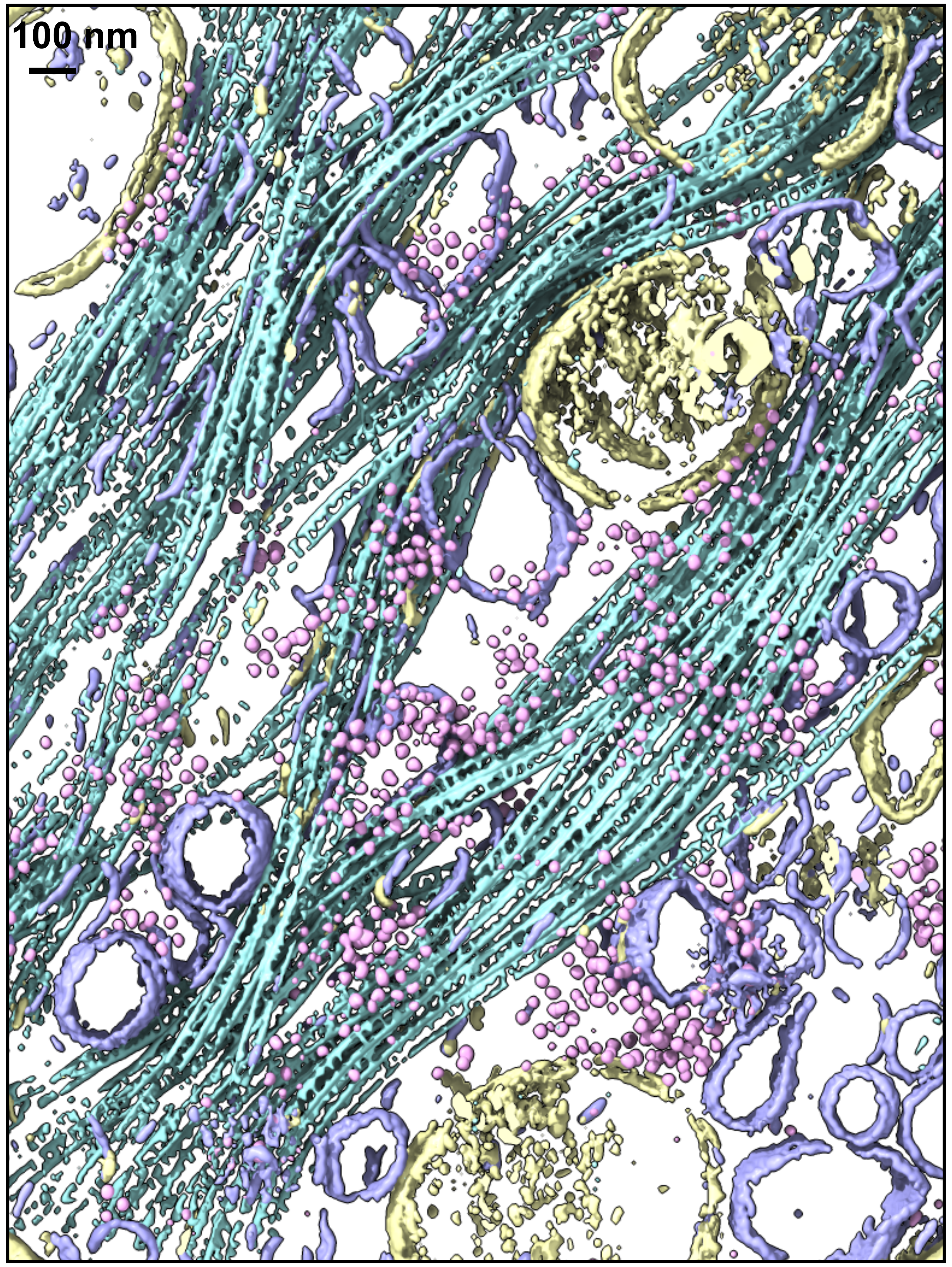
Tomogram annotation

Muyuan Chen
2019-05

Feature annotation in cellular tomograms

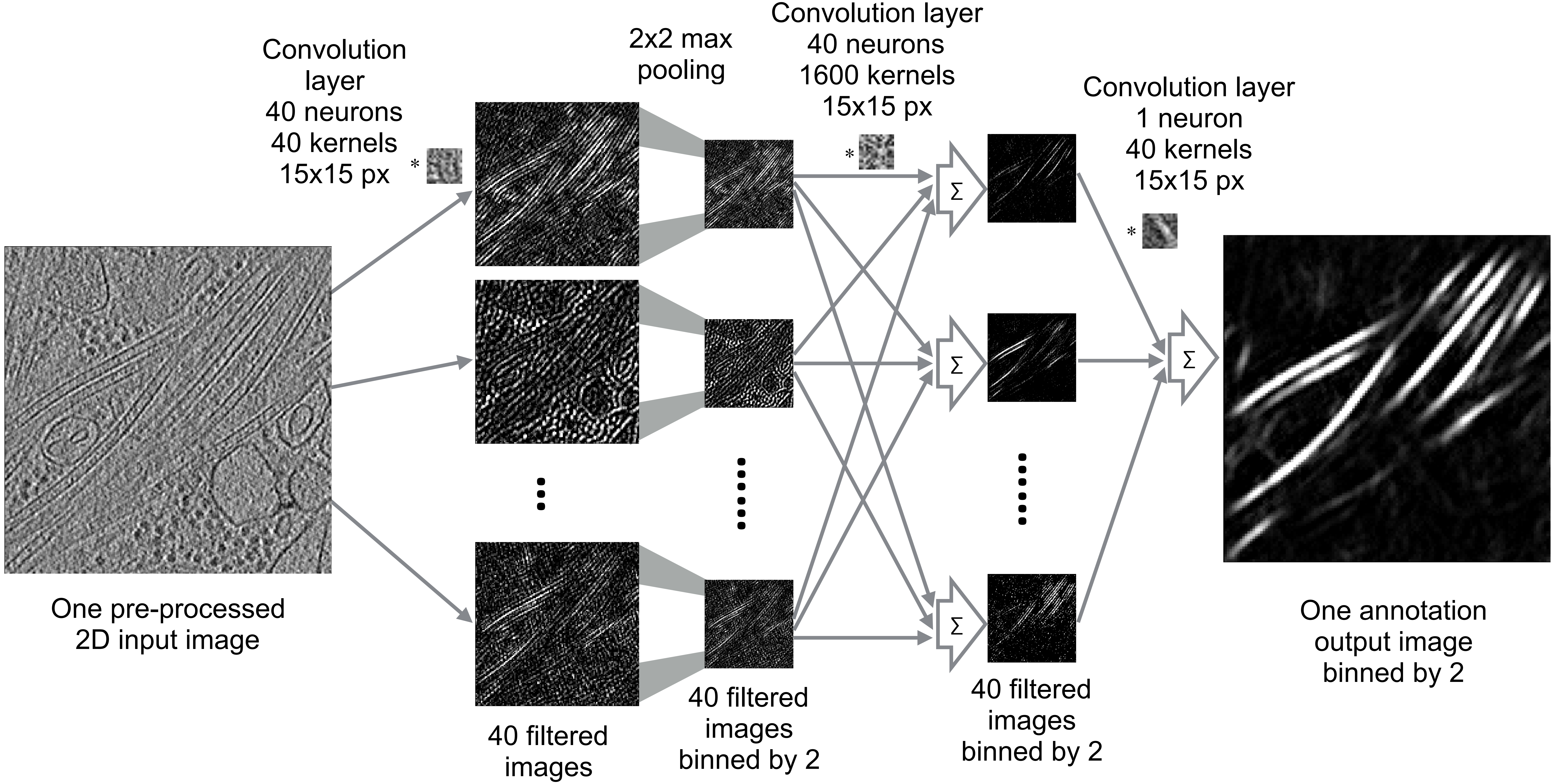


PC12 cell

W. Dai, Rutergers

Convolutional Neural Networks

Convolutional Neural Networks



Convolution layer
40 neurons
40 kernels
15x15 px

2x2 max pooling

Convolution layer
40 neurons
