Seeing is believing, Quantifying is convincing

Transferable image-based computational software for content quantification

AGUZZI's Lab
PhD, Caredio Davide

Date: 27-10-2020

WHY A COMPUTATIONAL APPROACH

1. Computational quantification aims to reduce subjective interpretation

2. Need to process ever-larger sets of images at high-throughput (thousands of images can be processed in parallel)

3. Computations makes the results reproducible

4. Algorithm can detect pixel variations that the human eye cannot see

COMPUTATIONAL SOFTWARE FOR CONTENT QUANTIFICATION

High demand

- As analysis tasks became more common, several tools were created to improve accessibility through a graphical front-end.

Time-consuming procedure

- Counting thousands of cells or (contents) with a certain marker, drawing the outlines of cells to quantify their shape and assess the strength of a reporter

Project-specific software

- They don’t allow training on new data, thus restricting their application domain to a small range of datasets

Trainable Software Package need

- Generic software package that can adapt itself autonomously to the task from appropriate data

COMPUTATIONAL SOFTWARE FOR CONTENT QUANTIFICATION

High demand

- As analysis tasks became more common, several tools were created to improve accessibility through a graphical front-end.

Time-consuming procedure

- Counting thousands of cells or (contents) with a certain marker, drawing the outlines of cells to quantify their shape and assess the strength of a reporter

Project-specific software

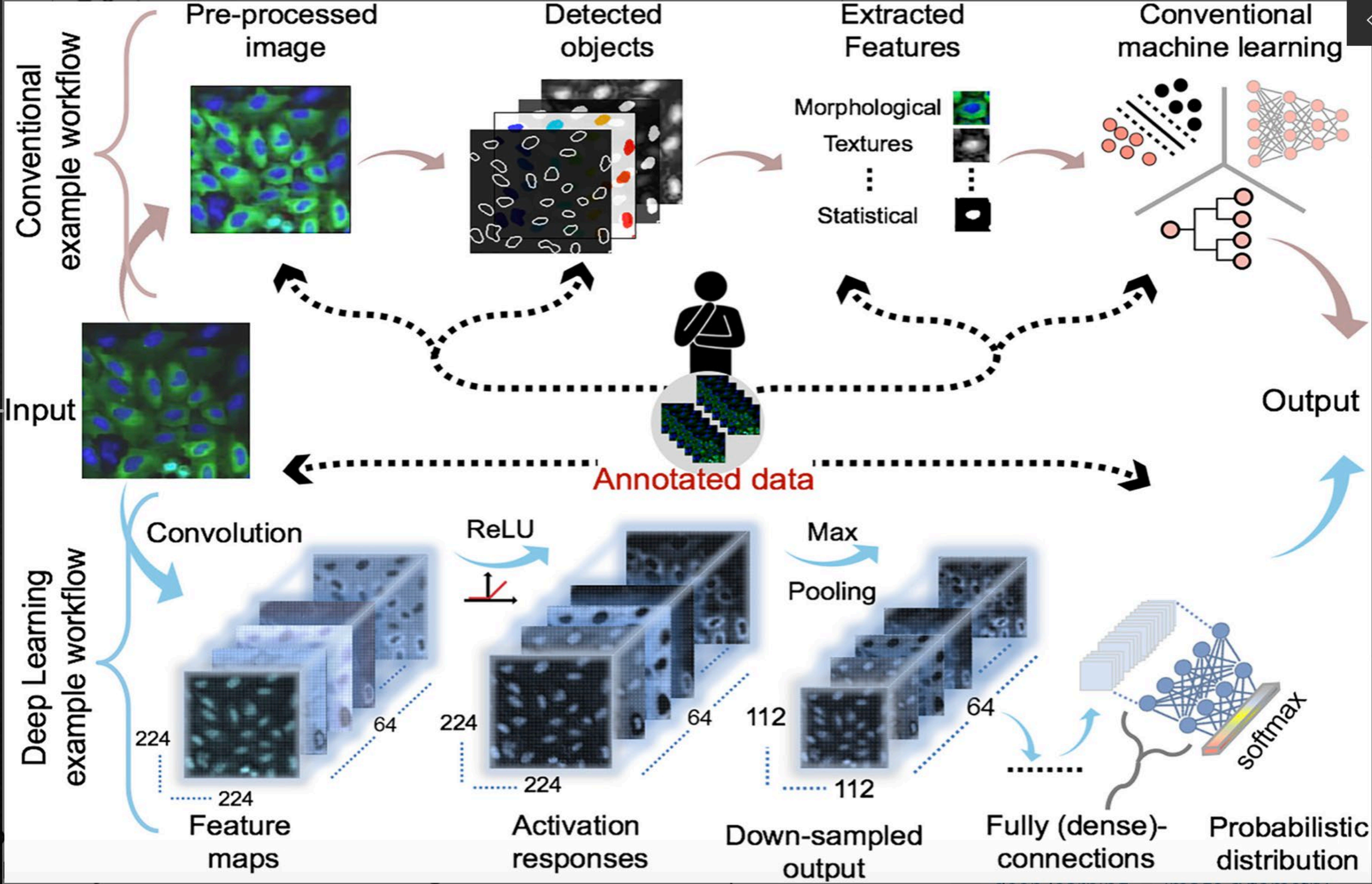
- They don’t allow training on new data, thus restricting their application domain to a small range of datasets

Trainable Software Package need

- Generic software package that can adapt itself autonomously to the task from appropriate data

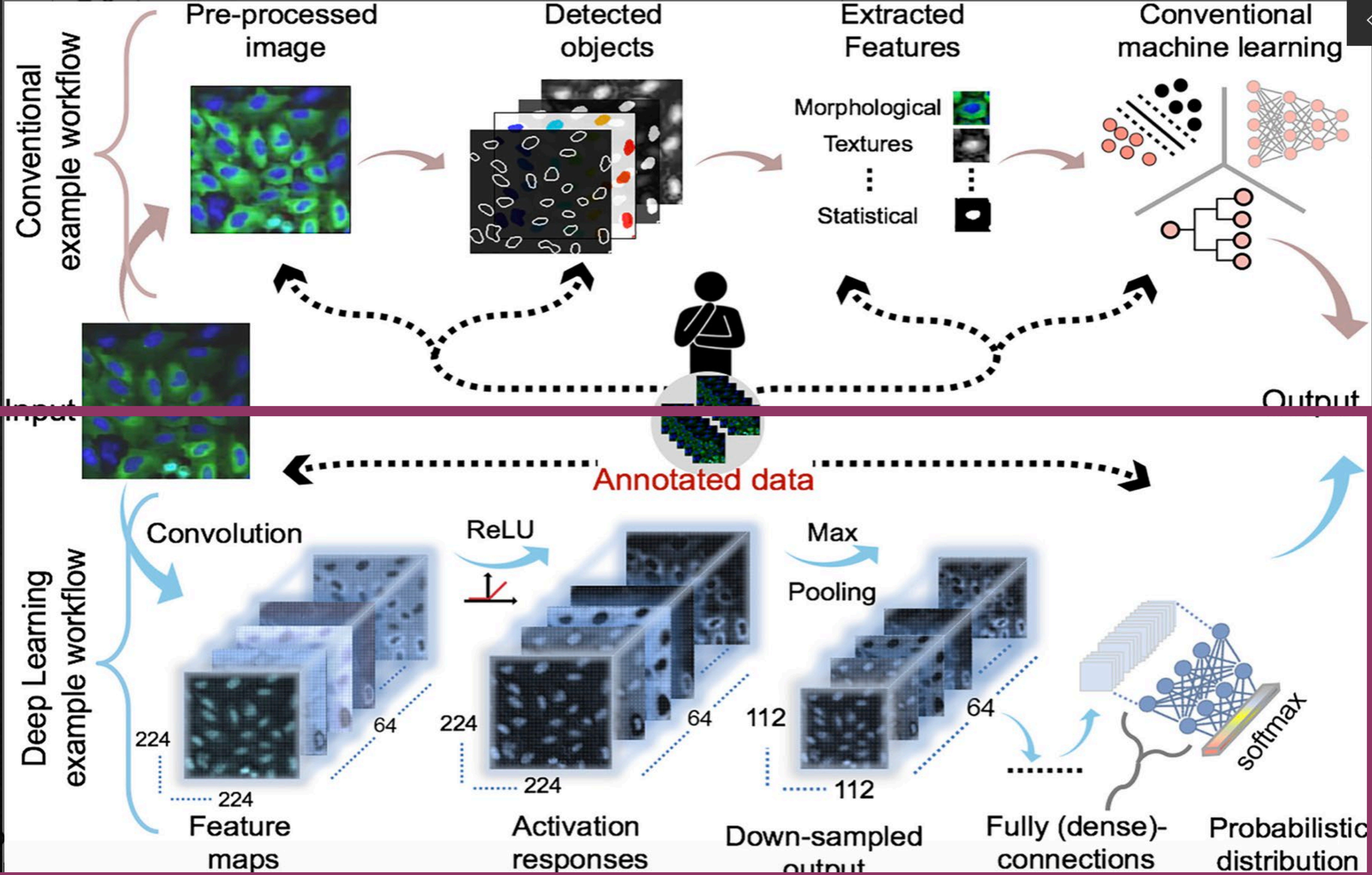
NOT TRAINABLE

TRAINABLE



NOT TRAINABLE

TRAINABLE



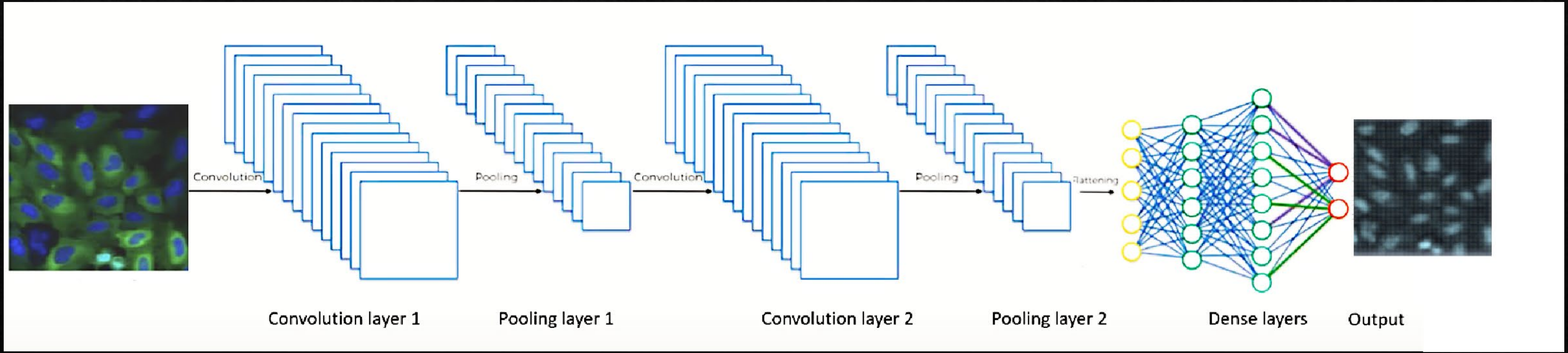
INTRODUCTION – an easy DCNN's structure description

1. From a biological perspective, a CNN approximately emulates the primate brain's visual system.

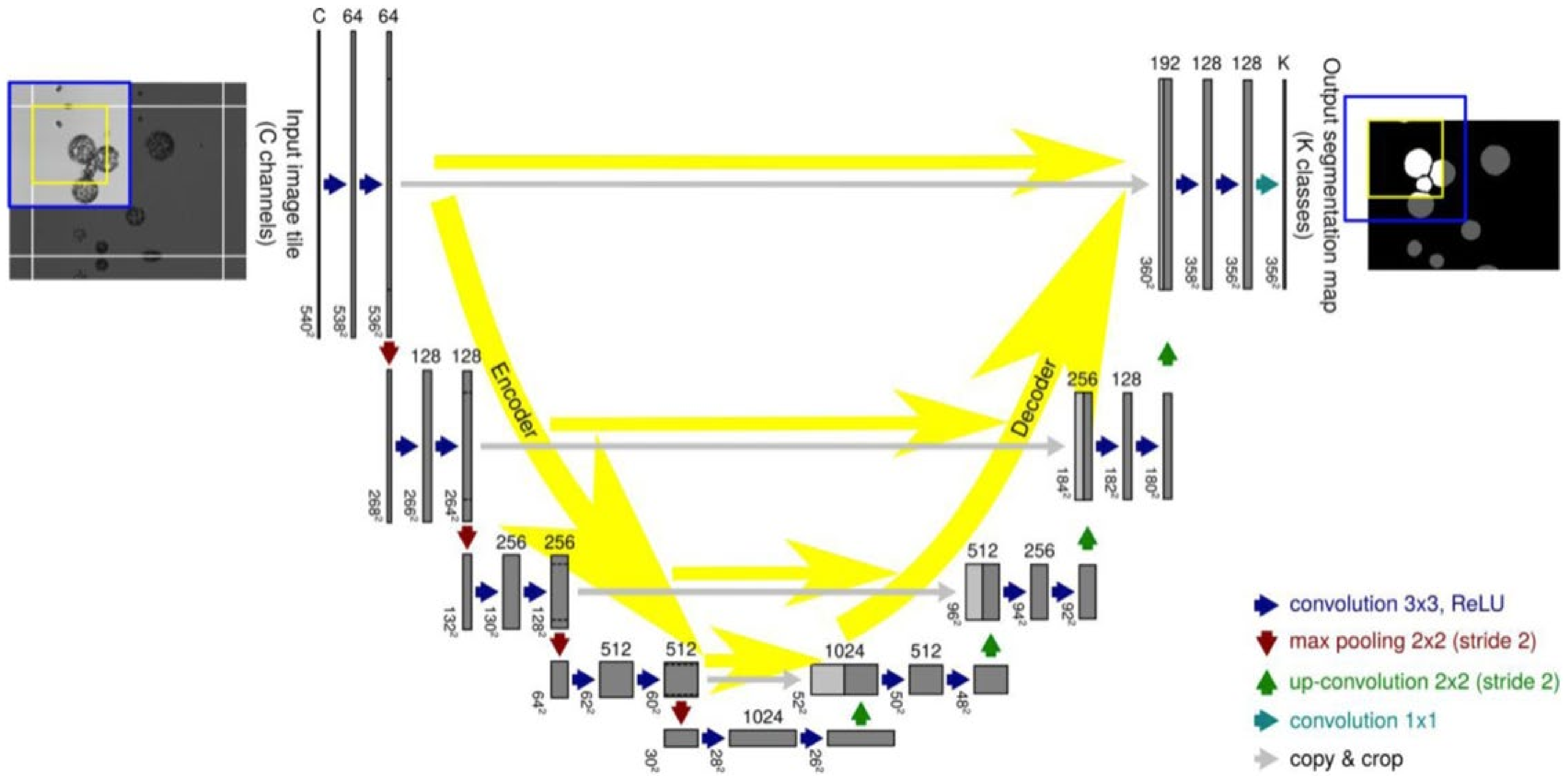
2. Manually labeled image (annotations) generates weighted filters which will determine the feature-extractions

3. Images must be converted first into numbers before they are accessible for statistical analysis.

4. It employs a combination of convolutional and pooling layers before the dense layers to progressively encode richer representations in an image

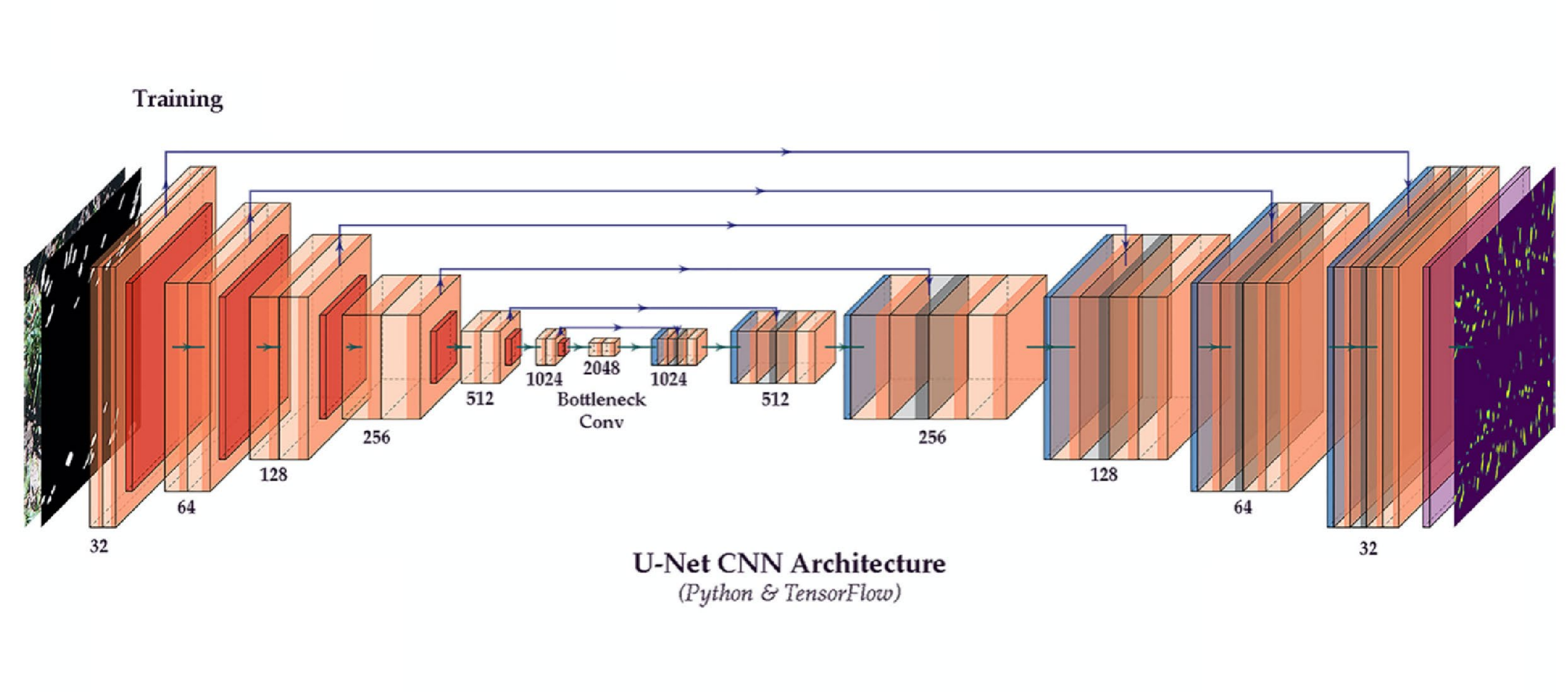


INTRODUCTION – U-Net DCNN structure



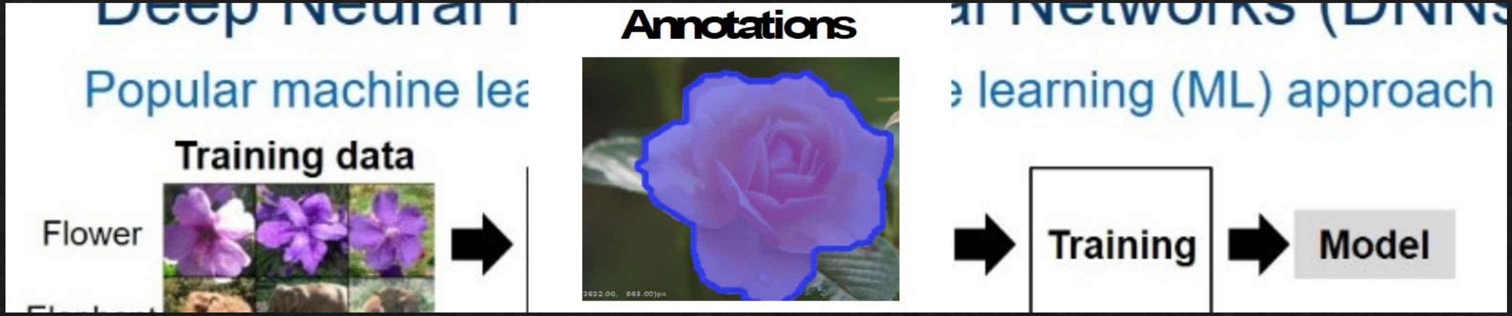
(left) Input: An image tile with 540×540 pixels and C channels (blue box). **(right)** Output: The K-class soft-max segmentation with 356×356 pixels (yellow box). Blocks show the computed feature hierarchy. Numbers atop each network block: number of feature channels; numbers left to each block: spatial feature map shape in pixels. Yellow arrows: Data flow

