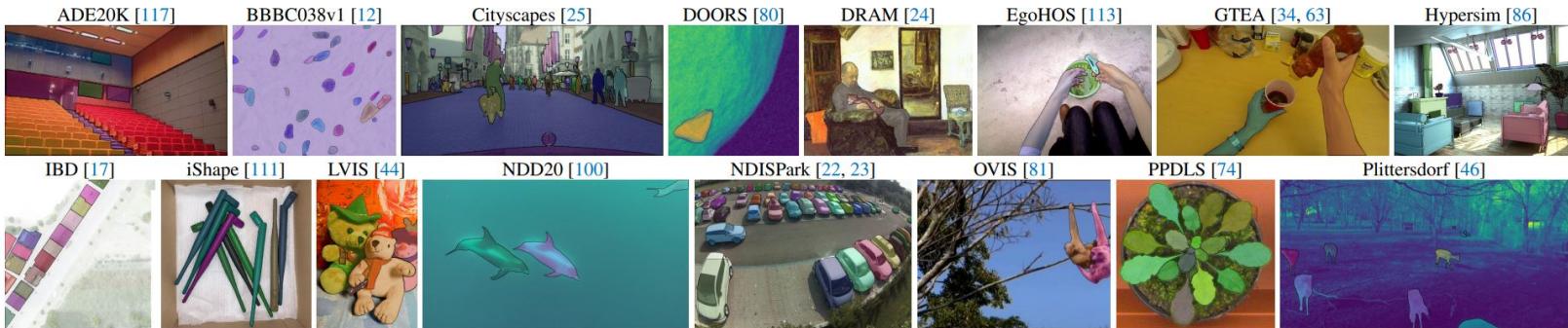
Trained by Meta AI

1 Billion masks, 11 Million images

Diverse and high-resolution images

Manual to automatic annotation process

Vision Transformer-based Architecture  
(ViT)



# BiaPy

Create and train AI based workflows without code

Graphical Interface or Python notebooks

2D and 3D



The screenshot shows the BiaPy graphical user interface. At the top right is a window title bar with a close button. Below it is a section titled "External links" with icons for GitHub, Forum, Documentation, Templates, Notebooks, and Citation. The main area has a sidebar on the left with icons for Home, Wizard, Workflow, Generic options, Train, Test, and Run Workflow. To the right of the sidebar are sections for "Docker dependency" (showing "Docker installation found") and "GPU dependency" (showing "2 NVIDIA GPU cards found"). A central message says "First timers: talk to the Wizard!" with a cartoon character. At the bottom right are buttons for "Create new workflow" and "Load and modify workflow".

**BiaPy: A unified framework for versatile bioimage analysis with deep learning**

Daniel Franco-Barranco, Jesús A. Andrés-San Román, Iván 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

doi: <https://doi.org/10.1101/2024.02.03.576026>

# DeepImageJ

Run pre-trained AI models without code

Models should be in Bioimage.io format

deepImageJ



[nature](#) > [nature methods](#) > [brief communications](#) > [article](#)

Brief Communication | Published: 30 September 2021

## DeepImageJ: A user-friendly environment to run deep learning models in ImageJ

[Estibaliz Gómez-de-Mariscal](#), [Carlos García-López-de-Haro](#), [Wei Ouyang](#), [Laurène Donati](#), [Emma Lundberg](#), [Michael Unser](#), [Arrate Muñoz-Barrutia](#) & [Daniel Sage](#)

[Nature Methods](#) 18, 1192–1195 (2021) | [Cite this article](#)

23k Accesses | 308 Altmetric | [Metrics](#)

The screenshot shows the deepImageJ-3.1.0-SNAPSHOT window. At the top, there is a logo of a flask with red dots and the text "deepImageJ" and "The Fiji/ImageJ Plugin for AI". Below the logo, there are two tabs: "Search" and "Bioimage.io". The main area has a "Local" tab selected, displaying three thumbnail images: "2D\_Ganglia\_RGB\_v2" (a grayscale image with green and magenta overlays), "B. Sutilist bacteria segment..." (a grayscale image with yellow outlines), and "fluorescence\_nuclei\_and\_ce..." (a grayscale image with red and green overlays). Below these thumbnails is a "Cover Image" section showing a grayscale image of bacteria with yellow outlines. To the right, there is a "Model Information" panel for the "B. Sutilist bacteria segmentation - Widefield microscopy - 2D UNet" model. It includes fields for "Nickname" (placid-llama), "Description" (a detailed text about the trained 2D U-Net model), "Folder name" (B. Sutilist bacteria segmentation - Widefield microscopy - 2D UNet\_25112024\_003825), and "Author(s)". At the bottom right are buttons for "Cancel", "Run on test", and "Run".

# DeepImageJ - Installation

Manage Update Sites

Act...	Name	URI	Host	Dir...	Description
<input checked="" type="checkbox"/>	DeepImageJ	https://sites.imagej.net/...	webdav:c...		Bioimage processing using (pre-)train...
<input type="checkbox"/>	Gut Analysis Toolbox	https://sites.imagej.net/...			Gut Analysis Toolbox enables semi-autom...

Add Unlisted Site Remove Validate URLs Apply and Close

Add deepImageJ in Fiji Update Site

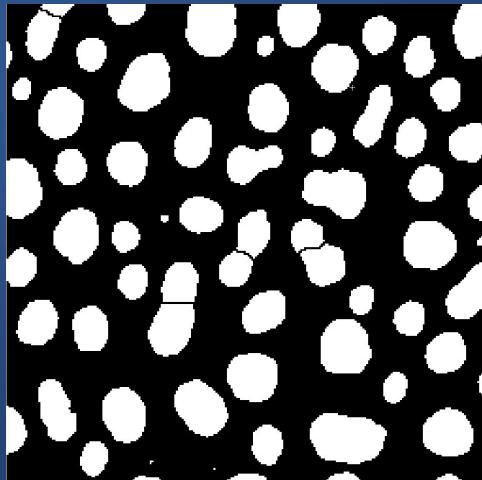
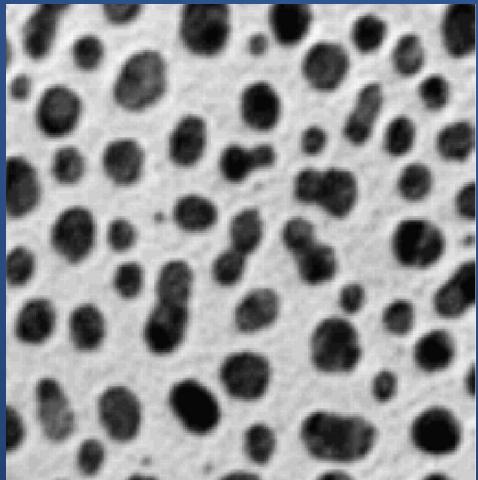
Start deepImageJ to initiate automatic installations



