

Thesis Title

Thesis Subtitle

Author Name

A thesis presented for the degree of
Doctor of Philosophy

Department Name

University Name

Country

Date

1 Abstract

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Contents

| | |
|--|-----------|
| 1 Abstract | 1 |
| 2 Introduction | 5 |
| 3 Methods and Materials | 5 |
| 3.1 Experimental procedures | 5 |
| 3.1.1 biostuff | 5 |
| 3.1.2 Timelapse recording | 5 |
| 3.2 Data analysis | 6 |
| 3.2.1 Timelapse datasets | 6 |
| 3.2.2 Initial timelapse processing | 6 |
| 3.2.3 Axon growth cone labelling | 6 |
| 3.2.4 Timelapse data preprocessing | 7 |
| 3.2.5 Growth cone detection model | 7 |
| 3.2.5.1 Temporal context frames | 7 |
| 3.2.5.2 Tiling | 9 |
| 3.2.5.3 Detection output format | 9 |
| 3.2.5.4 Training procedure | 9 |
| 3.2.5.5 Data association | 10 |
| 4 Results | 11 |
| 4.1 Tracking performance | 11 |
| 5 Discussion | 13 |
| 6 Supplementary Information | 14 |

List of Figures

| | | |
|---|--|----|
| 1 | Growth cone detection model overview | 8 |
| 2 | Growth cone tracking model performance | 12 |

List of Supplementary Figures

| | | |
|---|--|----|
| 1 | Initial timelapse preprocessing steps | 15 |
| 2 | Axon growth cone labelling | 16 |
| 3 | Labelled training data examples | 17 |
| 4 | Axon reconstruction from growth cone track | 18 |

List of Tables

| | | |
|---|--|----|
| 1 | Overview of timelapse recordings screening PDMS micro structures | 6 |
| 2 | Minimum cost flow hyperparameters | 11 |

List of Abbreviations

CLSM - Confocal laser scanning microscope PDMS CNN MOTA

2 Introduction

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

3 Methods and Materials

3.1 Experimental procedures

3.1.1 biostuff

3.1.2 Timelapse recording

Fast solid state storage media was used to prevent a frame rate bottleneck from an insufficiently fast network connection.

| | Acquired | n PDMS designs | T [min] | n frames | Length [days] | Model usage |
|----------|----------|----------------|---------|----------|---------------|-------------|
| Dataset1 | 20.12.20 | | 1 | 40 | 37 | 1 |
| Dataset2 | 27.08.21 | | 2 | 31 | 210 | 4.5 |
| Dataset3 | 27.10.21 | | 18 | 31 | 210 | 4.5 |
| Dataset4 | 07.10.21 | | 21 | 32 | 242 | 5.4 |

Table 1: Overview of timelapse recordings screening PDMS micro structures. n PDMS designs refers to the number of unique PDMS micros tructure designs composing the dataset. T refers to the temporal period of the recording. White space between rows indicates different experiments.

3.2 Data analysis

3.2.1 Timelapse datasets

This work incorporates three PDMS micros tructure timelapse recordings that were acquired at different timepoints. Accordingly, the data used in this project is based on three experiments, where each experiment was performed with 8-14 rat embryos (compare ?? for details). One of the three timelapses was solely obtained for generating model training data, thus the presented results are based on two experiments with a total of 16-28 biological replicates. A summery of timelapse datasets is given in Table 1.

3.2.2 Initial timelapse processing

The proprietary `.oir` files produced by the CLSM were converted to three dimensional `.tif` files using Python's `bioformats` package, which relies on a java virtual machine implemented within the `javabridge` package. Additionally, `.tif` frame sequences were rendered to `.mp4` video using `scikit-image` and `open-cv` (Suppl. Figure 1 A). These videos were used for initial evaluation of the timelapse, validating for example absence of undergrowth. The transmission channel of each PDMS micro structure timelapse was then loaded into napari, a python based n-dimensional image viewer (Sofroniew et al. 2021). Using `skimage.filters` to segment the micro channels in the PDMS designs, edge magnitude was detected with `prewitt()`, gaussian smoothing was performed with `gaussion()` ($\sigma = 1$), thresholding was done with `threshold_otsu()`, and finally, the segmentation was cleaned up using `skimage.morphology.binary_closing()` (diameter = 4) (Suppl. Figure 1 B). Subsequently, the segmentation of the PDMS micro channels was manually cleaned up, mainly using the bucket tool to fill areas enclosed by detected edges. As a last step, both the final output channel and the first 100 μm of the channels exiting the source wells were segmented and saved as binary masks (Suppl. Figure 1 C).

3.2.3 Axon growth cone labelling

The axon growth cone tracking model was trained using Dataset1, and Dataset2 which included timelapse recordings of three unique PDMS designs (compare Table 1). The labelling of these three image sequences was performed by one human expert using the napari image viewer (Suppl. Figure 2 A). The workflow for obtaining the four dimensional label of `FrameID - AxonID - X coordinates - Y coordinate` was as follows:

1. Load timelapse sequence.
2. Create empty set of axon identities.

3. Inspect short time slice of 3-6 frames for distinct, coherently moving blob.
4. Identify axon identity by its growth cone.
5. Trace axon identify over adjacent frames until unidentifiable.