INTRODUCTION – U-Net DCNN structure



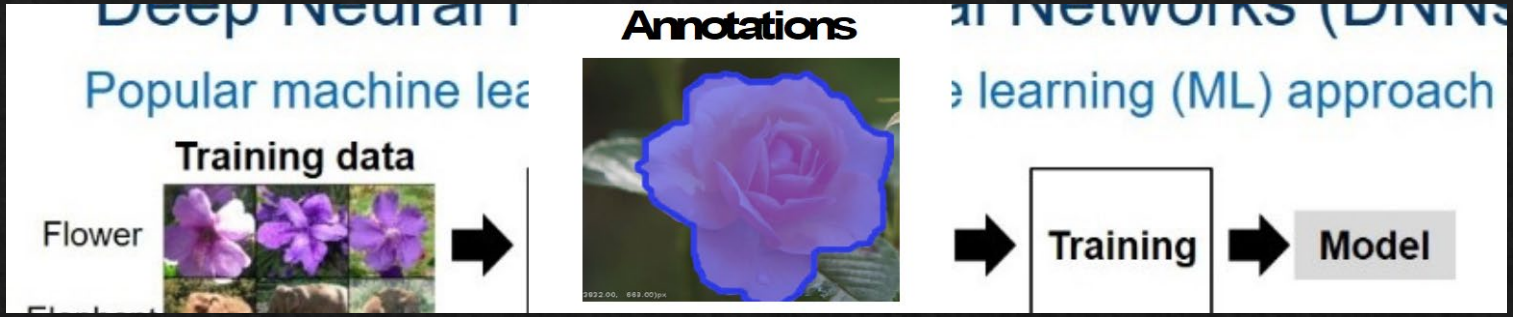
GENERIC WORKFLOW & TERMINOLOGY

1. TRAINING A MODEL/SOFTWARE

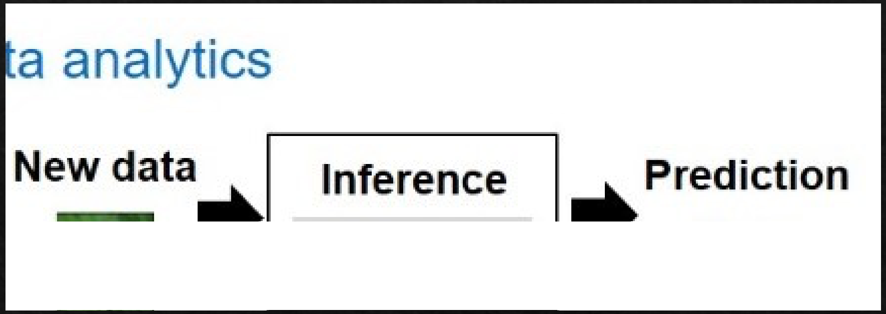


GENERIC WORKFLOW & TERMINOLOGY

1. TRAINING A MODEL/SOFTWARE

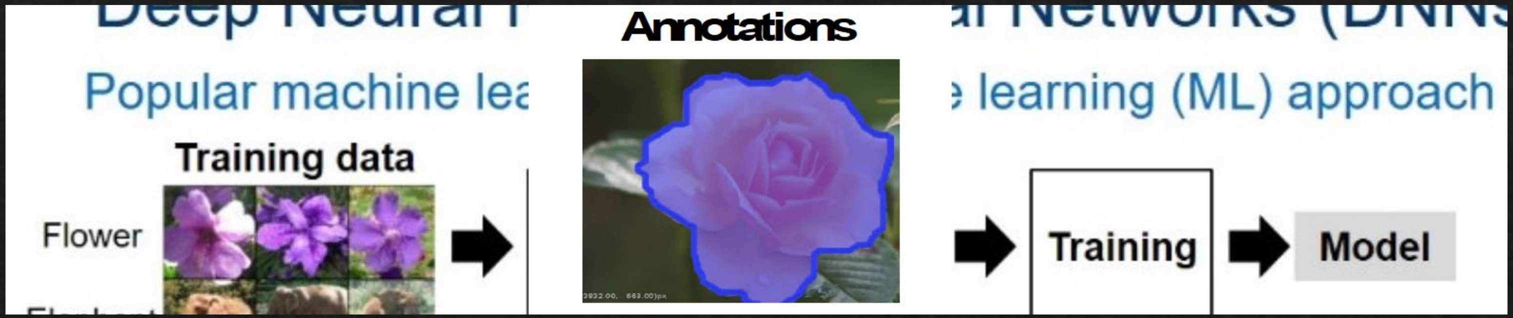


2. PREDICTION OUTCOME

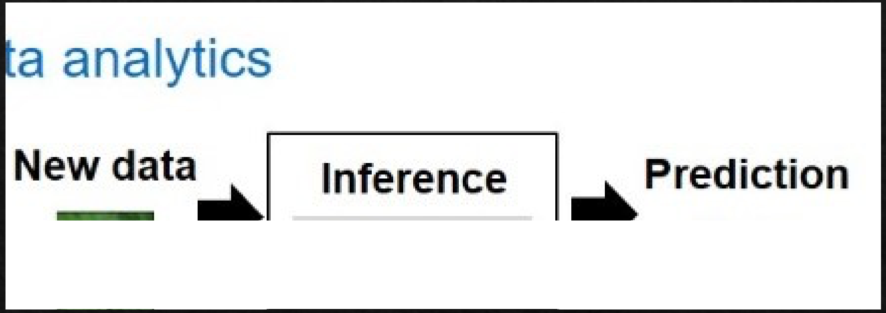


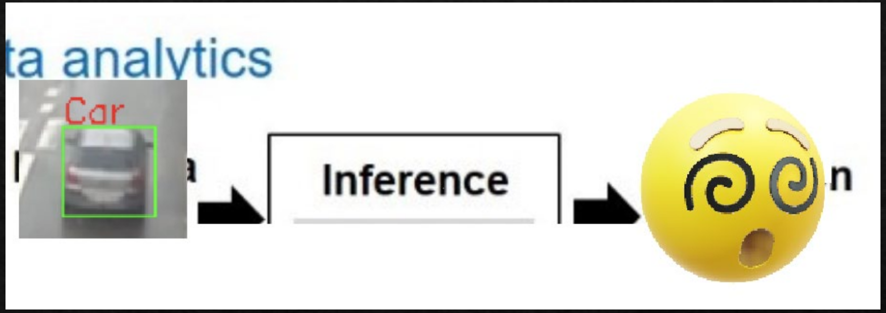
GENERIC WORKFLOW & TERMINOLOGY

1. TRAINING A MODEL/SOFTWARE



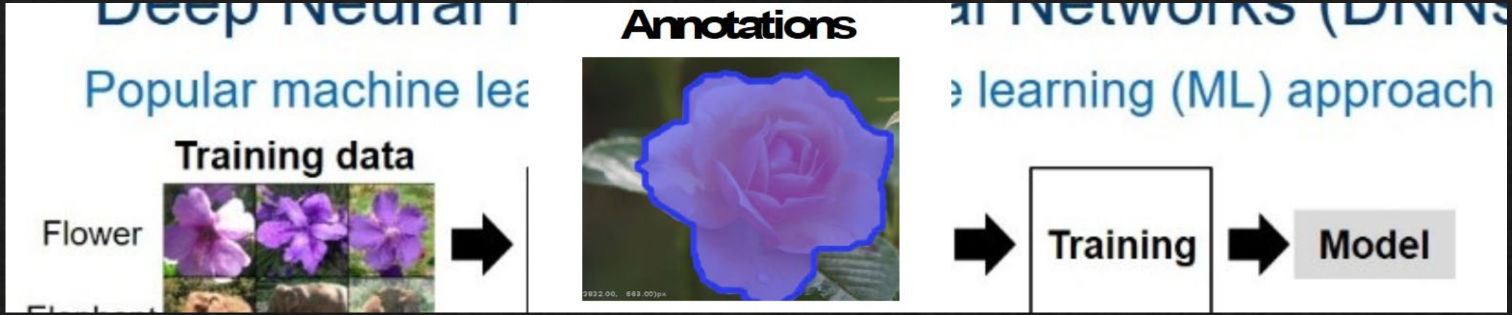
2. PREDICTION OUTCOME





GENERIC WORKFLOW & TERMINOLOGY

1. TRAINING A MODEL/SOFTWARE



Deep Neural I

Annotations

al Networks (DNNs

Popular machine lea

e learning (ML) approach

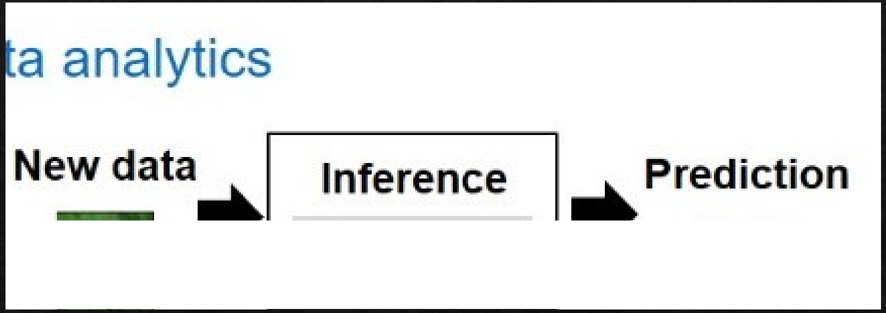
Training data

Flower

Training

Model

2. PREDICTION OUTCOME

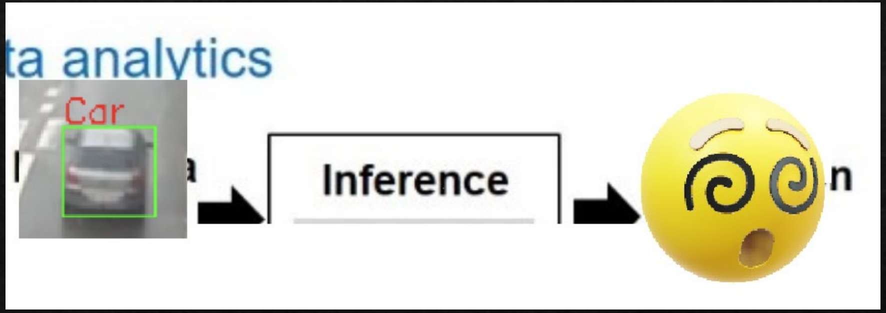


ta analytics

New data

Inference

Prediction

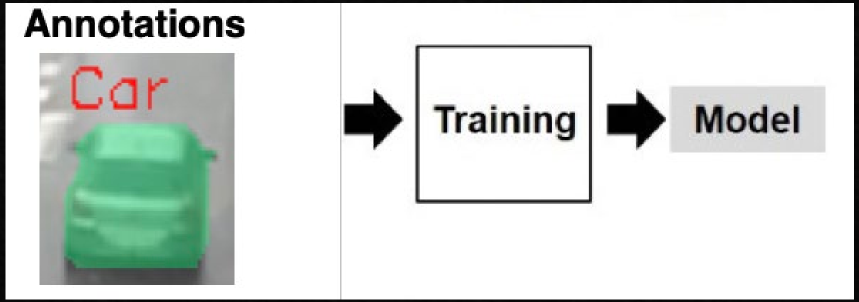


ta analytics

Car

Inference

3. TRANSFER LEARNING

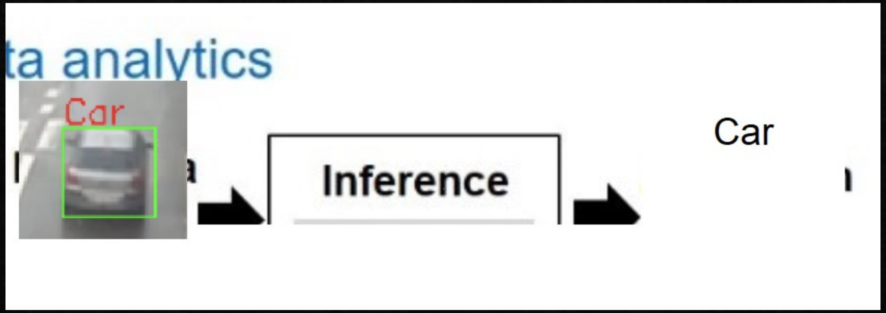


Annotations

Car

Training

Model



ta analytics

Car

Inference

Car

ADVANTAGES

1. It learns discriminative representations (or features) from image examples
2. Detection performance increase with the n° of iterations
3. A pre-trained model from one domain can be adapted through “fine-tuning”

DISADVANTAGES

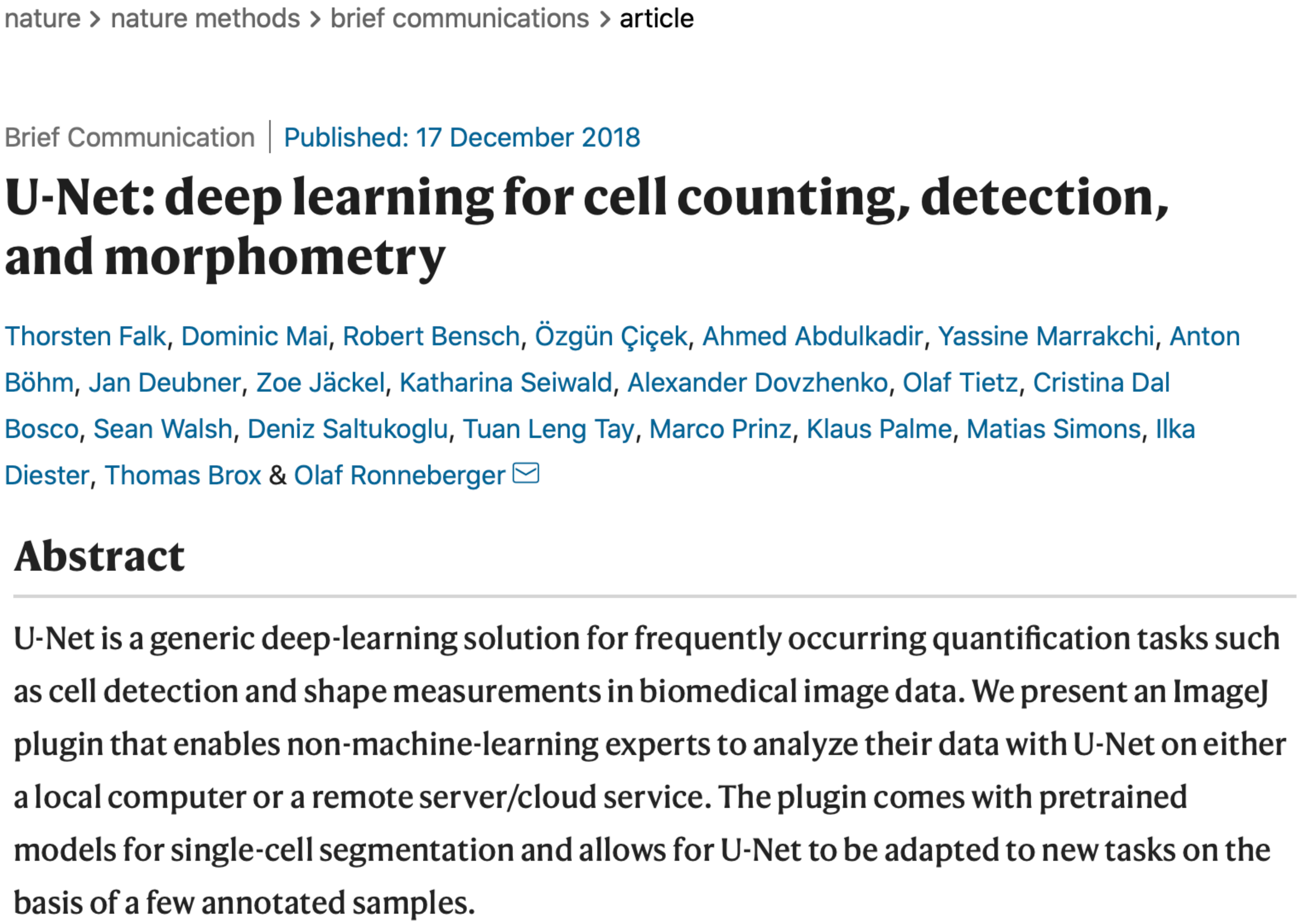
1. Machine-learning experts are often required
2. High amount of annotations (time-consuming)
3. Getting segmented images does not mean you nailed it: time-consuming for checking biases and eventually start back the annotations from the beginning.

DCNN IN IMAGE CYTOMETRY

1. Image pre-processing
2. Classification and detection (classify objects within the image)
3. Image segmentation (sliding windows for volumetric imaging)
4. Transfer learning and Domain adaptation

DCNN APPLICATIONS

1. High content screening
2. Cytology and histopathology
3. Time-lapse imaging analysis



nature > nature methods > brief communications > article

Brief Communication | Published: 17 December 2018

U-Net: deep learning for cell counting, detection, and morphometry

Thorsten Falk, Dominic Mai, Robert Bensch, Özgün Çiçek, Ahmed Abdulkadir, Yassine Marrakchi, Anton Böhm, Jan Deubner, Zoe Jäckel, Katharina Seiwald, Alexander Dovzhenko, Olaf Tietz, Cristina Dal Bosco, Sean Walsh, Deniz Saltukoglu, Tuan Leng Tay, Marco Prinz, Klaus Palme, Matias Simons, Ilka Diester, Thomas Brox & Olaf Ronneberger

Abstract

U-Net is a generic deep-learning solution for frequently occurring quantification tasks such as cell detection and shape measurements in biomedical image data. We present an ImageJ plugin that enables non-machine-learning experts to analyze their data with U-Net on either a local computer or a remote server/cloud service. The plugin comes with pretrained models for single-cell segmentation and allows for U-Net to be adapted to new tasks on the basis of a few annotated samples.

INTRODUCTION

ImageJ plug-in that enables non-machine-learning experts to analyse their data with U-Net.

Generic deep-learning-based software package for cell detection and cell segmentation

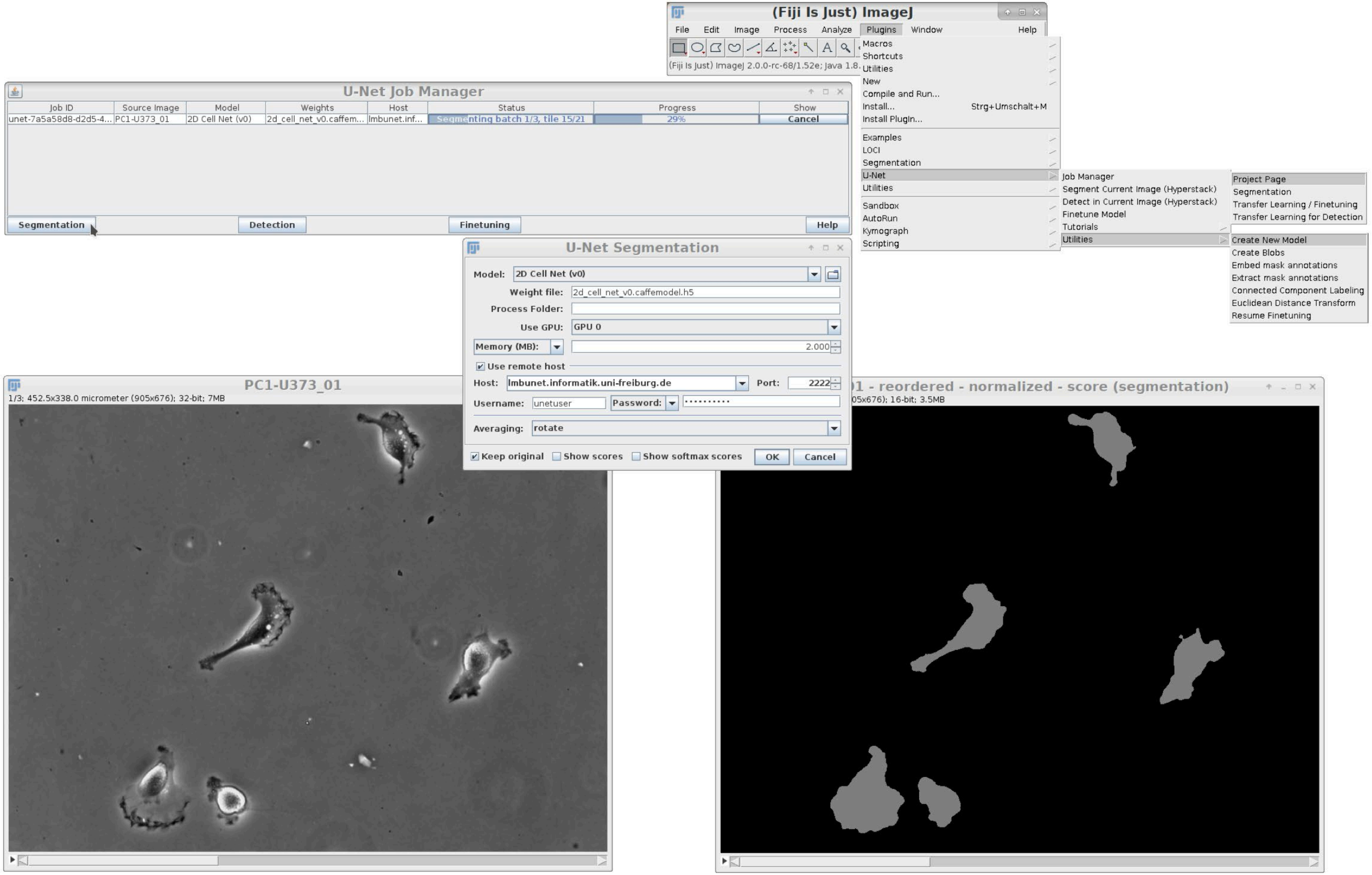
Tuned for new tasks adaptation with few annotated samples

SOFTWARE FEATURES

- The U-Net segmentation plugin interface
- New annotations for different expected results
- Cell detection, classification, and counting
- Pretrained annotations tuned for U-Net segmentation plugin

SOFTWARE FEATURES

- The U-Net segmentation plugin interface
- New annotations for different expected results
- Cell detection, classification, and counting
- Pretrained annotations tuned for U-Net segmentation plugin



Job ID	Source Image	Model	Weights	Host	Status	Progress	Show
unet-7a5a58d8-d2d5-4...	PC1-U373_01	2D Cell Net (v0)	2d_cell_net_v0.caffem...	lmbunet.inf...	Segmenting batch 1/3, tile 15/21	29%	Cancel

Model: 2D Cell Net (v0)

Weight file: 2d_cell_net_v0.caffemodel.h5

Process Folder:

