Supplementary Material:
FISBe: A real-world benchmark dataset for instance segmentation of long-range thin filamentous structures

Lisa Mais[1,2,4,*,✉], **Peter Hirsch**[1,2,*], **Claire Managan**[3], **Ramya Kandarpa**[3],
Josef Lorenz Rumberger[1,2], **Annika Reinke**[2,5], **Lena Maier-Hein**[2,5],
Gudrun Ihrke[3], **Dagmar Kainmueller**[1,2,4,✉]
[1] Max-Delbrueck-Center for Molecular Medicine in the Helmholtz Association (MDC) [2] Helmholtz Imaging
[3] HHMI Janelia Research Campus [4] University of Potsdam [5] German Cancer Research Center (DKFZ)
✉ {firstname.lastname}@mdc-berlin.de * shared first authors

A. Appendix

A.1. Dataset Documentation

A.1.1 Datasheet

In this section we answer the Datasheet for Datasets questionnaire [5] to document FISBe, the FlyLight Instance Segmentation Benchmark dataset. It contains information about motivation, composition, collection, preprocessing, usage, licensing as well as hosting and maintenance plan.

A.1.1.1 Motivation

For what purpose was the dataset created? Was there a specific task in mind? Was there a specific gap that needed to be filled? Please provide a description.

Segmenting individual neurons in multi-neuron light microscopy (LM) recordings is intricate due to the long, thin filamentous and widely branching morphology of individual neurons, the tight interweaving of multiple neurons, and LM-specific imaging characteristics like partial volume effects and uneven illumination. These properties reflect a current key challenge for deep-learning models across domains, namely to efficiently capture long-range dependencies in the data. While methodological research on this topic is buzzing in the machine learning community, to date, respective methods are typically benchmarked on synthetic datasets. To fill this gap, we created the FlyLight Instance Segmentation Benchmark dataset, to the best of our knowledge, the first publicly available multi-neuron LM dataset with pixel-wise ground truth and the first real-world benchmark dataset for instance segmentation of long thin filamentous objects.

Who created this dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)?
This dataset was created in a collaboration of the Max-Delbrueck-Center for Molecular Medicine in the Helmholtz Association (MDC) and the Howard Hughes Medical Institute Janelia Research Campus. More precisely, the Kainmueller lab at the MDC and the Project Technical Resources Team at Janelia.

Who funded the creation of the dataset? If there is an associated grant, please provide the name of the grantor and the grant name and number.

Howard Hughes Medical Institute Janelia Research Campus and Max-Delbrueck-Center for Molecular Medicine in the Helmholtz Association (MDC) funded the creation of the dataset.

A.1.1.2 Composition

What do the samples[1] that comprise the dataset represent (e.g., documents, photos, people, countries)? Are there multiple types of samples (e.g., movies, users, and ratings; people and interactions between them; nodes and edges)? Please provide a description.

The dataset consists of 3d multi-neuron multicolor light microscopy images and their respective pixel-wise instance segmentation masks. The ”raw” light microscopy data shows neurons of the fruit fly Drosophila Melanogaster acquired with a technique called MultiColor FlpOut (MCFO) [15, 18]. Fruit fly brains of different transgenic lines (e.g.

[1] We changed *instances* to *samples* when refering to images of the dataset to not use the term ambiguously; instead we only use *instances* to refer to *object instances* in images.

GAL4 lines [9]) were imaged, each transgenic line tags a different set of neurons. There are multiple MCFO images of the same transgenic line, where each MCFO image expresses (shows) a stochastic subset of the tagged neurons. The neurons contained in each image were manually annotated by trained expert annotators. The dataset is split into a *completely* labeled (all neurons in the image are manually segmented) and a *partly* labeled (a subset of neurons in the image is manually segmented) set.

How many samples/instances are there in total (of each type, if appropriate)?

The *completely* labeled set comprises 30 images with 139 labeled neurons in total, and the *partly* labeled set comprises 71 images with 451 labeled neurons in total.

Does the dataset contain all possible samples or is it a subset (not necessarily random) of samples from a larger set? If the dataset is a sample, then what is the larger set? Is the sample representative of the larger set (e.g., geographic coverage)? If so, please describe how this representativeness was validated/verified. If it is not representative of the larger set, please describe why not (e.g., to cover a more diverse range of instances, because instances were withheld or unavailable).

The dataset contains a subset of 101 images from the ”40x Gen1” set of [15]. The full ”40x Gen1” set consists of 46,791 images of 4575 different transgenic lines. From this set, we selected relatively sparse images in terms of number of expressed neurons which seemed feasible for manual annotation. Thus, our dataset is not representative for the full ”40x Gen1” MCFO collection.

What data does each sample consist of? “Raw” data (e.g., unprocessed text or images) or features? Is there a label or target associated with each sample? Please provide a description.

Each sample consists of a single 3d MCFO image of neurons of the fruit fly. For each image, we provide a pixel-wise instance segmentation for all separable neurons. Each sample is stored as a separate *zarr* file (”Zarr is a file storage format for chunked, compressed, N-dimensional arrays based on an open-source specification.” https://zarr.readthedocs.io). The image data (”raw”) and the segmentation (”gt_instances”) are stored as two arrays within a single zarr file. The segmentation mask for each neuron is stored in a separate channel. The order of dimensions is CZYX. In Python the data can, for instance, be opened with:

```
import zarr
raw = zarr.open(
  <path_to_zarr>,
  path="volumes/raw")
seg = zarr.open(
  <path_to_zarr>,
  path="volumes/gt_instances")
```

Zarr arrays are read lazily on-demand. Many functions that expect numpy arrays also work with zarr arrays. The arrays can also explicitly be converted to numpy arrays with:

```
import numpy as np
raw_np = np.array(raw)
```

Is any information missing from individual samples? If so, please provide a description, explaining why this information is missing (e.g., because it was unavailable). This does not include intentionally removed information, but might include, e.g., redacted text.

Not all neuronal structures could be segmented within all images of the provided dataset. Mainly, there are two reasons: (1) there are overlapping neurons with the same or a similar color that could not be separated due to the partial volume effect, and (2) some neuronal structures cannot be delineated correctly in the presence of noisy background in the same color as the neuron itself. In the *completely* labeled set all neuronal structures have been segmented, in the *partly* labeled set some structures are missing.

Are relationships between individual samples made explicit (e.g., users’ movie ratings, social network links)? If so, please describe how these relationships are made explicit.

Yes, one transgenic line is often imaged multiple times as only a stochastic subset of all tagged neurons is visible per MCFO image. Moreover, the same neuron might be tagged in multiple transgenic lines.

Are there recommended data splits (e.g., training, development/validation, testing)? If so, please provide a description of these splits, explaining the rationale behind them.

Yes, we provide a recommended data split for training, validation and testing. The files in the provided download are presorted according to this recommendation. When splitting the data into sets, we made sure that images of the same transgenic lines are in the same split and paid attention to having similar proportions of images with overlapping neurons as well as having a similar average number of neurons per image in each split.

Are there any errors, sources of noise, or redundancies in the dataset? If so, please provide a description.

There might be uneven illumination resulting in gaps within neurons in the raw microscopy images as well as the corresponding annotations. This is intrinsic to this kind of light microscopy images.

Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)? If it links to or relies on external resources, a) are there guarantees that they will exist, and remain constant, over time; b) are there official archival versions of the complete dataset (i.e., including the external resources as they existed at the time the dataset was created); c) are there any restrictions (e.g., licenses, fees) associated with any of the external resources that might apply to a future user? Please provide descriptions of all external resources and any restrictions associated with them, as well as links or other access points, as appropriate.

Yes, the dataset is self-contained. There is an external, additional source of raw images that could potentially be used for self-supervised learning. The raw images in our dataset are a subset of the released MCFO collection of the FlyLight project [15]. The whole collection can be downloaded at https://gen1mcfo.janelia.org. Note though that there are no segmentation masks available for these images.

Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctor-patient confidentiality, data that includes the content of individuals non-public communications)? If so, please provide a description.

No.