In the scenario where two separate growth cones converge forming a single observable growth cone, one of the two identities was arbitrarily chosen to be continued while the other one was terminated. Hence, the underlying number of axons for a given growth cone label may be larger than one. It should also be considered that there is some degree of uncertainty in the ground truth labels. Especially when the PDMS micro channels become largely filled, distinguishing between GFP-protein trafficking along existing axons versus new growth cones becomes challenging. The annotations here were consistently done more conservatively, weighting the avoidance of false positives higher than missing true positives. Following this conservative labelling methodology, an axon identity was only considered if it appeared over more than three frames. From three concatenated PDMS micro structure timelapses, 300 growth cones were identified over $N=327$ frames where the average axon identity lifetime was 24 frames. An overview of the identify lifetime is given in Suppl. Figure 2 B; four labelled example frames are shown in Suppl. Figure 3.

3.2.4 Timelapse data preprocessing

The CLSM 12bit gray scale intensity values saved as `16bit unsigned integers` were first converted to a scale of 0 to 1 using `skimage.util.img_as_float()`. For image sequences that had an offset in the intensity profile, this offset was subtracted such that the minimal intensity was always 0. Next, the segmentation of the micro channels was used to mask the image sequence (see Suppl. Figure 1 C for example mask). The resulting initial distribution of intensity values for both training and inference data is shown in (Figure 1 A top left). In the next step, intensity values below threshold = 0.00083 were clipped and set to 0 (Figure 1 A top right). Next, the intensity profile I_{in} was stretched using `skimage.exposure.adjust_log()` function with a $gain = 1$ which transformed the distribution according to formula 1 (Figure 1 A bottom left).

$$I_{out} = gain * \log(1 + I_{in}) \quad (1)$$

Finally, the intensity distribution was divided by the global standard deviation across the entire training image sequence, ensuring unit variance in the model input data (Figure 1 A bottom right). Both frame-wise, and mean-related standardizations were omitted since their application resulted in decreased detection performance. The intensity distributions from train- and inference data do not overlap in Figure 1 A because the sparsity differs vastly across frames. Train intensity values do not increase from t_0 to t_N because t_N corresponds to a different timelapse video (Dataset2) which is more sparse than t_0 (Dataset1).

3.2.5 Growth cone detection model

3.2.5.1 Temporal context frames

The growth cone detection model implemented in PyTorch follows the general approach of YOLO (You Only Look Once) (Redmon et al. 2015) where the detections are obtained by a single pass through the network (Figure 1 B). The first aspect in which it deviates from the original is that instead of inputting an RGB image, the network receives a temporal stack of five gray scale images. Concretely, to detect growth cones at frame