Use GPU: GPU 0

Memory (MB): 2.000

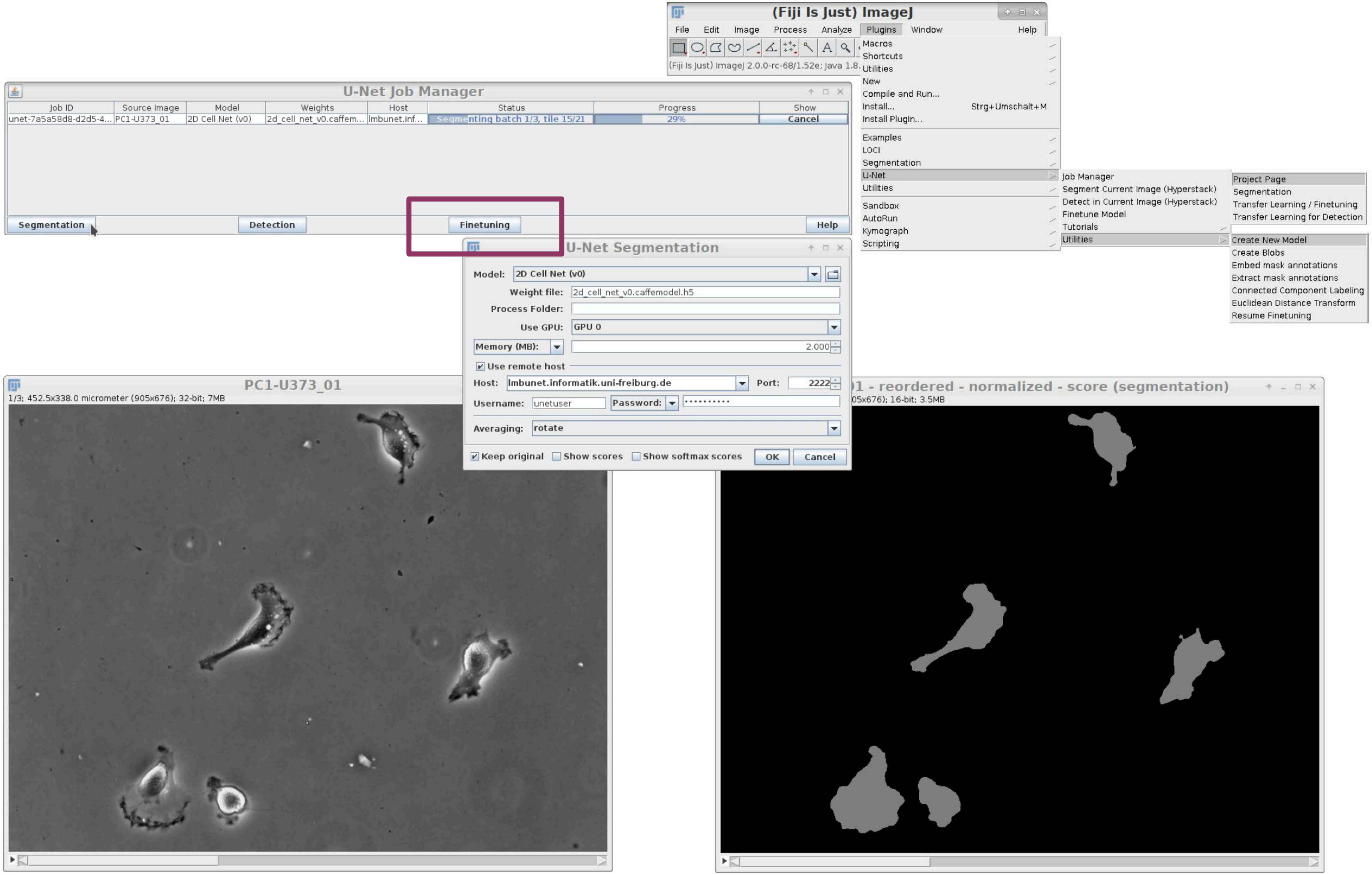
☑ Use remote host

Host: lmbunet.informatik.uni-freiburg.de Port: 2222

Username: unetuser Password: ••••••••••

Averaging: rotate

☑ Keep original ☐ Show scores ☐ Show softmax scores OK Cancel



U-Net Job Manager

Job ID	Source Image	Model	Weights	Host	Status	Progress	Show
unet-7a5a58d8-d2d5-4...	PC1-U373_01	2D Cell Net (v0)	2d_cell_net_v0.caffem...	lmbunet.inf...	Segmenting batch 1/3, tile 15/21	29%	Cancel

U-Net Segmentation

Model: 2D Cell Net (v0)

Weight file: 2d_cell_net_v0.caffemodel.h5

Process Folder:

Use GPU: GPU 0

Memory (MB): 2.000

☑ Use remote host

Host: lmbunet.informatik.uni-freiburg.de Port: 2222

Username: unetuser Password: •••••••••

Averaging: rotate

☑ Keep original ☐ Show scores ☐ Show softmax scores

PC1-U373_01

1/3; 452.5x338.0 micrometer (905x676); 32-bit; 7MB

01 - reordered - normalized - score (segmentation)

05x676); 16-bit; 3.5MB

SOFTWARE FEATURES

- The U-Net segmentation plugin interface
- New annotations for different expected results
- Cell detection, classification, and counting
- Pretrained annotations tuned for U-Net segmentation plugin

U-Net software guidelines - *New annotations for different expected results*

- Be consistent

- Quality over quantity

- Capture the whole range of visual appearances (not important the amount of the annotations, but the variability of objects within the dataset)

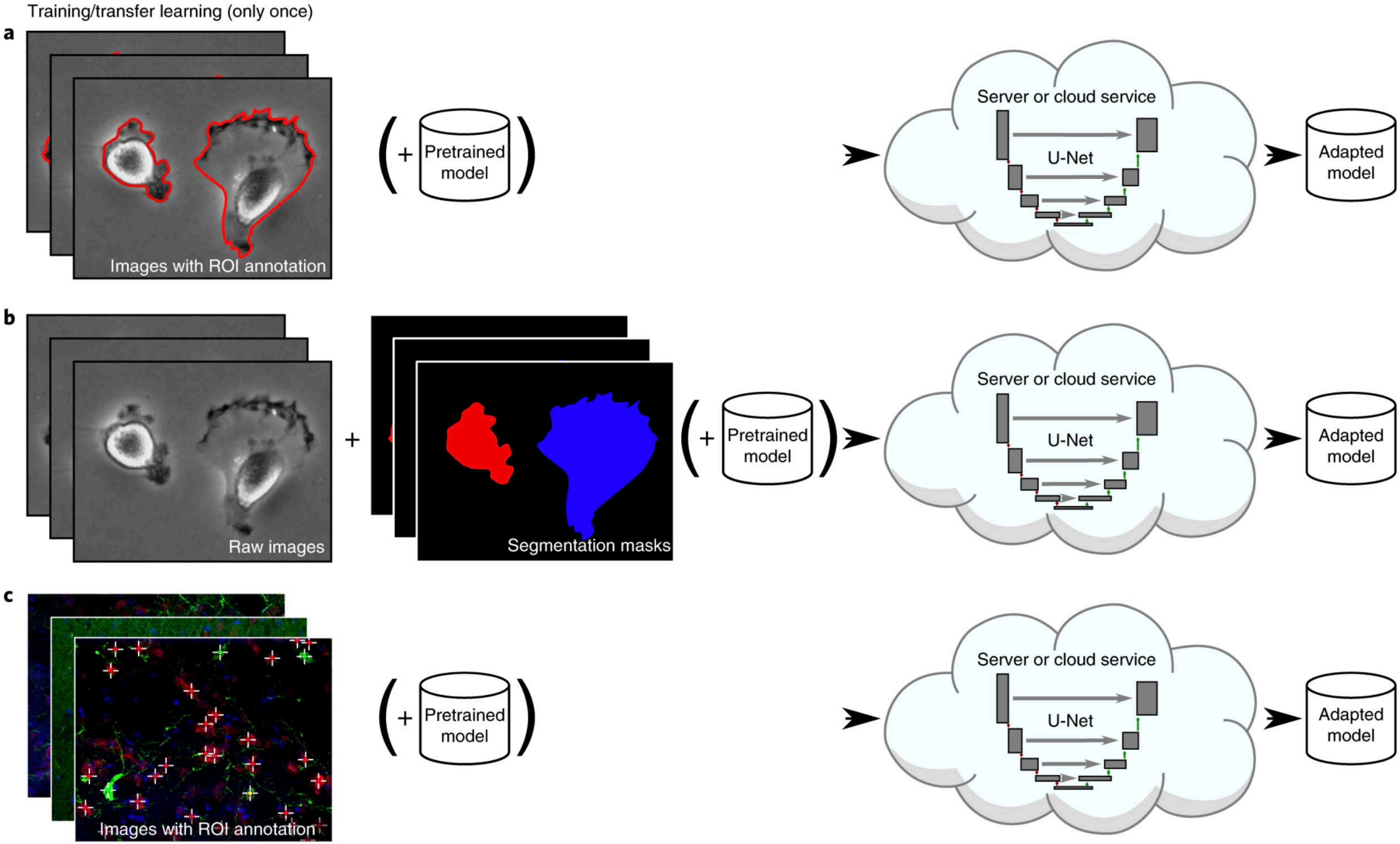
U-Net software guidelines - *New annotations for different expected results*

Segmentation (2D)

with ROI

with segmentation mask

Detection with multipoint annotations (2D)

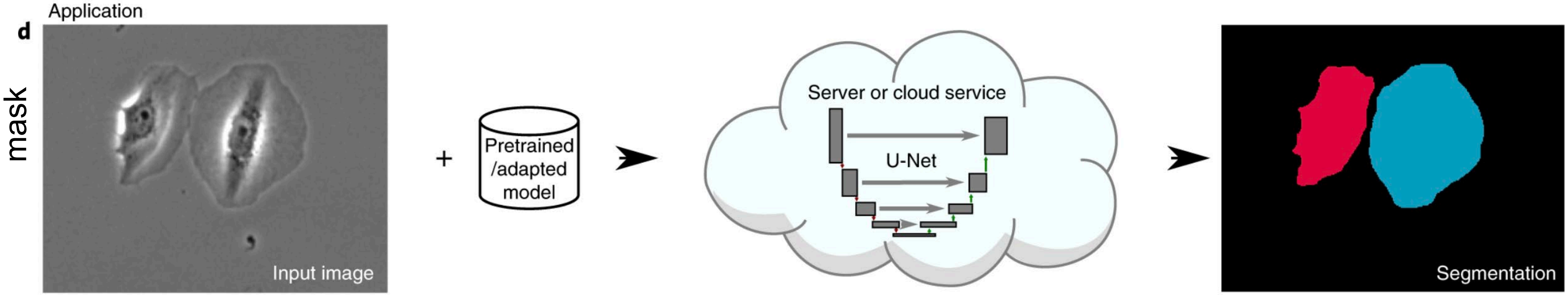


SOFTWARE FEATURES

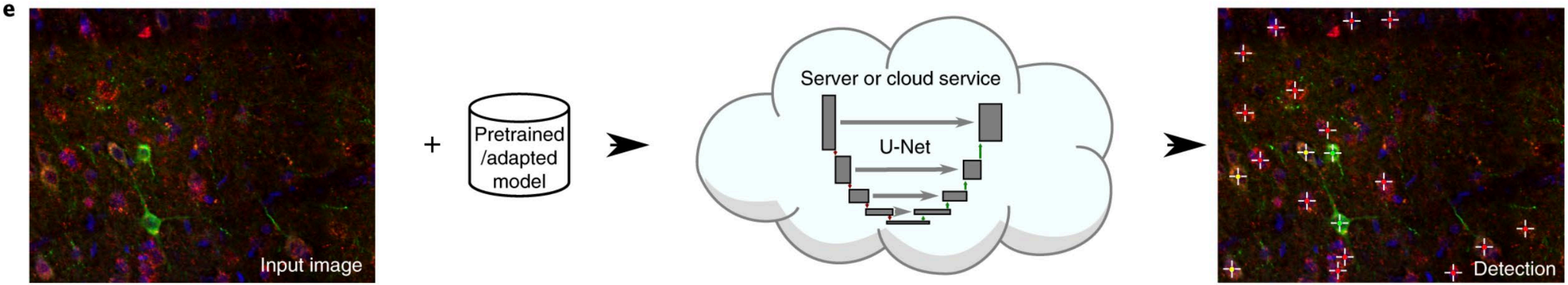
- The U-Net segmentation plugin interface
- New annotations for different expected results
- Cell detection, classification, and counting
- Pretrained annotations tuned for U-Net segmentation plugin

U-Net software guidelines - *Cell detection, classification, and counting*

Segmentation (2D) with segmentation mask



Detection with multipoint annotations (2D)



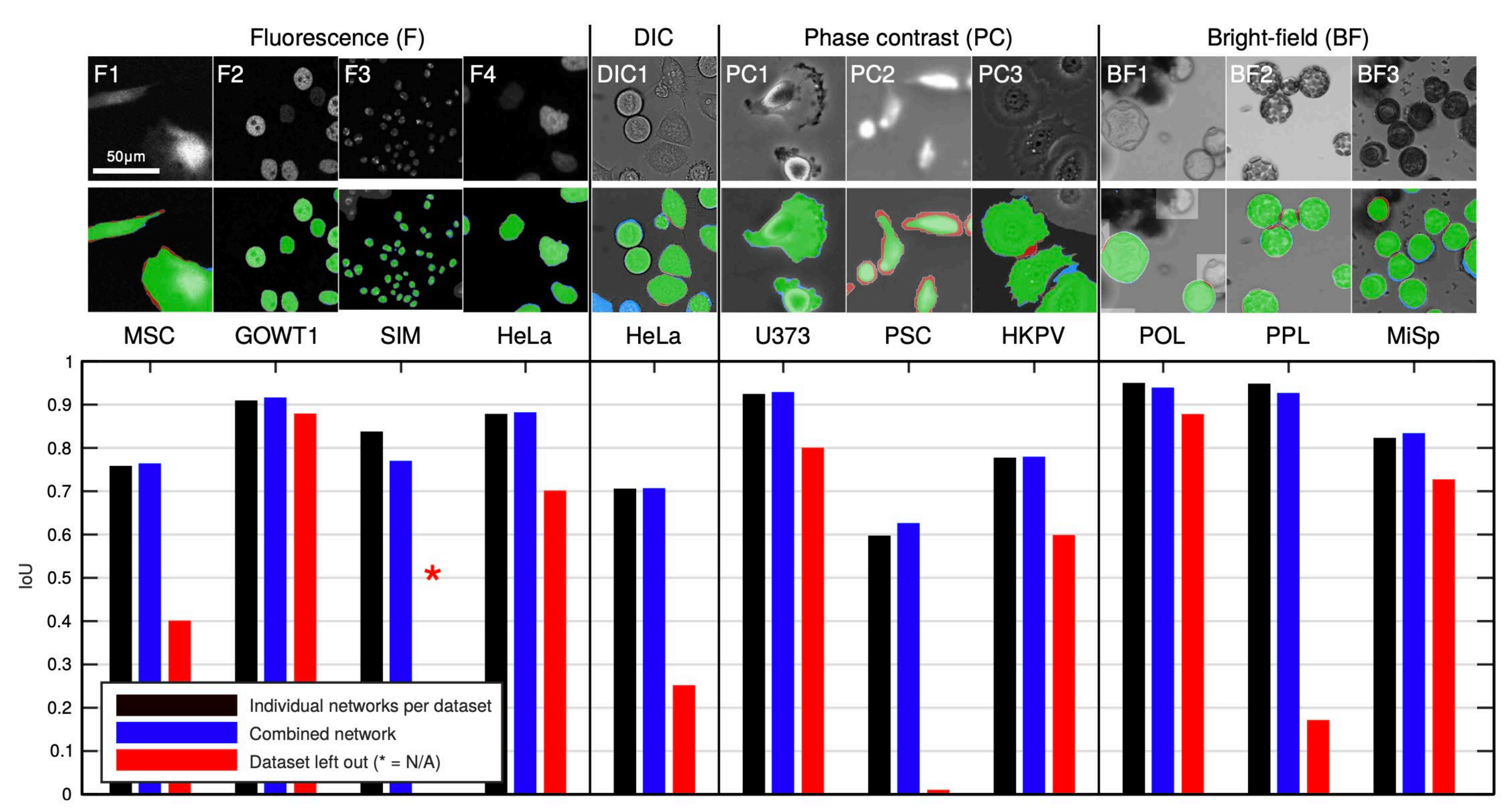
SOFTWARE FEATURES

- The U-Net segmentation plugin interface
- New annotations for different expected results
- Cell detection, classification, and counting
- Pretrained annotations tuned for U-Net segmentation plugin

Included pretrained annotations from experiment images

- **Segmentation** mask pretrained images
  - segmentation of single cells recorded with various common imaging modalities

(2) Segmentation of single cells recorded with various common imaging modalities

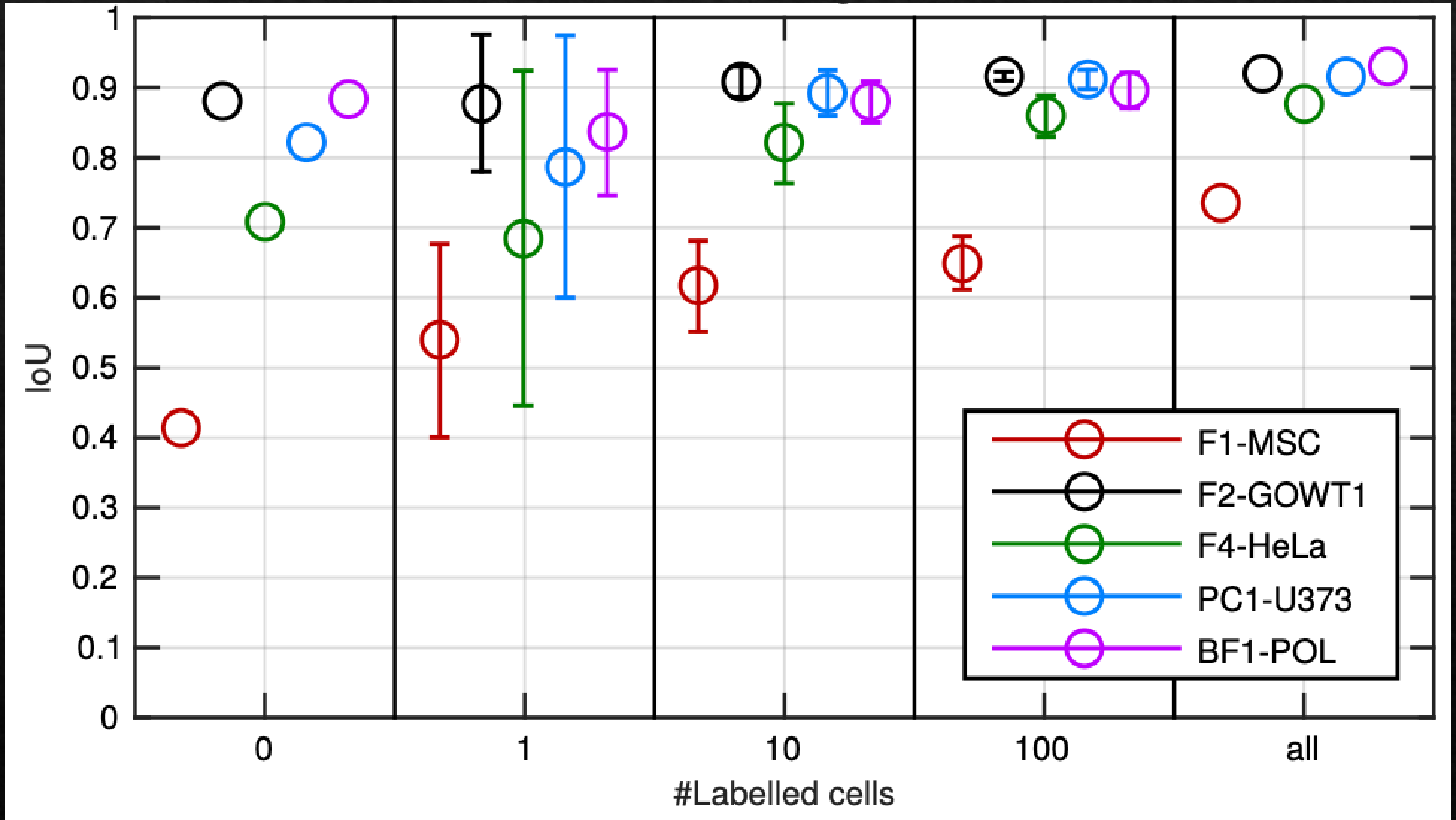


Intersection over union (IoU) evaluates the overlap between the ground truth for each cell and the segmentation computed by the network.
A value of 0 corresponds to no overlap at all, a value of 1 to a perfect pixel-by-pixel match. Values above 0.8 indicate a very good segmentation.

(2) Segmentation of single cells recorded with various common imaging modalities

TRANFER LEARNING

When your cell type to be analysed is morphologically different from these 11 pre-trained cells



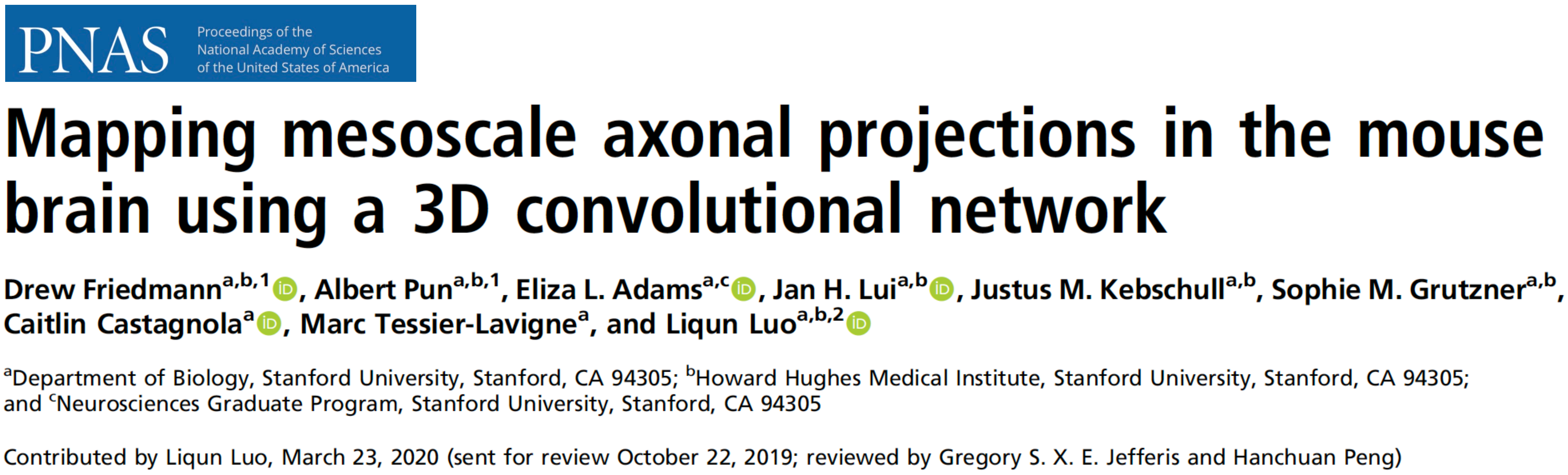
CONCLUSION

The integration of the software in ImageJ and a step-by-step protocol tutorial make deep learning available to scientists without a computer-science background

U-Net segmentation plugin can be directly used to count or segment your input image

Transfer learning must be used when your cells to be analysed morphologically differ from the pre-trained ones (11 showed before)

Applicable model for a High content screening



PNAS Proceedings of the National Academy of Sciences of the United States of America

Mapping mesoscale axonal projections in the mouse brain using a 3D convolutional network

Drew Friedmann[a,b,1], Albert Pun[a,b,1], Eliza L. Adams[a,c], Jan H. Lui[a,b], Justus M. Kebschull[a,b], Sophie M. Grutzner[a,b], Caitlin Castagnola[a], Marc Tessier-Lavigne[a], and Liqun Luo[a,b,2]

[a]Department of Biology, Stanford University, Stanford, CA 94305; [b]Howard Hughes Medical Institute, Stanford University, Stanford, CA 94305; and [c]Neurosciences Graduate Program, Stanford University, Stanford, CA 94305

Contributed by Liqun Luo, March 23, 2020 (sent for review October 22, 2019; reviewed by Gregory S. X. E. Jefferis and Hanchuan Peng)

Significance

Simple, high-resolution methods for visualizing complex neural circuitry in 3D in the intact mammalian brain are revolutionizing the way researchers study brain connectivity and function. However, concomitant development of robust, open-source computational tools for the automated quantification and analysis of these volumetric data has not kept pace. We have developed a method to perform automated identifications of axonal projections in whole mouse brains. Our method takes advantage of recent advances in machine learning and outperforms existing methods in ease of use, speed, accuracy, and generalizability for axons from different types of neurons.