1600 kernels
15x15 px

Convolution layer
1 neuron
40 kernels
15x15 px

One pre-processed 2D input image

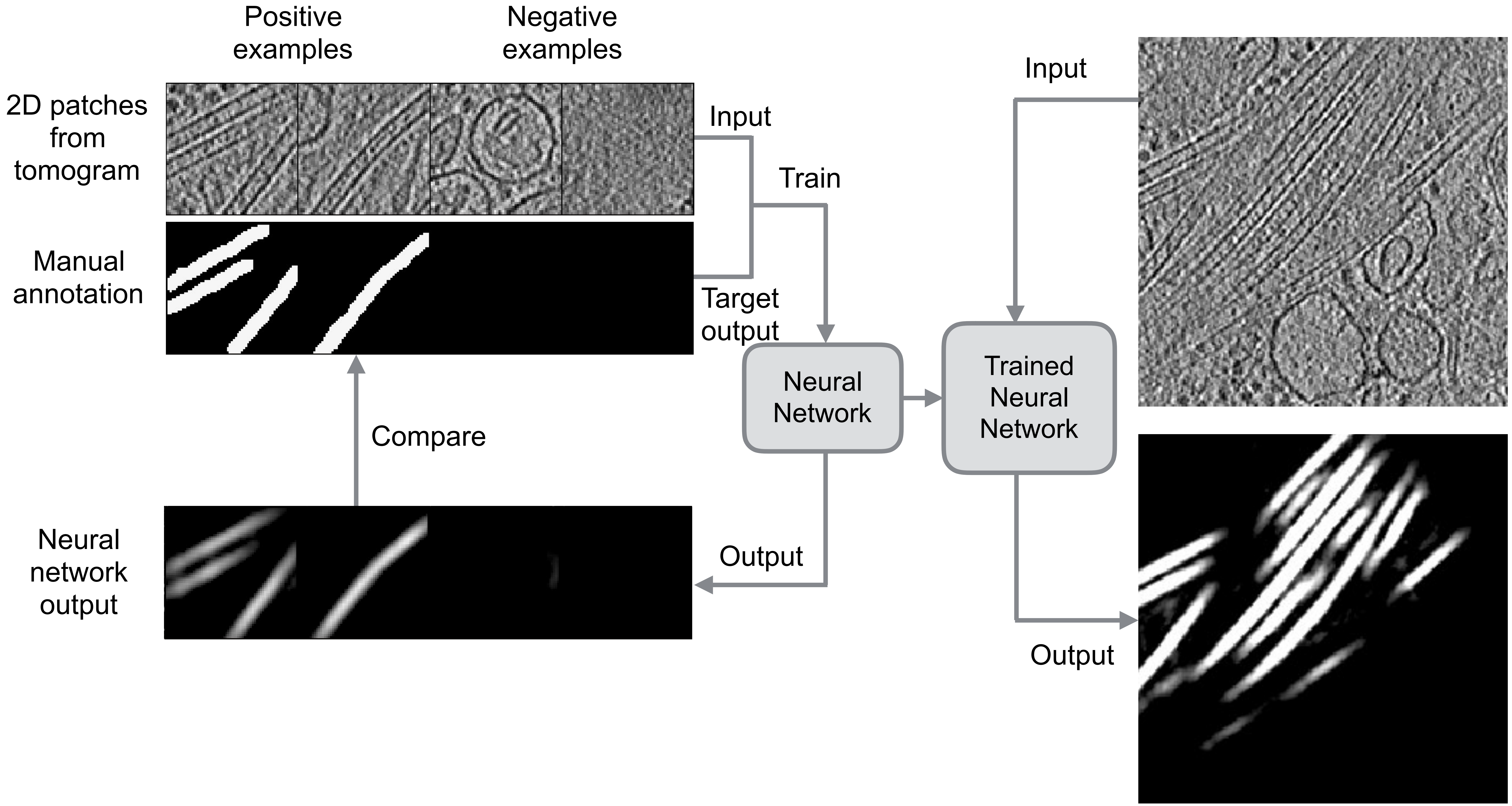
40 filtered images

40 filtered images binned by 2

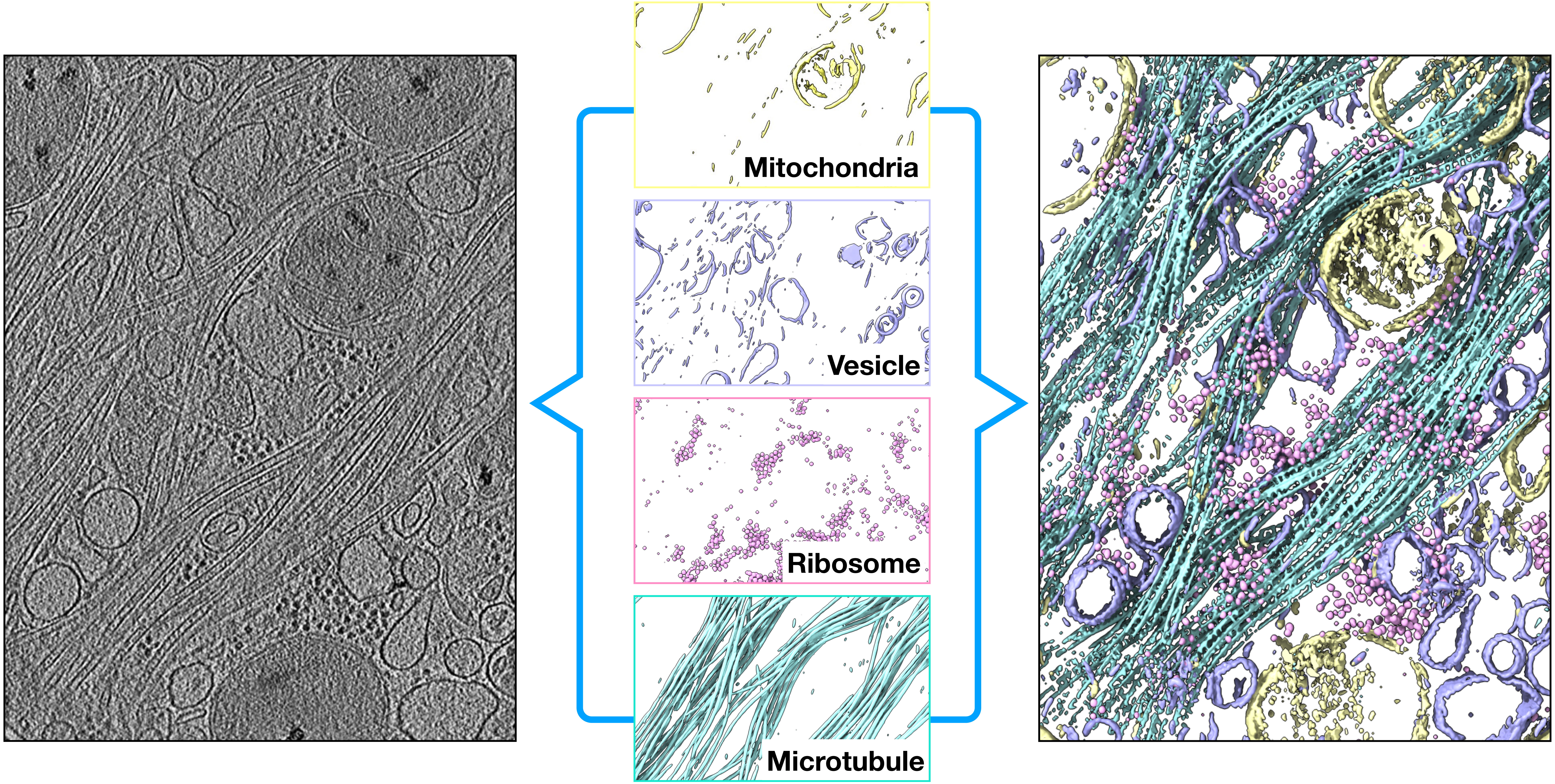
40 filtered images binned by 2

One annotation output image binned by 2

Training of the Neural Network



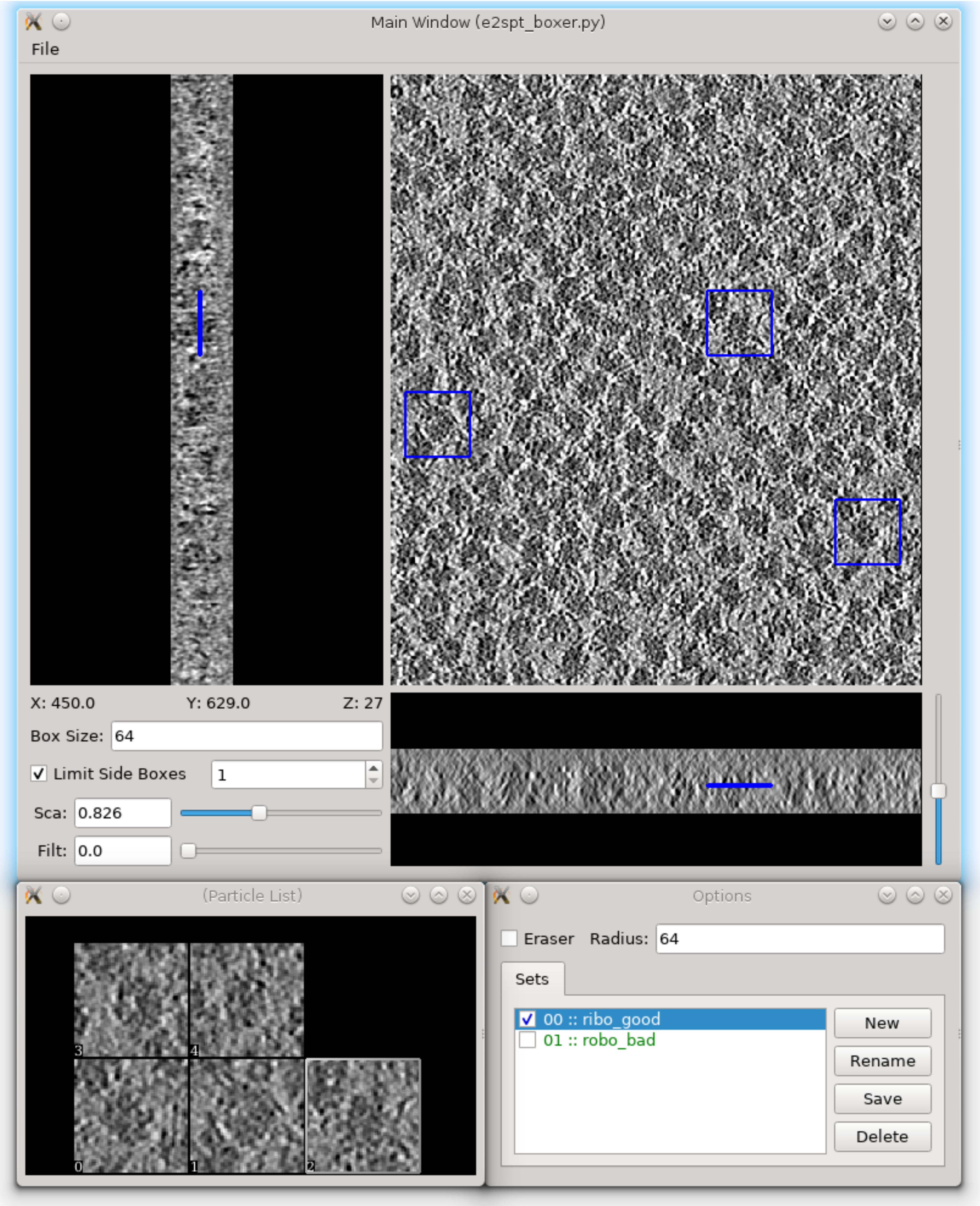
Competitive merging of multiple features



Demo on the ribosome dataset

Select references

- **Segmentation -> Box training references**
- Select one tomogram and click **Launch**
- Select a few ribosome particles (5 is enough in this case), and name the set **ribo_good**
- Click **Save** in the set panel when only **ribo_good** is visible



