



# deepImageJ

Bridging Deep Learning to ImageJ

[deepimagej.github.io/deepimagej/](https://deepimagej.github.io/deepimagej/)

E. Gómez-de-Mariscal<sup>1</sup>, C. García-López de Haro<sup>1</sup>, A. Muñoz-Barrutia<sup>1</sup>, D. Sage<sup>2</sup>

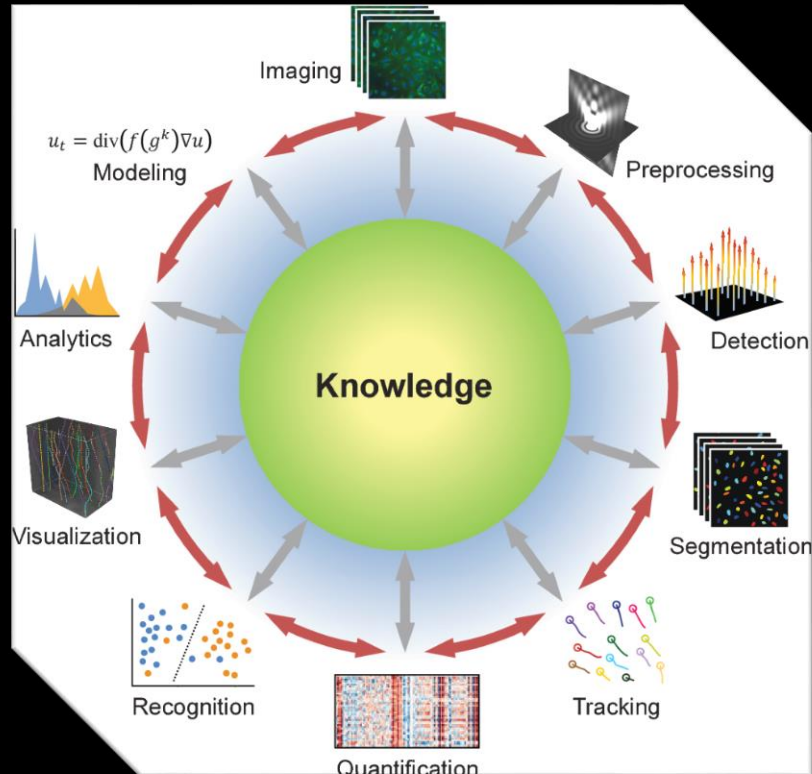
<sup>1</sup>Bioeng. and Aerospace Eng. Dept., Universidad Carlos III de Madrid (UC3M)  
Instituto de Investigación Sanitaria Gregorio Marañón, Spain

<sup>2</sup>Biomedical Imaging Group, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland



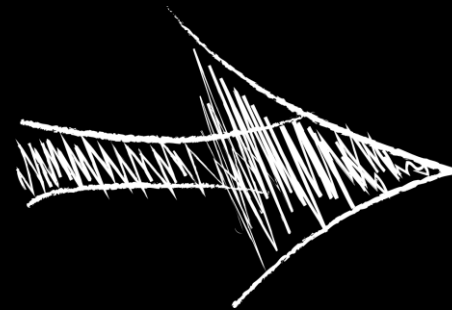
# CHALLENGE FOR DL SOLUTIONS:

Integration of image analysis pipelines



E. Meijering et al., Nature Biotechnology 2016

user friendly



+

Open source tools



ilastik



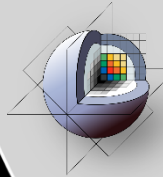
#ZeroCostDL4Mic



ImJoy



CSBDeep



3DSlicer



colab

**CellProfiler**  
cell image analysis software

# Run Deep Learning models in ImageJ



## ➔ Functional

- Integrate new models
- Process new data

## ➔ General

- Compatible with different CNN architectures




deepImageJ

## Easy-to-use!

- ImageJ's plugin-like: macro recordable
- unifying interface for TensorFlow - Pytorch models