INTRODUCTION

Volumetric imaging to visualize neurons in intact mouse brain tissue has become a widespread technique

A growing need for computational tools to analyse the resultant large datasets in three dimensions

Without a computational method for quantifying axon content, researchers must still select and score representative two-dimensional (2D) optical sections

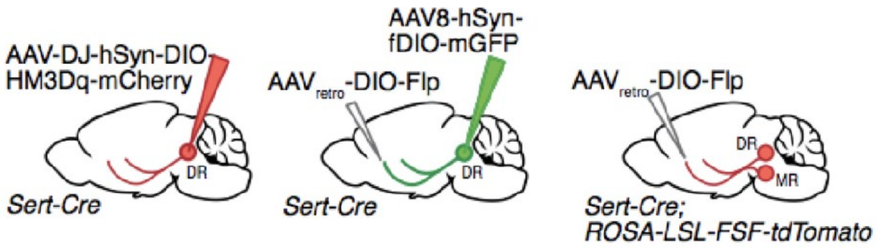
AIMS

The automated identification and segmentation of axons from 3D images should circumvent these limitations

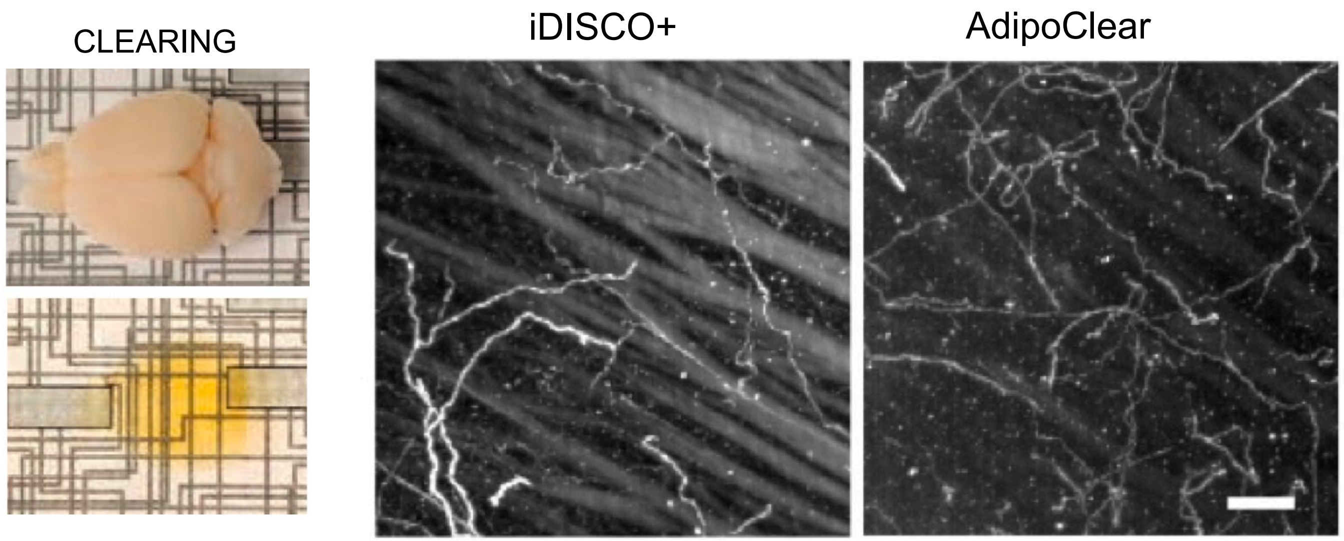
Accurate “projectome” reconstruction (also for dense labeling)

Mitigation of artifacts that could contaminate the samples (myelin autofluorescence and non-specific antibody labeling)

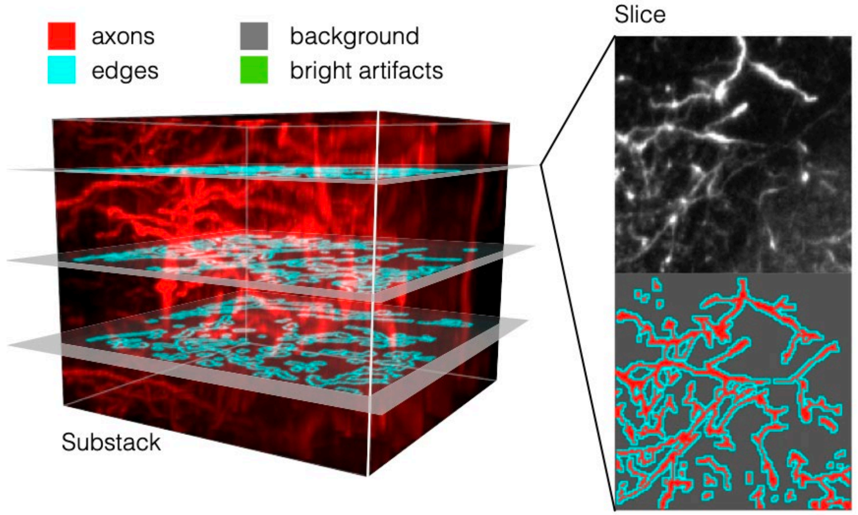
Generic software package that can adapt itself autonomously to the task from appropriate data



18 separate intact brains containing fluorescently labelled serotonergic axons were imaged



Autofluorescent fiber tracts in striatum



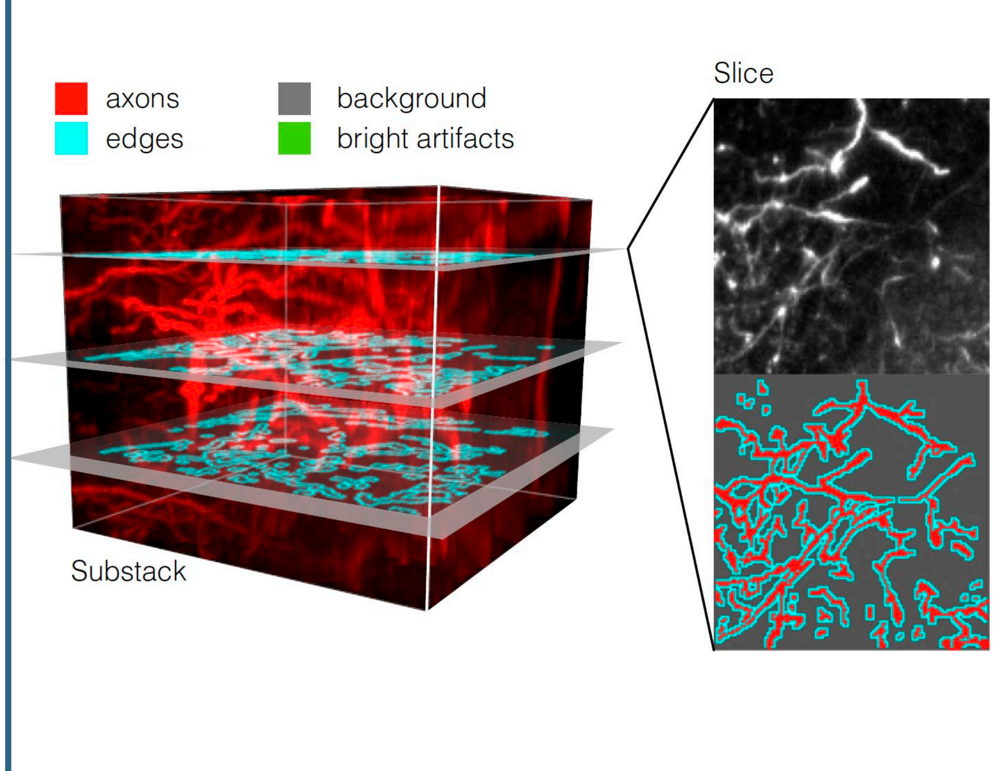
36 substucks manually labeled for axons

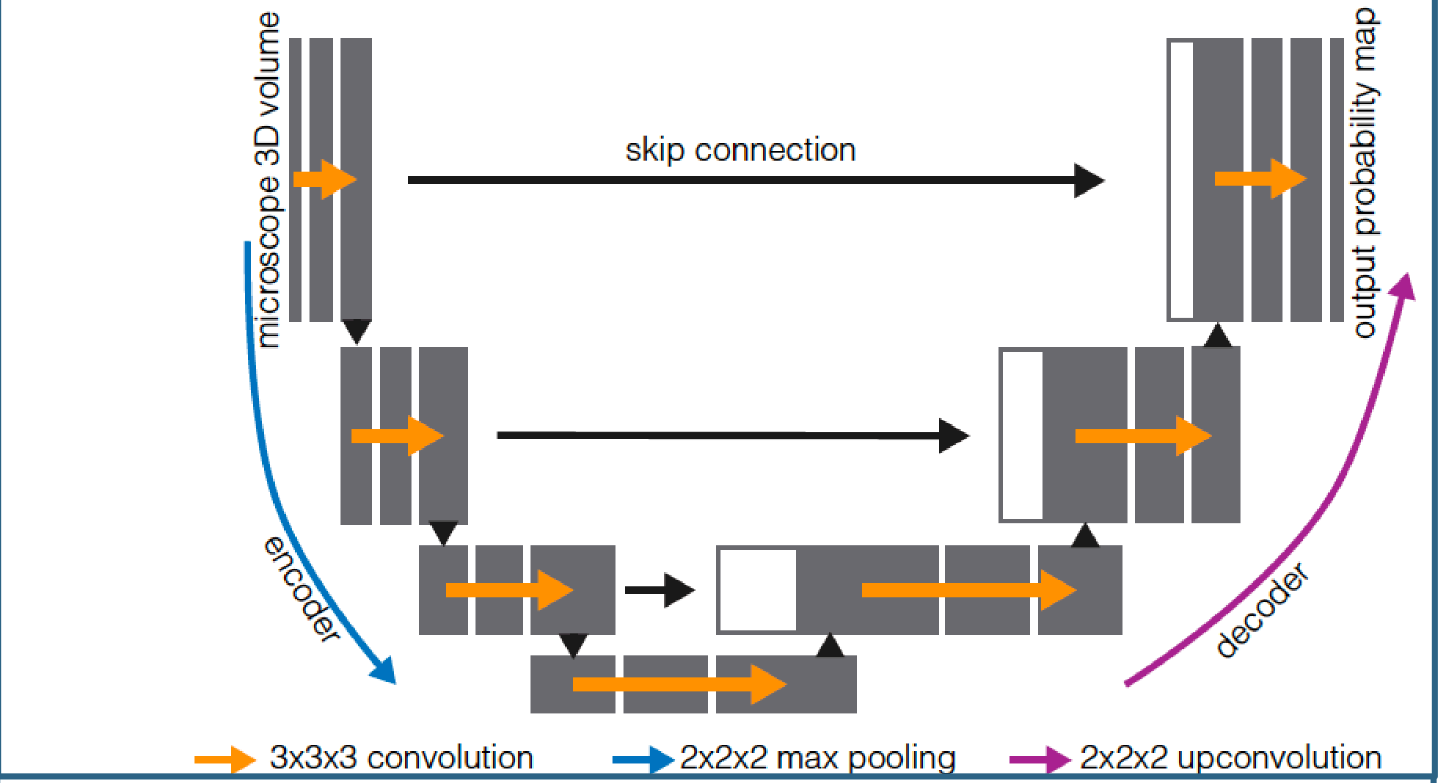
40 substacks manually labeled for artifacts

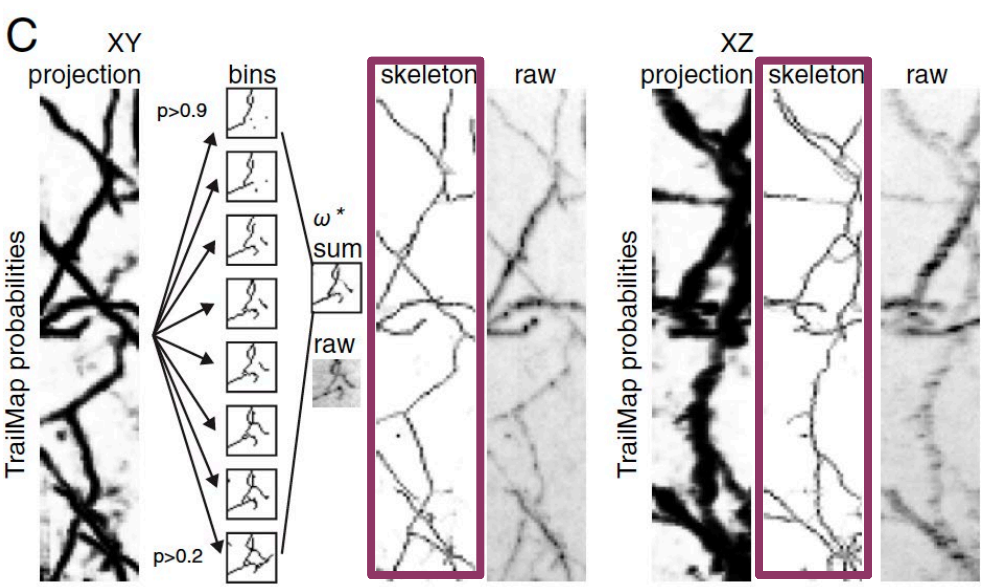
- Substacks were manually selected to represent the diversity of possible brain regions, background levels, and axon morphology.

- They annotated 3 to 10 individual XY planes within each substack, at a spacing of 80 to 180 μm between labeled slices

- In the same XY slice, a second label surrounding the axon annotation ("edges") was automatically generated, and the remaining unlabeled voxels in the slice were given a label for "background."



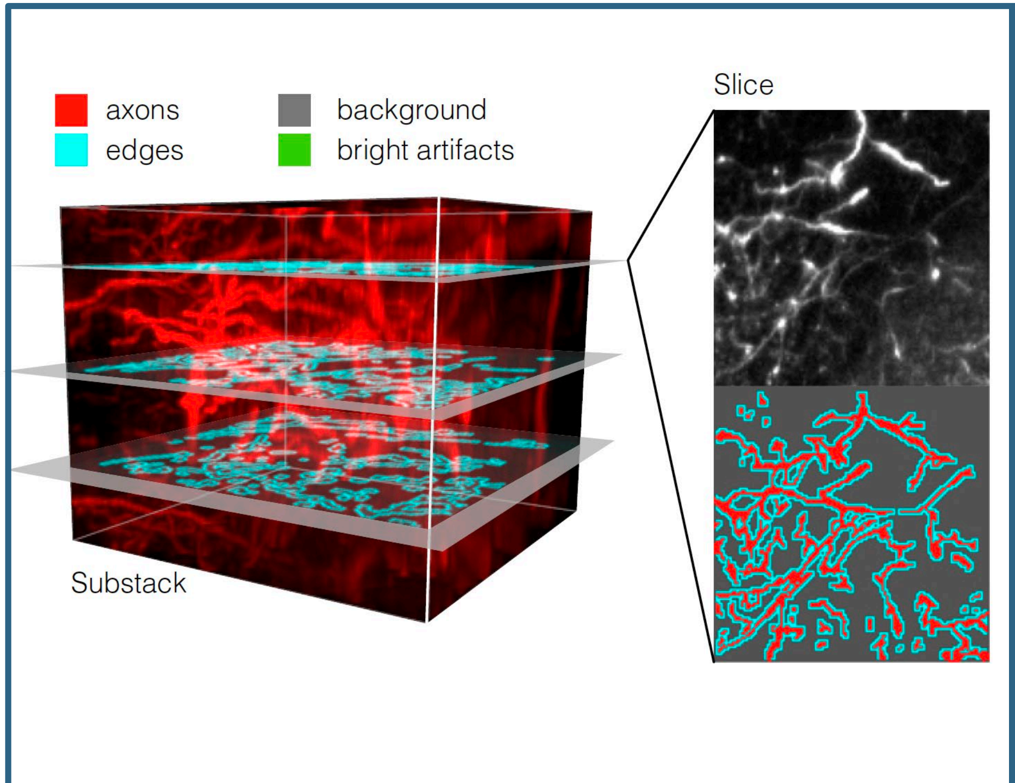


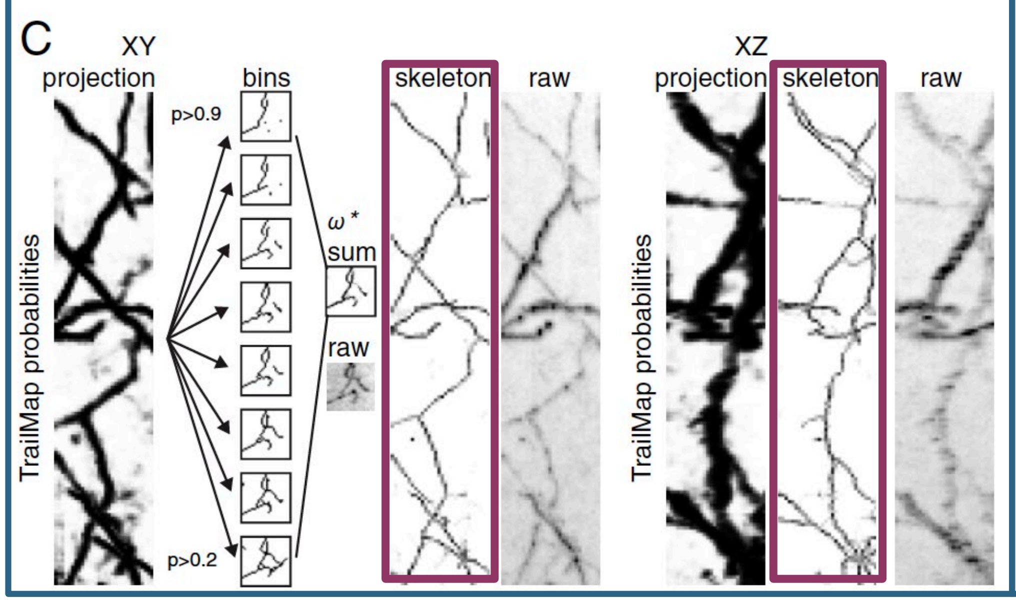


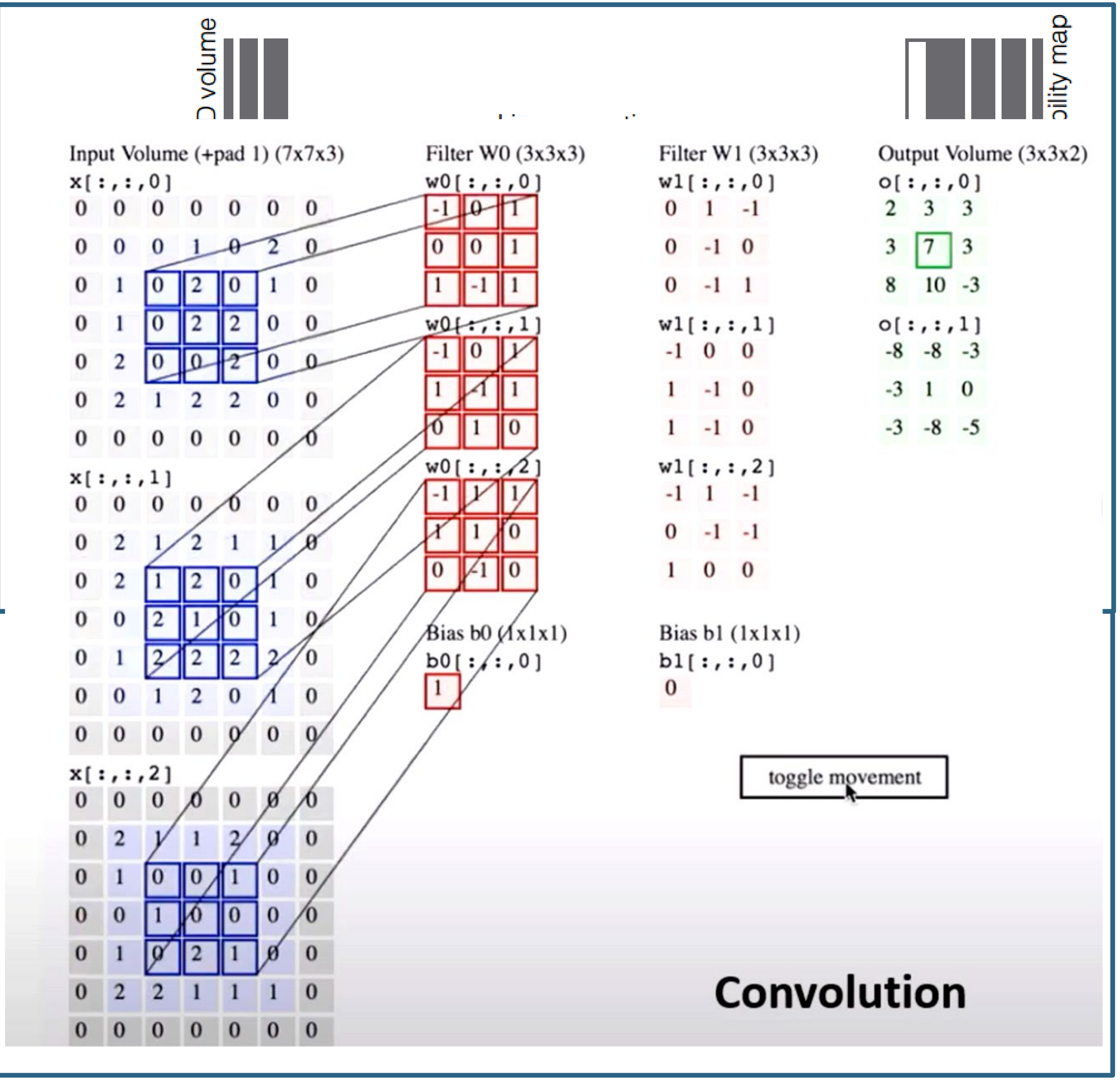
- For a given input cube, the network outputted a 36 × 36 × 36 volume containing voxel-wise axon predictions ($0 < P < 1$)

- Large volumes, including intact brains, were processed with a **sliding window strategy**.

