**Part 1: Make manual segmentation**

1. Make two folders in a directory: “Original” and “ROI”
2. Save the images for segmentation in a folder named “Original”
3. Open one image in ImageJ (version 1.51w)
4. Zoom in on linker upper corner, e.g. with the arrows on your keyboard. You can move the window that is being shown by pressing the space bar and drag the window in the preferred direction.
5. Select ‘Freehand selections’ in the ImageJ default icon menu (4th from the left):Graphical user interface, text, application

   Description automatically generated
6. Encircle one nucleus as precise as you can: Graphical user interface, text

   Description automatically generated
7. Click the <t> key on your keyboard, to add this Region of Interest (ROI) to the ROI manager
8. Now, the ROI manager appears with your first item. To save your ROIs, click More>>Save.. to choose the “ROI” directory you created in step 1 to save your ROIs. The default name is RoiSet.zip. **Rename the set such that it also contains your image number at the start: <image\_nr>\_RoiSet.zip, e.g. 001\_RoiSet.zip.**Graphical user interface, application, table

   Description automatically generated
9. Make one RoiSet per image!

**Part 2: Create segmentation maps with ImageJ macro**

1. Open image in ImageJ (version 1.51w)
2. Add an empty folder named “Original”. The folder structure then becomes:  
   Trainingset
   1. Original
   2. ROI
   3. Segmentation
3. Open 00\_Crop\_original.ijm in ImageJ
4. Click Run
5. In the first pop-up, select the input directory “Original”
6. In the second pop-up, select the (still empty) output directory “Segmentation”
7. Close the macro and check whether cropped\_original now appeared in the segmentation folder
8. Open 00\_Crop\_ROI.ijm in ImageJ and run it
9. In the first pop-up, select the input directory “ROI”
10. In the second pop-up, select again the output directory “Segmentation”
11. In the last pop-up, select one original image as input, e.g. the first image in the Original folder
12. Close the macro and check whether you have greyscale segmentation images in the folder cropped\_greyscale

**Part 3: Train model**

1. Open a browser and go to google colab.
2. Upload the script instructions.ipynb and open it
3. Create four folders in the Files section: original, segmentation, predict, output
4. Upload the pre-trained model cell\_seg\_model\_001.pth
5. Upload the files in cropped\_original to ‘original’ folder
6. Upload the files in cropped\_segmentation to the ‘segmentation’ folder
7. Run the notebook. Start the training with about 30 epochs.
8. Tune the epoch parameter: if validation loss >> training loss you can call it overfitting. A little overfitting is no problem. Set the epoch value as the first epoch at which the validation loss is somewhat bigger than training loss.
9. Run the <train\_models> function to train the models.
10. Save the newly created models. You have now a trained model that can be used for segmentation

**Part 4: Segmentation**

1. Get yourself access to the ‘nerva’ server from DDS4.
2. Upload the trained models to the ‘nerva’ server from DDS4 in a folder called ‘models’.
3. Upload your images that you want to segment to the ‘nerva’ server in a folder called ‘input’
4. Create a directory named ‘tiff\_segm’.
5. Run the nnsegmentation.py script followed by arguments i) path to the models, ii) input directory and iii) output directory:

name@nerva: python nnsegmentation.py “models/” “input/” “output/”

It is best to do this in a TMUX terminal multiplexer to prevent your script from being disrupted.

1. Download the output.zip file to your local PC and unzip its content

**Part 5: Tracking**

1. Install and open CellProfiler version 3 or 4.
2. Upload the raw images and the segmented images in CellProfiler
3. Under <NamesAndTypes>, use the ‘objects’ option for the segmentated images to identify the segmented nuclei as objects directly. You do not have to use the modules IdentifyPrimaryObjects anymore.
4. Include modules MeasureObjectSizeShape and MeasureObjectIntensity to measure the size and intensity of all channels, so including the images with the Hoechst signal (usually channel c1).
5. Include TrackObjects and SaveImages.
6. Export the data as SQLite database using the ExportToDatabase module.

**Part 6: Fix tracks**

1. Start CPtrackR Shiny app from Gerhard.
2. Upload the .db database that came out of CellProfiler by pressing <Browse…>
3. Select the Tracking tab.
4. Click <Download all fixed tracks>
5. Save the fixed\_tracks.csv file.