## Ready out-of-the-box!

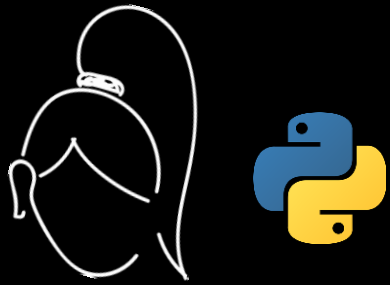
- One-click installation
- Runs on a laptop / CPU / GPU
- Proof of concept for a Model-Repository → 

## Make your models accessible!

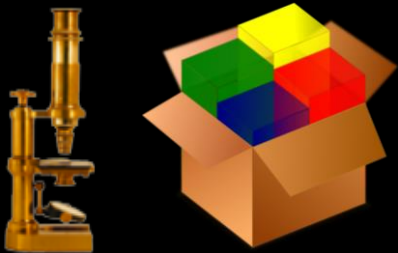
- Easy model sharing
- Ready to use models
- GUI to build the bundled model

New release of  
**deepImageJ**

# Developer meets the user



Model developer



Bundled model



Share the model

deepImageJ

Home About Download Bundled Models Contribute Tutorials

## DeepImageJ Bundled Models

Ready-to-use trained models. Most of them are also available at Bioimage.IO 🐮

- Unzip the zip file and store it into the folder **models** of ImageJ.
- If it is a first-time use, create the folder **models** inside ImageJ (`./ImageJ/models/`).
- **IMPORTANT NOTE:** We strongly recommend the user to review the results obtained after processing any image. Please, note that these models were trained using some specific data and their use cannot be generalized. If a user wants to process new data, the model has to be trained using images with exactly the same properties. Do not hesitate to **contact** us if you have any question.

### U-Net Pancreatic Segmentation

Binary segmentation - Phase contrast microscopy

DeepImageJ & Ignacio Arganda-Carreras, EHU.

[Open in Colab](#)

[U-Net Pancreatic Segmentation](#)

### FRU-Net sEV Segmentation

Instance segmentation - TEM

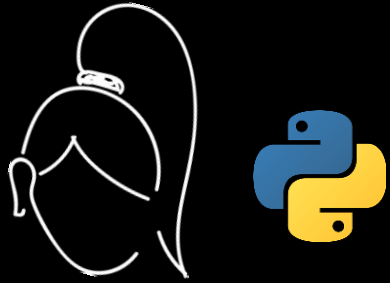
Estibaliz Gómez-de-Mariscal et al., [Deep-Learning-Based Segmentation of Small Extracellular Vesicles in Transmission Electron Microscopy Images](#), Scientific Reports, 2019.

[Code](#)

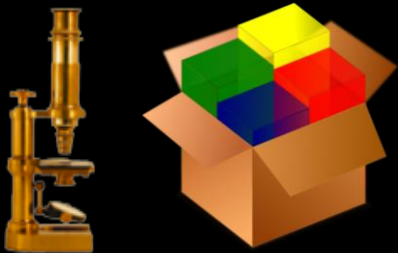
[FRU-Net sEV Segmentation](#)

[deepimagej.github.io/deepimagej/](https://deepimagej.github.io/deepimagej/)