deepImageJ

# Exercise - Semantic Segmentation

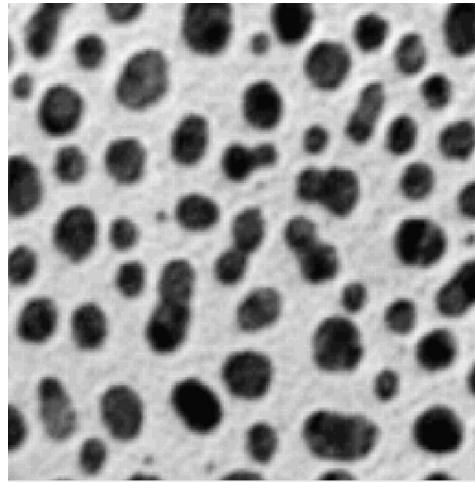
---



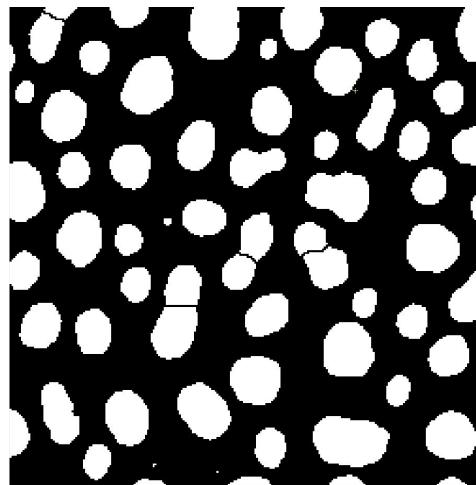
# Semantic segmentation

---

*Classifying every pixel in an image by category.  
For example: **background vs foreground***



Original image

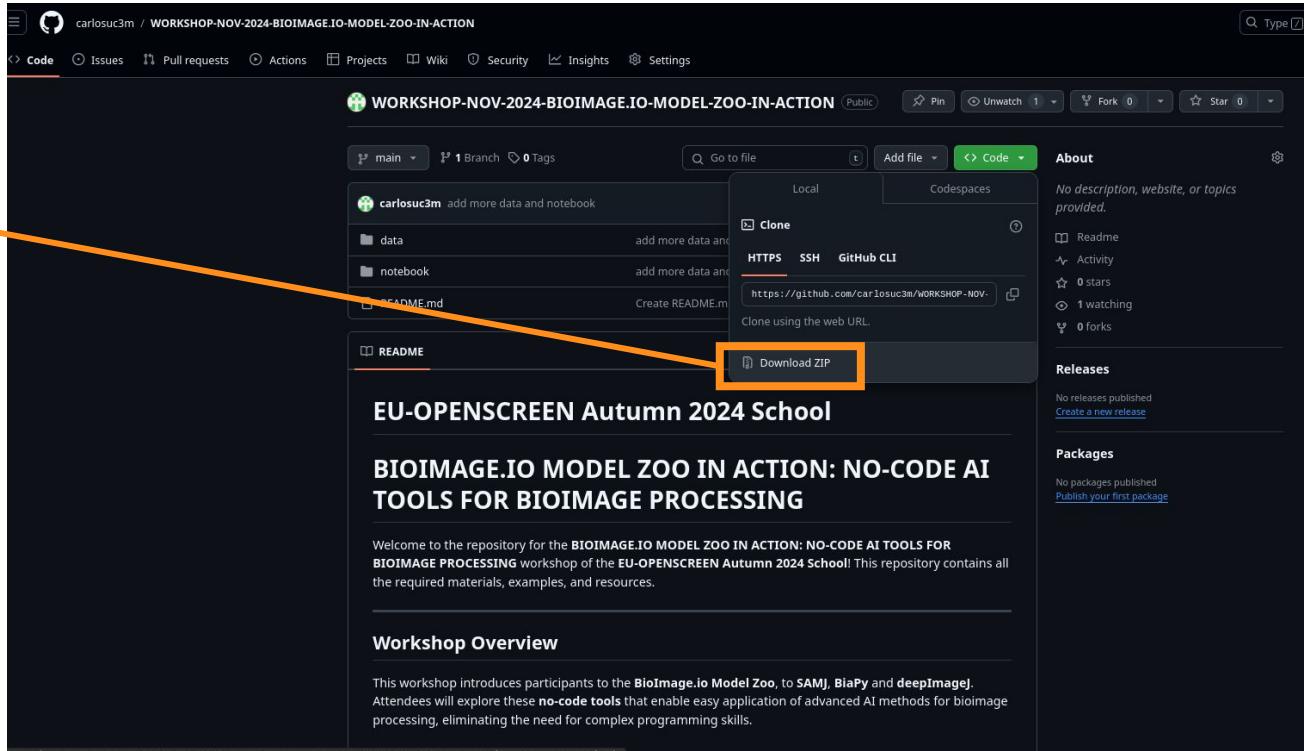


Semantic segmentation  
(pixel classification)