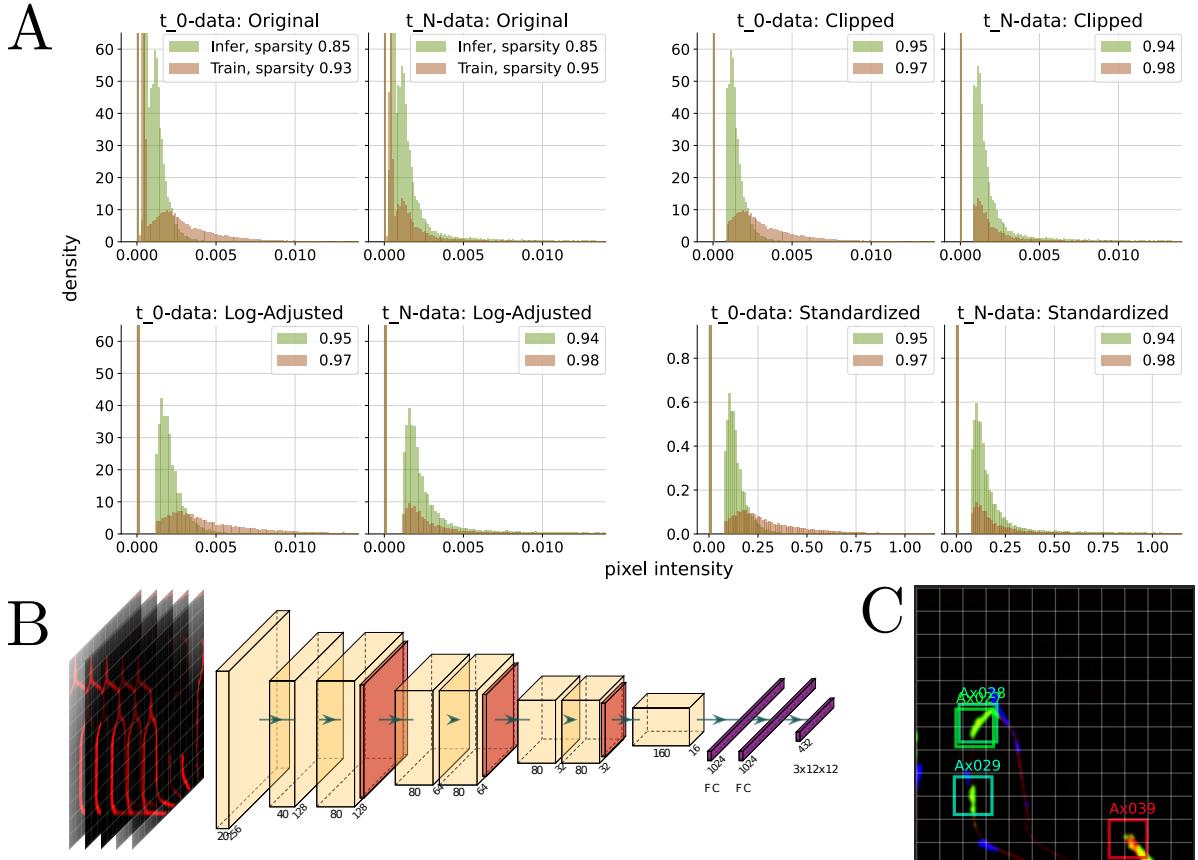


Figure 1: Growth cone detection model overview. **A** Pixel intensity distribution of training-, and inference data at t_0 and t_N over three major preprocessing steps. The two plots top left show the initial pixel intensity distributions, top right after clipping, bottom left after log adjusting, and bottom right after standardization. Histograms on the left were obtained from sampling 10^6 pixel values from the first frame, histograms on the right from the last respective frame. The inference data to produce the histograms (brown) was taken from a representative image sequence from Dataset3 (see Table 1). The number in the legend indicates the proportion of image values equal to zero (sparsity). Note that outlier intensity values are not shown. **B** CNN architecture. Each yellow block in represents a sequence of 2D-convolution, batch normalization (Ioffe and Szegedy 2015), and Leaky ReLU (Maas, Hannun, and Ng 2013). Orange layers stand for maximum pooling operations. FC stands for fully connected layers. **C** Example tile illustrating the YOLO label format. Each grid box can be predicted to contain a growth cone. In this example, 4 of 12 x 12 grid boxes are positive. Green colors in the tile represent positive motion (pixel intensity increased from frame t_0 to t_1), blue represents negative motion. Grid box size = 26 μm .

t_0 , frame $t_{-2}, t_{-1}, t_0, t_1, t_2$ are fed into the network. This architecture aims to imitate the strategy of human labelling: by inspecting single frames, growth cone identification is highly uncertain; only when scanning sequences of frames, coherently moving blobs of particular shape and dynamics can be linked to growth cones and thus an axon identity. To always provide full temporal context, frame t_0, t_1, t_{N-1}, t_N were omitted from detecting growth cones in the image sequence. Computing the motion between frames manually by subtraction yielded decreased detection performance over the implicit approach of passing temporal context frames. An illustration of the motion computation is shown in Figure 1 C.

3.2.5.2 Tiling

Each yellow block in Figure 1 B represents a sequence of 2D-convolution, batch normalization (Ioffe and Szegedy 2015), and Leaky ReLU (Maas, Hannun, and Ng 2013). Orange layers stand for maximum pooling operations. As performing detection on the original resolution of 3868 x 1972 was computationally intractable, the timelapse frames were split into 512 x 512 tiles (see Figure 1 C and grid in Suppl. Figure 3). The CNN computes a convolutional feature map of 16 x 16 x 160, thus a single *feature pixel* represents a region of $\frac{512}{16} = 32$ pixels in the original 512 x 512 input image. The CNN output resolution was a relevant consideration for its architecture, as the detection objects of interest are small and potentially locally clustered (using microscopy settings described in ??, growth cones are between 4-26 pixels). If the same CNN feature output resolution was to be achieved using original timelapse frames, the CNN output would be of shape of $\frac{3868}{32} \times \frac{1972}{32} \times 160 \approx 120 \times 61 \times 160$. Storing the weights between this high-resolution CNN feature map and the first fully connected layer exceeded GPU memory. An additional computational benefit is achieved by skipping empty tiles. From visual inspection, the discontinuities between tiles did not seem to result in decreased detection performance for growth cones near the tile edges.

3.2.5.3 Detection output format

Following the general YOLO label format, the network is trained to find a mapping from a single tile CNN feature map to a 12 x 12 x 3 array. Here, the first two dimensions represent a grid of the input tile, the last dimension refers to the confidence of the respective grid box containing a growth cone, and X-, Y grid box coordinates referring to the relative location of a growth cone within the box (Figure 1 C). This representation results in the limitation, that only one growth cone can be detected per grid box. As the example in Figure 1 C shows, close growth cones may still be detected as two separate identities if their centers are located in different grid boxes. In the worst case scenario, the spatial detection resolution of multiple growth cones is limited by the grid box size which is equal to $\frac{512}{12} = 43$ pixels or 26 μm . This resolution was sufficient for the application of our model as densely grouped growth cones were the exception.

To drop overlapping detections, non max suppression was applied to the final detection output according to (Bodla et al. 2017) using a minimum euclidean distance of 23 pixels.