A.1.1.3 Collection Process

How was the data associated with each sample acquired? Was the data directly observable (e.g., raw text, movie ratings), reported by subjects (e.g., survey responses), or indirectly inferred/derived from other data (e.g., part-of-speech tags, model-based guesses for age or language)? If data was reported by subjects or indirectly inferred/derived from other data, was the data validated/verified? If so, please describe how.

The content of the raw images was directly recorded using confocal microscopes. The annotations were created manually.

What mechanisms or procedures were used to collect the data (e.g., hardware apparatus or sensor, manual human curation, software program, software API)? How were these mechanisms or procedures validated?

Imaging was performed using eight Zeiss LSM 710 or 780 laser scanning confocal microscopes (for more information on the imaging process see [15]). Two trained expert annotators manually segmented and proofread each other to segment the neurons in these images using the interactive rendering tool VVD Viewer (https://github.com/JaneliaSciComp/VVDViewer).

If the dataset is a sample from a larger set, what was the sampling strategy (e.g., deterministic, probabilistic with specific sampling probabilities)?
We manually selected images from the larger ”40x Gen1” collection. We chose images that contained a sparse set of neurons and that contained neurons that preferably were not contained in previously selected images.

Who was involved in the data collection process (e.g., students, crowdworkers, contractors) and how were they compensated (e.g., how much were crowdworkers paid)?
The data collection process was done by full time employees at the Howard Hughes Medical Institute Janelia Research Campus and the Max-Delbrueck-Center for Molecular Medicine in the Helmholtz Association (MDC).

Over what timeframe was the data collected? Does this timeframe match the creation timeframe of the data associated with the samples (e.g., recent crawl of old news articles)? If not, please describe the timeframe in which the data associated with the samples was created.

MCFO selection and manual annotation were mainly done in 2018 and 2019. The respective acquisition date of the MCFO sample is noted within the sample name in ”YYYYMMDD” format. Most samples of our dataset were acquired in 2017 and 2018.

Were any ethical review processes conducted (e.g., by an institutional review board)? If so, please provide a description of these review processes, including the outcomes, as well as a link or other access point to any supporting documentation.

There was no ethical review process conducted as we did not record any new animal data, the dataset does not relate to people and it does not contain confidential data.

Does the dataset relate to people? If not, you may skip the remaining questions in this section.

No.

A.1.1.4 Preprocessing/cleaning/labeling

Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)? If so, please provide a description. If not, you may skip the remainder of the questions in this section.

The following preprocessing was done for each image: The central brain and part of the ventral nerve cord (VNC) were recorded in tiles by the light microscope. The tiles were stitched together and distortion corrected (for more information see [26]).

Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)? If so, please provide a link or other access point to the "raw" data.

The original images are available at https://gen1mcfo.janelia.org.

Is the software used to preprocess/clean/label the samples available? If so, please provide a link or other access point.

The image processing, such as distortion correction and stitching, is done by using the open-source software Janelia Workstation [21].

A.1.1.5 Uses

Has the dataset been used for any tasks already? If so, please provide a description.

In [13], an earlier, unpublished version of our dataset has been used to qualitatively evaluate PatchPerPix, a deep learning-based instance segmentation method. The trained model was then applied to ∼40.000 samples of the MCFO collection [14, 15] to search for given neuronal structures extracted from electron microscopy (EM) data [23]. PatchPerPix is also used as one of three baselines to showcase this published version of our dataset.

Is there a repository that links to any or all papers or systems that use the dataset? If so, please provide a link or other access point.

As they are getting published, we will reference them at https://kainmueller-lab.github.io/fisbe

What (other) tasks could the dataset be used for?
The dataset can be used for a wide range of method development tasks such as capturing long-range dependencies, segmentation of thin filamentous structures, self- and semi-supervised training or denoising. Advances in these areas can in turn facilitate scientific discoveries in basic neuroscience by providing improved neuron reconstructions for morphological and functional analyses.

A.1.1.6 Distribution

Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created? If so, please provide a description.

The dataset will be publicly available.

How will the dataset be distributed (e.g., tarball on website, API, GitHub) Does the dataset have a digital object identifier (DOI)?

The dataset will be distributed through zenodo (DOI: 10.5281/zenodo.10875063) and our project page https://kainmueller-lab.github.io/fisbe.

When will the dataset be distributed?
With publication of the accompanying paper.

Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)? If so, please describe this license and/or ToU, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms or ToU, as well as any fees associated with these restrictions.

The dataset will be distributed under the Creative Commons Attribution 4.0 International (CC BY 4.0) license (https://creativecommons.org/licenses/by/4.0/).

Have any third parties imposed IP-based or other restrictions on the data associated with the samples? If so, please describe these restrictions, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms, as well as any fees associated with these restrictions.

All MCFO images have previously been made publicly available by [15] under the same license (CC BY 4.0) at https://gen1mcfo.janelia.org.

A.1.1.7 Maintenance

Who will be supporting/hosting/maintaining the dataset?
Lisa Mais supports and maintains the dataset.

How can the owner/curator/manager of the dataset be contacted (e.g., email address)?
Lisa Mais and Dagmar Kainmueller can be contacted at {firstname.lastname}@mdc-berlin.de.

Is there an erratum? If so, please provide a link or other access point.

Errata will be published at https://kainmueller-lab.github.io/fisbe.

Will the dataset be updated (e.g., to correct labeling errors, add new samples, delete samples)? If so, please describe how often, by whom, and how updates will be communicated to users (e.g., mailing list, GitHub)?

The dataset will be updated to correct erroneous segmentation and potentially to add new samples and annotations. It will be updated when a relevant number of updates has accumulated. Updates will be communicated through https://kainmueller-lab.github.io/fisbe.

Will older versions of the dataset continue to be supported/hosted/maintained? If so, please describe how. If not, please describe how its obsolescence will be communicated to users.

We publish our dataset on zenodo. Zenodo supports versioning, including DOI versioning. Older versions of the dataset will thus stay available.

If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so? If so, please provide a description. Will these contributions be validated/verified? If so, please describe how. If not, why not? Is there a process for communicating/distributing these contributions to other users? If so, please provide a description.

We welcome contributions to our dataset. Errata, new samples and annotations and other contributions can be contributed via *github issues* at https://kainmueller-lab.github.io/fisbe. We will verify such contributions and update the dataset accordingly.

A.1.2 How to Open and View Image Files

We recommend viewing the FISBe dataset with napari [17]. The following instructions have been tested with Linux. While they should also work for Windows and MacOS, they might require some small changes. Please follow the official installation instructions (https://napari.org/stable/):

```
conda create -y -n napari-env -c \
  conda-forge python=3.9
conda activate napari-env
pip install "napari[all]" zarr
```

Then save the following Python script (also included in the provided download of our dataset):

```
import zarr, sys, napari

raw = zarr.load(
    sys.argv[1], path="volumes/raw")
gts = zarr.load(
    sys.argv[1], path="volumes/gt_instances")

viewer = napari.Viewer(ndisplay=3)
for idx, gt in enumerate(gts):
    viewer.add_labels(
        gt, rendering='translucent',
        blending='additive', name=f'gt_{idx}'
            )
viewer.add_image(raw[0], colormap="red",
    name='raw_r', blending='additive')
viewer.add_image(raw[1], colormap="green",
    name='raw_g', blending='additive')
viewer.add_image(raw[2], colormap="blue",
    name='raw_b', blending='additive')
napari.run()
```

Execute it from the command line to view the image:

```
python <script_name.py> <path-to-file>/
    R9F03-20181030_62_B5.zarr
```

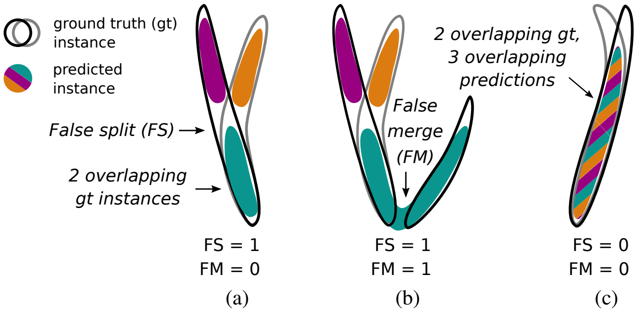

ground truth (gt) instance
predicted instance
False split (FS)
2 overlapping gt instances
FS = 1
FM = 0
(a)
False merge (FM)
FS = 1
FM = 1
(b)
2 overlapping gt, 3 overlapping predictions
FS = 0
FM = 0
(c)

Figure 5. Exemplary challenges for many-to-many matching with overlaps. (a) One predicted instance lies completely within an overlapping gt region, but it should only be assigned to one of them; (b) one predicted instance covers one gt and merges with an overlapping gt region, here it should be assigned to the single gt and one of the overlapping ones; and (c) three overlapping predicted instances cover two overlapping gt instances, here only two predicted instances should be matched to the two gt instances respectively (the other predicted instance should rather only count as false positive than as false split). As there are plenty of scenarios how gt and predicted instances can overlap, special treatment for overlapping regions is difficult and error-prone. However, our proposed algorithm (see Alg. 1 in the main paper) naturally handles such overlaps by keeping track of already matched pixels (as opposed to only on the level of instances).