X: 450.0 Y: 629.0 Z: 27

Box Size: 64

☑ Limit Side Boxes 1

Sca: 0.826

Filt: 0.0

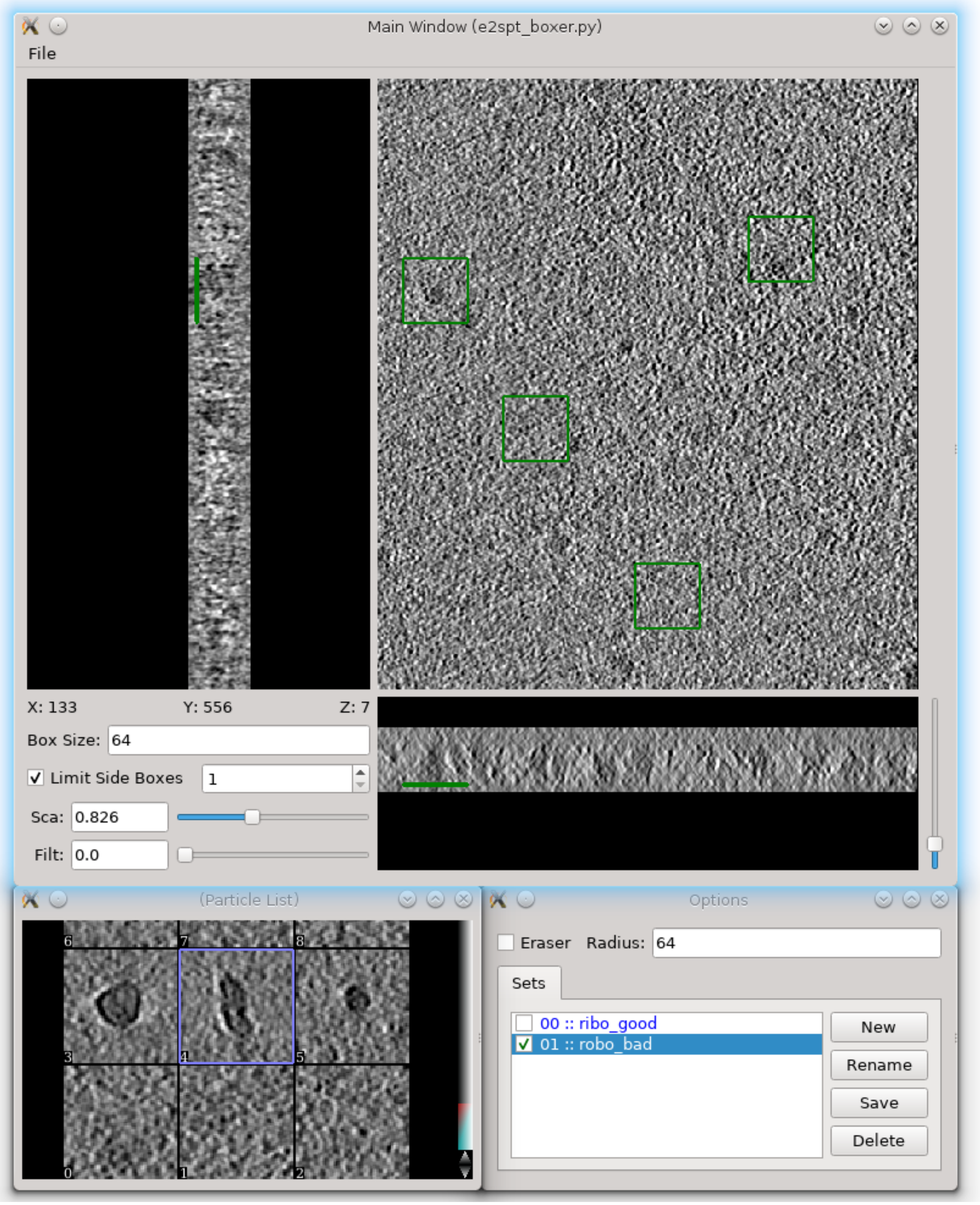
☐ Eraser Radius: 64

☑ 00 :: ribo_good

☐ 01 :: robo_bad

Select references

- Hide the **ribo_good** set and create a new set called **ribo_bad.**
- Select some patches that do not contain ribosomes (here 30 patches are used). Try including more diverse features you would like to exclude.
- Click **Save** in the set panel when only **ribo_bad** is visible


Main Window (e2spt_boxer.py)

X: 133 Y: 556 Z: 7

Box Size: 64

☑ Limit Side Boxes 1

Sca: 0.826

Filt: 0.0

(Particle List)

Options

□ Eraser Radius: 64

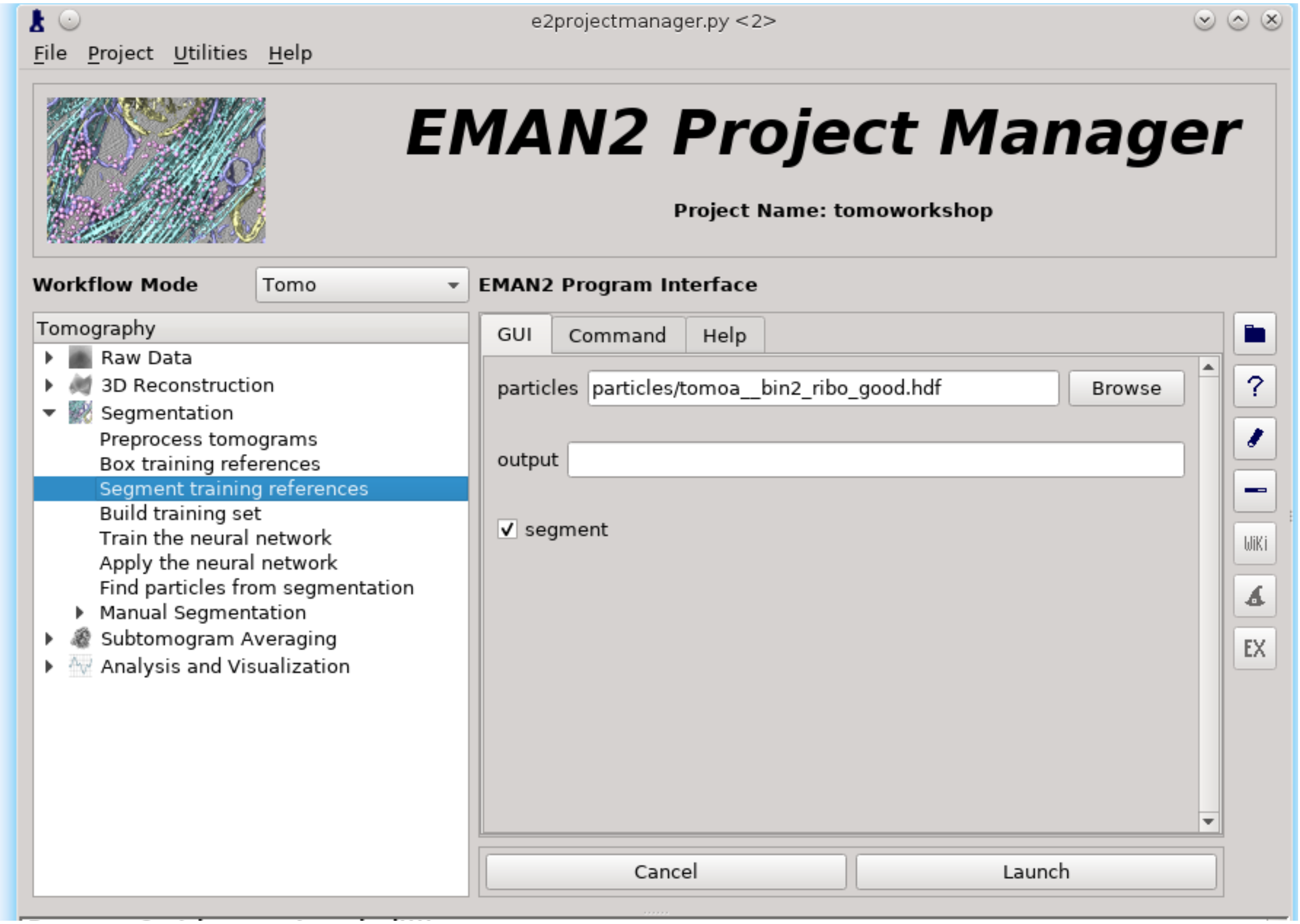
Sets

□ 00 :: ribo_good

☑ 01 :: robo_bad

Segment training references

- **Segmentation -> Segment training references**
- Select the particle file ends with **bin2_ribo_good.hdf**
- Click Launch
- In the new window, manually paint the ribosome density white



