

Using Google Colabs:

Please refer to the documentation provided here for an overview of basic features for using Google Colabs and installing python packages:

https://colab.research.google.com/notebooks/basic_features_overview.ipynb

https://colab.research.google.com/notebooks/snippets/importing_libraries.ipynb

Using Ilastik:

Adapted from:

<https://www.ilastik.org/documentation/nn/nn#neural-network-classification-demo>

Neural Network Classification

This workflow allows one to execute (run inference for) pre-trained convolutional neural networks (CNNs) on the data loaded into ilastik.

Neural Network Classification Demo:

<https://www.youtube.com/watch?v=PfXCKNhSOLw>

Prerequisites: Running locally or remotely

Unlike other ilastik workflows, the Neural Network workflow provides the possibility to run the computations on a different machine (presumably, on a remote machine with GPUs). If you only want to run locally, on the CPUs or GPUs of the machine where ilastik is installed, you don't need anything else, just skip to the next section.

If you want to execute the neural network predictions on a different machine, you need to install a special back-end called TikTorch (for ilastik+PyTorch). You have to install it on the server where you have the GPUs or ask your IT or facility administrators to install it for you. Luckily, the installation process is not difficult, just follow the instructions in the [github repo](#). Concerning permissions, you need to be able to ssh to the machine where TikTorch will run.

Neural Network Classification (local) - step by step

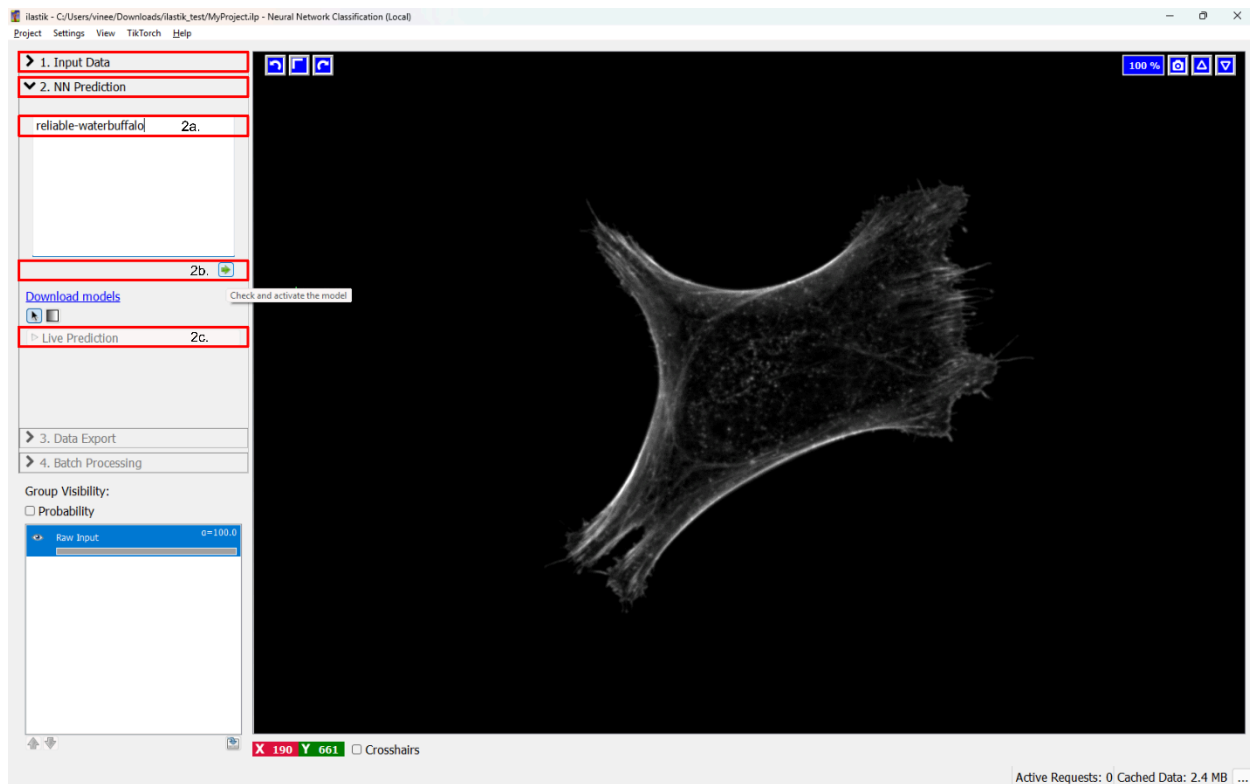
To run the local workflow, start up ilastik as you usually would.