A.2. Extended Metrics Information

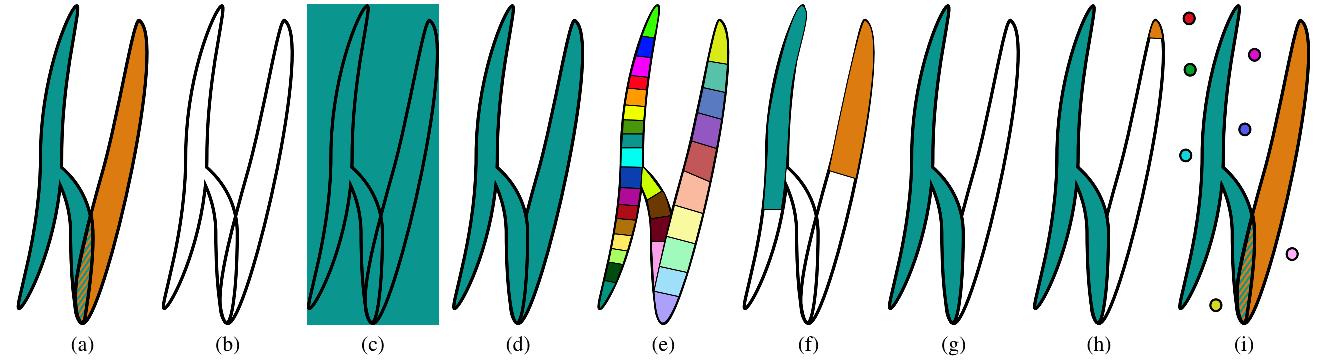
Table 2 summarizes all used metrics with their localization criterion and matching. Fig. 5 highlights some of the challenges of computing a consistent many-to-many matching for overlapping instances. Fig. 6 visualizes and quantifies a comprehensive set of different edge cases of our evaluation metrics.

Table 2. Overview of localization criterion and matching algorithm for used scores. Last line shows cardinalities of the ground truth-to-prediction relationships.

Score	avF1	C	FS	FM	clDice_{TP}
Loc.	clDice	clPrec.	clRecall	clRecall	clDice
Match.	greedy	greedy	greedy	greedy	greedy
	1:1	1:n	n:m	n:m	1:1

A.3. Extended Baseline Information and Results

We describe our three baseline methods in the following sections, namely PatchPerPix in Sec. A.3.1, Flood Filling Networks in Sec. A.3.2 and Duan et al.'s color clustering in Sec. A.3.3. The evaluation code is available here: https://github.com/Kainmueller-Lab/evaluate-instance-segmentation. Please see Table 3 and 4 for the extended quantitative results. Qualitative results for all three



	S	$avF1$	C	clDice$_{\text{TP}}$	#Pred	TP	FP	FS	FN	FM
(a)	1.0	1.0	1.0	1.0	2	2	0	0	0	0
(b)	0.0	0.0	0.0	0.0	0	0	0	0	2	0
(c)	0.0	0.0	0.0	0.0	1	0	1	0	2	1
(d)	0.47	0.44	0.5	0.67	1	1	0	0	1	1
(e)	0.5	0.0	1.0	0.0	31	0	31	30	2	0
(f)	0.58	0.67	0.51	0.68	2	2	0	0	0	0
(g)	0.58	0.67	0.5	1.0	1	1	0	0	1	0
(h)	0.5	0.5	0.52	1.0	2	1	1	0	1	0
(i)	0.68	0.36	1.0	1.0	9	2	7	0	0	0

	F1$_{0.1}$	F1$_{0.2}$	F1$_{0.3}$	F1$_{0.4}$	F1$_{0.5}$	F1$_{0.6}$	F1$_{0.7}$	F1$_{0.8}$	F1$_{0.9}$
(d)	0.67	0.67	0.67	0.67	0.67	0.67	0.0	0.0	0.0
(f)	1.0	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0
(g)	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67
(h)	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
(i)	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36

Figure 6. Quantitative assessment of a number of different edge cases of our evaluation metrics (outline: ground truth, color: predictions, th $= 0.5$), highlighting their applicability and validity for FISBe. In (a) we have a perfect prediction, the score is perfect and there are no errors. In (b) we have no prediction, the score is zero and we have as many FN as there are instances. In (c) we have one prediction that covers the whole image; as the clDice value is too low, there is no TP, so avF1 is still zero. When computing the clPrecision for the predicted instance, the corresponding skeleton will likely have the largest overlap with the ground truth background and will be matched to it. Thus, C will be zero as well. In (d) we have a perfect foreground segmentation but the two ground truth instances are merged; the predicted instance is assigned to one of the ground truth instances, resulting in C $= 0.5$. Assuming clDice $= 0.67$ for the one match (and thus clDice$_{\text{TP}}$ $= 0.67$), we have F1 $= 0.67$ for th < 0.7 and 0 otherwise. In (e) we again have a perfect foreground segmentation but there are many small instances; clDice for each pair of predicted and ground truth instances is < 0.1, thus avF1 $= 0$ and clDice$_{\text{TP}}$ $= 0$; however, C $= 1.0$ because both instances are completely covered (and multiple predicted instances can be matched to one ground truth instance). In (f), (g) and (h) overall slightly more than half of the total ground truth is covered; in (f) both instances are covered slightly more than half; in (g) one instance is covered completely and the other is not; in (h) one instance is covered completely and only a tiny part of the other; to distinguish the cases quantitatively, one has to look at the details: about the same amount of ground truth is covered, thus C has a similar value; in (f) clDice$_{\text{TP}}$ is worst as both predicted instances are counted as TP, yet both only cover just over half of their respective ground truth instance; furthermore, while avF1 is identical for (f) and (g), when looking at the full range of F1$_{\text{th}}$ values there are more differences: in (f) there are 2 TP for th < 0.7, resulting in F1 being equal to 1 for smaller thresholds and equal to 0 for larger thresholds; in (g) there is 1 TP for the full range of thresholds, but also 1 FN; in both cases this results in avF1 $= 0.67$; finally in (h) there is 1 TP for the full range of thresholds, 1 FN as in (g), but also 1 FP, resulting in avF1 $= 0.5$. In (i) we have a perfect prediction as in (a), but in addition we have a number of small FP, due to noise categorized as foreground; the coverage values are not affected, but the avF1 value drops. One could argue that (h) should be better than (g) as more is detected; however, if a prediction is too small, it is, in general, more likely to be noise. One could also argue that (d) should better than (g), as both neurons are detected, just merged; however, for downstream tasks having one fully correct instance that can directly be used is often more valuable than first having to manually fix errors.

baselines are shown in Fig. 7 and a visualization of typical error types for PatchPerPix is presented in Fig. 8.