- From this output, a **thinning strategy** was implemented to generate a skeletonized armature of the extracted axons







Input Volume (+pad 1) (7x7x3)

x[:,:,0]

0	0	0	0	0	0	0
0	0	0	1	0	2	0
0	1	0	2	0	1	0
0	1	0	2	2	0	0
0	2	0	0	2	0	0
0	2	1	2	2	0	0
0	0	0	0	0	0	0

x[:,:,1]

0	0	0	0	0	0	0
0	2	1	2	1	1	0
0	2	1	2	0	1	0
0	0	2	1	0	1	0
0	1	2	2	2	2	0
0	0	1	2	0	1	0
0	0	0	0	0	0	0

x[:,:,2]

0	0	0	0	0	0	0
0	2	1	1	2	0	0
0	1	0	0	1	0	0
0	0	1	0	0	0	0
0	1	0	2	1	0	0
0	2	2	1	1	1	0
0	0	0	0	0	0	0

Filter W0 (3x3x3)

w0[:,:,0]

-1	0	1
0	0	1
1	-1	1

w0[:,:,1]

-1	0	1
1	-1	1
0	1	0

w0[:,:,2]

-1	1	1
1	1	0
0	-1	0

Bias b0 (1x1x1)

b0[:,:,0]

1

Filter W1 (3x3x3)

w1[:,:,0]

0	1	-1
0	-1	0
0	-1	1

w1[:,:,1]

-1	0	0
1	-1	0
1	-1	0

w1[:,:,2]

-1	1	-1
0	-1	-1
1	0	0

Bias b1 (1x1x1)

b1[:,:,0]

0

Output Volume (3x3x2)

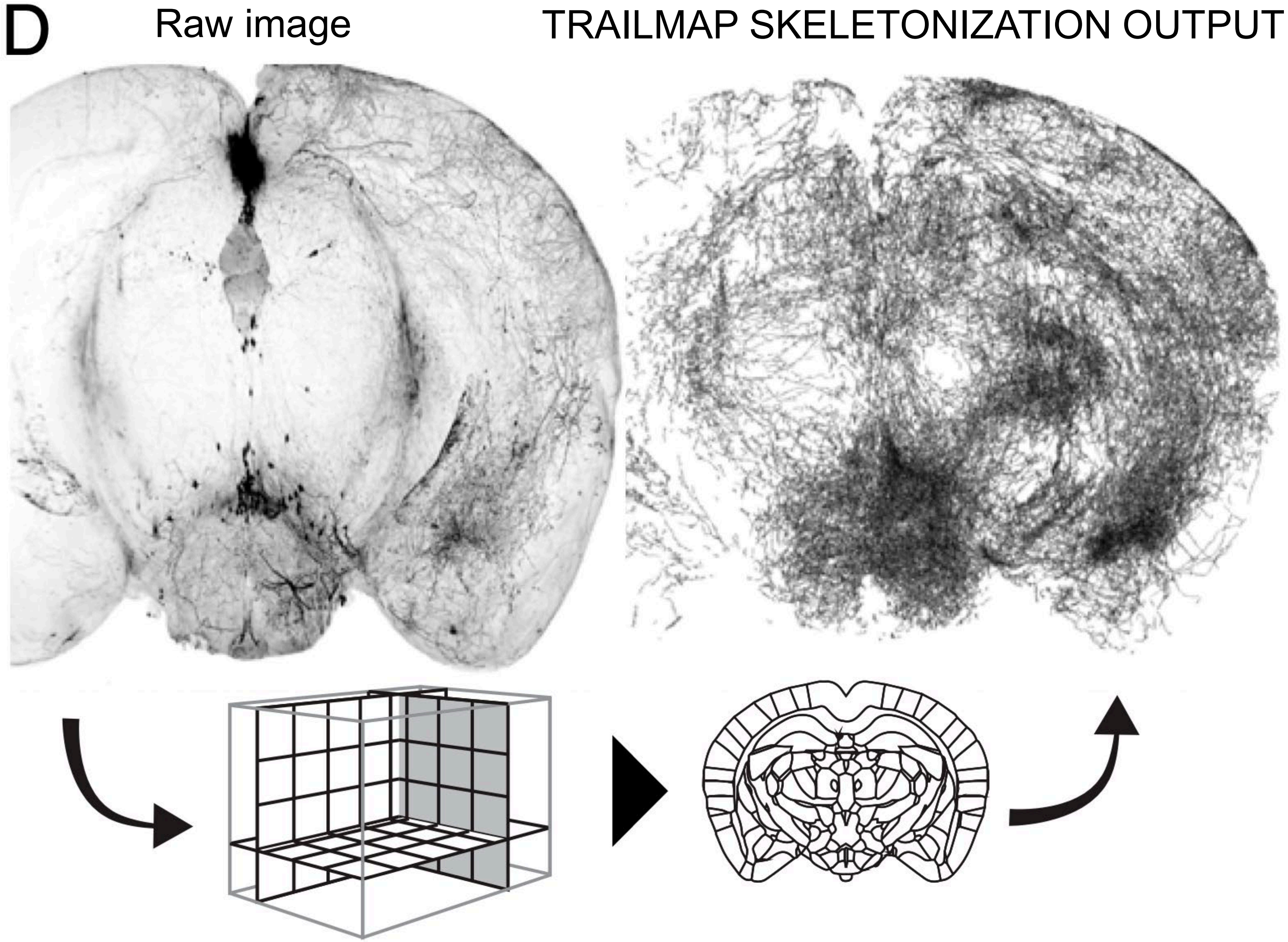
o[:,:,0]

2	3	3
3	7	3
8	10	-3

o[:,:,1]

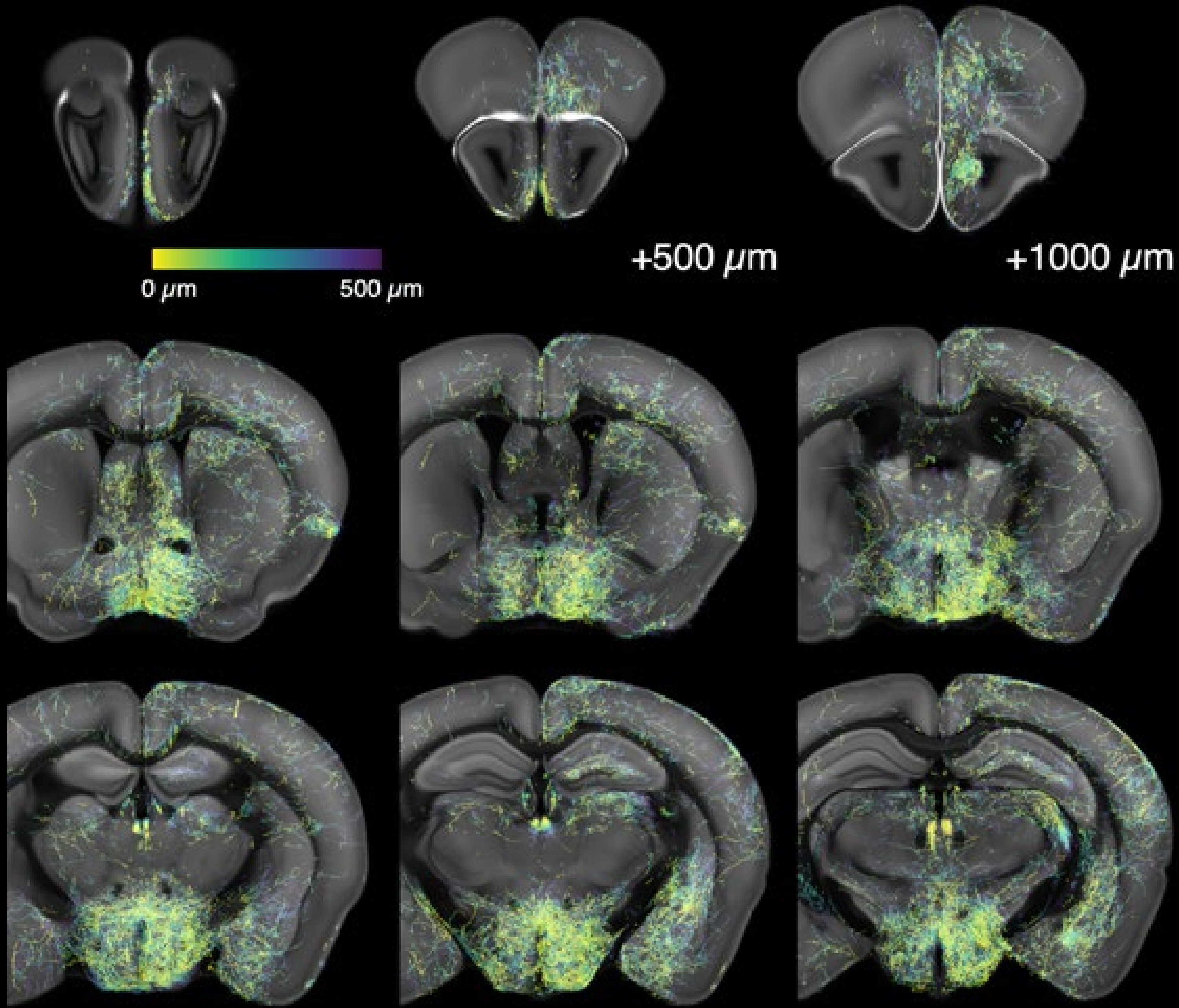
-8	-8	-3
-3	1	0
-3	-8	-5

Convolution



A 2-mm-thick volumetric coronal slab, before and after the TrailMap procedure, which includes **axon extraction**, **skeletonization**, and **alignment** to the Allen Brain Atlas Common Coordinate Framework

With the extracted axonal projectome transformed into the Allen Institute reference space, the axon armature could be overlaid on a template to better highlight their presence, absence, and structure in local subregions.

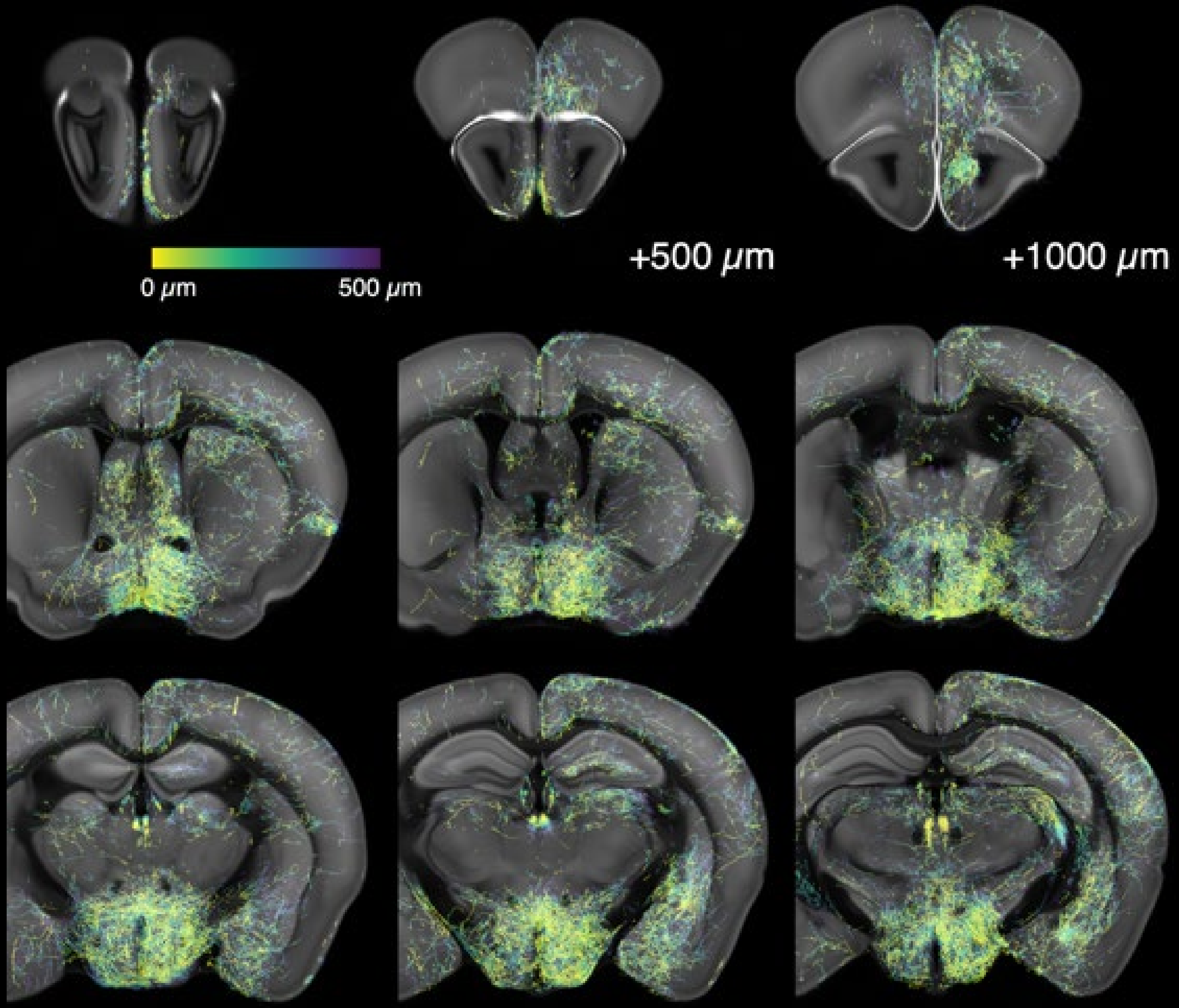


Each panel represents 500 μm of Z-depth in the coronal axis, color-coded by depth

With the extracted axonal projectome transformed into the Allen Institute reference space, the axon armature could be overlaid on a template to better highlight their presence, absence, and structure in local subregions.

↓

However, it was difficult to resolve steep changes in density or local hotspots of innervation.

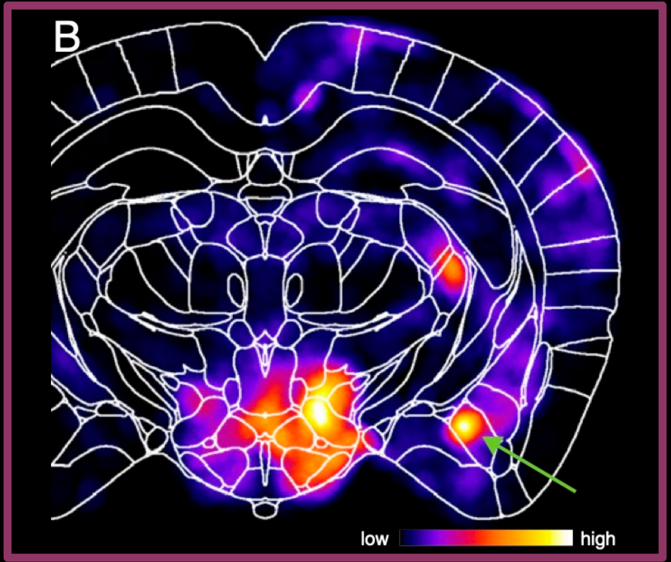


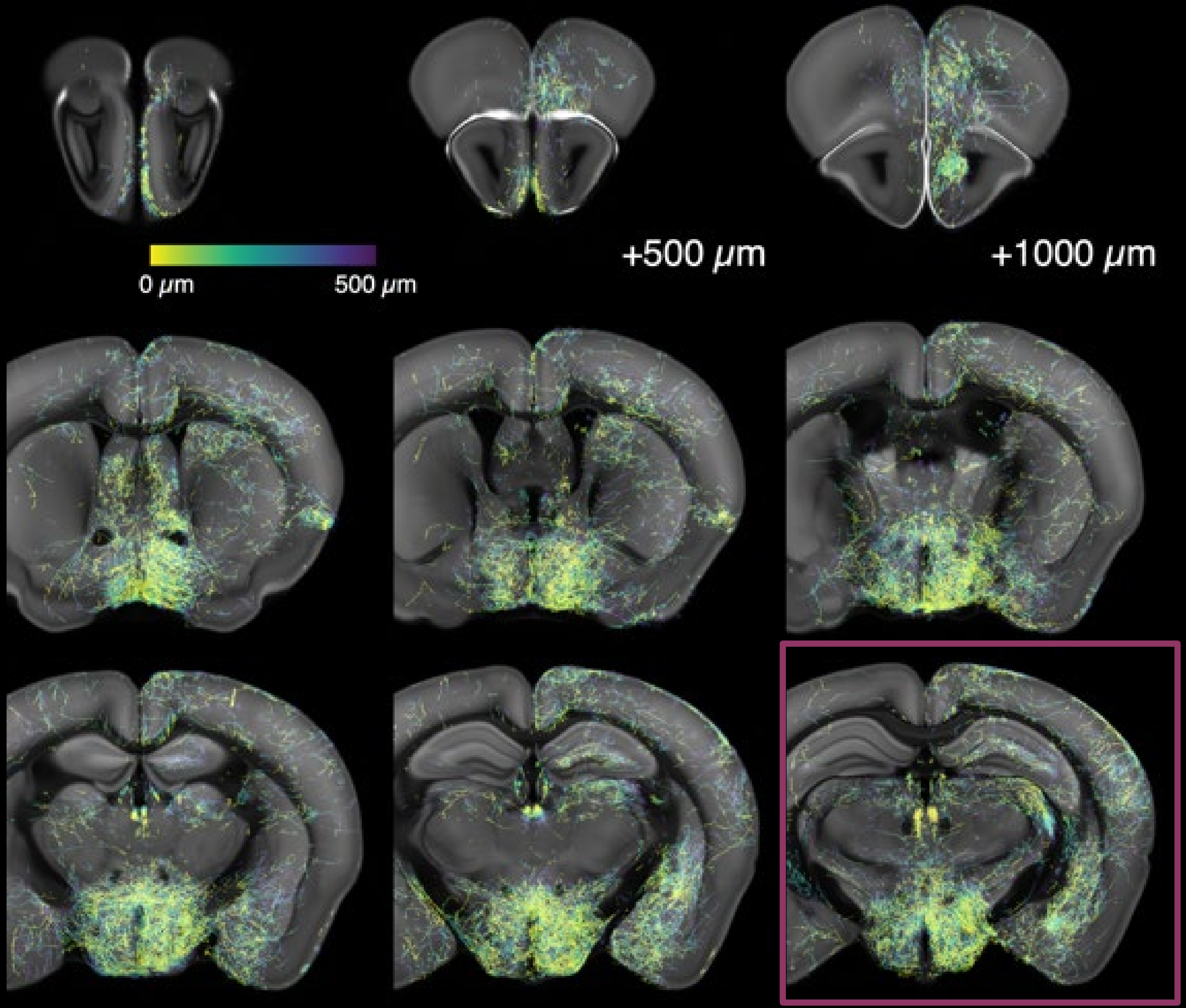
Each panel represents 500 μm of Z-depth in the coronal axis, color-coded by depth

With the extracted axonal projectome transformed into the Allen Institute reference space, the axon armature could be overlaid on a template to better highlight their presence, absence, and structure in local subregions.

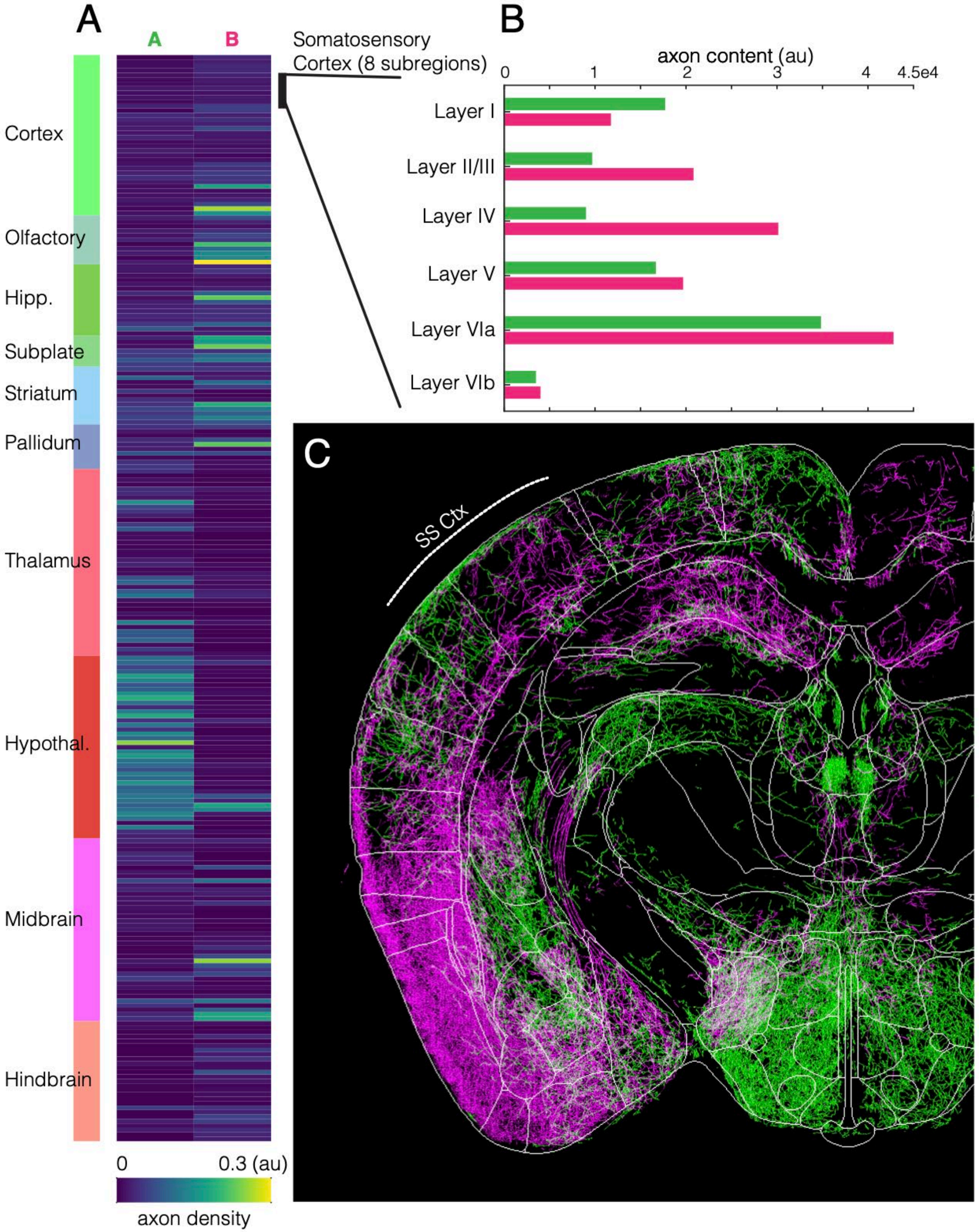
However, it was difficult to resolve steep changes in density or local hotspots of innervation.