0. Create Your Project

After starting up ilastik, click on *Neural Network Classification (local)* and save the project file.

1. Load Your Data

Load your raw data into ilastik as usual in the [Data Selection applet](#):



2. Load a Pre-trained Neural Network

Proceed to the next applet (NN Prediction). Pre-trained models can be found in the [ilastik Model Zoo at BioImage.IO](#).

The Bioimage Model Zoo

Together with community partners, we have already put some in the [ilastik Model Zoo at BioImage.IO](#). We will keep updating it with our networks; the format is also open, so please get in touch with us if you want to contribute models. The first step of using the Neural Network workflow is to find a network you would like to use in the Zoo. The best way is to put keywords into the search and visually check if any of the networks that come up do the task you need; we always show before/after images in the model

preview cards.

BioImage.IO + Upload Documentation About

BioImage Model Zoo

Advanced AI models in one-click

Integrated with Fiji, ilastik, ImJoy and more
Try model instantly with BioEngine
Contribute your models via Github
Link models to datasets and applications

[Explore the Zoo](#)

Community Partners

All **models** applications datasets

Search: Type a keyword and press enter Tags & Filters

Neuron Segmentation in EM ...
Neuron segmentation in EM, trained on the CREMI challenge data.
tags: unet, neurons, instance-segmentation, electron-microscopy
downloads: 15773 license: CC-BY-4.0

PlatynereisEMnucleiSegmen...
{organelle} segmentation in EM of platynereis.
tags: unet, nuclei, instance-segmentation, electron-microscopy
downloads: 13298 license: CC-BY-4.0

StarDist H&E Nuclei Segmen...
StarDist - Object Detection with Star-convex Shapes
tags: whole-slide-imaging, 2d, nuclei, tensorflow
downloads: 10814 license: BSD-3-Clause

Arabidopsis Leaf Segmentat...
Perform leaf segmentation in light microscopy images of Arabidop...
tags: zerocostdl4mic, deepimagej, segmentation, unet
downloads: 10101 license: MIT

StarDist Fluorescence Nucle...
StarDist - Object Detection with Star-convex Shapes
tags: fluorescence-light-microscopy, whole-slide-imaging, other, 2d
downloads: 9899 license: BSD-3-Clause

3D UNet Arabidopsis Apical ...
3D Unet trained on confocal images of Arabidopsis thaliana apica...
tags: plantseg, unet, 3d, semantic-segmentation
downloads: 8773 license: MIT

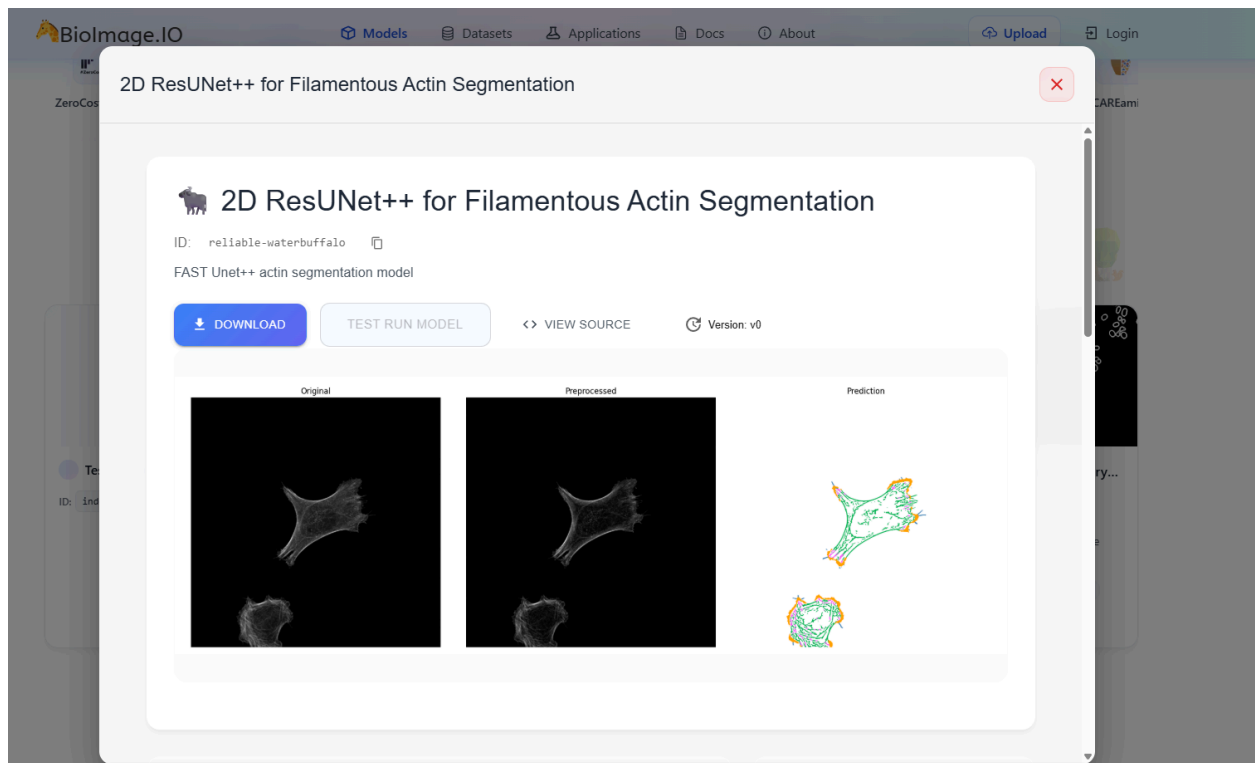
PlatynereisEMcellsSegment...
Cell segmentation in EM of platynereis.
tags: unet, cells, instance-segmentation, electron-microscopy
downloads: 8090 license: CC-BY-4.0