3.2.5.4 Training procedure

The training data was split into 287 train frames (0.87), and 40 (0.13) consecutive test frames which spanned two different PDMS micro structures. The final model used for inference was trained on the entire dataset. Using translation, rotation, horizontal and

vertical flipping as data augmentation, the model was trained up to convergence for 1000 epochs (Figure 2 A). The loss function below (2) is a slight modification from the original.

$$\lambda_{anchor} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \lambda_{obj} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} (c_i - \hat{c}_i)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{noobj} (c_i - \hat{c}_i)^2 \quad (2)$$

where $S = 12$ is the number of tiles, $B = 1$ is the number of detections per grid box, \mathbb{I}_{ij}^{obj} equals to 1 if a growth cone exists, 0 otherwise, \mathbb{I}_{ij}^{noobj} equals to 0 if a growth cone exists, 1 otherwise, c_i refers to the confidence that an object exists in the grid box, and x_i, y_i represent the grid box coordinates. $\hat{\cdot}$ stands for the ground truth label. The loss terms for predicting coordinates, object presence, and object absence are weighted according to $\lambda_{anchor} = 45$, $\lambda_{obj} = 54.25$, and $\lambda_{noobj} = 0.75$ respectively. The balancing of those terms is based on the proportion of positive grid boxes which is $\approx 0.7\%$. The initial learning rate was set to 0.0005 and decayed with a rate given in formula 3.

$$\gamma = e^{-\frac{1}{10}\sqrt{x}} \quad (3)$$

where γ is multiplied with the original learning rate, and x refers to the current epoch. Pytorch's Adam optimizer (Kingma and Ba 2014) was used for fitting the model with $\beta_1 = 0.9$, $\beta_2 = 0.999$.

3.2.5.5 Data association

The implemented growth cone tracking model follows a classical object tracking paradigm of splitting the problem into object detection and identity association. For this second step, the detections produced by the YOLO-like architecture need to be classified into unique growth cone identities that live over several video frames. In this work, identity assignment is framed as a graph problem where we seek minimum cost flow solutions (Wang, Wang, and Yu 2019). At a high level, nodes represent detections at particular frames, and edges represent identity associations between them (see illustration in Figure 2 E). Each detection confidence is also represented as an edge, which elegantly incorporates the detection model uncertainty into forming identity trajectories through the graph and avoids the setting of explicit detection confidence thresholds. At the basis of associating detections between frames is the cost we assign between them. This cost can be interpreted as the likelihood the two detections correspond to the same growth cone identity. In contrast to other domains where visual similarity is highly relevant, the edge costs here are completely based on the spatial distance between detections. Specifically, the A* distances (Hart, Nilsson, and Raphael 1968) between them computed on the segmented PDMS micro channel mask. This cost considers the constraint, that growth cones can only translocate within the micro channels and are limited in outgrowth speed. Solving the graph optimization problem includes the constraint, that a node can only receive and emit a single edge, or in other words, a node can only represent a single identity. As shown by Wang, Wang, and Yu 2019, the graph can be solved optimally and efficiently using integer linear programming. The implementation here is based on the open-source package `libmot`, using the build-in `MinCostFlowTracker`. The optimal hyperparameters listed in table 2 were identified with a grid-search algorithm on test data (see Figure 2 D).

| | | |
|--------------------------|---------------------------|----------------------|
| Edge cost threshold | Entry-exit cost | Miss rate |
| 0.7 | 2 | 0.6 |
| Maximum number of misses | Minimum network flow | Maximum network flow |
| 1 | 5 | 450 |
| Visual similarity weight | Confidence capping method | |
| 0 | <i>scale</i> | |

Table 2: Minimum cost flow hyperparameters. The edge cost threshold determines if an edge is pruned or kept, the entry-exit cost defines the cost of creating and terminating identities, the maximum number of misses indicates for how many frames an identity can be not detected, but still not terminated, the miss rate determines how much cost is incurred from missing detections (low means high cost), minimum and maximum network flow gives the minimum and maximum number of identities over all frames, visual similarity weight determines the degree to which visual similarity between detections contributes to the cost, and finally the confidence capping method sets the behavior for confidence values above 1, where *scale* means normalize to maximum confidence.

4 Results

4.1 Tracking performance

—Motivation why we build it—

The presented model splits the tracking problem into growth cone detection and identity association. A representative example of detections is shown in Figure 2 B. Given the temporal stack of five image tiles, the detection model accurately identifies growth cones in PDMS micro structure timelapse frames. False positive detections made by the model are often ambiguous image regions that may be interpreted as positives under less conservative ground truth labelling. On the test set, the detector reaches a precision of 0.73, and recall of 0.79. F1 score at a confidence threshold of 0.79 is 0.76 (Figure 2 C). A both deeper and wider CNN architecture did show decreased performance. The ensuing step of identity association was performed in a graph framework optimizing for minimum cost flow solutions. Matching detection performance in tracking is challenging as in addition to detection, identity switches, object occlusions, and suitable identity creation-, and termination need to be considered. Using the multiple object tracking benchmarks proposed in ??, our tracker achieves identity precision, recall and F1 score of 0.73, 0.68, 0.71, respectively, which is a reasonably small drop from detection performance (Figure 2 D). The commonly used MOTA (multiple object tracking accuracy) metric which considers the number of false positives, false negatives (including identity), and identity switches normalized to the number of ground truth labels was 0.61. A more intuitive measure of tracking performance is visualized by the top bar in Figure 2 D, indicating the proportion of growth cones mostly tracked (0.57), partially tracked (0.23), and mostly lost (0.2).

Although not utilized for our application of the tracking model, axons can be reconstructed from the growth cone track, assuming that outgrowth followed the shortest path between detections.