# Materials

Download ZIP

Unzip it



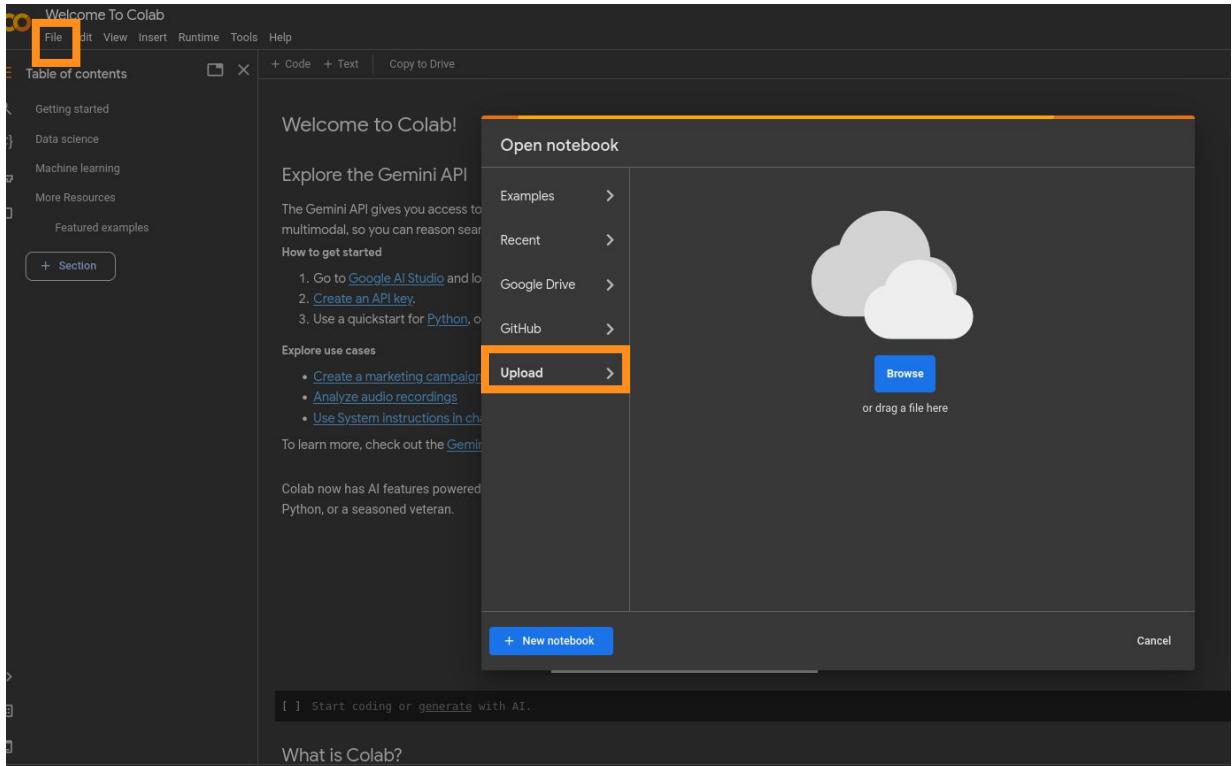
# BiaPy preparation

**File>Open Notebook>Upload**

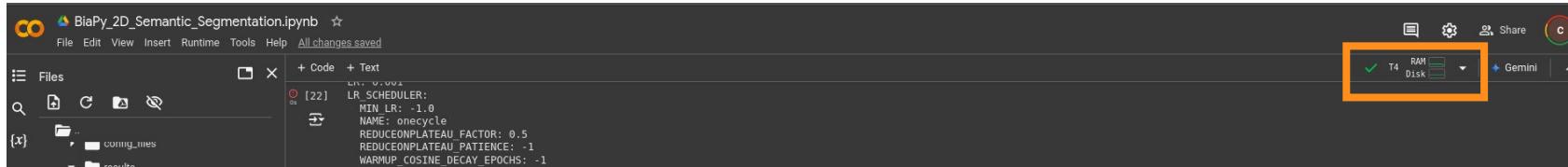
**Browse** and select the notebook we downloaded in the previous step

Inside the unzipped folder

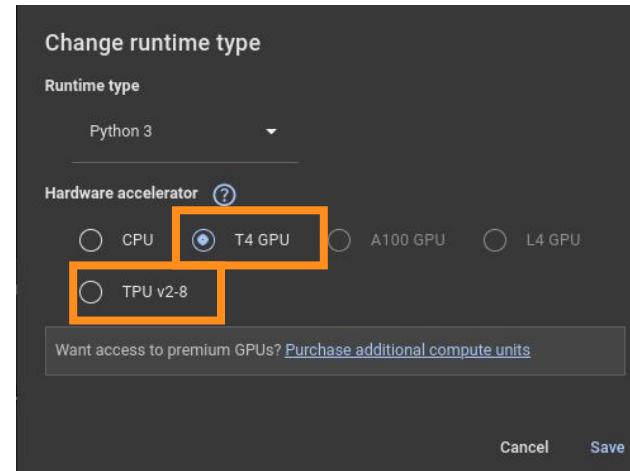
*notebooks/BiaPy\_2D\_Semantic\_Segmentation\_DL\_MODIFIED.ipynb*