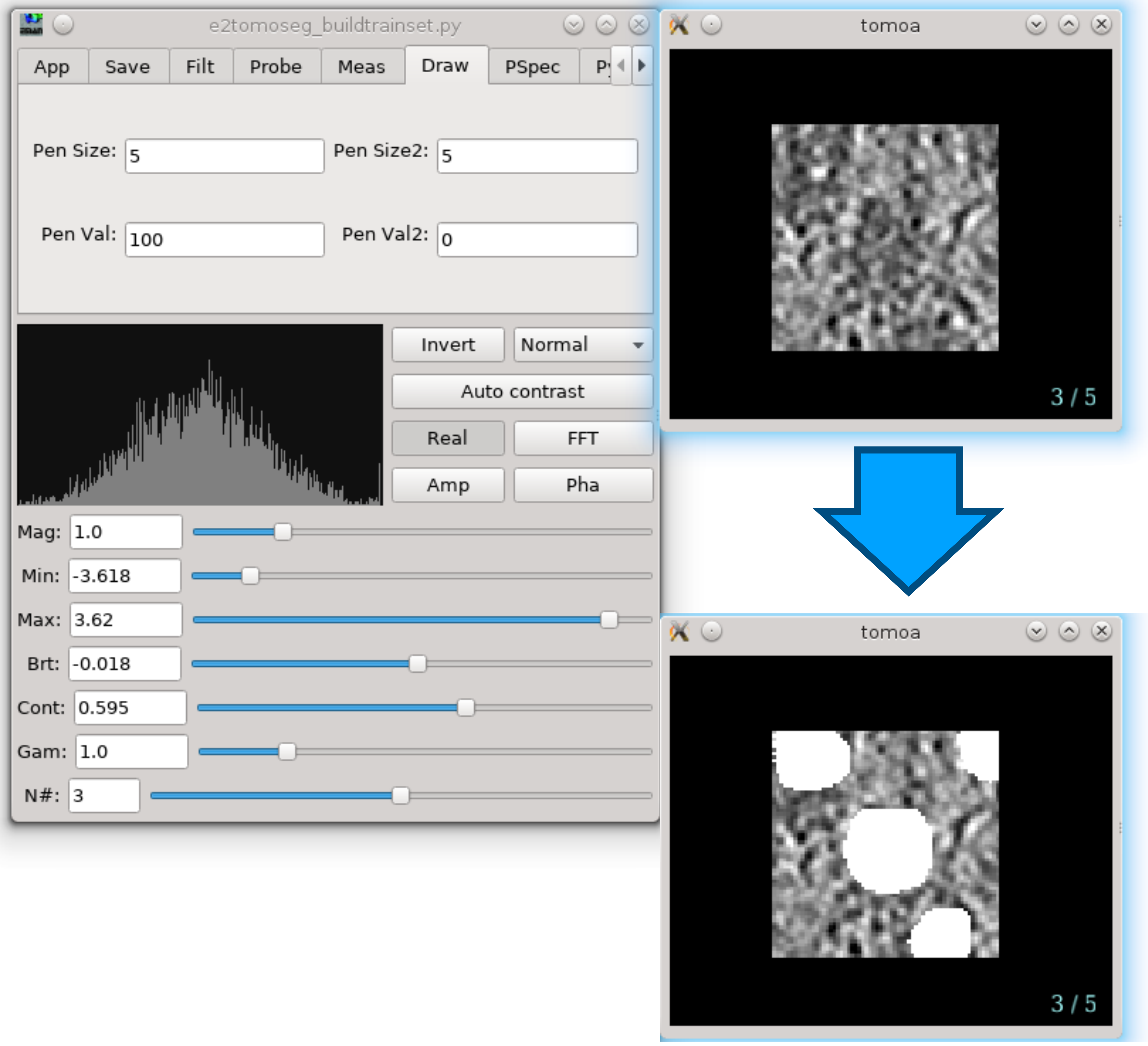
particles: particles/tomoa__bin2_ribo_good.hdf

output:

☑ segment

Segment training references

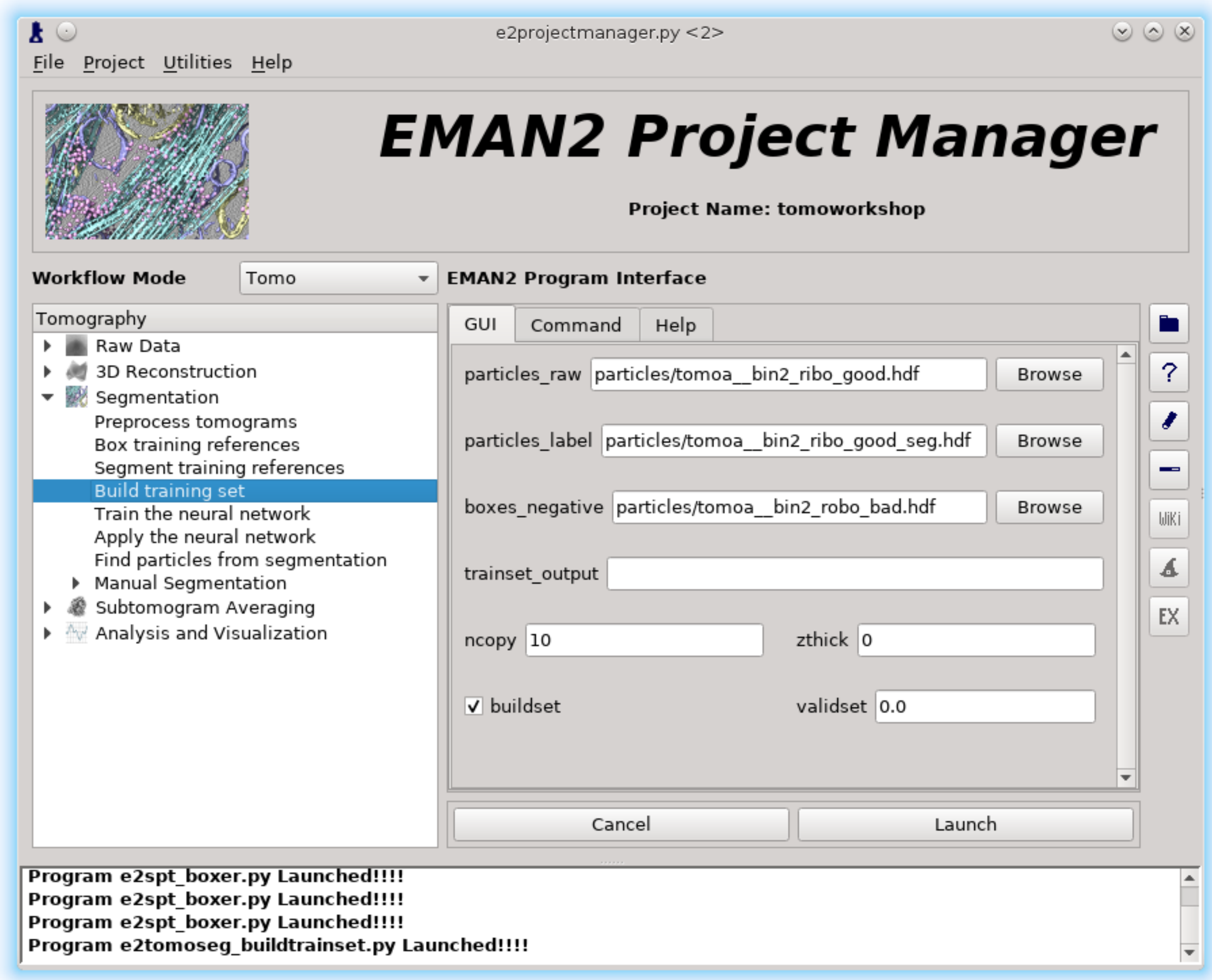
- **Segmentation -> Segment training references**
- Select the particle file ends with **bin2_ribo_good.hdf**
- Click Launch
- In the new window, manually paint the ribosome density white



Pen Size: 5
Pen Size2: 5
Pen Val: 100
Pen Val2: 0
Mag: 1.0
Min: -3.618
Max: 3.62
Brt: -0.018
Cont: 0.595
Gam: 1.0
N#: 3