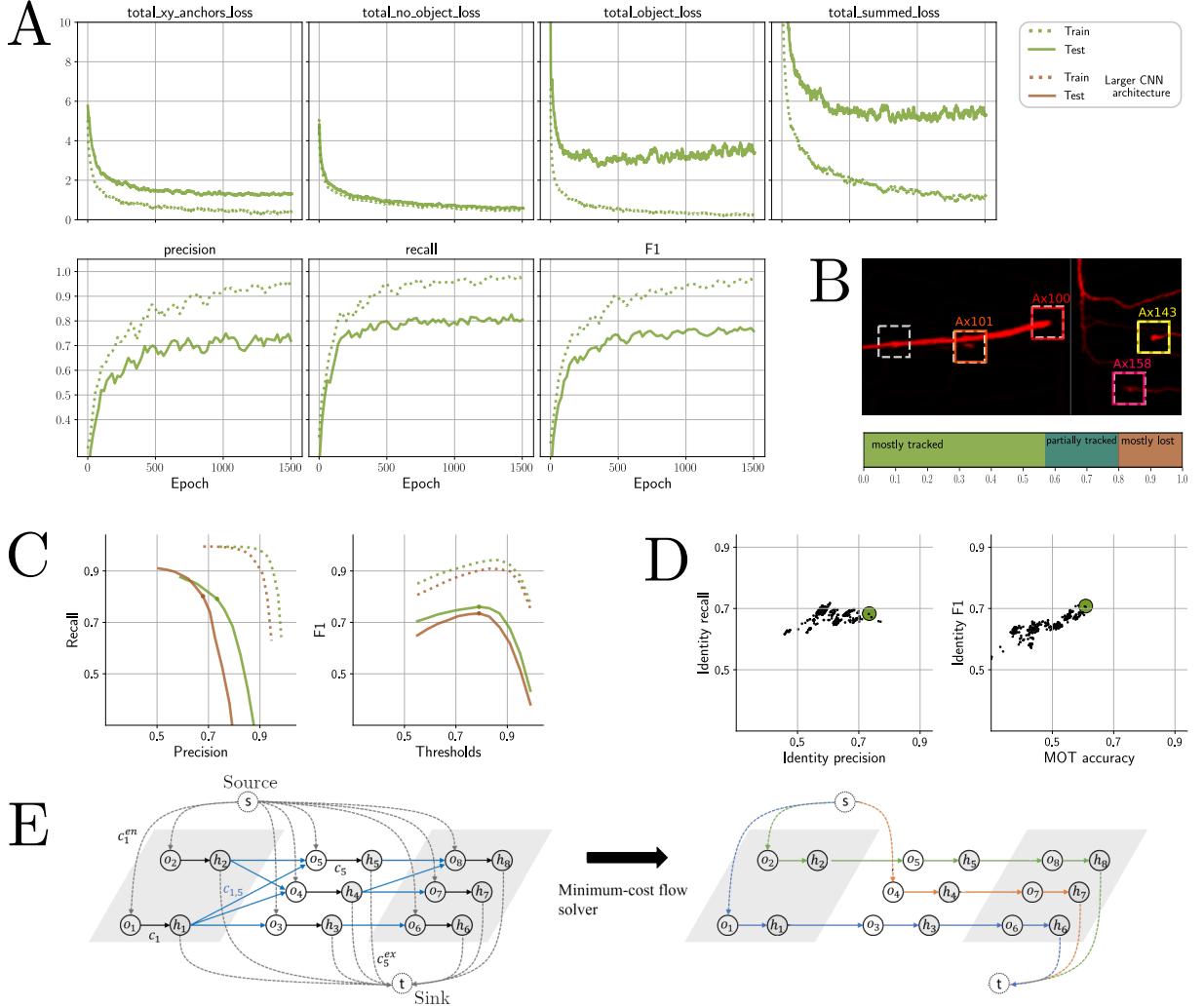
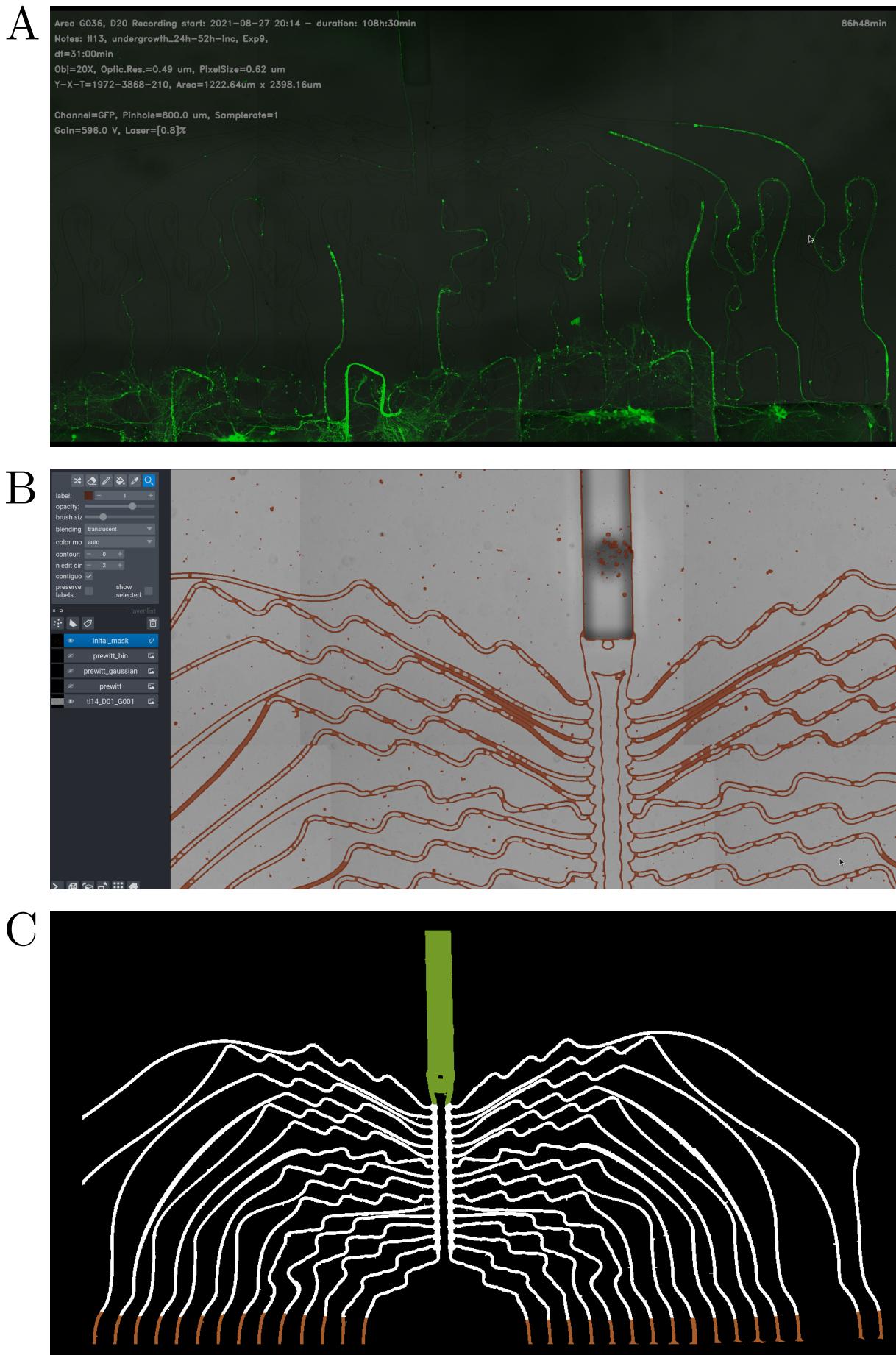