A.3.1 PatchPerPix

We use PatchPerPix [13] with a 3-level 3d U-Net [1, 22] with 20 initial feature maps, tripled at each downsampling layer. The predicted patches are of size 7 × 7 × 7 pixels. We use the base model without the additional patch decoder. In addition to the patches the model is trained to predict how many instances there are per pixel (*numinst* in the code) modelled as a categorical prediction task with the categories: zero, one and more than one instance. We use PyTorch [19] in combination with gunpowder [6] for training with the following standard random augmentations: Elastic, Intensity, Flipping. We add the following augmentations: Overlay (overlaying two random image crops to simulate denser images), Permute (randomly permute color channels), Hue (random rotation of the color wheel). We train the model only on the completely labeled data as the training is not directly applicable to partly labeled data.

The models are trained for 300k iterations with a learning rate of 0.0001 using Adam[10], storing weight checkpoints every 10k iterations. We use a batch size of 2 and train on random crops. As most images are in large part background, we sample foreground and areas where neurons overlap with higher probability. The exact ratios depend on the model and are detailed in the respective provided configuration files. The training code is available here: https://github.com/Kainmueller-Lab/PatchPerPix.

We select the best checkpoint, the best patch threshold and the best threshold for the *numinst* prediction based on the validation set and report both the validation results and the final results on the test set (both combined and separately for the completely labeled and the partly labeled dataset). We observed that the models tend to overestimate the case of a single neuron in a given region and underestimate background and neuron overlaps. To counter this, instead of using a simple $\arg\max$, we additionally select an optimal threshold based on the validation results.

PatchPerPix can only handle overlaps up to the size of the patch size. As its instance assembly step is computationally demanding for 3d data, the currently applicable patch size is restricted. In order to be able to handle larger overlaps, PatchPerPix needs to be scaled up in future work.

A.3.2 Flood Filling Networks

For Flood Filling Networks (FFN) we mainly follow the proposed architecture from [7] and the publicly available code[2]. We use 12 stacked convolution modules with skip connections in between, where each convolution module consists of two 3d convolution layers. The field of view (FoV) size, which corresponds to the spatial dimensions of the network's input and output size, is 33×33×33 and we adapted the network to work with three input channel. FFNs move their current FoV by a short distance after each update to be able to trace the entire object. For this, we use the cuboid movement policy described in [8] with step size 8 for each dimension. We apply standard data augmentation by flipping and permuting spatial axis. We train the models for 2m iterations with batch size 4. The sampling strategy is the same as in the original work.

During training we use seeds (starting position of the FoV) generated from ground truth. For prediction we create the seeds as follows: We convert the three channel input to a grayscale volume and threshold it to obtain a foreground mask, where we filter out small connected components. Finally, we take local maxima on the corresponding distance transform map. We determine both thresholds (foreground and small connected components size) during validation. Aside from that, we choose the best checkpoint, the best FoV movement threshold and the best final segmentation threshold based on validation.

We train and test FFNs both on only the completely labeled dataset and on the full dataset. In contrast to PatchPerPix, FFNs only consider one instance at a time, which means that there are no changes necessary to train FFNs on partly labeled samples.

A.3.3 Color Clustering

Duan et al. [2] propose a non-learnt color clustering algorithm based on [24] to segment mouse neurons in Brainbow [11] images. Brainbow is a stochastic labeling technique to image neurons in unique colors with light microscopy. This assumption does not hold for our FISBe dataset, where multiple neurons and abundant noise can be of the same color. Thus, some steps of the pipeline (denoising, supervoxel generation, color clustering, linkage bridging) need to be adapted to fit our dataset.[3]

Following the original work, we denoise our 3d images with bm4d [12]. We use $\sigma = 0.05$ as noise standard deviation, and normalize and denoise each channel separately. For supervoxel generation, we threshold the denoised image with foreground threshold $t_{fg} = 0.08$, apply distance transform, and run watershed transform with local maxima as seeds and the thresholded foreground as mask. All connected components smaller than threshold $t_{rm} = 800$ are removed. In the next step, all supervoxels are clustered with Gaussian Mixture Models (GMM). We create an adjacency matrix where supervoxel pairs have a value > 0, if their spatial and color distance is smaller than certain thresholds

[2] original code: https://github.com/google/ffn; and adapted for FISBe: https://github.com/Kainmueller-Lab/ffn

[3] code: https://github.com/Kainmueller-Lab/brainbow

Table 3. Quantitative results of our baseline models on the *combined*, *completely* and *partly* labeled FISBe datasets. We train models both only on the completely labeled data (ppp, FFN), and on the completely and the partly labeled data (FFN+partly). Note that the scores are not directly comparable to each other across datasets (combined, completely, partly), but they are comparable across splits (val, test) and methods within each dataset. We report mean and standard deviation (±) over three independent runs (except for Duan et al.'s as it is non-learnt). For all scores except FS and FM higher values are better. Continued in Table 4.