Build training set

- **Segmentation -> Build training set**
- In **particles_raw**, **particles_label** and **boxes_negative**, select the files with tag “**ribo_good**”, “**ribo_good_seg**”, and “**ribo_bad**” correspondingly.
- Click **Launch**



particles_raw: particles/tomoa__bin2_ribo_good.hdf

particles_label: particles/tomoa__bin2_ribo_good_seg.hdf

boxes_negative: particles/tomoa__bin2_robo_bad.hdf

trainset_output:

ncopy: 10

zthick: 0

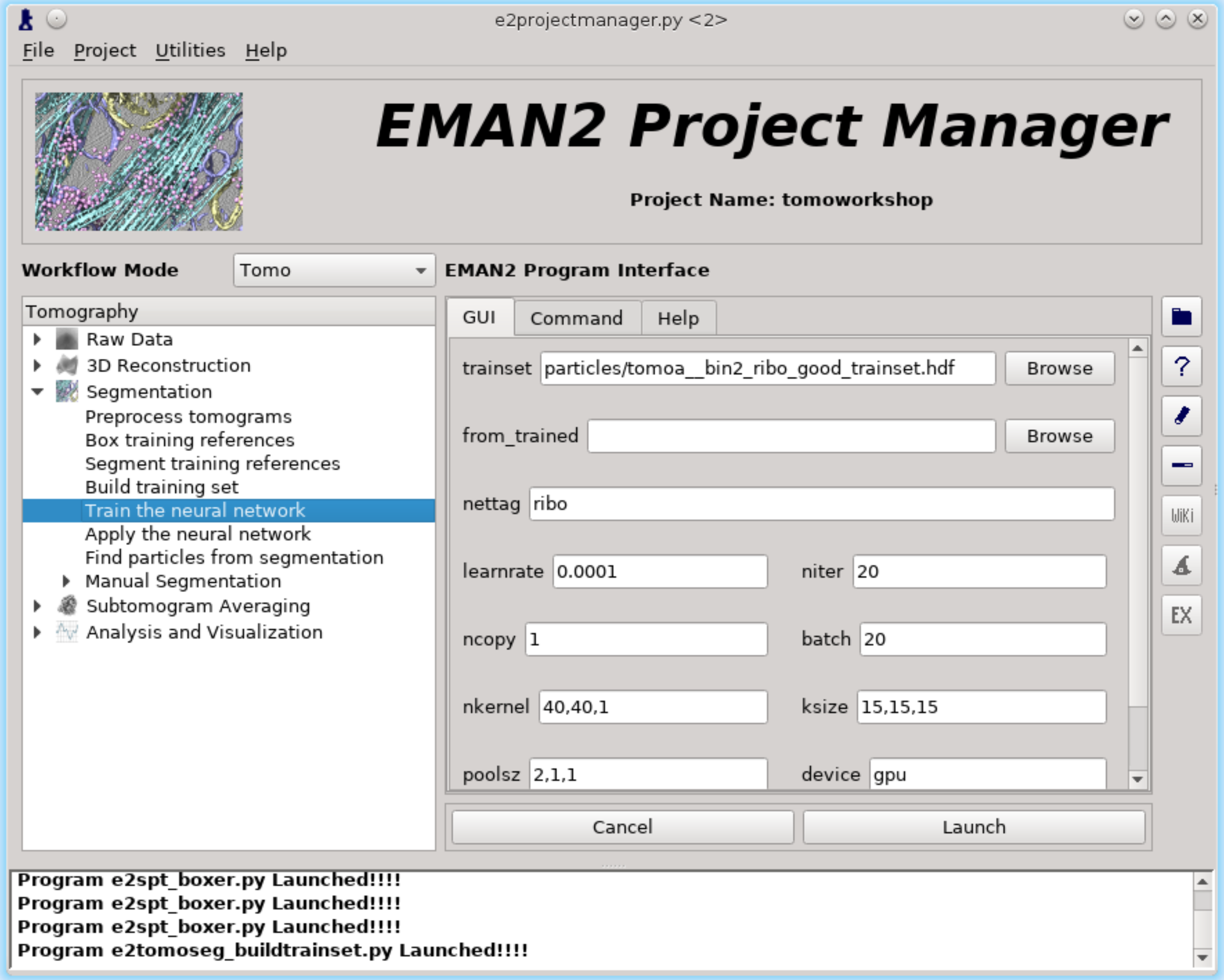
☑ buildset

validset: 0.0

```
Program e2spt_boxer.py Launched!!!!
Program e2spt_boxer.py Launched!!!!
Program e2spt_boxer.py Launched!!!!
Program e2tomoseg_buildtrainset.py Launched!!!!
```

Train neural network

- **Segmentation -> Train neural network**
- Select the file ends with “**_trainset.hdf**” for **trainset**
- Set **nettag** to **ribo**
- Click **Launch**



trainset particles/tomoa__bin2_ribo_good_trainset.hdf

from_trained

nettag ribo

learnrate 0.0001

niter 20

ncopy 1

batch 20

nkernel 40,40,1

ksize 15,15,15

poolsz 2,1,1

device gpu

```
Program e2spt_boxer.py Launched!!!!
Program e2spt_boxer.py Launched!!!!
Program e2spt_boxer.py Launched!!!!
Program e2tomoseg_buildtrainset.py Launched!!!!
```