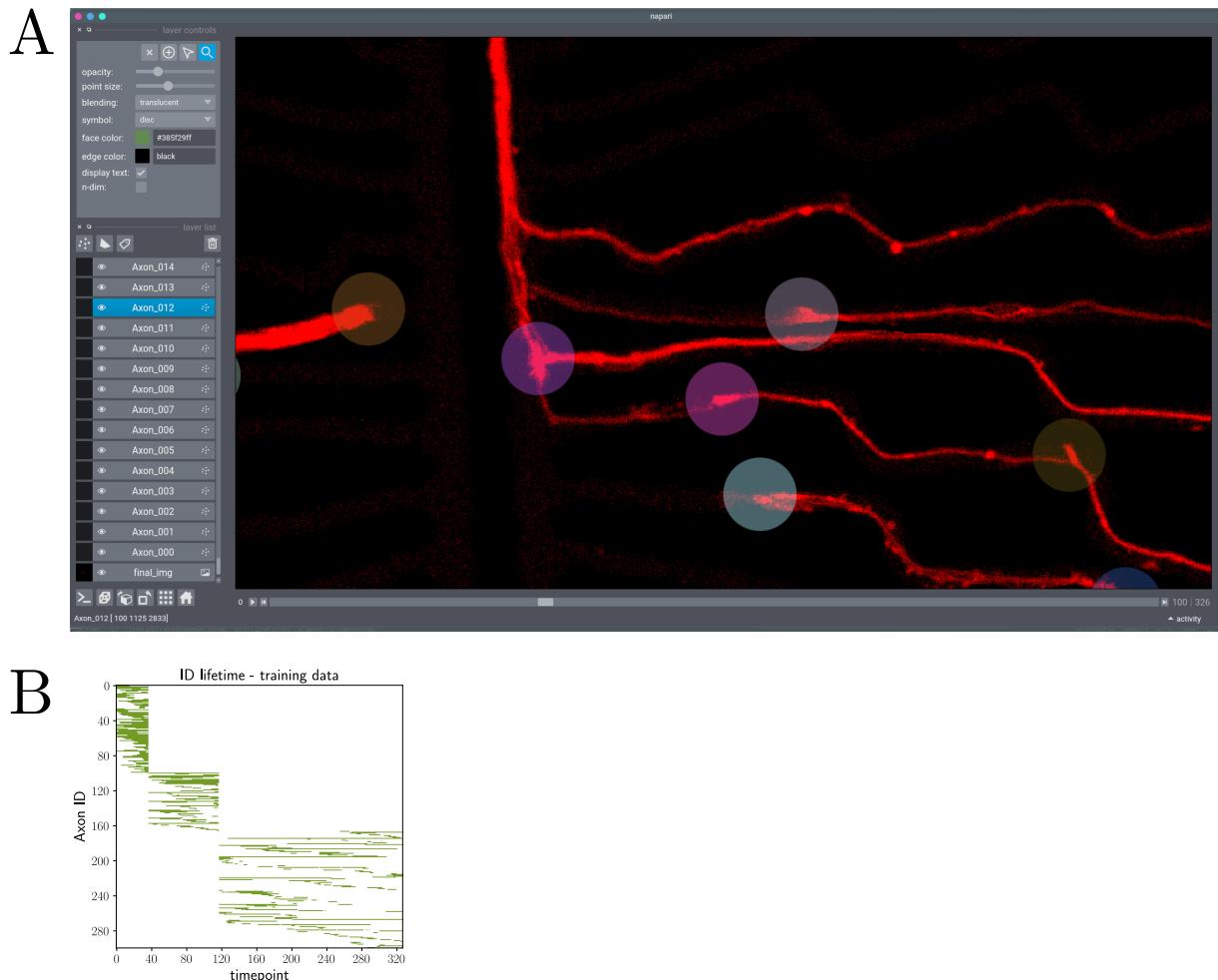
Figure 2: Growth cone tracking model performance. **A** Loss and performance over 1500 training epochs. Dotted line refers to train set, solid line to test. The plotted loss was smoothed with an exponentially decaying kernel over 25 epochs, precision, recall, and F1 over 60 epochs. **B** Representative growth cone detection example. Dashed boxes are predicted, colored ones are ground truth. Scale bar = 90 μm . **C** Detection performance. The maximum F1 score for varying confidence thresholds on test set is indicated by the dot. The brown line shows performance for a model with wider and deeper architecture. Legend in A applies. **D** Model growth cone tracking performance. Identity precision and recall incorporate classification of correct identity. Each black dot represents the performance using one set of hyperparameters, the green dot represents the highest scoring set (see Table 2) where identity F1 was 0.71, MOT accuracy 0.61. MOT accuracy measures the number of false positives, false negatives, and identity switches normalized to the number of ground truth labels. The top bar visualizes the proportion of growth cones that were mostly tracked (green, >80% identity lifetime tracked), mostly lost (brown, <20%), and partially tracked (dark green, between 20-80%). **E** Minimum cost flow optimization illustration adopted from (Wang, Wang, and Yu 2019). Frames are illustrated in gray and white background, detections within a frame are represented by a pre- (o_i), and post (h_i) node. Blue edges on the left represent costs between detections in adjacent frames. Coloured edges on the right indicate identity associations after solving the graph.