Split	Method	S	$avF1$	C	$clDice_{TP}$	FS	FM	C_{dim}	C_{ovlp}	tp	tp_{dim}	tp_{ovlp}
						Combined						
Val	ppp	$0.38_{\pm0.02}$	$0.41_{\pm0.02}$	$0.35_{\pm0.01}$	$0.75_{\pm0.02}$	$6.0_{\pm0.8}$	$24_{\pm1.6}$	$0.12_{\pm0.01}$	$0.38_{\pm0.04}$	$0.46_{\pm0.01}$	$0.16_{\pm0.04}$	$0.39_{\pm0.03}$
	FFN	$0.25_{\pm0.01}$	$0.27_{\pm0.01}$	$0.23_{\pm0.01}$	$0.79_{\pm0.01}$	$7.0_{\pm2.9}$	$12_{\pm2.0}$	$0.03_{\pm0.01}$	$0.30_{\pm0.01}$	$0.32_{\pm0.01}$	$0.04_{\pm0.01}$	$0.37_{\pm0.02}$
	FFN+partly	$0.27_{\pm0.01}$	$0.29_{\pm0.01}$	$0.24_{\pm0.01}$	$0.79_{\pm0.01}$	$7.7_{\pm2.6}$	$14_{\pm0.8}$	$0.02_{\pm0.01}$	$0.33_{\pm0.02}$	$0.34_{\pm0.03}$	$0.03_{\pm0.00}$	$0.38_{\pm0.04}$
	Duan et al.	0.24	0.26	0.22	0.70	14	13	0.02	0.28	0.37	0.03	0.42
Test	ppp	$0.35_{\pm0.00}$	$0.34_{\pm0.01}$	$0.35_{\pm0.01}$	$0.80_{\pm0.00}$	$19_{\pm2.9}$	$52_{\pm3.4}$	$0.16_{\pm0.03}$	$0.27_{\pm0.04}$	$0.36_{\pm0.01}$	$0.19_{\pm0.04}$	$0.19_{\pm0.03}$
	FFN	$0.25_{\pm0.03}$	$0.22_{\pm0.04}$	$0.29_{\pm0.02}$	$0.80_{\pm0.01}$	$17_{\pm1.7}$	$39_{\pm5.3}$	$0.03_{\pm0.01}$	$0.26_{\pm0.03}$	$0.32_{\pm0.03}$	$0.00_{\pm0.00}$	$0.24_{\pm0.05}$
	FFN+partly	$0.27_{\pm0.01}$	$0.24_{\pm0.02}$	$0.31_{\pm0.00}$	$0.80_{\pm0.01}$	$18_{\pm3.7}$	$36_{\pm3.6}$	$0.04_{\pm0.01}$	$0.28_{\pm0.01}$	$0.36_{\pm0.01}$	$0.03_{\pm0.00}$	$0.28_{\pm0.01}$
	Duan et al.	0.30	0.27	0.33	0.77	45	29	0.03	0.36	0.37	0.03	0.34
						Completely						
Val	ppp	$0.30_{\pm0.03}$	$0.34_{\pm0.04}$	$0.27_{\pm0.03}$	$0.72_{\pm0.02}$	$1.7_{\pm1.7}$	$3.7_{\pm1.3}$	$0.07_{\pm0.01}$	$0.41_{\pm0.07}$	$0.37_{\pm0.06}$	$0.06_{\pm0.04}$	$0.47_{\pm0.12}$
	FFN	$0.18_{\pm0.01}$	$0.21_{\pm0.01}$	$0.15_{\pm0.02}$	$0.78_{\pm0.02}$	$0.3_{\pm0.5}$	$0.0_{\pm0.0}$	$0.10_{\pm0.00}$	$0.28_{\pm0.02}$	$0.21_{\pm0.02}$	$0.00_{\pm0.00}$	$0.40_{\pm0.00}$
	FFN+partly	$0.20_{\pm0.01}$	$0.24_{\pm0.02}$	$0.17_{\pm0.00}$	$0.81_{\pm0.02}$	$0.7_{\pm0.5}$	$0.3_{\pm0.5}$	$0.00_{\pm0.00}$	$0.32_{\pm0.01}$	$0.22_{\pm0.02}$	$0.00_{\pm0.00}$	$0.40_{\pm0.00}$
	Duan et al.	0.16	0.17	0.15	0.65	2	1	0.00	0.25	0.24	0.00	0.40
Test	ppp	$0.34_{\pm0.02}$	$0.29_{\pm0.04}$	$0.40_{\pm0.02}$	$0.81_{\pm0.02}$	$3.0_{\pm0.8}$	$4.3_{\pm1.3}$	$0.14_{\pm0.05}$	$0.42_{\pm0.03}$	$0.45_{\pm0.01}$	$0.19_{\pm0.09}$	$0.38_{\pm0.10}$
	FFN	$0.18_{\pm0.04}$	$0.11_{\pm0.08}$	$0.26_{\pm0.01}$	$0.77_{\pm0.05}$	$2.0_{\pm0.8}$	$2.0_{\pm1.4}$	$0.02_{\pm0.02}$	$0.24_{\pm0.03}$	$0.31_{\pm0.06}$	$0.00_{\pm0.00}$	$0.25_{\pm0.10}$
	FFN+partly	$0.19_{\pm0.02}$	$0.10_{\pm0.03}$	$0.29_{\pm0.02}$	$0.80_{\pm0.01}$	$2.3_{\pm0.5}$	$1.7_{\pm0.5}$	$0.03_{\pm0.01}$	$0.32_{\pm0.04}$	$0.34_{\pm0.06}$	$0.02_{\pm0.03}$	$0.42_{\pm0.12}$
	Duan et al.	0.28	0.23	0.33	0.81	6	1	0.02	0.43	0.38	0.00	0.50
						Partly						
Val	ppp	$0.48_{\pm0.00}$	$0.52_{\pm0.00}$	$0.45_{\pm0.01}$	$0.79_{\pm0.00}$	$3.7_{\pm0.9}$	$20_{\pm0.0}$	$0.19_{\pm0.01}$	$0.35_{\pm0.01}$	$0.51_{\pm0.0}$	$0.25_{\pm0.02}$	$0.37_{\pm0.00}$
	FFN	$0.34_{\pm0.01}$	$0.36_{\pm0.01}$	$0.32_{\pm0.01}$	$0.79_{\pm0.01}$	$7.3_{\pm2.9}$	$12_{\pm1.6}$	$0.06_{\pm0.02}$	$0.33_{\pm0.00}$	$0.37_{\pm0.02}$	$0.06_{\pm0.02}$	$0.39_{\pm0.03}$
	FFN+partly	$0.34_{\pm0.02}$	$0.36_{\pm0.03}$	$0.32_{\pm0.01}$	$0.77_{\pm0.01}$	$8.0_{\pm1.6}$	$13_{\pm0.9}$	$0.04_{\pm0.02}$	$0.34_{\pm0.02}$	$0.38_{\pm0.03}$	$0.03_{\pm0.02}$	$0.39_{\pm0.04}$
	Duan et al.	0.32	0.35	0.30	0.74	12	12	0.04	0.31	0.41	0.04	0.43
Test	ppp	$0.35_{\pm0.01}$	$0.40_{\pm0.00}$	$0.31_{\pm0.02}$	$0.79_{\pm0.01}$	$15_{\pm1.9}$	$46_{\pm1.7}$	$0.15_{\pm0.02}$	$0.15_{\pm0.02}$	$0.33_{\pm0.01}$	$0.13_{\pm0.02}$	$0.17_{\pm0.02}$
	FFN	$0.33_{\pm0.02}$	$0.35_{\pm0.02}$	$0.32_{\pm0.02}$	$0.80_{\pm0.01}$	$16_{\pm2.5}$	$37_{\pm4.0}$	$0.03_{\pm0.01}$	$0.23_{\pm0.03}$	$0.34_{\pm0.02}$	$0.00_{\pm0.00}$	$0.23_{\pm0.04}$
	FFN+partly	$0.34_{\pm0.02}$	$0.36_{\pm0.03}$	$0.33_{\pm0.02}$	$0.82_{\pm0.01}$	$17_{\pm2.5}$	$35_{\pm3.7}$	$0.04_{\pm0.01}$	$0.25_{\pm0.03}$	$0.34_{\pm0.02}$	$0.05_{\pm0.00}$	$0.25_{\pm0.01}$
	Duan et al.	0.33	0.32	0.34	0.73	39	28	0.04	0.28	0.37	0.05	0.32

($\delta_s = 5$, $\delta_c = 14$). Moreover, we use the Bayes Information Criterion (BIC) to determine the number of clusters for the GMM clustering. Finally, as same colored, but not touching neurons are clustered together, we apply connected component analysis for each GMM cluster with a distance threshold ($\Delta_s = 20$). Please note, that differing from the original works, we omit supervoxel subdivision and merging, PCA as well as linking bridging, because these steps did not improve performance for our dataset. We determined σ, t_{fg}, t_{rm}, δ_s, δ_c and Δ_s during validation.

A.4. Biological Background and Motivation

This section gives a brief overview of the advances our dataset will facilitate in the field of basic neuroscience. Based on neuron instance segmentations in MCFO images, neurons can be studied threefold: (1) Clustering and subse-

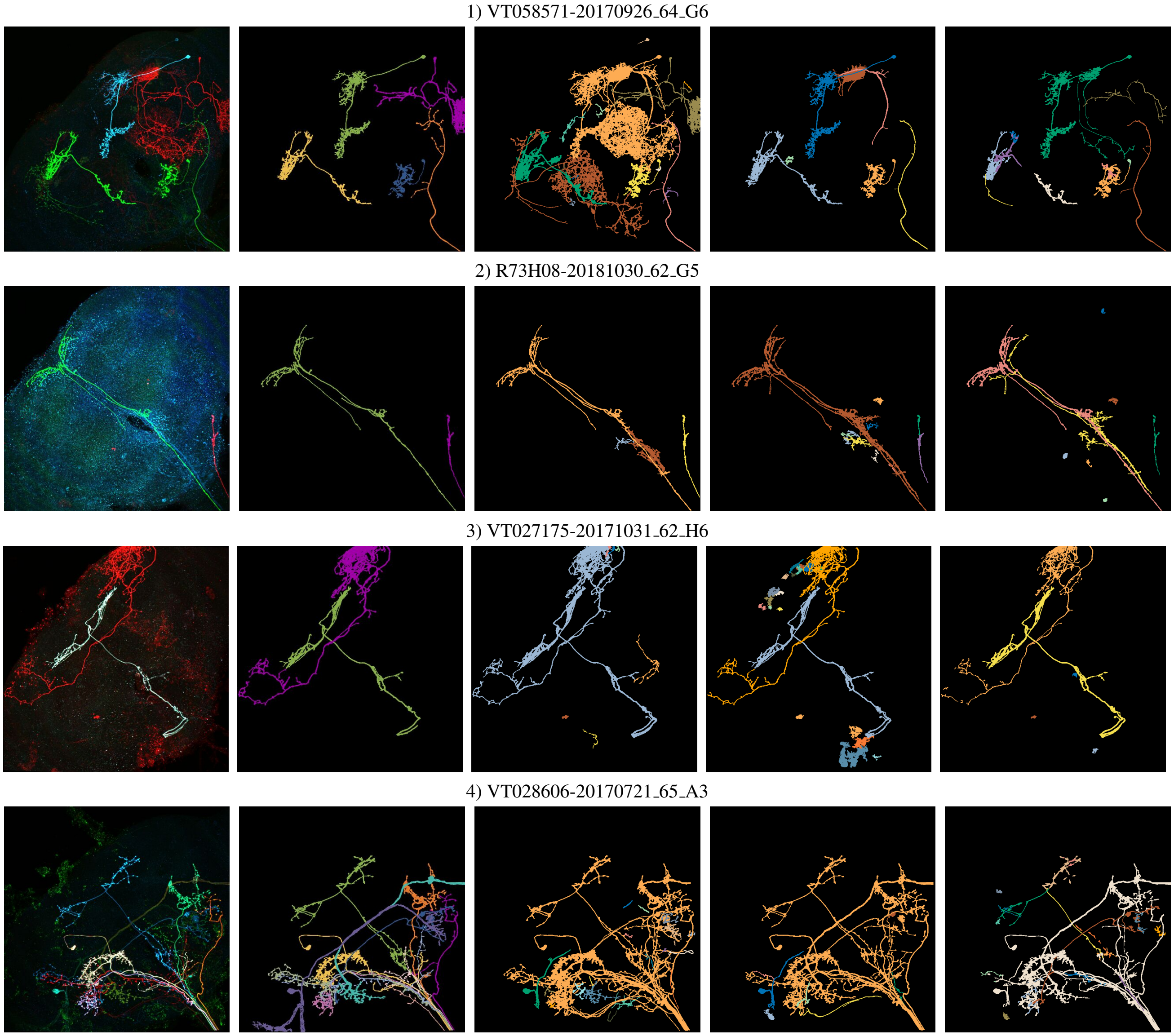
Table 4. Quantitative results of our baseline models on *combined*, *completely* and *partly* labeled FISBe datasets (continuation of Table 3).