# Developer meets the user



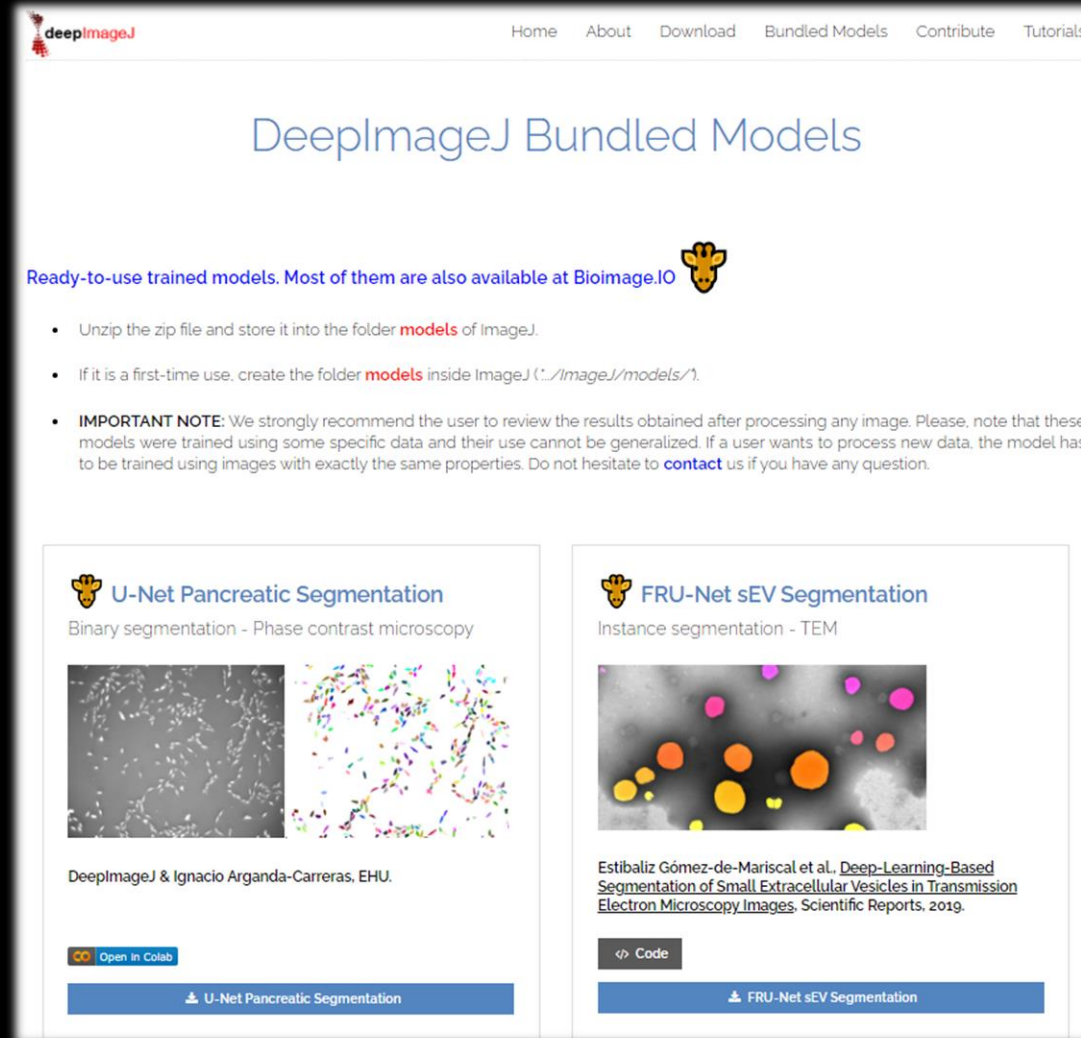
Model developer



Bundled model



Share the model



[deepimagej.github.io/deepimagej/](https://deepimagej.github.io/deepimagej/)



Model user



Install **deepImageJ**

update site:

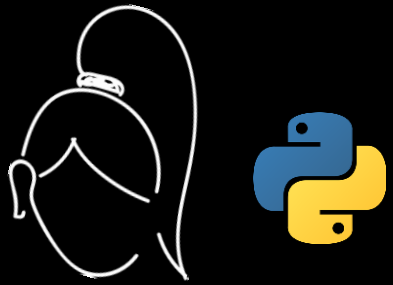
<https://sites.imagej.net/DeepImageJ/>



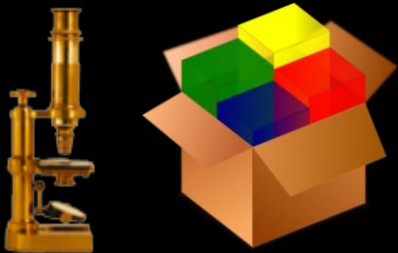
Download the  
model



# Developer meets the user



Model developer



Bundled model



Share the model



u-net_pancreatic_segmentation	7 items
variables	2 items
variables.data-00000-of-00001	11,6 MB
variables.index	9,6 kB
config.xml	1,9 kB
exampleImage.tiff	414,9 kB
postprocessing.txt	316 bytes
preprocessing.txt	156 bytes
resultImage.tiff	831,2 kB
saved_model.pb	1,2 MB