# BiaPy preparation - GPU

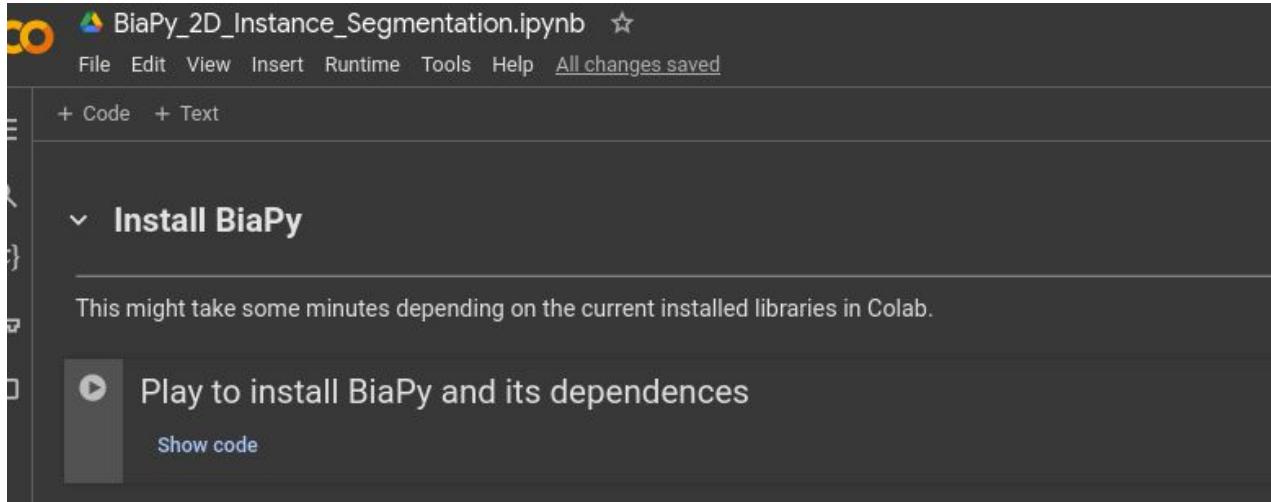


**Check that the runtime is not on CPU**



# BiaPy installation

---



**Click on the play button** to install dependencies (takes time)

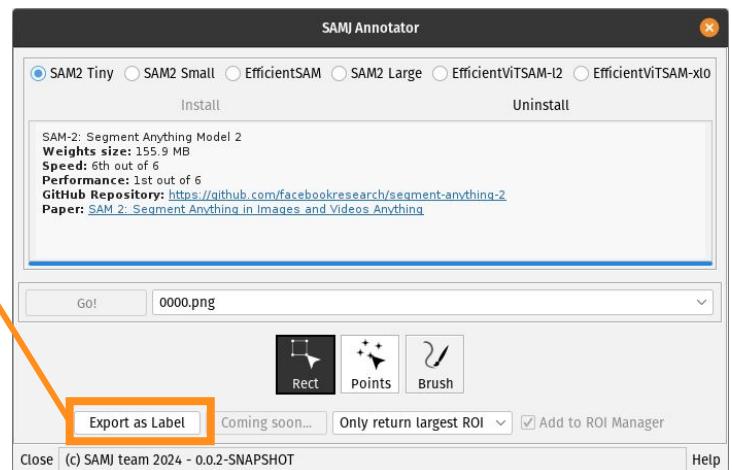
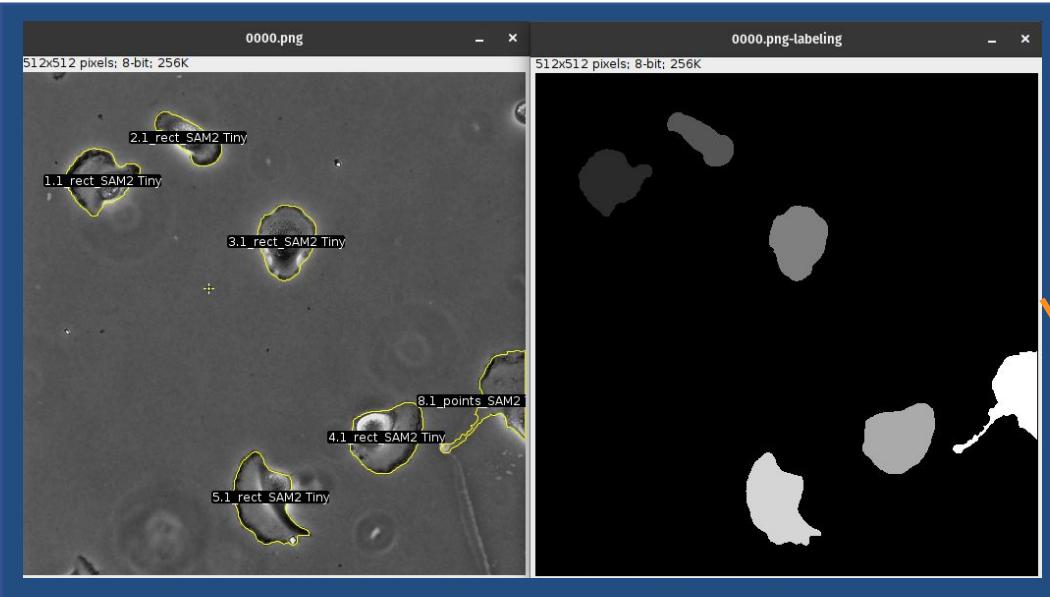
# Annotate your data with SAMJ

Open Fiji  
Open SAMJ

Open an image in Fiji from the folder **to\_annotate** in one of the samples datasets

Open SAMJ, annotate the image and then click on “Export as Label”

Save the image in the “**groundthruth**” folder



# Train and export a model with BiaPy

[Go back to Google Colab](#)

Click on every “play button”

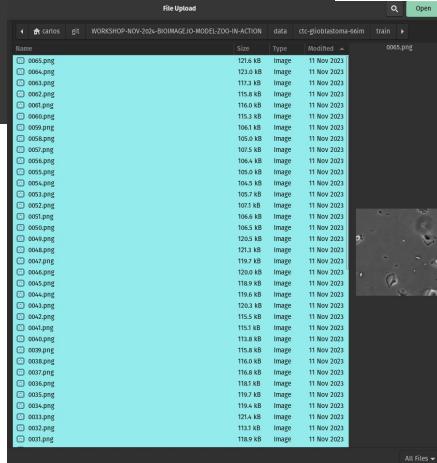
Upload Files from Your Local Machine

When you select this option, you'll be prompted to upload your files to Colab. Once uploaded, they will be stored in the `/content/input/` directory.

Play the cell to upload local files (train raw images)

Show code

... /content/input/train/x  
Browse... No files selected. Cancel upload



When asked to upload train raw images, upload the train sources

# Train and export a model with BiaPy

- ➊ Play the cell to upload local files (train label images)

Show code

```
... /content/input/train/y  
Browse... No files selected.
```

Cancel upload

**When asked to upload train label images, upload the train groundtruth**

**In the next cells, do the same but for test samples**

Name	Size	Type	Modified
0000.png	3.1 kB	Image	10:37 PM
0001.png	3.0 kB	Image	10:37 PM
0002.png	3.2 kB	Image	10:37 PM
0003.png	3.1 kB	Image	10:37 PM
0004.png	2.5 kB	Image	10:37 PM
0005.png	2.8 kB	Image	10:37 PM
0006.png	2.8 kB	Image	10:37 PM
0007.png	2.8 kB	Image	10:37 PM
0008.png	3.1 kB	Image	10:37 PM
0009.png	3.4 kB	Image	10:37 PM
0010.png	3.4 kB	Image	10:37 PM
0011.png	3.1 kB	Image	10:37 PM
0012.png	3.6 kB	Image	10:37 PM
0013.png	3.4 kB	Image	10:37 PM
0014.png	3.5 kB	Image	10:37 PM
0015.png	3.4 kB	Image	10:37 PM
0016.png	3.1 kB	Image	10:37 PM
0019.png	2.1 kB	Image	10:37 PM
0020.png	2.0 kB	Image	10:37 PM
0021.png	2.2 kB	Image	10:37 PM
0022.png	2.2 kB	Image	10:37 PM
0023.png	2.2 kB	Image	10:37 PM
0024.png	2.1 kB	Image	10:37 PM
0025.png	1.9 kB	Image	10:37 PM
0026.png	2.1 kB	Image	10:37 PM
0027.png	2.3 kB	Image	10:37 PM
0028.png	1.9 kB	Image	10:37 PM
0029.png	2.0 kB	Image	10:37 PM
0030.png	2.1 kB	Image	10:37 PM
0031.png	2.0 kB	Image	10:37 PM
0032.png	1.8 kB	Image	10:37 PM
0033.png	2.8 kB	Image	10:37 PM
0034.png	3.1 kB	Image	10:37 PM
0035.png	2.9 kB	Image	10:37 PM
0036.png	2.8 kB	Image	10:37 PM

# Train and export a model with BiaPy

---

## Define name and parameters

Only change the name

The screenshot shows a configuration interface for a machine learning model named "my\_2d\_instance\_segmentation\_2". The interface includes sections for Name of the model, Data management, Basic training parameters, Advanced training parameters, and other optional settings.

Parameter	Value
Name of the model:	model_name: "my_2d_instance_segmentation_2"
Data management:	test_ground_truth: <input type="checkbox"/> percentage_validation: 20
Basic training parameters:	input_channels: 1 number_of_epochs: 10 patience: 2
Advanced training parameters:	model_architecture: U-Net batch_size: 8 patch_size: 256 optimizer: ADAMW initial_learning_rate: 0.0001 problem_representation: Binary mask + Contours learning_rate_scheduler: None aggressive_data_augmentation: <input type="checkbox"/> test_time_augmentation: <input type="checkbox"/>