5 Discussion

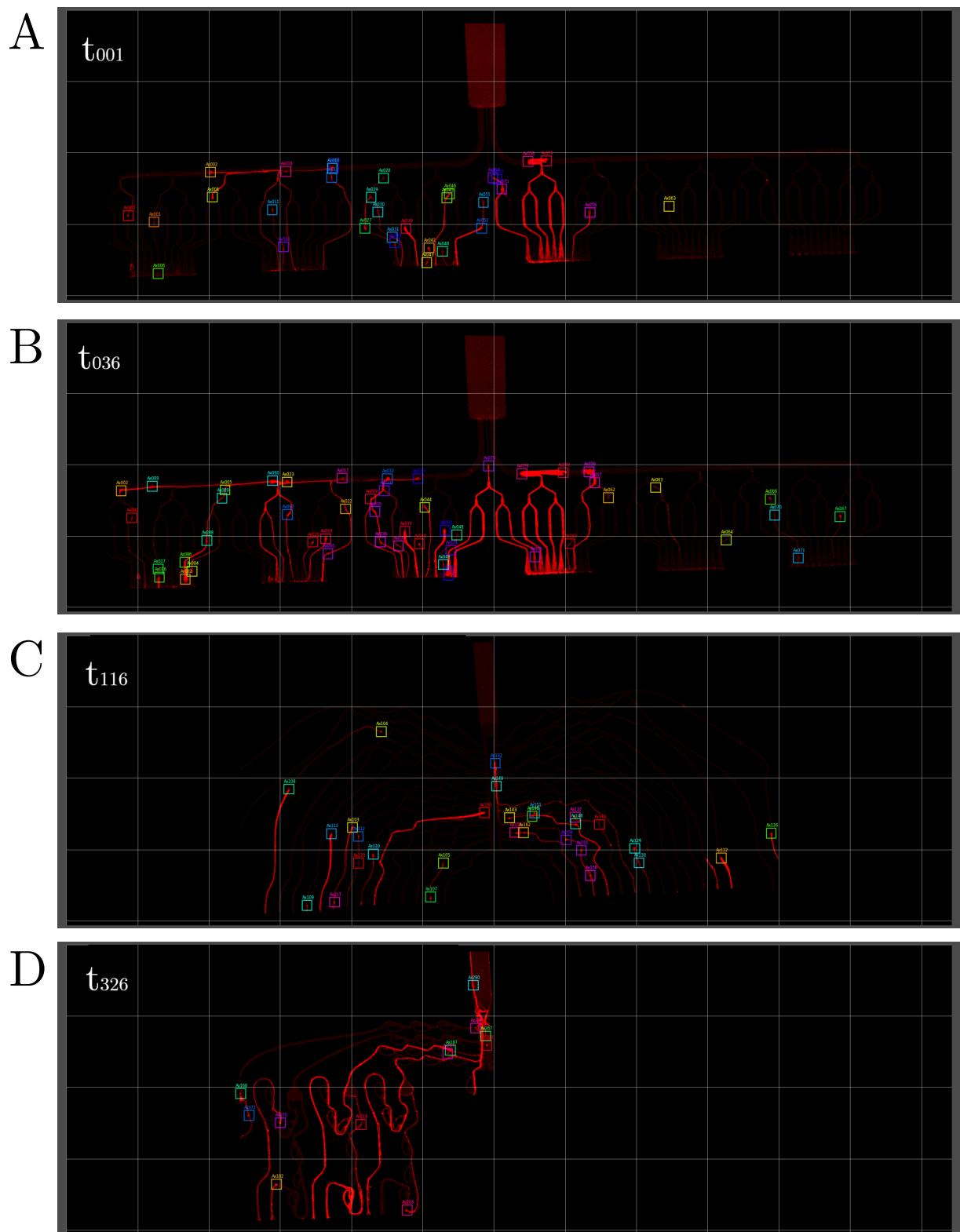
6 Supplementary Information



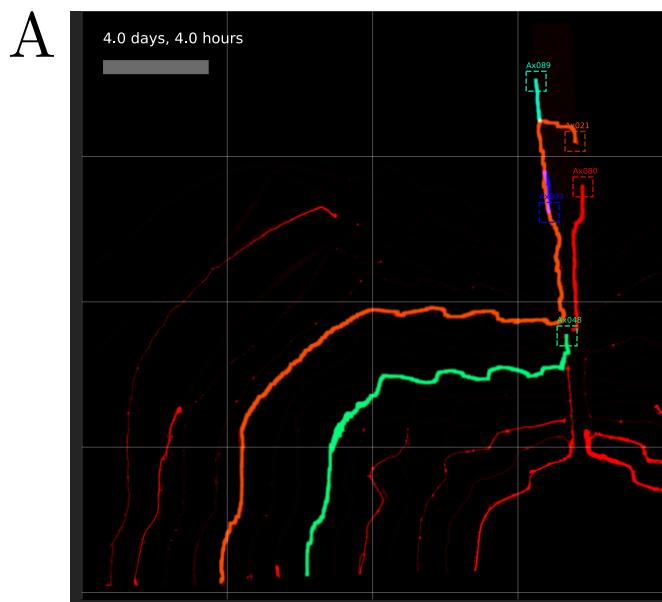
Supplementary Figure 1: Initial timelapse preprocessing steps. A shows a snapshot of the exported video including the relevant metadata of the recording. B shows the napari image viewer interface with the edge segmentation layer in brown. C shows the binary mask of the micro channels in non-black, the output channel in green, and the exiting channels in brown.



Supplementary Figure 2: Axon growth cone labelling. **A** shows a snapshot of the napari image viewer during labelling. Each circle represents a growth cone label, the color corresponds to the identity. Each axon identity is saved as a napari-Points layer which are listed on the left. **B** illustrates the axon identify lifetime. A green point on this pixel map indicates that a label exists for the matching axon identity and frame. The three clusters originate from the concatenation of three PDMS microstructure timelapse videos.



Supplementary Figure 3: Labelled training data examples. **A** shows the first frame of the training data sequence. Each box represents a growth cone, the color indicates the identity over consecutive frames. **B** shows the last frame of Dataset1. **C** shows the last frame of Dataset2, PDMS micro structure 1 (compare Table 1). **D** shows the last frame of Dataset2, PDMS micro structure 2. Gridsize = 317 μm .



Supplementary Figure 4: Axon reconstruction from growth cone track. **A** For clarity, only a subset of identified axons is drawn.

References

- Sofroniew, Nicholas et al. (Oct. 2021). “napari/napari: 0.4.12rc2”. In: DOI: 10.5281/ZENODO.5587893. URL: <https://zenodo.org/record/5587893>.