3D DENSITY HEATMAP



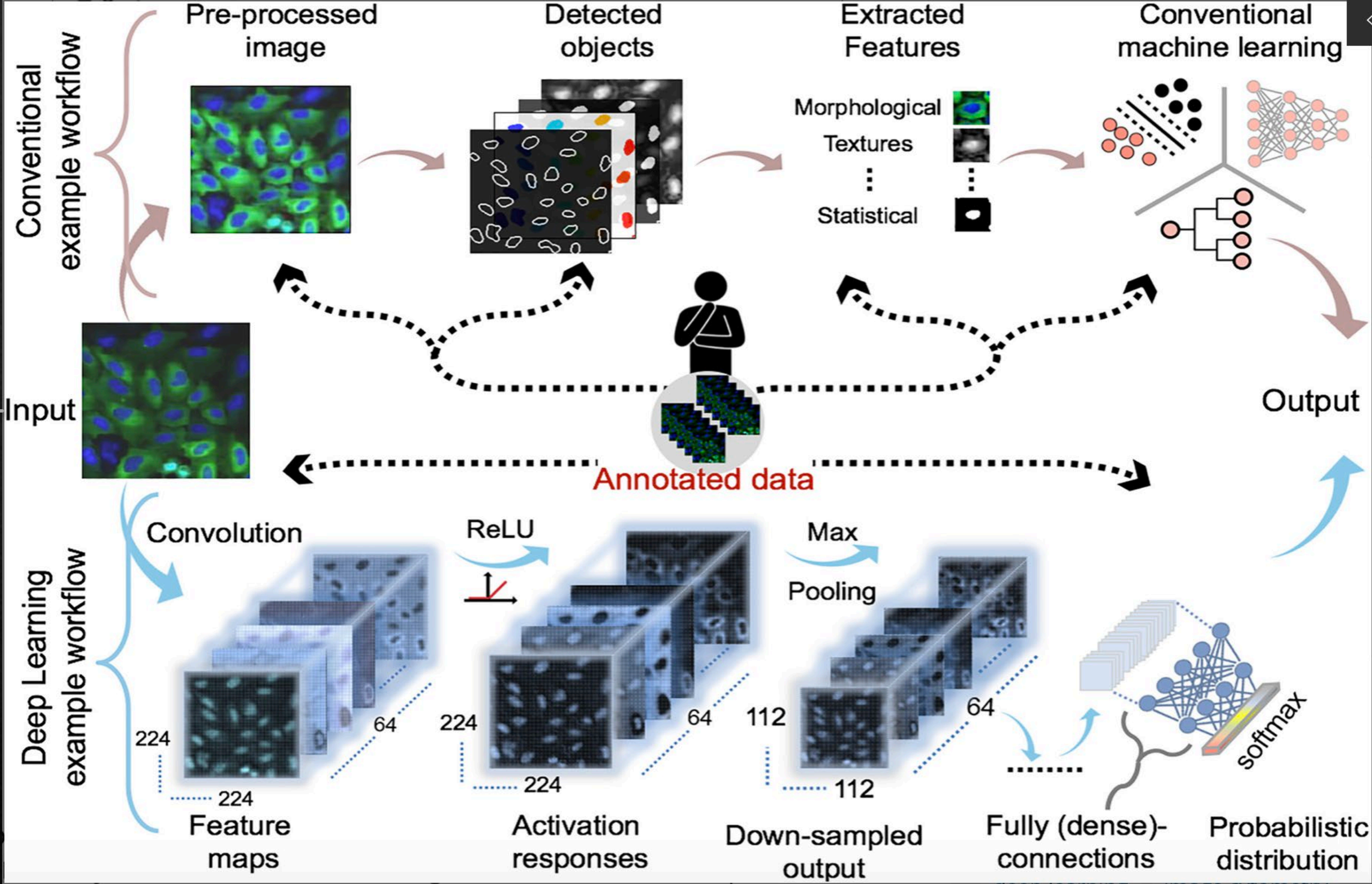


Each panel represents 500 µm of Z-depth in the coronal axis, color-coded by depth


A
A
B
Somatosensory
Cortex (8 subregions)
Cortex
Olfactory
Hipp.
Subplate
Striatum
Pallidum
Thalamus
Hypothal.
Midbrain
Hindbrain
0
0.3 (au)
axon density
B
axon content (au)
0
1
2
3
4
4.5e4
Layer I
Layer II/III
Layer IV
Layer V
Layer VIa
Layer VIb
C
SS Ctx

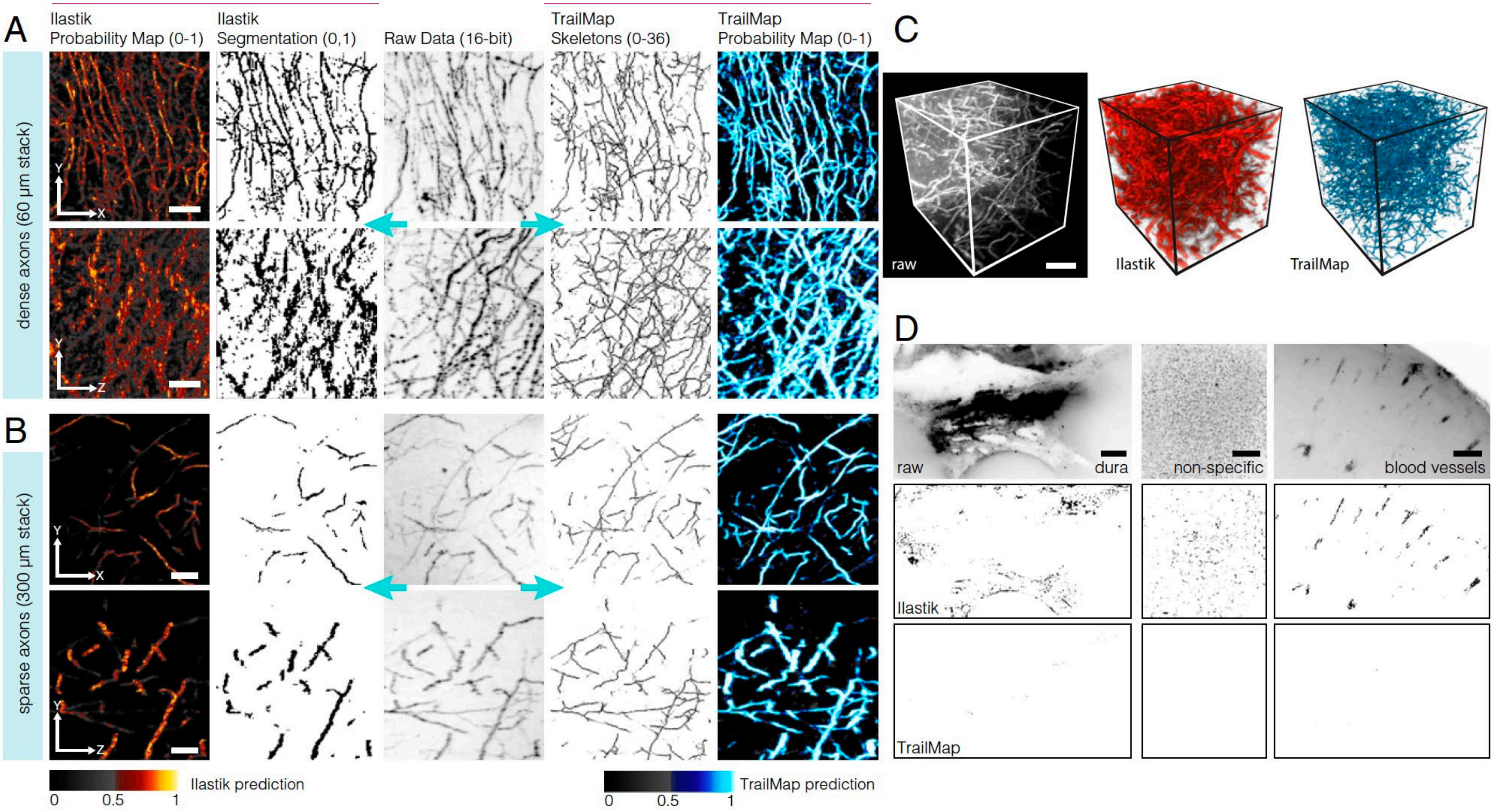
NOT TRAINABLE

TRAINABLE



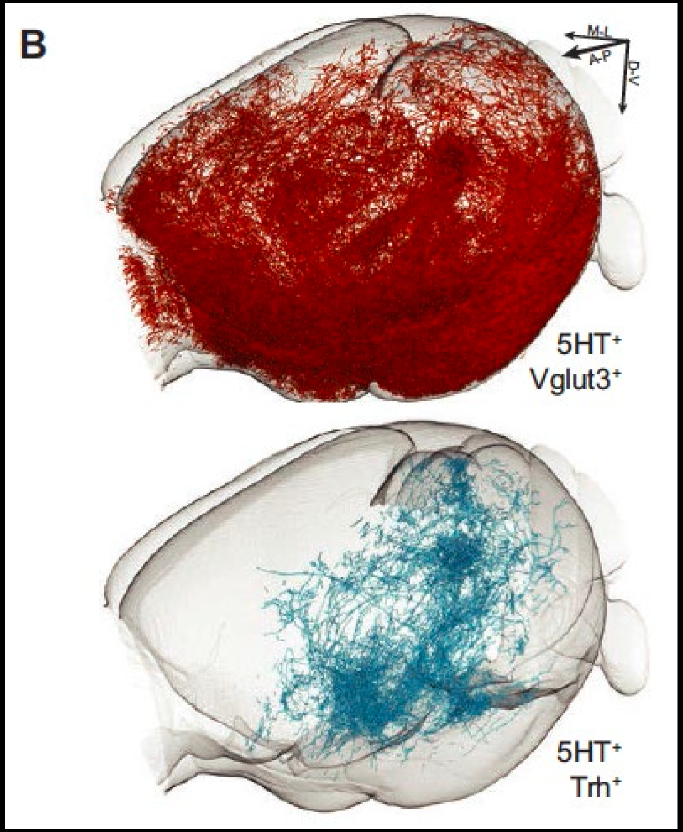
Conventional machine learning

TRAILMAP DCNN

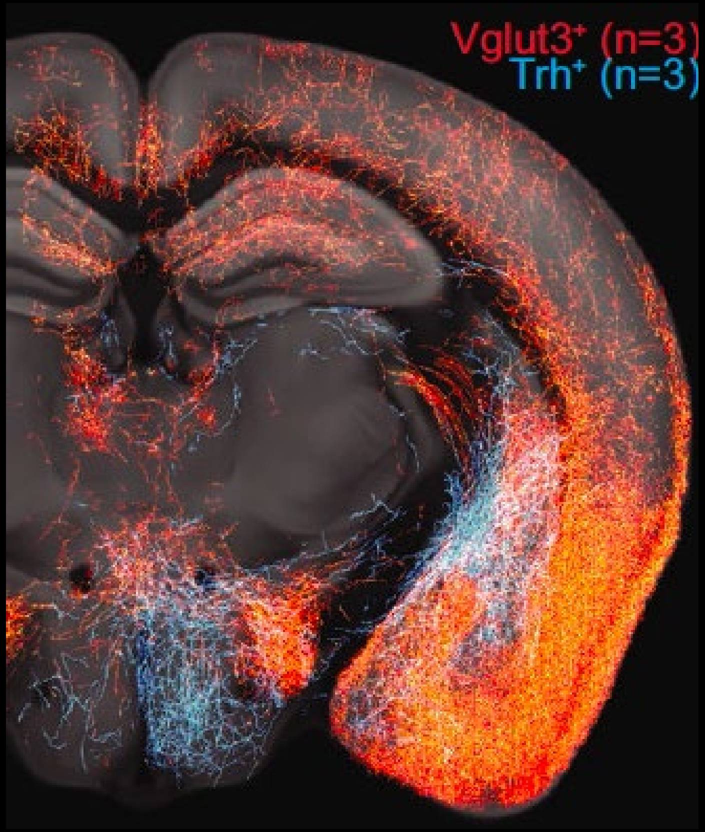


TRAILMAP application

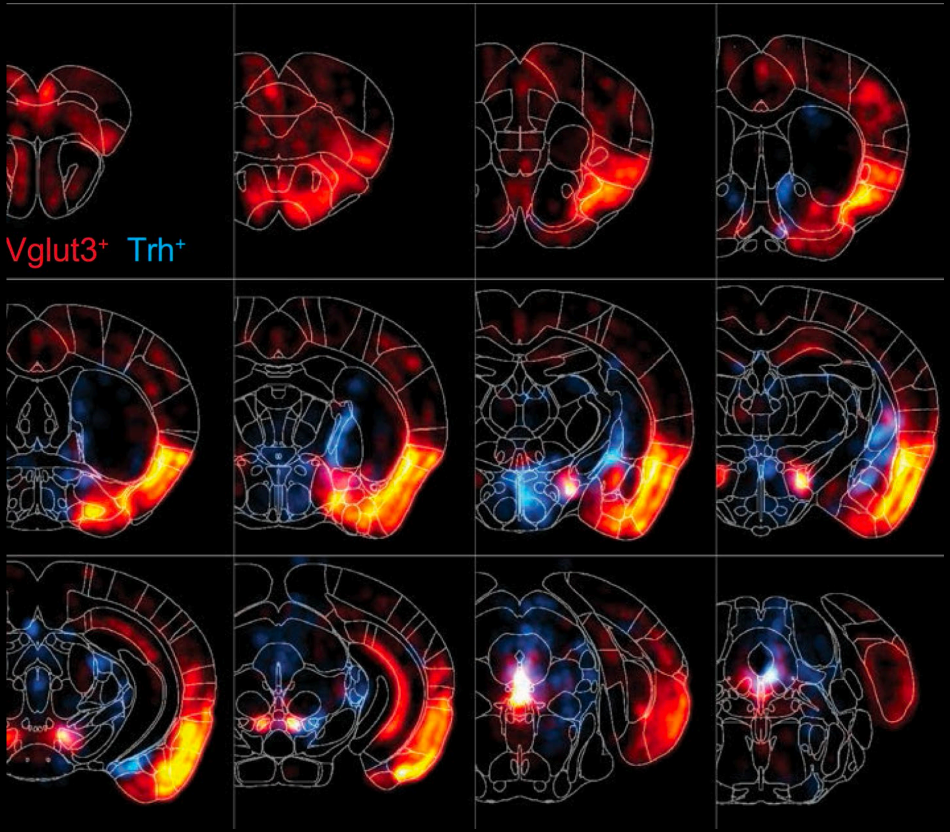
Ren et al. eLife 2019 - Single-cell transcriptomes and whole-brain projections of serotonin neurons in the mouse dorsal and median raphe nuclei



3D U-Net-based CNN Skeletonization



Coronal Z-projection showing axonal innervation patterns of 6 aligned brains

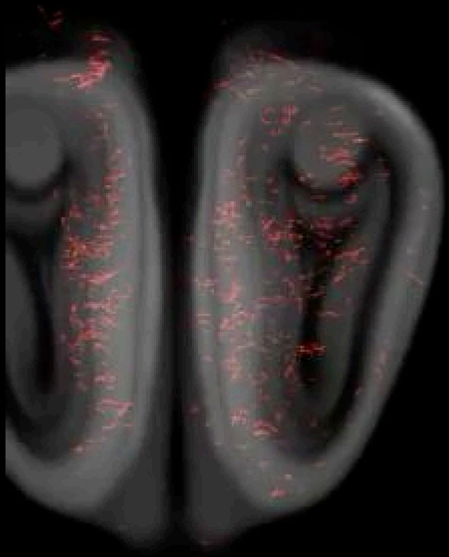


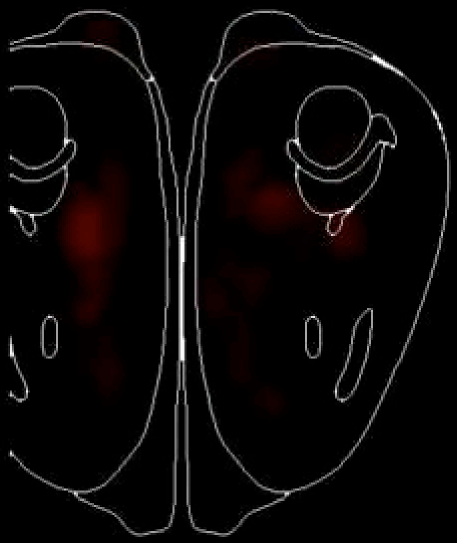
Coronal heatmaps of axonal innervation patterns at 12 positions along the rostral–caudal axis for the same six brains as seen in the image before.

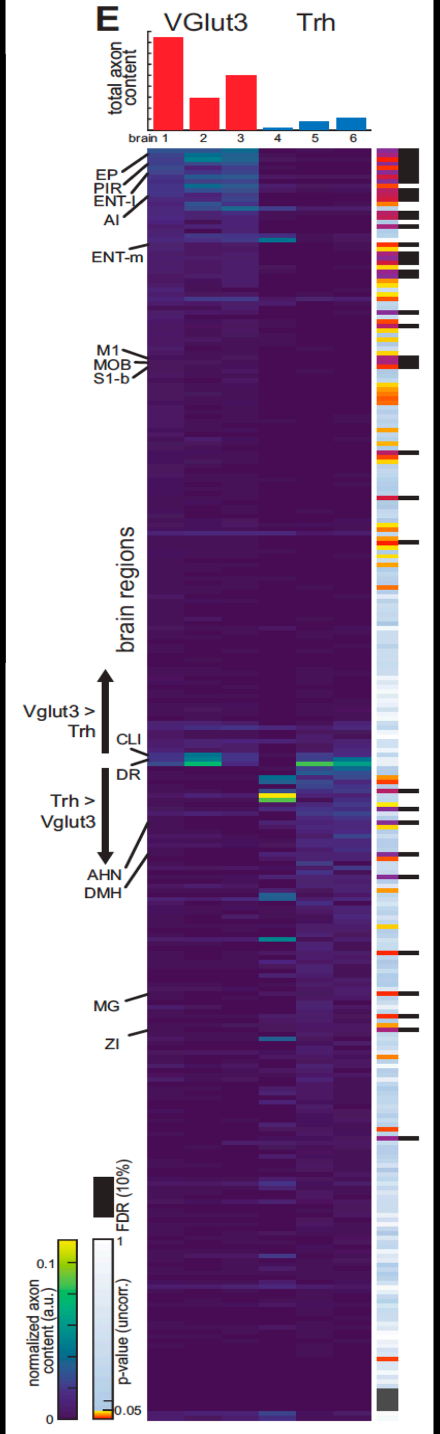
Weightings for individual voxels represent axonal content within a radius of 225 μm.

Strong segregation of axonal projection patterns between 5HT$^+$-VGlut3$^+$ and 5HT$^+$- Trh$^+$ in cDRN

Vglut3^{+} axons project preferentially in anterolateral cortical regions and adjacent structures
Trh^{+} axons preferentially project in thalamic and hypothalamic regions







TRAILMAP on-going in-house application

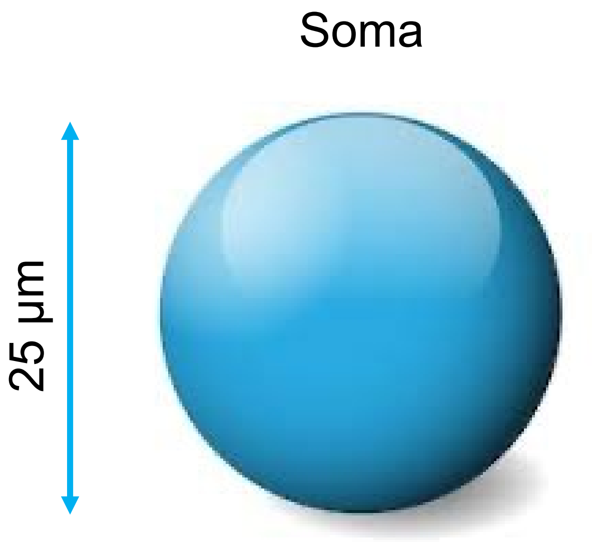
TRAILMAP training for detecting cell-bodies

Pv positive neurons (from Yingjun)

PvCre mouse x Ai6 reporter mouse

As they are larger than axons, I modified the annotation strategy and the script by changing the autogenerated “edge” label from 1 to 2-pixels

Neuron’s soma (diameter) stands from 4-100 μm **BUT** they labeled slices 80 to 180 μm far from each other.