# Train and export a model with BiaPy

Click on train

## Play to train the model

Show code

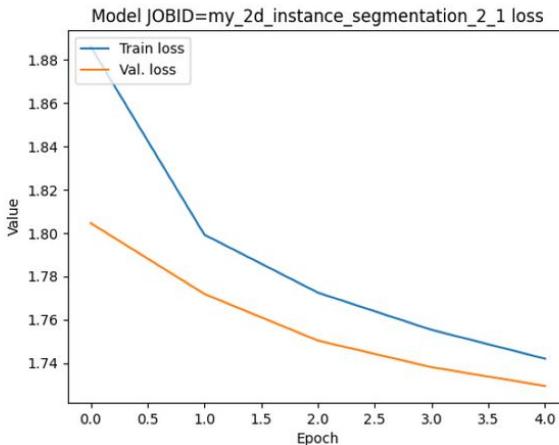
```
...  
DA_CRAFT_TRAIN: /content/output/my_2d_instance_segmentation_2/results/my_2d_instance_segmentation_2_1/DA_CRAFT  
CHARTS: /content/output/my_2d_instance_segmentation_2/results/my_2d_instance_segmentation_2_1/charts  
CHECKPOINT: /content/output/my_2d_instance_segmentation_2/checkpoints  
CHECKPOINT_FILE:  
DA_SAMPLES: /content/output/my_2d_instance_segmentation_2/results/my_2d_instance_segmentation_2_1/aug  
GEN_CHECKS: /content/output/my_2d_instance_segmentation_2/results/my_2d_instance_segmentation_2_1/gen_check  
GEN_MASK_CHECKS: /content/output/my_2d_instance_segmentation_2/results/my_2d_instance_segmentation_2_1/gen_mask_check  
LWR_X_FILE: /content/output/my_2d_instance_segmentation_2/checkpoints/lower_bound_X_perc.npy  
LWR_Y_FILE: /content/output/my_2d_instance_segmentation_2/checkpoints/lower_bound_Y_perc.npy  
MAE_OUT_DIR: /content/output/my_2d_instance_segmentation_2/results/my_2d_instance_segmentation_2_1/MAE_checks  
MEAN_INFO_FILE: /content/output/my_2d_instance_segmentation_2/checkpoints/normalization_mean.npy  
PROB_MAP_DIR: /content/output/my_2d_instance_segmentation_2/prob_map  
PROB_MAP_FILENAME: prob_map.npy  
PROFILER: /content/output/my_2d_instance_segmentation_2/results/my_2d_instance_segmentation_2_1/profiler  
RESULT_DIR:  
AS_3D_STACK: /content/output/my_2d_instance_segmentation_2/results/my_2d_instance_segmentation_2_1/as_3d_stack  
AS_3D_STACK_BIN: /content/output/my_2d_instance_segmentation_2/results/my_2d_instance_segmentation_2_1/as_3d_stack_binarized  
AS_3D_STACK_POST_PROCESSING: /content/output/my_2d_instance_segmentation_2/results/my_2d_instance_segmentation_2_1/as_3d_stack_post_processing  
DET_ASSOC_POINTS: /content/output/my_2d_instance_segmentation_2/results/my_2d_instance_segmentation_2_1/point_associations  
DET_LOCAL_MAX_COORDS_CHECK: /content/output/my_2d_instance_segmentation_2/results/my_2d_instance_segmentation_2_1/per_image_local_max_check  
FULL_IMAGE: /content/output/my_2d_instance_segmentation_2/results/my_2d_instance_segmentation_2_1/full_image  
FULL_IMAGE_BIN: /content/output/my_2d_instance_segmentation_2/results/my_2d_instance_segmentation_2_1/full_image_binarized  
FULL_IMAGE_INSTANCES: /content/output/my_2d_instance_segmentation_2/results/my_2d_instance_segmentation_2_1/full_image_instances  
FULL_IMAGE_POST_PROCESSING: /content/output/my_2d_instance_segmentation_2/results/my_2d_instance_segmentation_2_1/full_image_post_processing  
INST_ASSOC_POINTS: /content/output/my_2d_instance_segmentation_2/results/my_2d_instance_segmentation_2_1/instance_associations  
PATH: /content/output/my_2d_instance_segmentation_2/results/my_2d_instance_segmentation_2_1  
PER_IMAGE: /content/output/my_2d_instance_segmentation_2/results/my_2d_instance_segmentation_2_1/per_image  
PER_IMAGE_BIN: /content/output/my_2d_instance_segmentation_2/results/my_2d_instance_segmentation_2_1/per_image_binarized  
PER_IMAGE_INSTANCES: /content/output/my_2d_instance_segmentation_2/results/my_2d_instance_segmentation_2_1/per_image_instances  
PER_IMAGE_POST_PROCESSING: /content/output/my_2d_instance_segmentation_2/results/my_2d_instance_segmentation_2_1/per_image_post_processing  
STD_INFO_FILE: /content/output/my_2d_instance_segmentation_2/checkpoints/normalization_std_value.npy  
TEST_FULL_GT_H5: /content/output/test/yh5  
TEST_INSTANCE_CHANNELS_CHECK: /content/output/my_2d_instance_segmentation_2/results/my_2d_instance_segmentation_2_1/test_BC_instance_channels  
TRAIN_INSTANCE_CHANNELS_CHECK: /content/output/my_2d_instance_segmentation_2/results/my_2d_instance_segmentation_2_1/train_BC_instance_channels  
UPR_X_FILE: /content/output/my_2d_instance_segmentation_2/checkpoints/upper_bound_X_perc.npy
```

# Train and export a model with BiaPy - Validation

Needs to go down

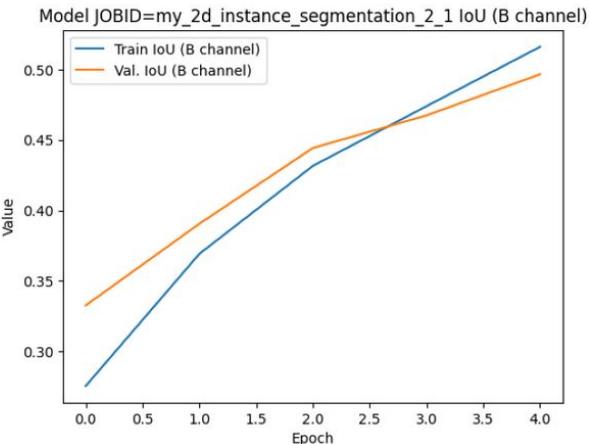
Play the cell to show a plot of training error vs. epoch number and IoU vs epoch number  
Show code