Model user



Install **deepImageJ**

update site:

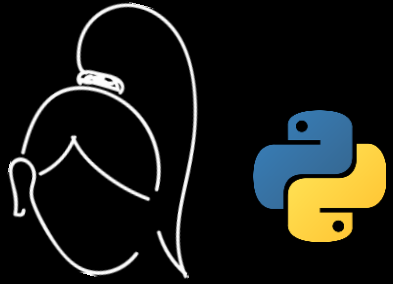
<https://sites.imagej.net/DeepImageJ/>



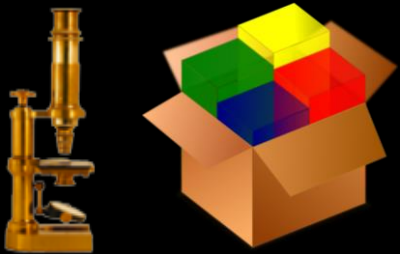
Download the  
model



# Developer meets the user



Model developer



Bundled model



Share the model

Install the model

1. Unzip

2. Move to ImageJ/models/

3. Process your image with **deepImageJ**



Model user



Install **deepImageJ**

update site:

<https://sites.imagej.net/DeepImageJ/>



Download the  
model<sub>8</sub>



# Use deepImageJ as a standard plugin

Preprocessing

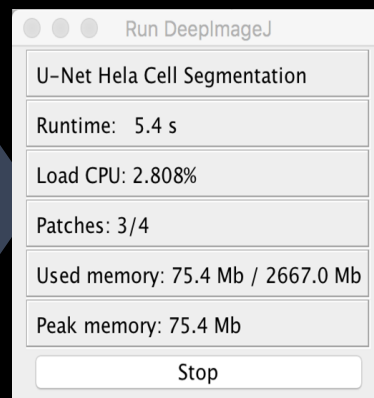
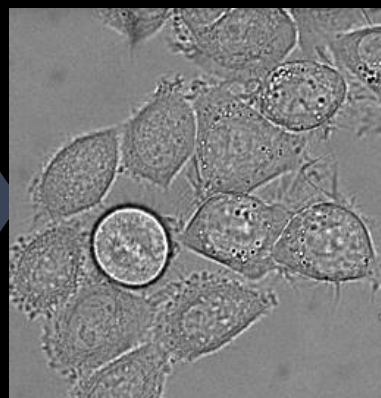
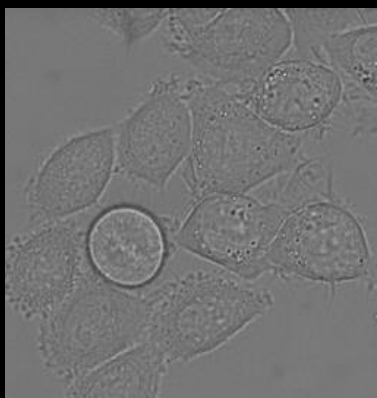