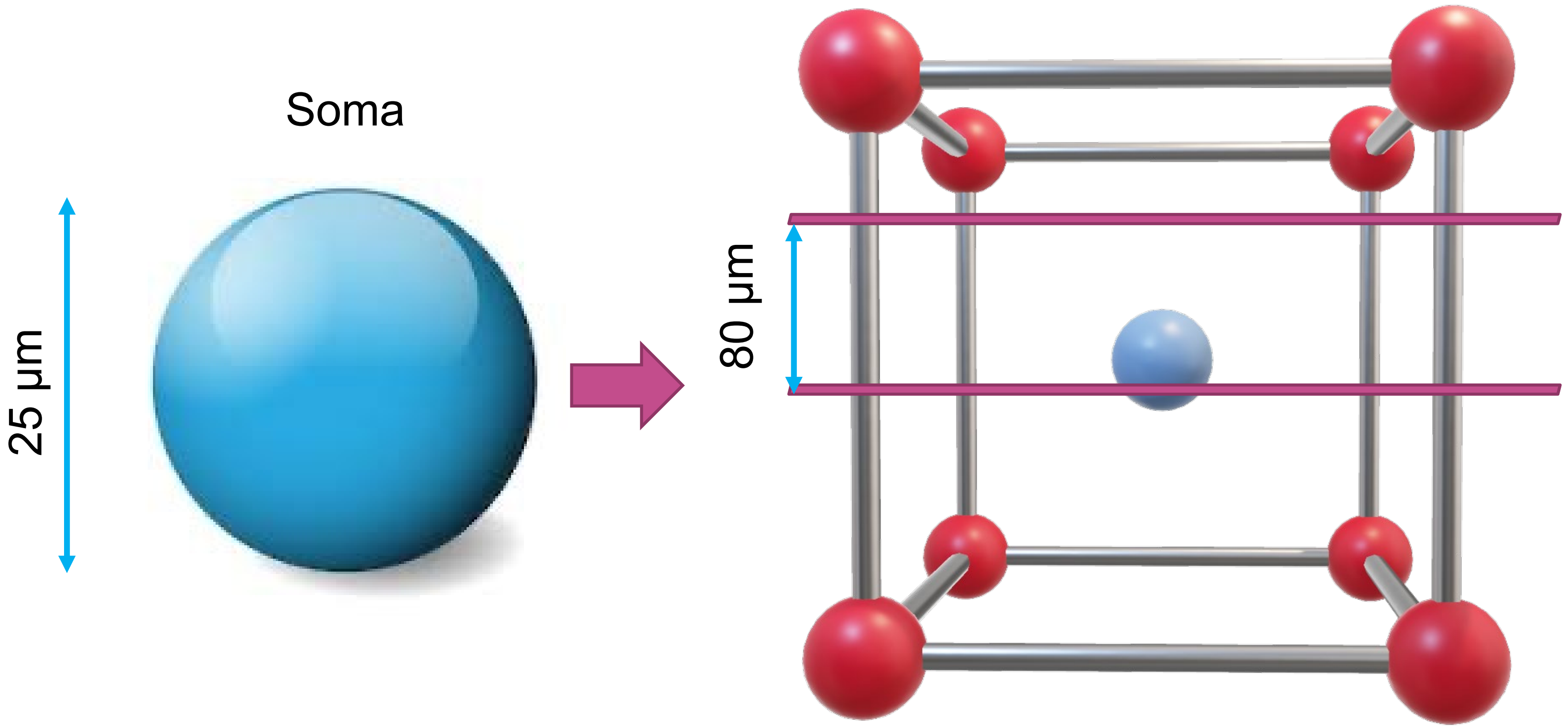
TRAILMAP training for detecting cell-bodies

Pv positive neurons (from Yingjun)

PvCre mouse x Ai6 reporter mouse

As they are larger than axons, I modified the annotation strategy and the script by changing the autogenerated "edge" label from 1 to 2-pixels

Neuron's soma (diameter) stands from 4-100 μm **BUT** they labeled slices 80 to 180 μm far from each other.



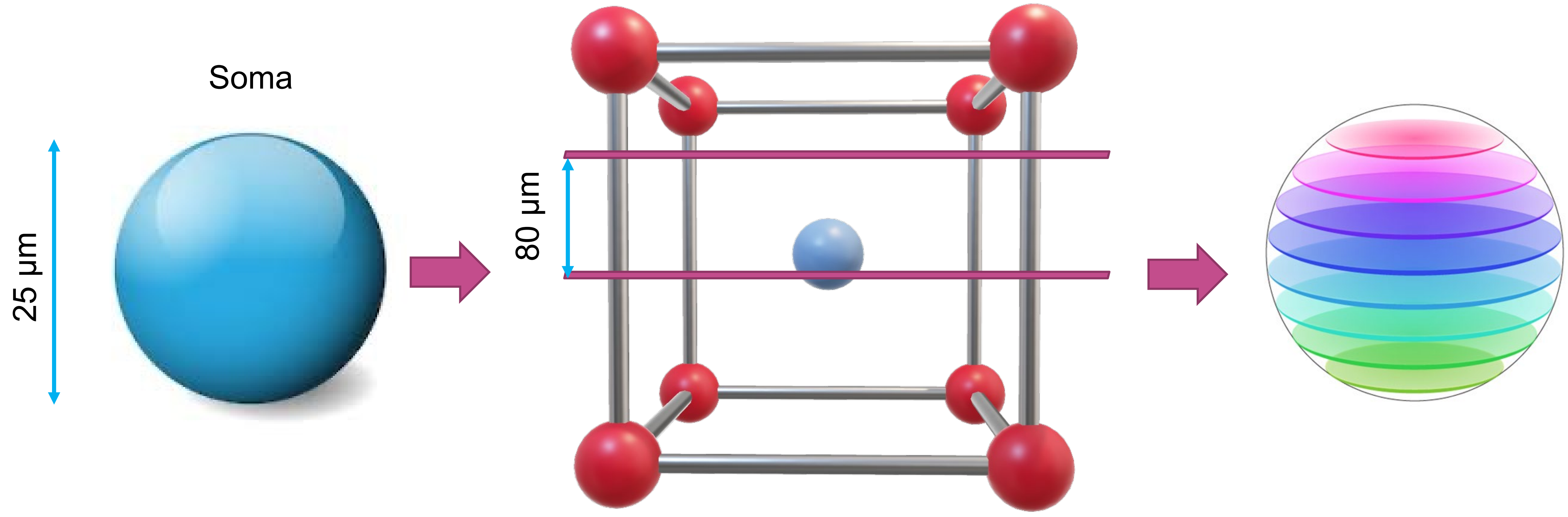
TRAILMAP training for detecting cell-bodies

Pv positive neurons (from Yingjun)

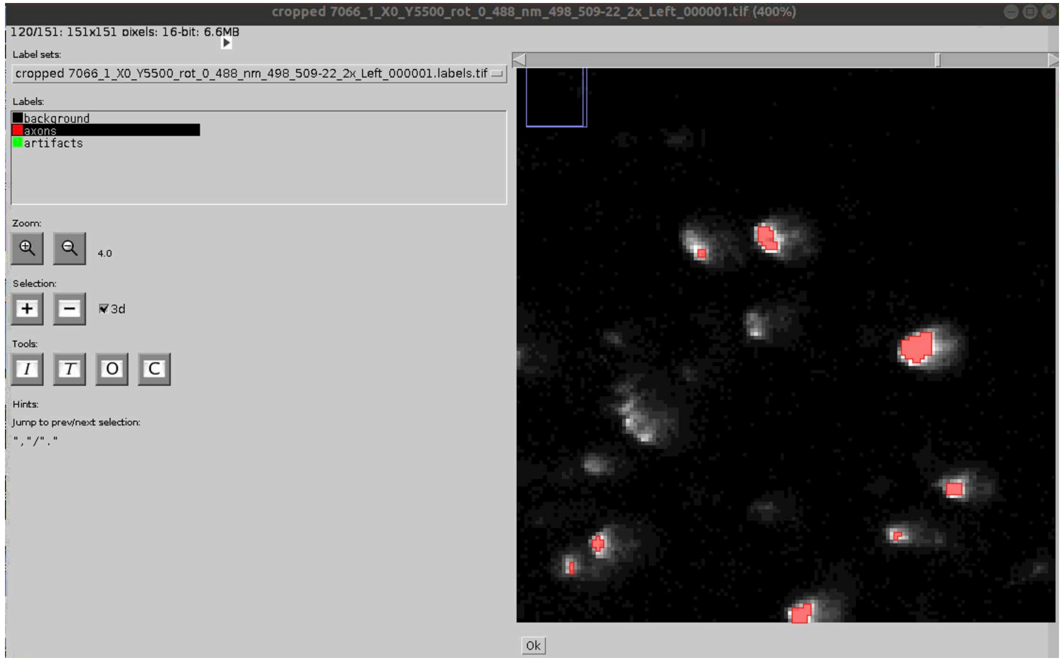
PvCre mouse x Ai6 reporter mouse

As they are larger than axons, I modified the annotation strategy and the script by changing the autogenerated “edge” label from 1 to 2-pixels

Neuron’s soma (diameter) stands from 4-100 µm **BUT** they labeled slices 80 to 180 µm far from each other.



(1) SEGMENTATION EDITOR PLUGIN



cropped 7066_1_X0_Y5500_rot_0_488_nm_498_509-22_2x_Left_000001.tif (400%)

120/151: 151x151 pixels: 16-bit: 6.6MB

Label sets: cropped 7066_1_X0_Y5500_rot_0_488_nm_498_509-22_2x_Left_000001.labels.tif

Labels:

- ■ background
- 🟥 axons
- 🟩 artifacts

Zoom: 4.0

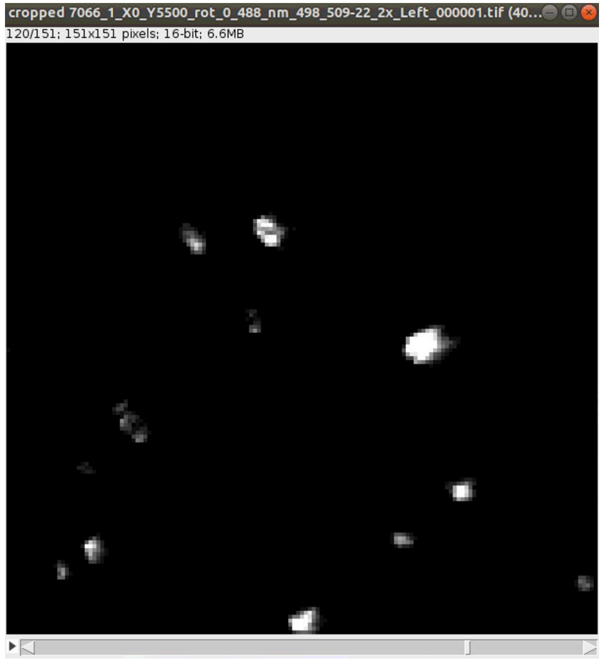
Selection: ☑ 3d

Hints:

Jump to prev/next selection:

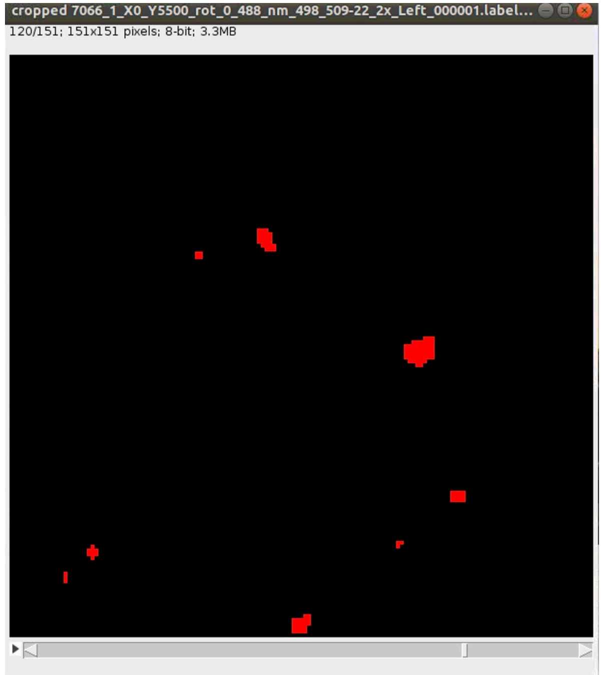
",","/","."

within training-original/volumes



cropped 7066_1_X0_Y5500_rot_0_488_nm_498_509-22_2x_Left_000001.tif (40...

120/151; 151x151 pixels; 16-bit; 6.6MB

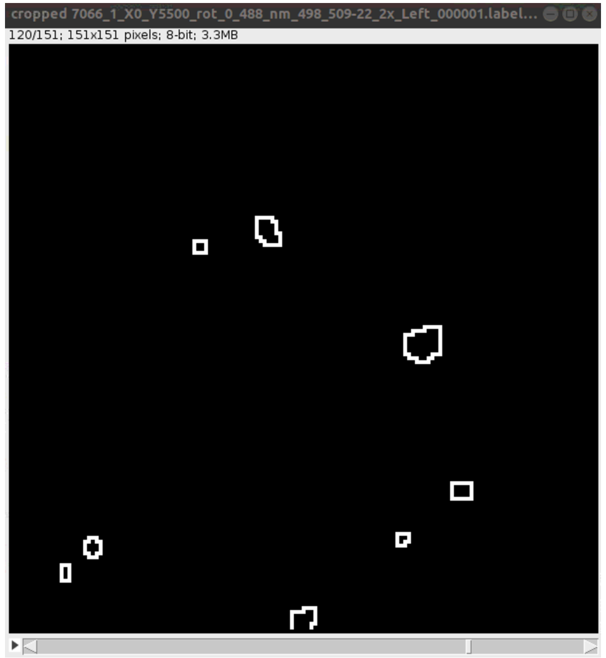


cropped 7066_1_X0_Y5500_rot_0_488_nm_498_509-22_2x_Left_000001.label...

120/151; 151x151 pixels; 8-bit; 3.3MB

within training-original/temp

→



cropped 7066_1_X0_Y5500_rot_0_488_nm_498_509-22_2x_Left_000001.label...

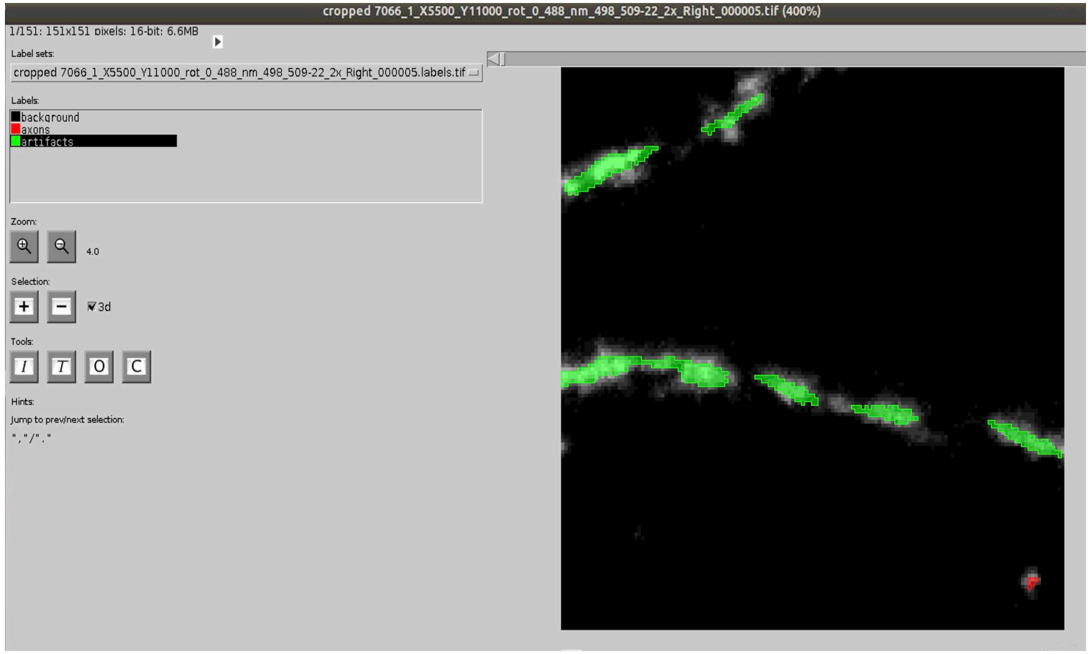
120/151; 151x151 pixels; 8-bit; 3.3MB

within training-original/processed-labels

↑

Copied within training-original/labels

(1) SEGMENTATION EDITOR PLUGIN



cropped 7066_1_X5500_Y11000_rot_0_488_nm_498_509-22_2x_Right_000005.tif (400%)

1/151: 151x151 pixels: 16-bit: 6.6MB

Label sets: cropped 7066_1_X5500_Y11000_rot_0_488_nm_498_509-22_2x_Right_000005.labels.tif

Labels:
- 🟥 background
- 🟥 axons
- 🟩 artifacts

Zoom: 4.0

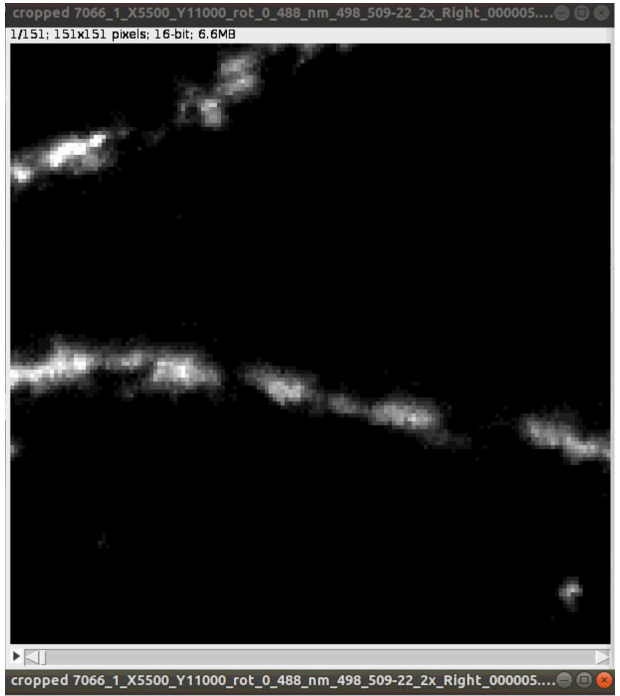
Selection: ☑ 3d

Hints:

Jump to prev/next selection:

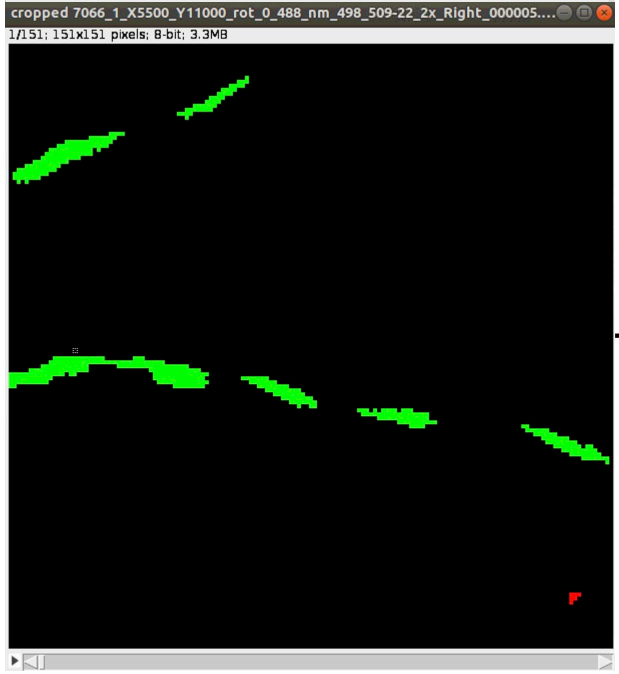
",","/"."

within training-original/volumes



cropped 7066_1_X5500_Y11000_rot_0_488_nm_498_509-22_2x_Right_000005....

1/151; 151x151 pixels; 16-bit; 6.6MB

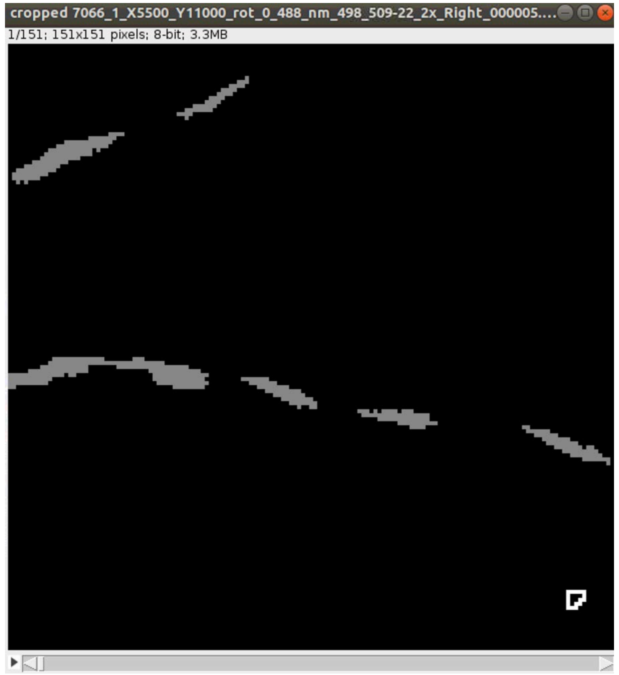


cropped 7066_1_X5500_Y11000_rot_0_488_nm_498_509-22_2x_Right_000005....

1/151; 151x151 pixels; 8-bit; 3.3MB

within training-original/temp

Copied within training-original/labels



cropped 7066_1_X5500_Y11000_rot_0_488_nm_498_509-22_2x_Right_000005....

1/151; 151x151 pixels; 8-bit; 3.3MB

within training-original/processed-labels

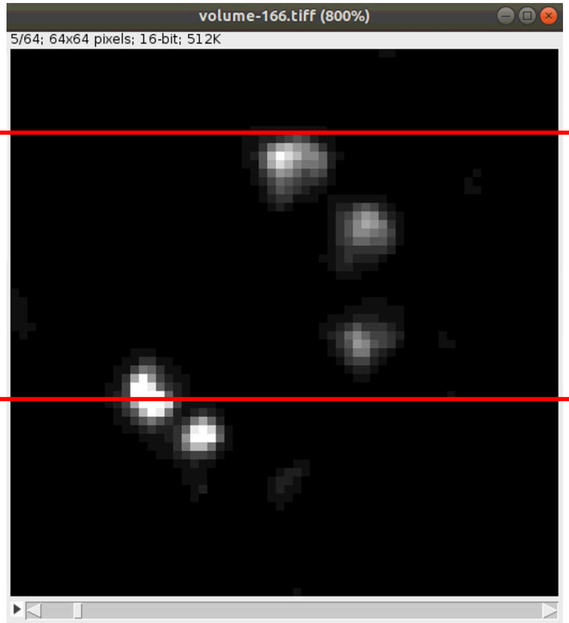
(2) TRAINING-SET

within training-set/volumes

within training-set/labels

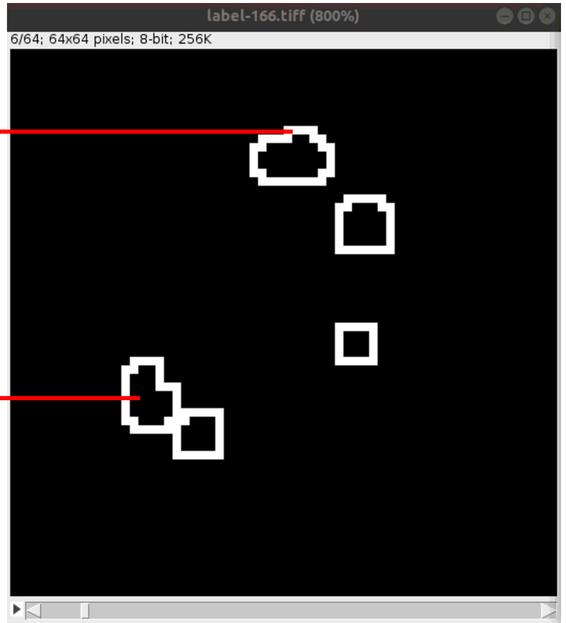
They seem bigger than they should be. Is it fine at this stage?