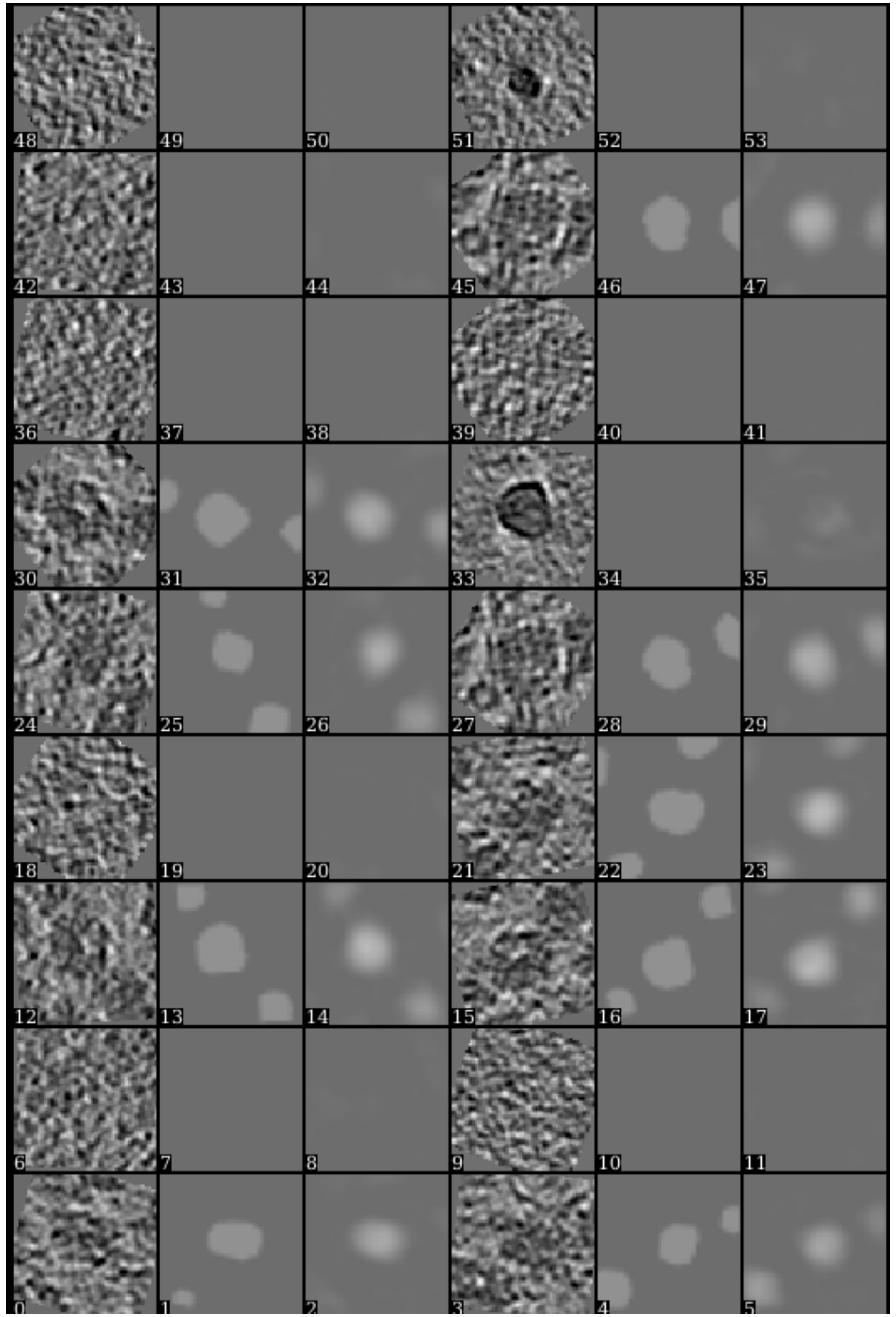
Train neural network

- To look at the training result, open **neuralnets/trainout_nnet_save__ribo.hdf** from the e2display browser

- Every three images in the stack are input 2D patch, manual annotation and neural network output.

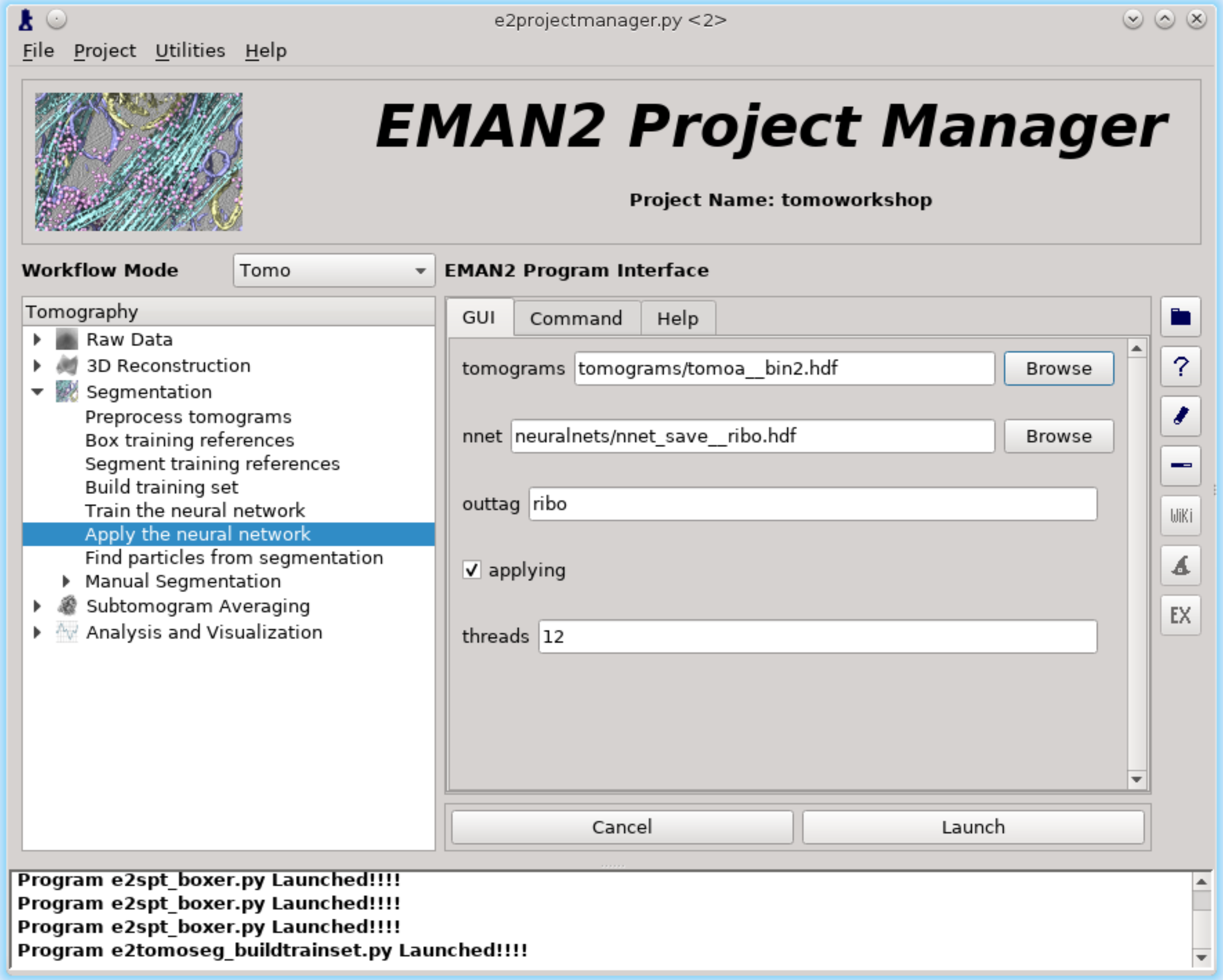
- Ideally the third image should be similar to the second one.

- The actual network content is saved in **neuralnets/nnet_save__ribo.hdf**, which does not make much senses to human eyes.



Apply neural network

- **Segmentation -> Apply neural network**
- Select a tomogram and the network we just trained (**neuralnets/nnet_save__ribo.hdf**)
- Set **outtag** to **ribo** and click **Launch**
- Check **Segmentations/ tomoa__ribo.hdf** for results.



tomograms tomograms/tomoa__bin2.hdf

nnet neuralnets/nnet_save__ribo.hdf

outtag ribo

☑ applying

threads 12

```
Program e2spt_boxer.py Launched!!!!
Program e2spt_boxer.py Launched!!!!
Program e2spt_boxer.py Launched!!!!
Program e2tomoseg_buildtrainset.py Launched!!!!
```