Prediction

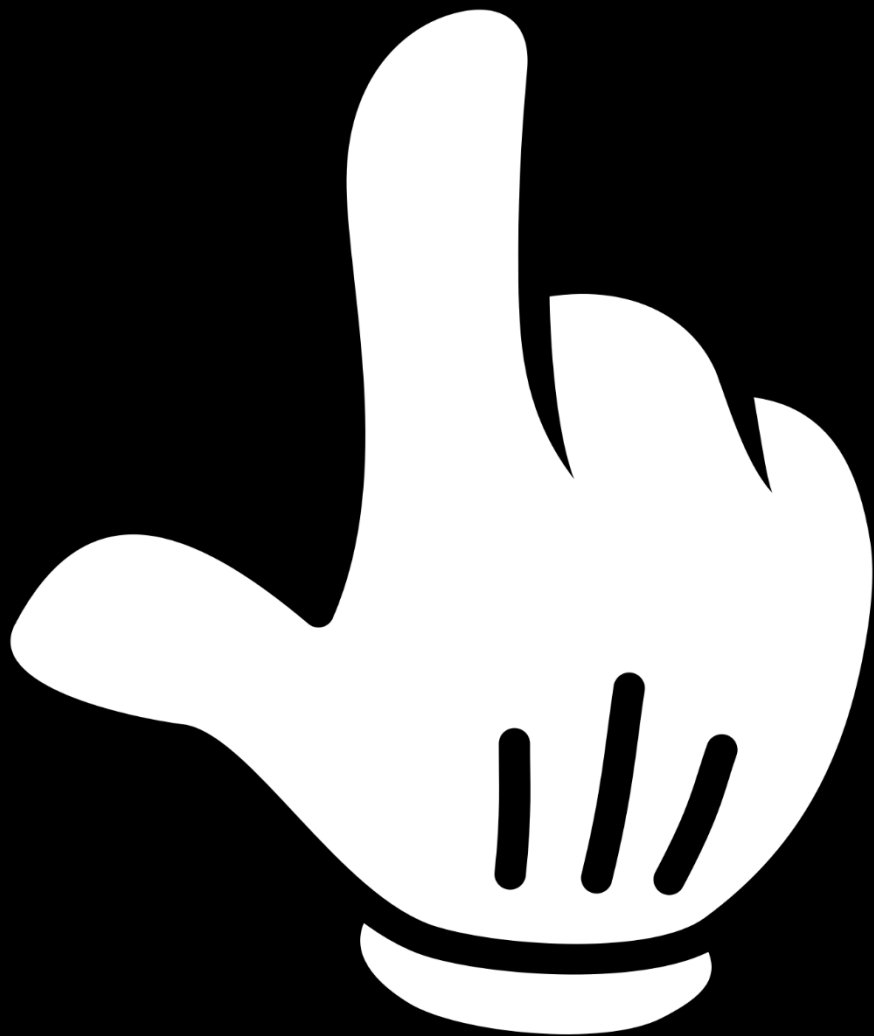
Java code



Postprocessing

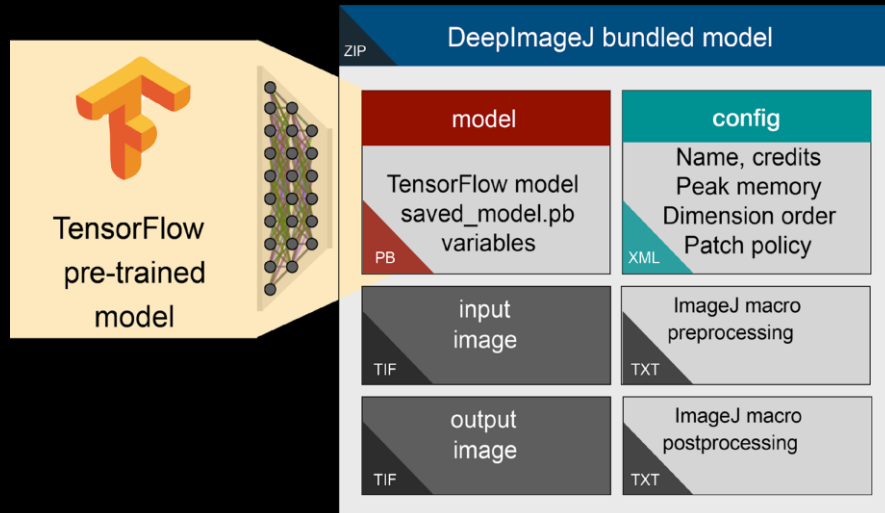


ZERO-CODE  
SOLUTION



Hey!  
Let's try it

# The magic is in the configuration file



## config.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Model>
  <ModelInformation>
    <Name>U-Net Hela Cell Segmentation</Name>
    <Author>João Soares Lopes</Author>
    <URL>n/a</URL>
    <Credit>Biomedical Imaging Group, School of Engineering, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland</Credit>
    <Version>n/a</Version>
    <Date>2019</Date>
    <Reference>n/a</Reference>
  </ModelInformation>
  <ModelTest>
    <InputSize>256x256</InputSize>
    <OutputSize>256x256</OutputSize>
    <MemoryPeak>268.0 Mb</MemoryPeak>
    <Runtime> 5.5 s</Runtime>
  </ModelTest>
  <ModelCharacteristics>
    <ModelTag>tf.saved_model.tag_constants.SERVING</ModelTag>
    <SignatureDefinition>tf.saved_model.signature_constants.DEFAULT_SERVING_SIGNATURE_DEF_KEY</SignatureDefinition>
    <InputTensorDimensions>,-1,256,256,1,</InputTensorDimensions>
    <NumberOfInputs>1</NumberOfInputs>
    <InputNames0>input</InputNames0>
    <InputOrganization0>NHWC</InputOrganization0>
    <NumberOfOutputs>1</NumberOfOutputs>
    <OutputNames0>output</OutputNames0>
    <OutputOrganization0>NHWC</OutputOrganization0>