Split	Method	$\mathrm{F1}_{0.1}$	$\mathrm{F1}_{0.2}$	$\mathrm{F1}_{0.3}$	$\mathrm{F1}_{0.4}$	$\mathrm{F1}_{0.5}$	$\mathrm{F1}_{0.6}$	$\mathrm{F1}_{0.7}$	$\mathrm{F1}_{0.8}$	$\mathrm{F1}_{0.9}$
						Combined				
Val	ppp	$0.62_{\pm 0.03}$	$0.57_{\pm 0.03}$	$0.53_{\pm 0.01}$	$0.50_{\pm 0.02}$	$0.49_{\pm 0.02}$	$0.43_{\pm 0.01}$	$0.28_{\pm 0.03}$	$0.22_{\pm 0.05}$	$0.10_{\pm 0.02}$
	FFN	$0.38_{\pm 0.01}$	$0.36_{\pm 0.01}$	$0.34_{\pm 0.01}$	$0.32_{\pm 0.00}$	$0.30_{\pm 0.01}$	$0.27_{\pm 0.01}$	$0.22_{\pm 0.02}$	$0.17_{\pm 0.01}$	$0.07_{\pm 0.01}$
	FFN+partly	$0.40_{\pm 0.02}$	$0.38_{\pm 0.02}$	$0.37_{\pm 0.02}$	$0.35_{\pm 0.01}$	$0.34_{\pm 0.01}$	$0.31_{\pm 0.01}$	$0.24_{\pm 0.01}$	$0.18_{\pm 0.01}$	$0.07_{\pm 0.01}$
	Duan et al.	0.38	0.37	0.34	0.33	0.33	0.27	0.18	0.09	0.03
Test	ppp	$0.50_{\pm 0.01}$	$0.48_{\pm 0.01}$	$0.44_{\pm 0.01}$	$0.41_{\pm 0.02}$	$0.35_{\pm 0.02}$	$0.29_{\pm 0.02}$	$0.26_{\pm 0.01}$	$0.19_{\pm 0.02}$	$0.12_{\pm 0.01}$
	FFN	$0.34_{\pm 0.05}$	$0.31_{\pm 0.04}$	$0.28_{\pm 0.04}$	$0.25_{\pm 0.05}$	$0.22_{\pm 0.04}$	$0.20_{\pm 0.04}$	$0.17_{\pm 0.03}$	$0.12_{\pm 0.01}$	$0.07_{\pm 0.01}$
	FFN+partly	$0.36_{\pm 0.02}$	$0.32_{\pm 0.02}$	$0.30_{\pm 0.02}$	$0.27_{\pm 0.03}$	$0.25_{\pm 0.03}$	$0.21_{\pm 0.03}$	$0.18_{\pm 0.02}$	$0.15_{\pm 0.02}$	$0.09_{\pm 0.01}$
	Duan et al.	0.43	0.38	0.35	0.33	0.31	0.29	0.20	0.12	0.06
						Completely				
Val	ppp	$0.57_{\pm 0.07}$	$0.50_{\pm 0.05}$	$0.46_{\pm 0.02}$	$0.43_{\pm 0.05}$	$0.41_{\pm 0.07}$	$0.34_{\pm 0.03}$	$0.19_{\pm 0.02}$	$0.14_{\pm 0.07}$	$0.04_{\pm 0.02}$
	FFN	$0.31_{\pm 0.04}$	$0.28_{\pm 0.04}$	$0.26_{\pm 0.03}$	$0.24_{\pm 0.01}$	$0.24_{\pm 0.01}$	$0.22_{\pm 0.03}$	$0.16_{\pm 0.03}$	$0.13_{\pm 0.01}$	$0.04_{\pm 0.03}$
	FFN+partly	$0.32_{\pm 0.03}$	$0.30_{\pm 0.01}$	$0.28_{\pm 0.02}$	$0.26_{\pm 0.04}$	$0.26_{\pm 0.04}$	$0.24_{\pm 0.03}$	$0.23_{\pm 0.02}$	$0.15_{\pm 0.02}$	$0.08_{\pm 0.03}$
	Duan et al.	0.24	0.24	0.24	0.24	0.24	0.20	0.10	0.00	0.00
Test	ppp	$0.40_{\pm 0.04}$	$0.38_{\pm 0.03}$	$0.37_{\pm 0.03}$	$0.34_{\pm 0.05}$	$0.30_{\pm 0.04}$	$0.27_{\pm 0.06}$	$0.23_{\pm 0.05}$	$0.18_{\pm 0.04}$	$0.11_{\pm 0.01}$
	FFN	$0.16_{\pm 0.11}$	$0.16_{\pm 0.11}$	$0.15_{\pm 0.10}$	$0.14_{\pm 0.10}$	$0.12_{\pm 0.08}$	$0.09_{\pm 0.06}$	$0.08_{\pm 0.05}$	$0.05_{\pm 0.04}$	$0.02_{\pm 0.02}$
	FFN+partly	$0.15_{\pm 0.04}$	$0.15_{\pm 0.04}$	$0.14_{\pm 0.04}$	$0.12_{\pm 0.04}$	$0.11_{\pm 0.04}$	$0.10_{\pm 0.03}$	$0.08_{\pm 0.02}$	$0.07_{\pm 0.03}$	$0.03_{\pm 0.01}$
	Duan et al.	0.31	0.29	0.27	0.27	0.27	0.27	0.20	0.14	0.06
						Partly				
Val	ppp	$0.73_{\pm 0.01}$	$0.68_{\pm 0.01}$	$0.65_{\pm 0.01}$	$0.61_{\pm 0.01}$	$0.59_{\pm 0.00}$	$0.54_{\pm 0.02}$	$0.41_{\pm 0.03}$	$0.28_{\pm 0.00}$	$0.19_{\pm 0.01}$
	FFN	$0.49_{\pm 0.01}$	$0.47_{\pm 0.01}$	$0.45_{\pm 0.02}$	$0.41_{\pm 0.01}$	$0.40_{\pm 0.01}$	$0.37_{\pm 0.01}$	$0.29_{\pm 0.03}$	$0.22_{\pm 0.02}$	$0.10_{\pm 0.02}$
	FFN+partly	$0.50_{\pm 0.04}$	$0.47_{\pm 0.04}$	$0.46_{\pm 0.05}$	$0.42_{\pm 0.05}$	$0.41_{\pm 0.06}$	$0.36_{\pm 0.05}$	$0.26_{\pm 0.02}$	$0.22_{\pm 0.00}$	$0.09_{\pm 0.02}$
	Duan et al.	0.51	0.49	0.44	0.42	0.42	0.34	0.25	0.18	0.06
Test	ppp	$0.62_{\pm 0.02}$	$0.58_{\pm 0.01}$	$0.51_{\pm 0.01}$	$0.47_{\pm 0.00}$	$0.40_{\pm 0.01}$	$0.33_{\pm 0.01}$	$0.28_{\pm 0.01}$	$0.21_{\pm 0.02}$	$0.17_{\pm 0.01}$
	FFN	$0.55_{\pm 0.04}$	$0.49_{\pm 0.01}$	$0.44_{\pm 0.02}$	$0.40_{\pm 0.02}$	$0.36_{\pm 0.02}$	$0.32_{\pm 0.03}$	$0.27_{\pm 0.02}$	$0.21_{\pm 0.01}$	$0.11_{\pm 0.01}$
	FFN+partly	$0.57_{\pm 0.06}$	$0.51_{\pm 0.04}$	$0.47_{\pm 0.03}$	$0.40_{\pm 0.04}$	$0.36_{\pm 0.03}$	$0.31_{\pm 0.03}$	$0.28_{\pm 0.03}$	$0.23_{\pm 0.02}$	$0.14_{\pm 0.02}$
	Duan et al.	0.55	0.48	0.43	0.39	0.35	0.31	0.19	0.09	0.06