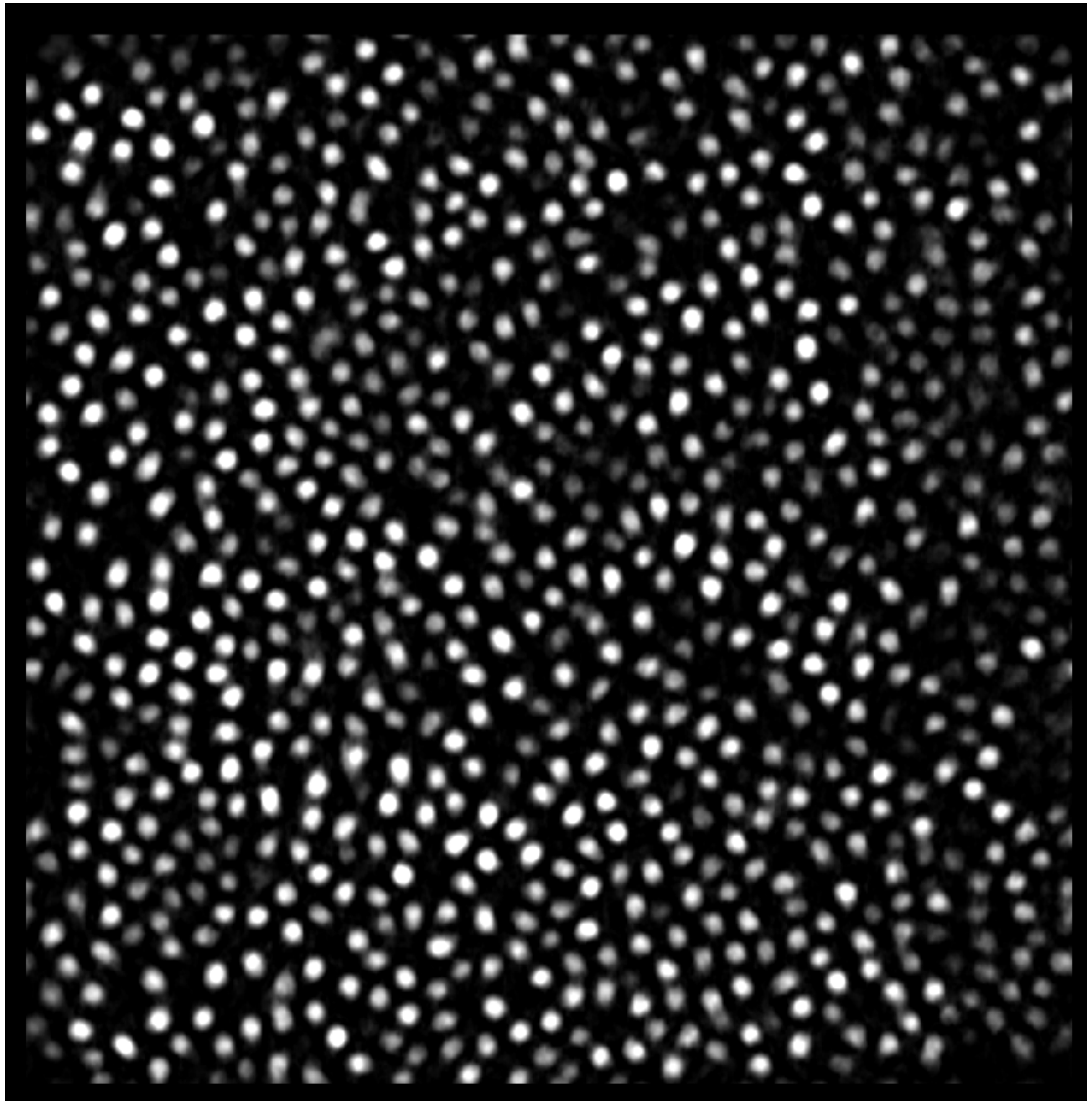
Apply neural network

- **Segmentation -> Apply neural network**

- Select a tomogram and the network we just trained (**neuralnets/nnet_save__ribo.hdf**)

- Set **outtag** to **ribo** and click **Launch**

- Check **Segmentations/ tomoa__ribo.hdf** for results.



Find particles from segmentations

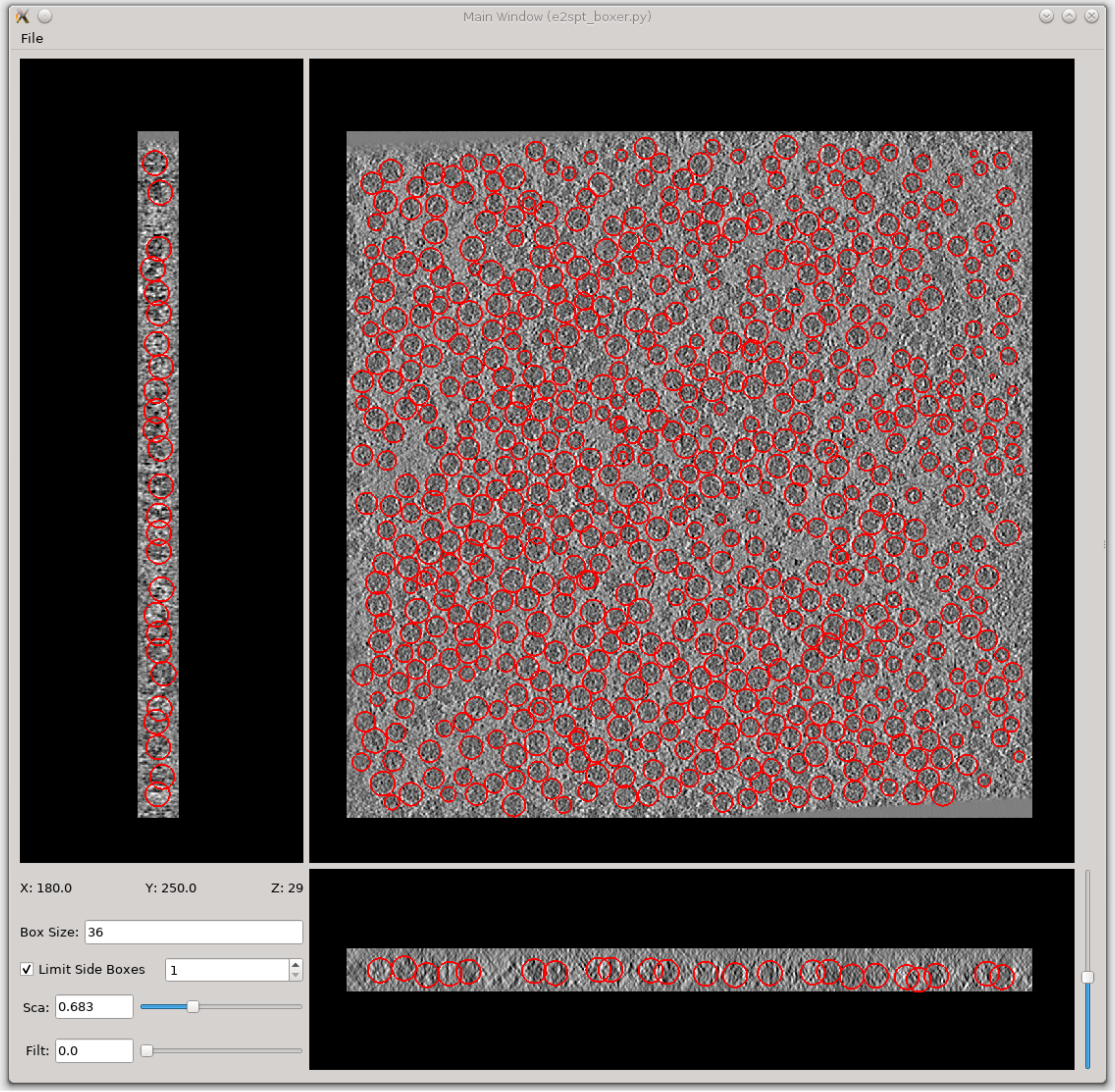
- **Segmentation -> Find particles from segmentation**

- Select the tomogram and corresponding segmentation

- Set **featurename** to **ribo_nn** (since the **ribo** label has been used during template matching)

- Click **Launch**

- Check the particles from the boxer window



X: 180.0 Y: 250.0 Z: 29

Box Size: 36

☑ Limit Side Boxes 1

Sca: 0.683

Filt: 0.0