    <Channels>1</Channels>
    <FixedPatch>true</FixedPatch>
    <MinimumSize>1</MinimumSize>
    <PatchSize>256</PatchSize>
    <FixedPadding>true</FixedPadding>
    <Padding>64</Padding>
    <PreprocessingFile>preprocessing.txt</PreprocessingFile>
    <PostprocessingFile>postprocessing.txt</PostprocessingFile>
    <slices>1</slices>
  </ModelCharacteristics>
</Model>
```

After you worked on it for a while...



The Big Umbrella by Amy June Bates

You see the combined effort of the whole community to democratize Deep Learning







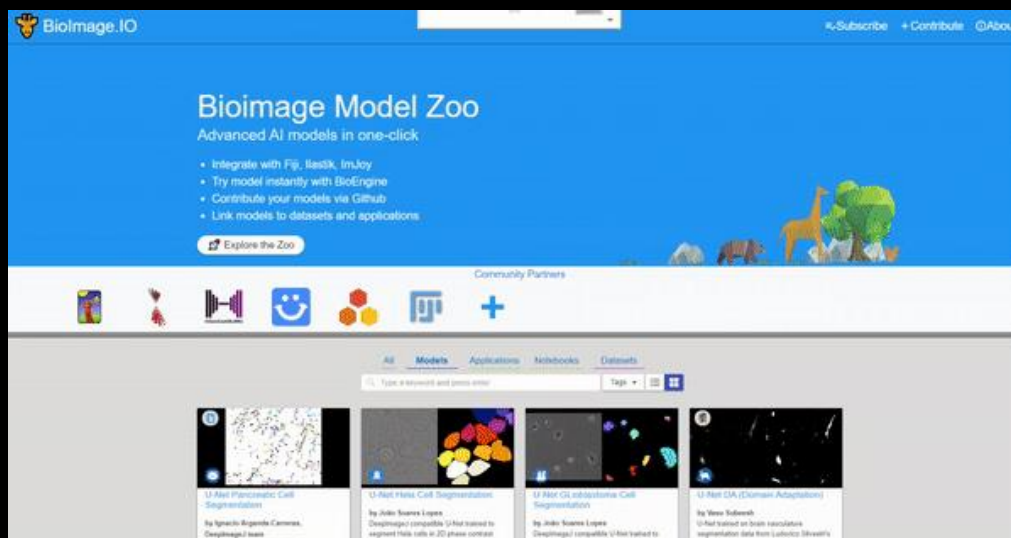
# Hackathon on BioImage Analysis, Dresden, 2019

"how to build bridges among the  
worlds of Java, C++, Python, and  
JavaScript."

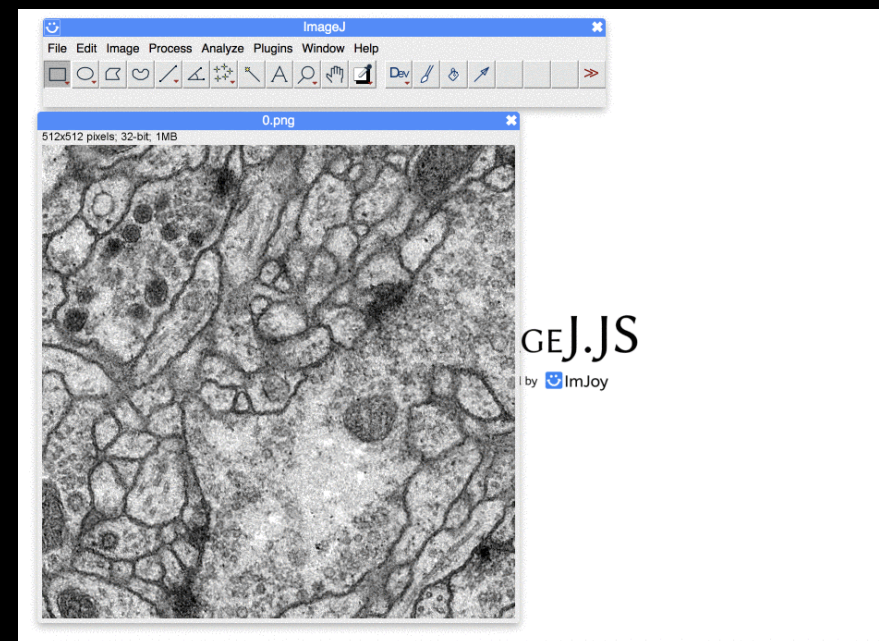


The magic is in the configuration file...

... which in the universal language is the YAML file



<https://bioimage.io/>





# Call for trained models for image processing



**deepImageJ**

Bridging Deep Learning to ImageJ

[deepimagej.github.io/deepimagej/](https://deepimagej.github.io/deepimagej/)



# Further tutorials and documentation

- E. Gómez-de-Mariscal, C. García-López-de-Haro, et al., bioRxiv, 2019. <https://doi.org/10.1101/799270>
- DeepImageJ web page: <https://deepimagej.github.io/deepimagej/>
- I. Arganda-Carreras, NEUBIAS Analyst School 2020:  