- Ioffe, Sergey and Christian Szegedy (Feb. 2015). “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: *32nd International Conference on Machine Learning, ICML 2015* 1, pp. 448–456. URL: <https://arxiv.org/abs/1502.03167v3>.
- Maas, Andrew L, Awni Y Hannun, and Andrew Y Ng (2013). “Rectifier Nonlinearities Improve Neural Network Acoustic Models”. In.
- Redmon, Joseph et al. (June 2015). “You Only Look Once: Unified, Real-Time Object Detection”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2016-December, pp. 779–788. ISSN: 10636919. DOI: 10.1109/CVPR.2016.91. URL: <https://arxiv.org/abs/1506.02640v5>.
- Bodla, Navaneeth et al. (Apr. 2017). “Soft-NMS – Improving Object Detection With One Line of Code”. In: *Proceedings of the IEEE International Conference on Computer Vision* 2017-October, pp. 5562–5570. ISSN: 15505499. DOI: 10.1109/ICCV.2017.593. URL: <https://arxiv.org/abs/1704.04503v2>.
- Kingma, Diederik P. and Jimmy Lei Ba (Dec. 2014). “Adam: A Method for Stochastic Optimization”. In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. URL: <https://arxiv.org/abs/1412.6980v9>.
- Wang, Congchao, Yizhi Wang, and Guoqiang Yu (Nov. 2019). “Efficient Global Multi-object Tracking Under Minimum-cost Circulation Framework”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1. ISSN: 0162-8828. DOI: 10.1109/tpami.2020.3026257. URL: <https://arxiv.org/abs/1911.00796v2>.
- Hart, Peter E., Nils J. Nilsson, and Bertram Raphael (1968). “A Formal Basis for the Heuristic Determination of Minimum Cost Paths”. In: *IEEE Transactions on Systems Science and Cybernetics* 4 (2), pp. 100–107. ISSN: 21682887. DOI: 10.1109/TSSC.1968.300136.