MitochondriaEMSegmentati...
Mitochondria segmentation for electron microscopy.
tags: 3d, electron-microscopy, mitochondria, instance-segmentation
downloads: 7892 license: CC-BY-4.0

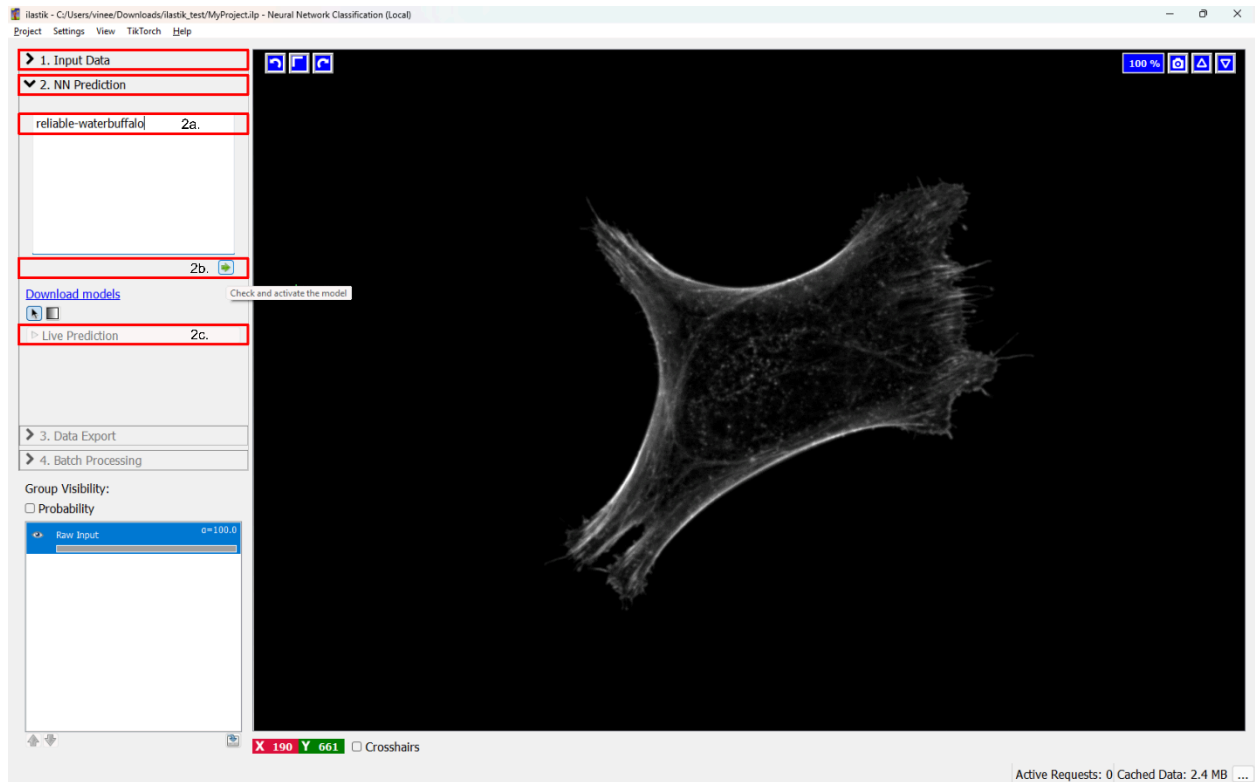
If you found a model you like, there are multiple ways to get it into ilastik.