quent statistical analysis of neuron morphologies may yield insights into neuronal cell types and their variability [3, 4]. (2) Locating a neuron morphology of interest in multiple MCFO images facilitates the creation of novel transgenic lines that sparsely express the neuron of interest, which in turn facilitates functional analyses of individual neurons of interest in vivo [20]. (3) Information on neuron connectivity and neuron function can be fused by locating neurons segmented from electron microscopy (EM) data in MCFO images and subsequent in vivo studies as in (2) [15, 16, 25]. For these tasks, instance segmentations do not necessarily need to cover all true neurons: Given that MCFO images express only a random subset of neurons in the first place, missing some dim neurons in an instance segmentation, while further reducing recall, does not introduce a categorically new source of error. More specifically, segmenting a subset of neurons with high individual clDice score is preferable to segmenting all neurons but only partly. We acknowledge these application-derived preferences in our selection of metrics (see Sec. 3 of the main paper).

A.5. Sample Information and Visualization

Table 5 provides a list of the included MCFO acquisitions in the completely labeled FISBe dataset including informa-



1) VT058571-20170926_64_G6

2) R73H08-20181030_62_G5

3) VT027175-20171031_62_H6

4) VT028606-20170721_65_A3

(a) MCFO (b) gt (c) ppp (d) FFN (e) Duan et al.

Figure 7. Qualitative results for our three baseline methods: PatchPerPix (ppp), Flood Filling Networks (FFN) and Duan et al.'s color clustering. The columns depict the following: (a): Maximum intensity projection of MCFO sample, (b): ground truth segmentation, (c): ppp prediction, (d): FFN prediction, and (e): Duan et al.'s result. In (1) all three methods yield some correctly segmented neurons, but ppp merges the blue one and one of the red ones, FFN does not segment most of the red ones and Duan et al.'s merges the blue neuron with parts of the red ones; FFN and Duan et al.'s have lower coverage. In (2) the noisy blue channel leads to false positives. In (3) ppp merges the two neurons whereas FFN and Duan et al.'s split them correctly; FFN additionally segments some noise. In (4) all three methods merge multiple neurons of different color; Duan et al.'s has lower coverage.

tion on the split (train/val/test) in which each sample was used. Table 6 provides a list of the included MCFO acquisitions in the partly labeled FISBe dataset including information on the split (train/val/test) in which each sample was used. Fig. 9 shows orthographic view for an exemplary sample. It highlights the thin structures and overall sparseness of foreground. Fig. 10 shows maximum intensity projections together with the gt instance segmentation of all samples in the completely labeled set, separated by train/val/test. Fig. 12 shows maximum intensity projections together with the gt instance segmentation of all samples in the partly labeled set, separated by train/val/test.

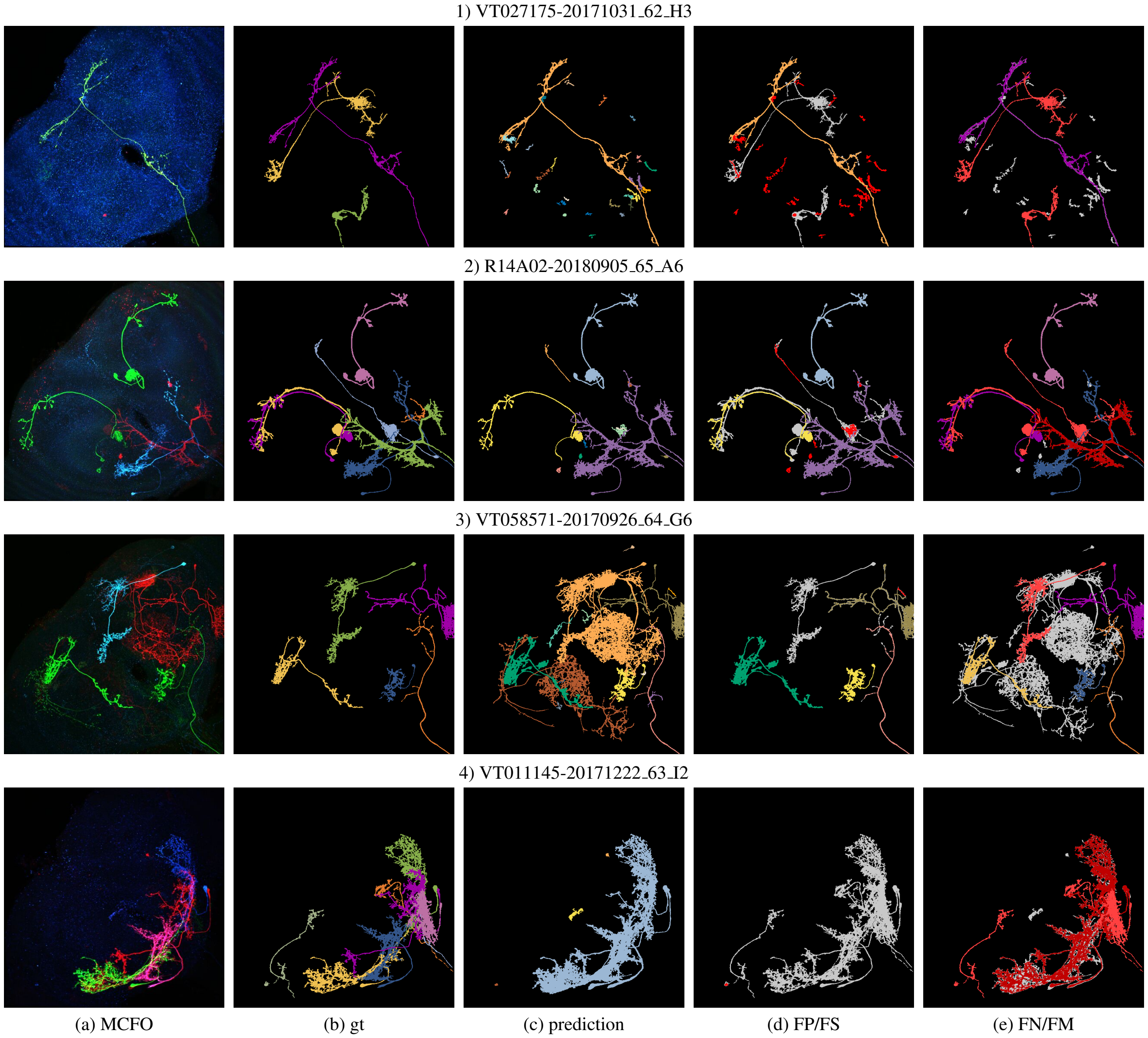

1) VT027175-20171031_62_H3
2) R14A02-20180905_65_A6
3) VT058571-20170926_64_G6
4) VT011145-20171222_63_I2
(a) MCFO
(b) gt
(c) prediction
(d) FP/FS
(e) FN/FM