Training error vs epoch number

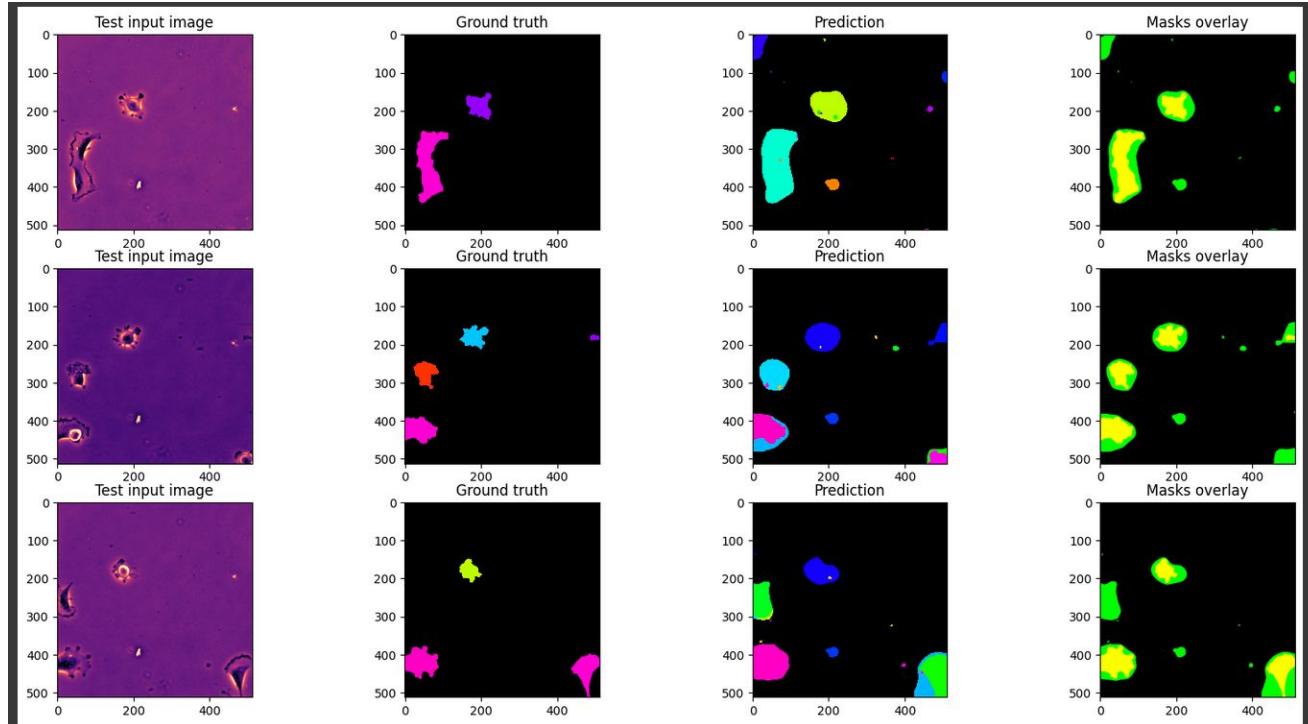


Needs to go up

Intersection over Union (IoU) vs epoch number



# Train and export a model with BiaPy - Validation



# Train and export a model with BiaPy - Export in BMZ format

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

## Option 1: Reuse previous BioImage Model Zoo model configuration

With this option, if you were using a model from BioImage 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.

```
trained_model_name: "my_model_biap  
trained_model_authors: "[student1]  
trained_model_authors_github_user: "[student1]  
trained_model_description: "Model trained during the EU-OS Autumn School  
trained_model_license: "CC-BY-4.0  
trained_model_references: ["Ronneberger et al. arXiv in 2015", "Franco-Barranco, Daniel, et al. ISBI in 2023"]  
trained_model_references_DOI: ["10.1007/978-3-319-24574-4_28", "10.1109/ISBI53787.2023.10230593"]  
trained_model_tags: ["tag-1", "tag-2"]  
trained_model_documentation: "/content/README.md
```

Show code

# Train and export a model with Biapy - Export in BMZ format



Play to download a zip file with your [Bioimage Model Zoo](#) exported model

Show code

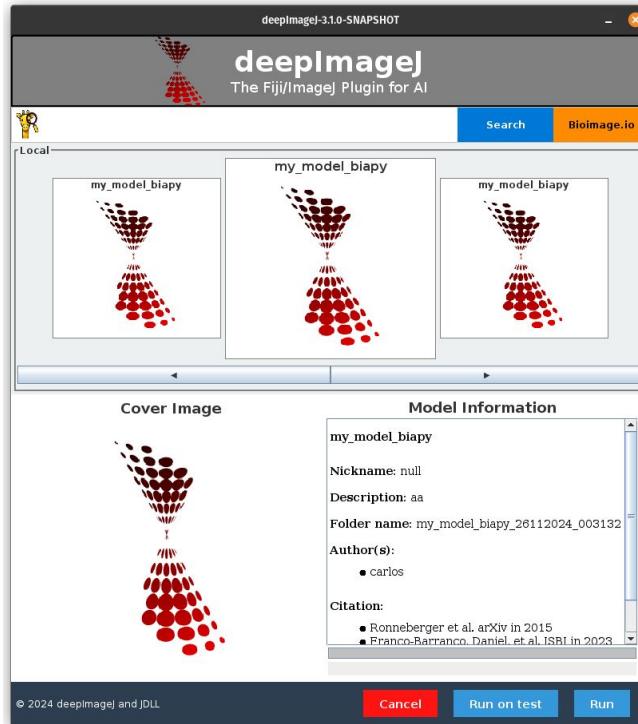
```
computing SHA256 of 6a45f6a50343cc346b6391c3856b0f5f-unet.py (result: 7f5b15948e8e2c91f78dcff34fbf30af517073e91ba487f3edb982b948d099b3): 100% [██████████] 20934/20934 [00:00<00:00, 8102201.71it/s]
2024-11-25 23:26:19.759 | Level 30 | bioimageio.spec.model.v0_5:_validate_documentation:2106 - documentation: No '# Validation' (sub)section found in /content/README.md.
2024-11-25 23:26:19.798 | Level 30 | bioimageio.spec.model.v0_5:_add_default_cover:2422 - covers: Failed to generate cover image(s): Failed to construct cover image from shape (1, 2, 512, 512)
/usr/local/lib/python3.10/dist-packages/pydantic/_internal/_serializers.py:42: UserWarning: Pydantic serializer warnings:
PydanticSerializationUnexpectedValue: Expected 'TimeOutputAxis' but got 'SpaceOutputAxis' with value 'SpaceOutputAxis(size=512,...', unit=None, scale=1.0)' - serialized value may not be as expected
PydanticSerializationUnexpectedValue: Expected 'TimeOutputAxisWithHalo' but got 'SpaceOutputAxis' with value 'SpaceOutputAxis(size=512,...', unit=None, scale=1.0)' - serialized value may not be as expected
v = handler(item, index)
/usr/local/lib/python3.10/dist-packages/pydantic/_internal/_serializers.py:42: UserWarning: Pydantic serializer warnings:
PydanticSerializationUnexpectedValue: Expected 'TimeOutputAxis' but got 'SpaceOutputAxis' with value 'SpaceOutputAxis(size=512,...', unit=None, scale=1.0)' - serialized value may not be as expected
PydanticSerializationUnexpectedValue: Expected 'TimeOutputAxisWithHalo' but got 'SpaceOutputAxis' with value 'SpaceOutputAxis(size=512,...', unit=None, scale=1.0)' - serialized value may not be as expected
PydanticSerializationUnexpectedValue: Expected 'TimeOutputAxis' but got 'SpaceOutputAxis' with value 'SpaceOutputAxis(size=512,...', unit=None, scale=1.0)' - serialized value may not be as expected
PydanticSerializationUnexpectedValue: Expected 'TimeOutputAxisWithHalo' but got 'SpaceOutputAxis' with value 'SpaceOutputAxis(size=512,...', unit=None, scale=1.0)' - serialized value may not be as expected
v = handler(item, index)
[23:26:19.80003] Created 'my_model.biapy'
computing SHA256 of test-input.npy (result: 083ade68924c93e5e8ab8ca916b534b18d2aa5dafde86433231349e8ef0c7eb):  0%|          | 0/1048704 [00:00<?, ?it/s]
computing SHA256 of test-output.npy (result: 89fe9e850a22147f23a3bf92f49facdf8377dc6c6ed8267903783978d9d41c0):  0%|          | 0/2097280 [00:00<?, ?it/s]
computing SHA256 of 6a45f6a50343cc346b6391c3856b0f5f-unet.py (result: 7f5b15948e8e2c91f78dcff34fbf30af517073e91ba487f3edb982b948d099b3):  0%|          | 0/20934 [00:00<?, ?it/s]
computing SHA256 of weights.pt (result: 3bd9c518c8473f1e35abb7624f82f3aa92f1015e66fb1f6a9d08444e1f2f5698):  0%|          | 0/115785997 [00:01<?, ?it/s]
[23:26:26.508838] Package path: /content/output/my_2d_instance_segmentation_2/results/my_2d_instance_segmentation_2_1/bmz_model/my_model.biapy.zip
[23:26:26.510697] FINISHED JOB my_2d_instance_segmentation_2_1 !!
```