Why isn't it labelled inside?



volume-166.tiff (800%)

5/64; 64x64 pixels; 16-bit; 512K

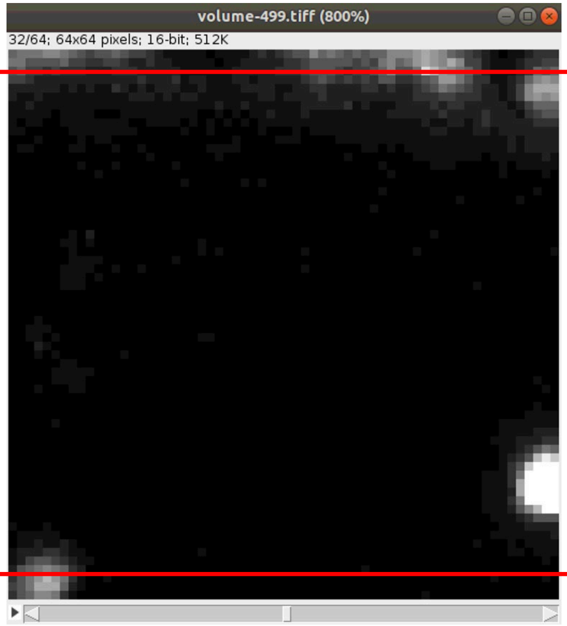


label-166.tiff (800%)

6/64; 64x64 pixels; 8-bit; 256K

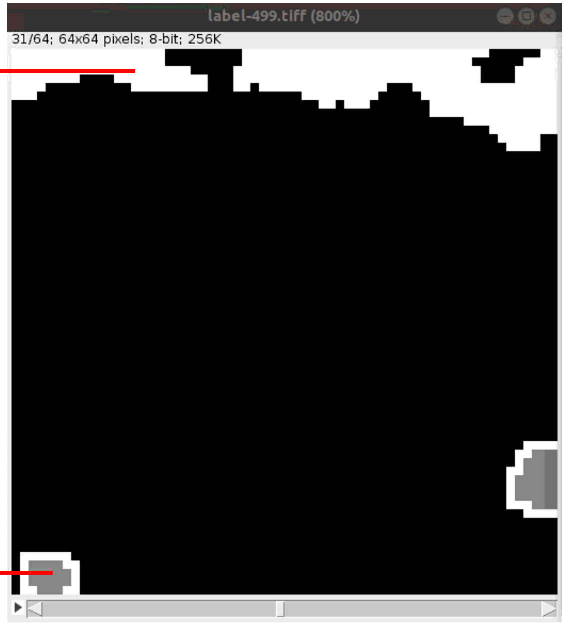
Why are the artifacts still recognized?

Why is it labelled inside with grey and not empty as in the first image?



volume-499.tiff (800%)

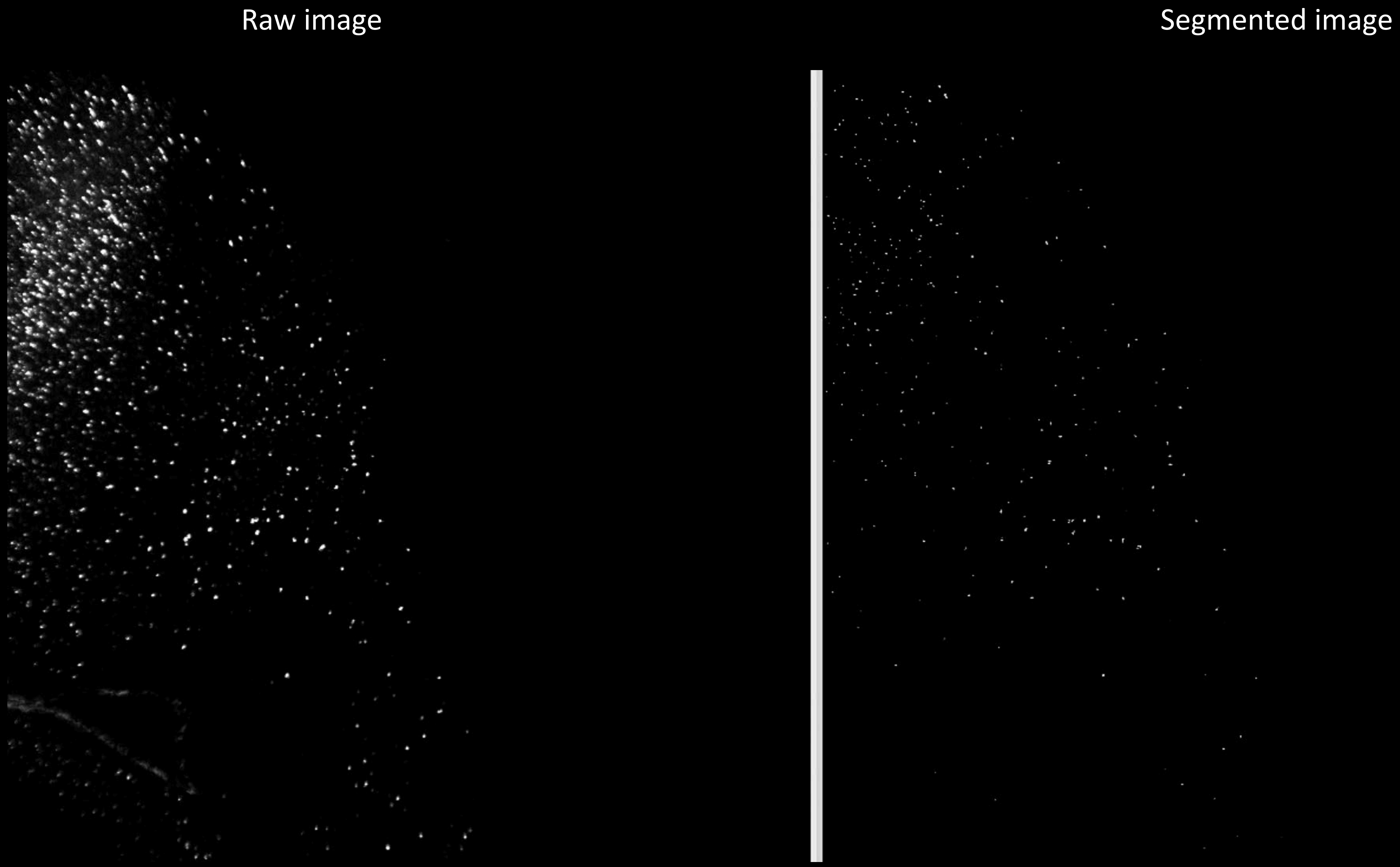
32/64; 64x64 pixels; 16-bit; 512K



label-499.tiff (800%)

31/64; 64x64 pixels; 8-bit; 256K

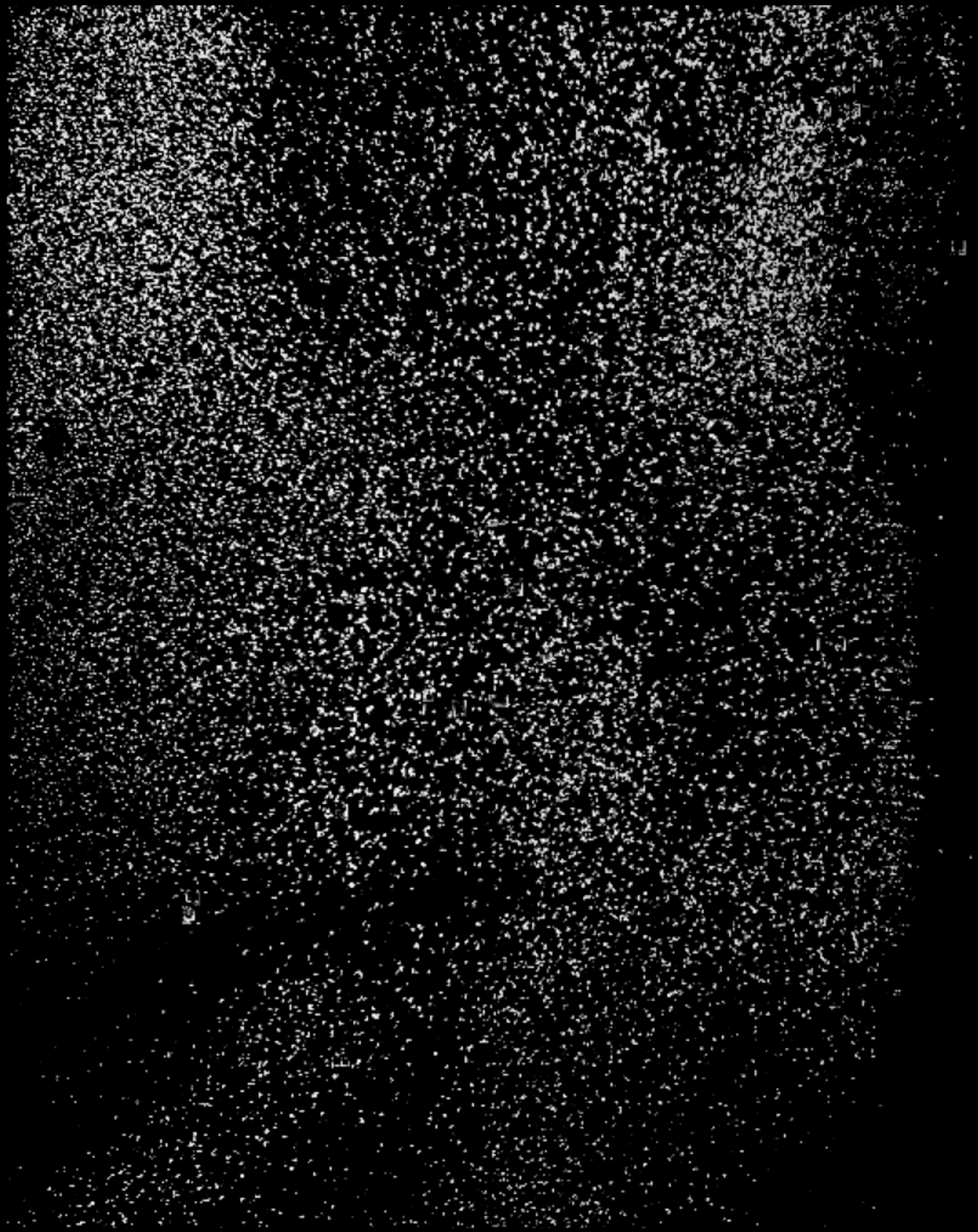
(3) Slice of the brain-TILE of PV-Ai6



Raw image

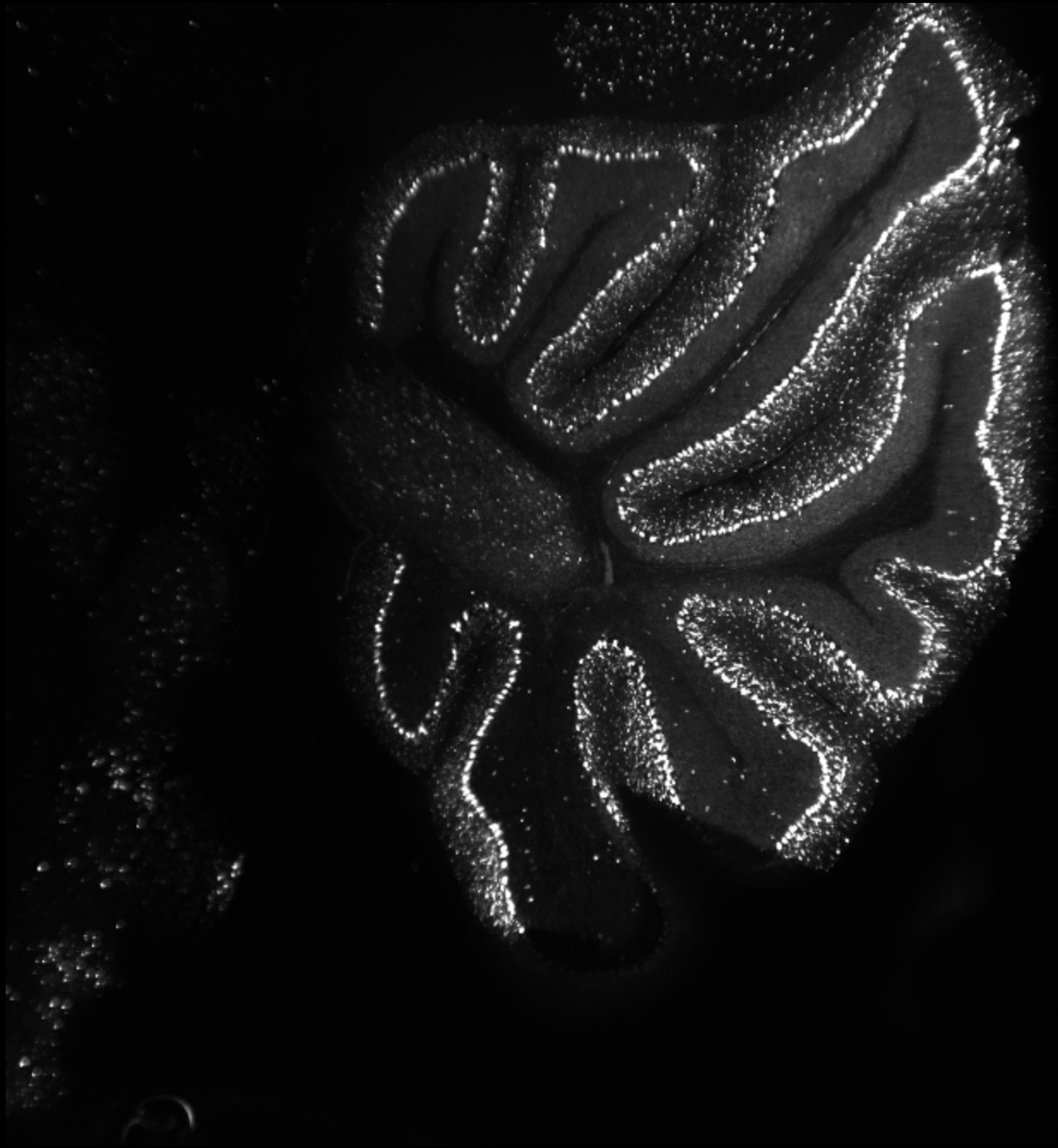
Segmented image

(3) 3D projection TILE OF PV-Ai6

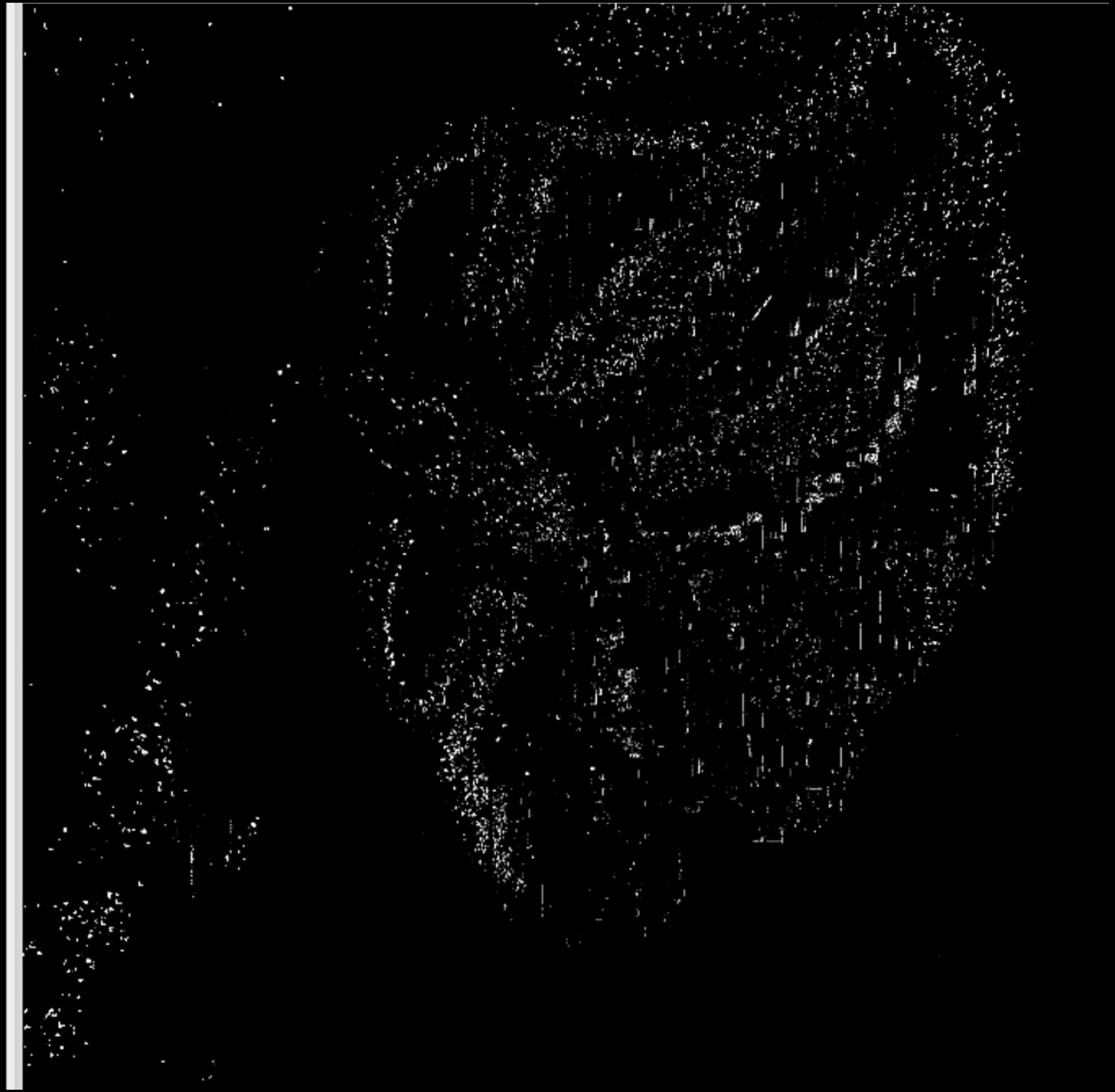


(3) Slice of the brain-TILE of PV-Ai6 - Cerebellum

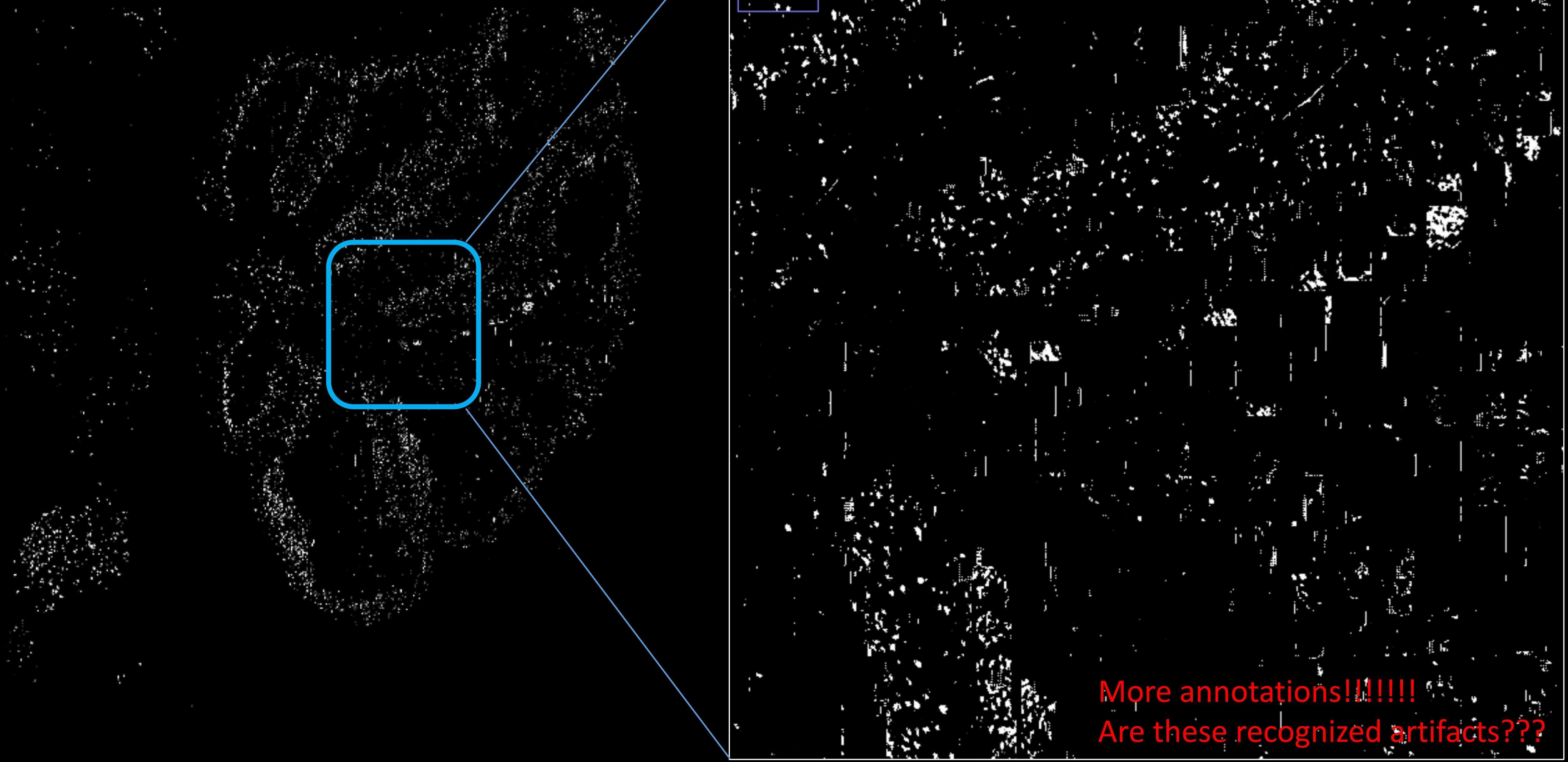
Raw image



Segmented image



(3) 3D maximal projection TILE of PV-Ai6 Cerebellum



More annotations!!!!!!!

Are these recognized artifacts???

CONCLUSION

It reliably extracts mesoscale projectomes rather than reconstructions of axons from sparsely labeled individual neurons

Mitigations of artifacts biases (reduction of false positive)

Thinning strategy allows to construct an armature of predicted axons: it reduces false breaks of dim axonal segments

GitHub repository (AlbertPun/TRAILMAP) is available and comes with a README.txt for easy-to-use applications

THANK YOU SO MUCH!