The easiest way to load a model from the model zoo in ilastik is to click on the model title (1) to open the detail view of the model card and copy either the doi or the

nickname (2) of the model.



Then, in ilastik, paste the doi or nickname to the model text field (1) and press the arrow-button (2). This will initiate the download and initialization of the network.

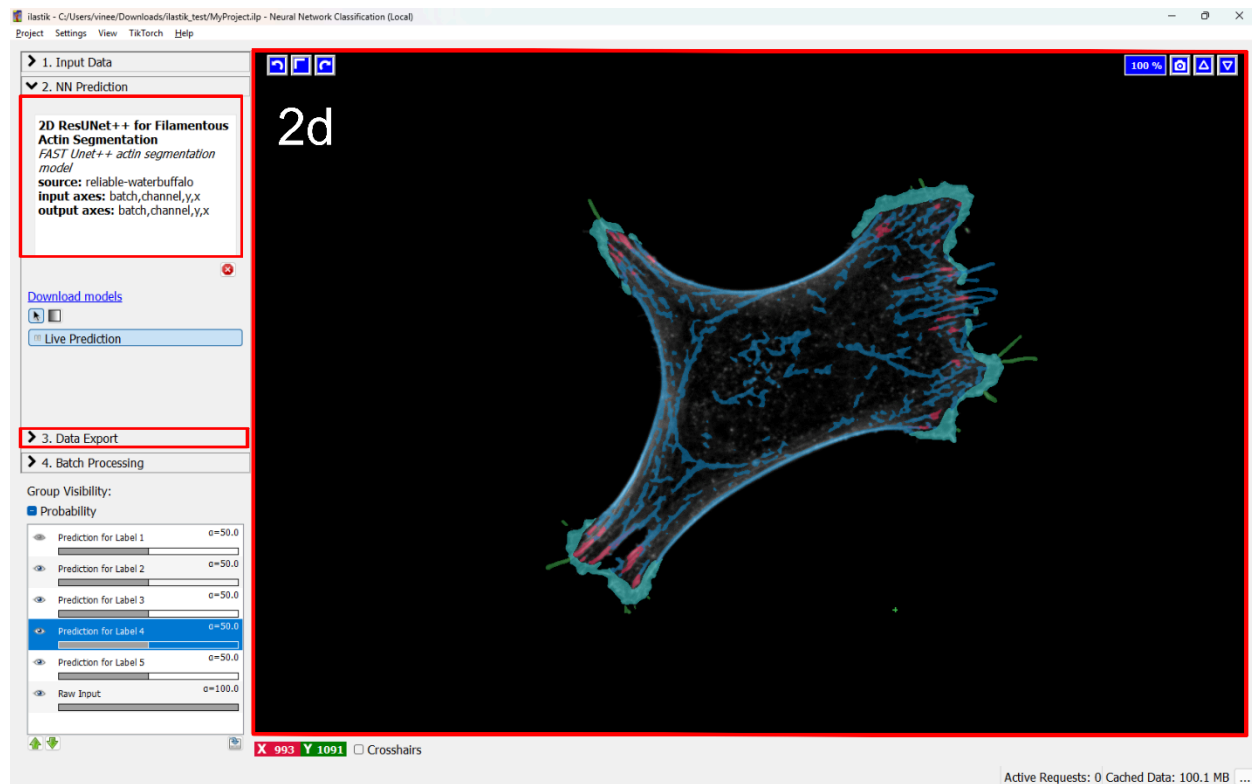


Alternatively, you can download a zip archive of the model by clicking on “Download (format)”. The server will then package the model with the correct weight format for ilastik consumption.

Once the file is downloaded, you can either drag and drop the model zip file from your file explorer window into the text field or click the arrow button while the text field is empty to bring up a load file dialog.