# Run the model with deeplImageJ

Unzip the model in the models folder

Run it on your image

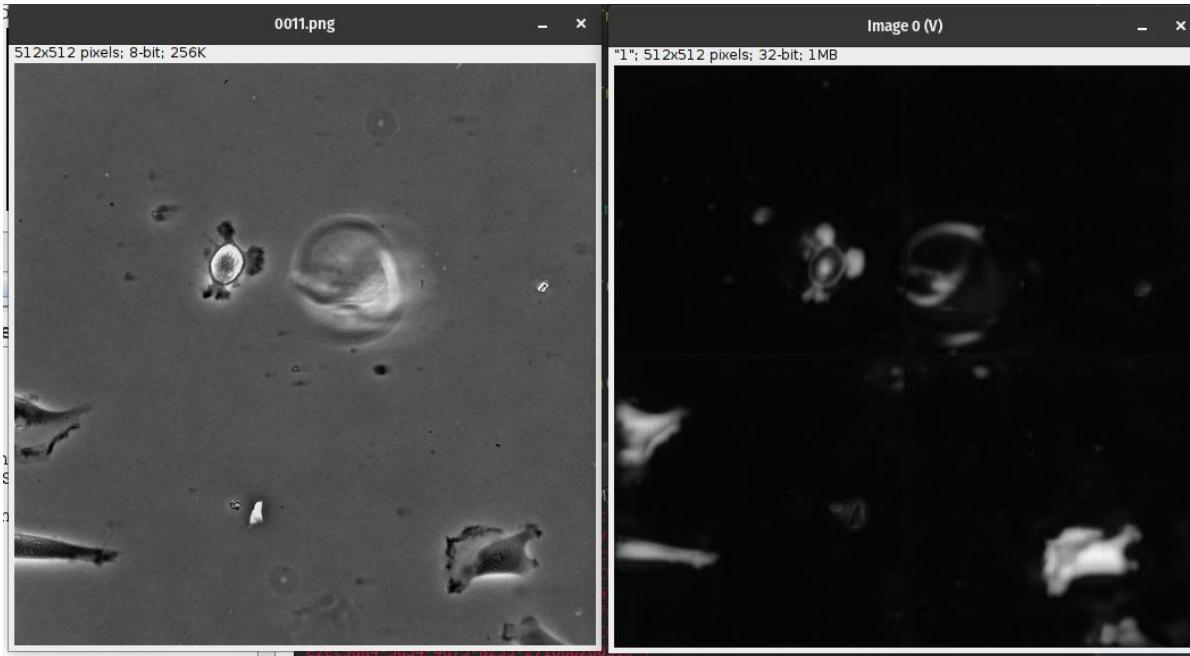
Create a macro to run it on all the images of a folder