[https://github.com/miura/NEUBIAS\\_AnalystSchool2020/tree/master/Ignacio](https://github.com/miura/NEUBIAS_AnalystSchool2020/tree/master/Ignacio)
- I. Arganda-Carreras, Intro to Machine Learning-DeepLearning-DeepImageJ - NEUBIASAcademy@Home:  
<https://youtu.be/ovTbs08vnuo>
- E. Gómez de Mariscal, et al., Neubias Springer Book 2021 ([https://github.com/NEUBIAS/neubias-springer-book-2021/tree/master/Ch03\\_Building\\_a\\_Bioimage\\_Analysis\\_Workflow\\_using\\_Deep\\_Learning](https://github.com/NEUBIAS/neubias-springer-book-2021/tree/master/Ch03_Building_a_Bioimage_Analysis_Workflow_using_Deep_Learning))



Carlos  
García-López  
De-Haro



Daniel  
Sage



Arrate  
Muñoz-Barrutia



The Big Umbrella by Amy June Bates

- Donati, Laurène, EPFL
- Unser, Michael, EPFL
- Soares Lopes, João Luís, EPFL student
- Pengo, Thomas, University of Minnesota
- M. Gordaliza, Pedro, UC3M
- Arganda-Carreras, Ignacio, Univ. del País Vasco
- M. Douglass, Kyle, EPFL, Switzerland
- Yair Rivenson, UCLA
- Hongda, Wang, UCLA
- Schmidt, Deborah, MPI, Dresden,
- Jug, Florian MPI, Dresden
- Eglinger, Jan, FMI Basel
- Reuden, Curtis LOCI Lab
- Ouyang, Wei SciLifeLab, KTH
- Henriques, Ricardo, Instituto Gulbenkian de Ciência, Portugal
- F. Laïne, Romain, UCL (MRC-LMCB), England
- Tosí, Sébastien, IRB Barcelona, Spain