3. Run the Network

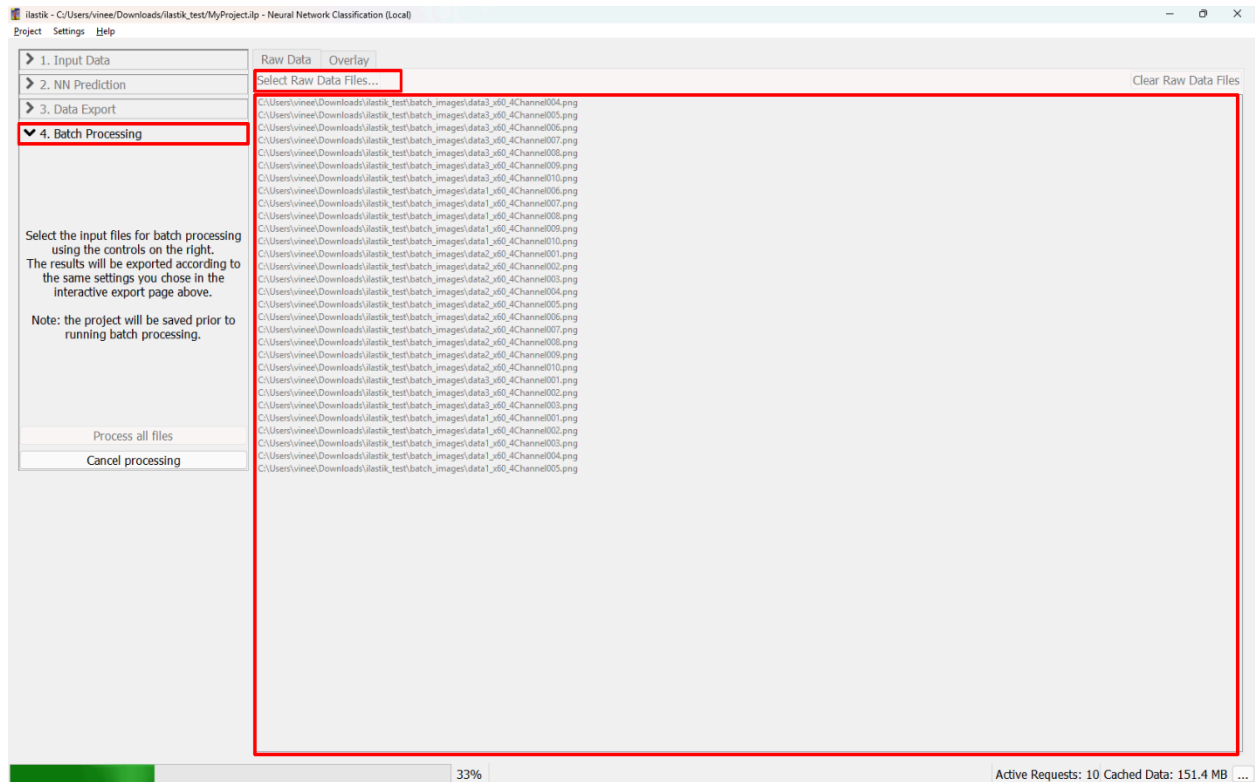
Only one thing left to do: press the “Live Predict” button to make the network predict:



If you prefer a different color for the display of the predictions, you can right-click on the corresponding layer [in the layerstack](#) and select *Change color*.

4. Export the Result

If you like the results, proceed to export them in the [Data Export applet](#). If you don't, try out another model. You can do so by first pressing the red cross below the text box to unload the model and then the red cross inside the text box to clear your previously selected model.



Neural Network Classification (remote)

When running remotely, the heavy lifting of neural network computations will happen on a remote machine, while you are loading the data and selecting the network on your local machine.

On the remote machine

1. Make sure the tiktorch server and neural network framework you want to run are installed via mamba. Installation instructions can be found [in the tiktorch github repository](#).
2. Activate the tiktorch environment and start the server giving the --addr 0.0.0.0 flag to make it accessible.
3. Take note of the ip/hostname of the remote machine and the port (watch out for the Starting server on 0.0.0.0:<port> output).

On your local machine/laptop

The steps are similar to the ones of the local workflow, except:

1. Create Your Project with the *Neural Network Classification (remote)* workflow.
2. Connect to the server. In the *Server Configuration* applet, enter the address in the form <ip/hostname>:<port>, e.g., myserver.mydomain.edu:5567 and give it a descriptive name. Click on *Get Devices* and select the device you want to run on (often, you will have the choice between *cpu* or *cuda* devices, if the server has a gpu). Click on *save* to finish the configuration.
3. From here on, you can proceed to load the data and go through the rest of the workflow like in the local variant.

Using Supervisely:

Please refer to the official documentation on using Supervisely provided here <https://docs.supervisely.com/>