# Final results

---

**Undertrained model with little data  
Results are still not good**



# THANK YOU!

---

AUTUMN TRAINING  
SCHOOL 2024



<https://eu-openscreen.eu>



<https://www.eu-openscreen.eu/impulse>

# THANK YOU!

---

AUTUMN TRAINING  
SCHOOL 2024



<https://eu-openscreen.eu>



<https://www.eu-openscreen.eu/impulse>

# THANK YOU!

---

AUTUMN TRAINING  
SCHOOL 2024



<https://eu-openscreen.eu>



<https://www.eu-openscreen.eu/impulse>

# THANK YOU!

---

AUTUMN TRAINING  
SCHOOL 2024



<https://eu-openscreen.eu>



<https://www.eu-openscreen.eu/impulse>







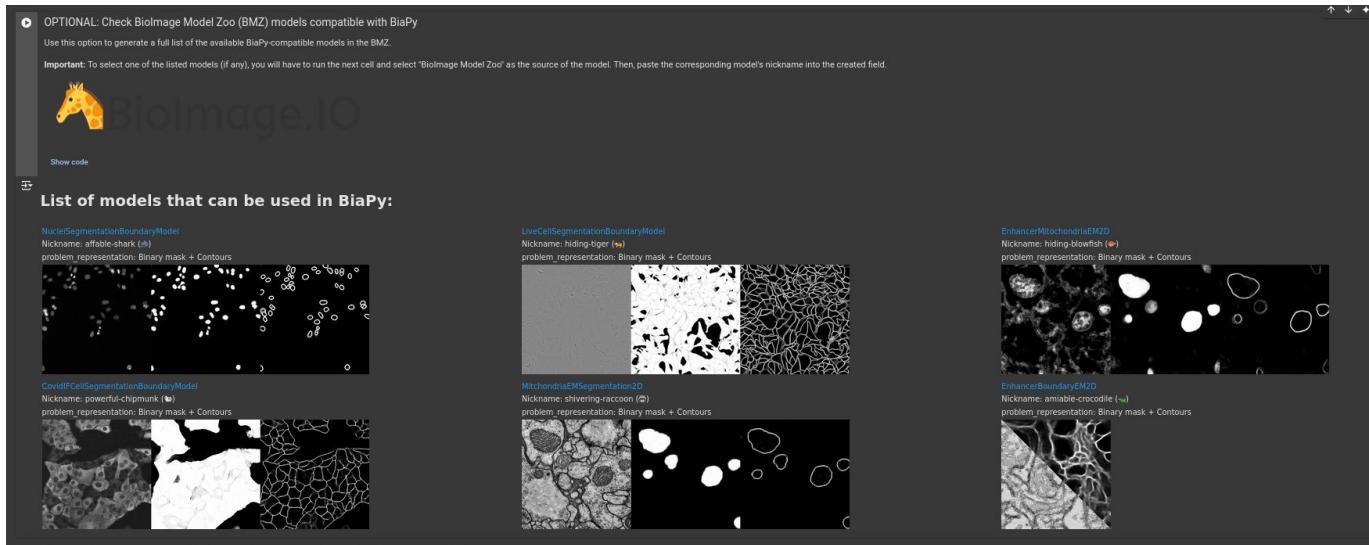




# Train and export a model with BiaPy - Fine-tuning

Select a model already pre-trained from the model zoo

Fine-tuning is faster than training if the pre-trained model is similar



# Train and export a model with BiaPy - Fine-tuning

---

Select powerful-chumpunk

Fine-tuning is faster than training if the pre-trained model is similar

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

**BiaPy:** to use the models implemented in BiaPy.

**Torchvision:** to use models from [Torchvision](#).

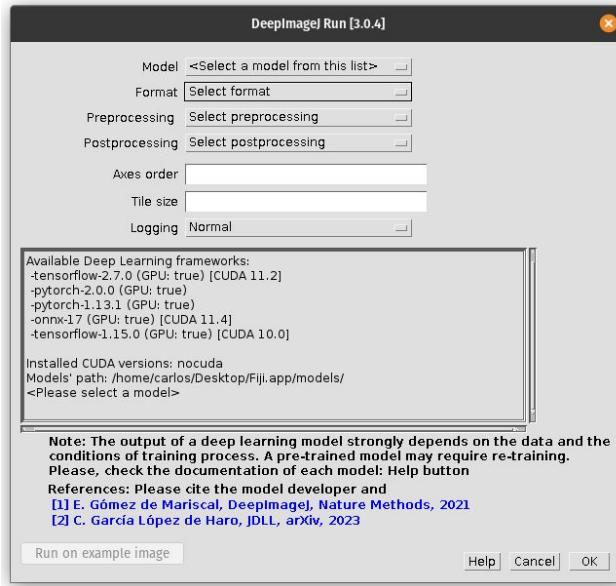
**Bioimage Model Zoo (BMZ):** to use models from the [BMZ repository](#). You can run the above cell to generate an

Show code

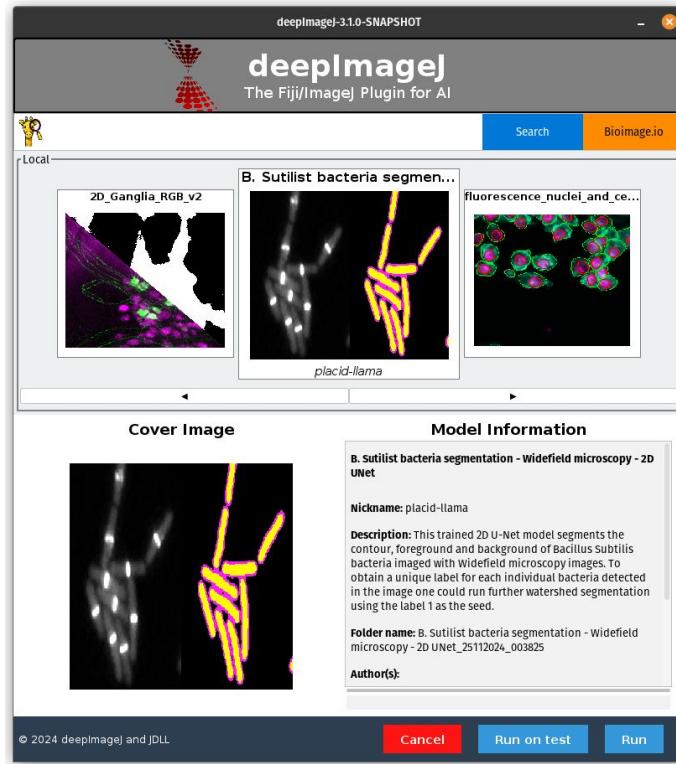
Source:  BiaPy  Torchvision  BioImage Model Zoo

ID:

# DeeplImageJ



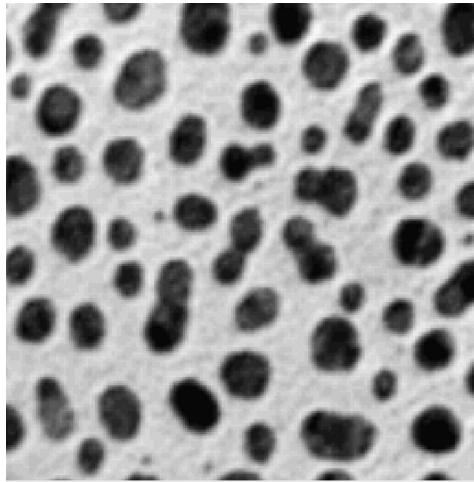
New things will be coming soon!



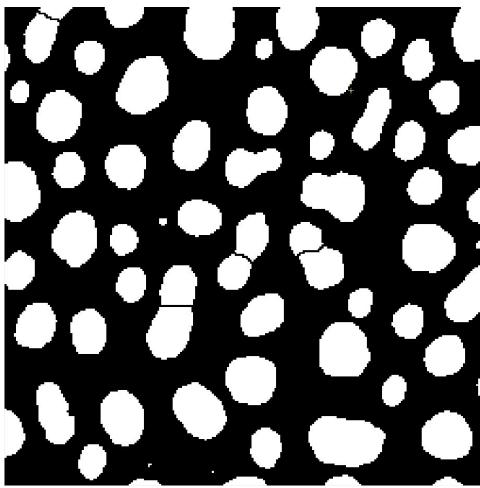
# Instance segmentation

---

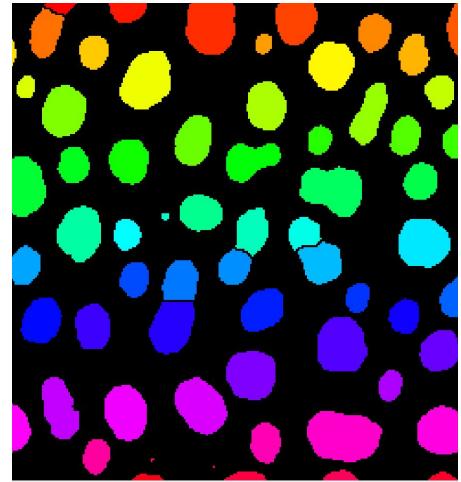
*Outline each object in an image at the pixel level*



Original image



Semantic segmentation  
(pixel classification)



Instance segmentation