Figure 8. Visualization of the results of our PatchPerPix baseline model for four samples of our completely labeled test set, ground truth-prediction matches are shown for a clDice threshold of 0.5. ((a): Maximum intensity projection of MCFO sample; (b): ground truth segmentation; (c): predicted segmentation; (d): TP predicted instances in same color code as in (c), FP and FS in red, ground truth in grey; (e): TP ground truth instances in same color code as in (b), FN in light red, FM in dark red, prediction mask in grey.) In (1) the bright green neuron is nicely segmented (see (c), in orange). However, there are two more, very dim neurons in the image, these were missed (see (e) in red). In addition, there is a large number of FP (see (d) in red). In (2) the bright green ones are again nicely segmented (see (c), in yellow and light blue). The purple prediction (see (c)) covers most of the blue and the red neurons, unfortunately resulting in a false merge (FM), despite having very different colors. The blue one still counts as a TP, the red one though as a FN, more precisely a FM (shown in dark red in (e)). There is a very dim red neuron next to the left green one that is missed completely (shown in bright red in (e)). There is a dim blue neuron between the two green ones that is mostly missed resulting in a few FS (shown in red in (d)). In (3) there are a number of unlabeled neurons. There are two bright red neurons, only one is labeled (shown by the absence of a label in (b)). There are a couple of somewhat dim neurons of different colors (there are still visible relatively well when zooming in). We can see that our prediction, as desired, includes the unlabeled neurons (see (c)). We can also see that, as they are not shown in (d) and again as desired, they are counted neither as TP nor as FP. There are again some FM, the bright blue neuron is segmented well but unfortunately the prediction is merged with other neurons (note that it is not shown in dark red (FM) in (e) as it is merged with unlabeled neurons, thus it is not possible to automatically tag it as a FM). (4) shows an extreme FM case. There is a cluster of multiple overlapping bright neurons in different colors (see (a) and (b)). In the prediction they are all merged (shown in dark red in (e)), thus there is no TP (shown by the absence of colored segmentation masks in (d)). In addition there are a number of dim neurons that have been overlooked by the model (shown in bright red in (e)).

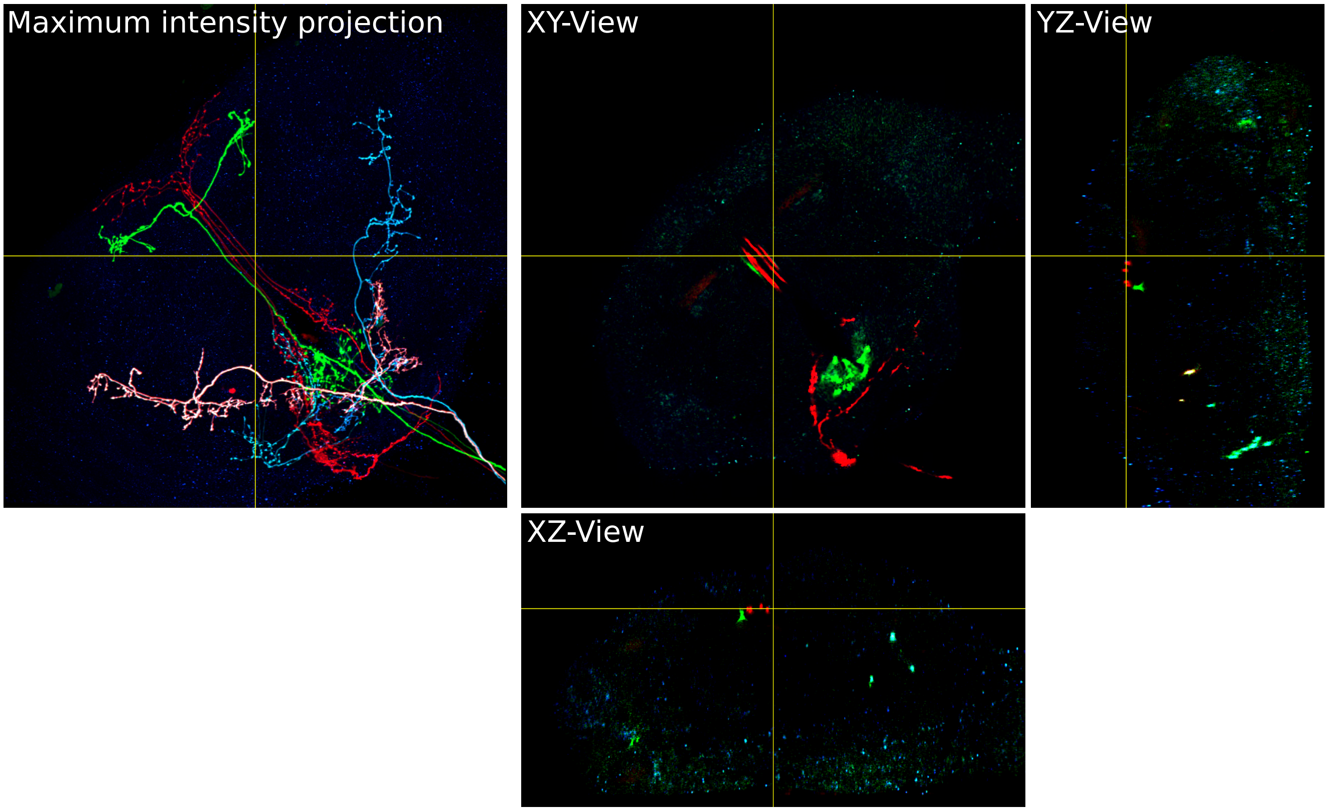


Figure 9. Maximum intensity projection and orthographic views for sample VT007080-20170517_61_A2. The orthographic views highlight the overall sparseness of the foreground and thinness of the neuronal structures.

Table 5. List of MCFO acquisitions in completely labeled FlyLight Instance Segmentation dataset with sample name (<*GAL4 line*>-<*slide code*>), number of annotated neurons, density category and split.

Sample name	Neurons	Cat.	Split	Sample name	Neurons	Cat.	Split
R38F04-20181005_63_G3	2	2	train	VT047848-20171020_66_J2	5	2	train
R38F04-20181005_63_G5	3	2	train	VT047848-20171020_66_I5	12	2	train
R38F04-20181005_63_H1	3	2	train	VT061467-20180911_62_E5	4	2	train
R53A10-20181019_64_A4	1	2	train	R22C03-20180918_66_J2	2	2	val
R75E01-20181030_64_D1	3	2	train	VT012403-20171128_61_B2	5	2	val
VT008647-20171222_63_D2	6	3	train	VT033614-20171124_64_H5	3	3	val
VT008647-20171222_63_D1	7	3	train	VT033614-20171124_64_H1	4	3	val
VT008647-20171222_63_E1	8	3	train	VT041298-20171114_63_C3	7	2	val
VT019303-20171013_65_B6	3	2	train	JRC_SS04989-20160318_24_A2	3	2	test
VT019307-20171013_65_F1	6	3	train	R14A02-20180905_65_A6	7	3	test
VT033051-20171128_61_E4	2	2	train	R54A09-20181019_64_H1	1	2	test
VT033051-20171128_61_E2	4	2	train	VT011145-20171222_63_I1	9	3	test
VT040433-20170919_63_D6	8	2	train	VT027175-20171031_62_H3	3	2	test
VT047848-20171020_66_I3	3	2	train	VT027175-20171031_62_H4	6	2	test
VT047848-20171020_66_I2	4	2	train	VT050157-20171110_61_C1	5	2	test

Table 6. List of MCFO acquisitions in partly labeled FlyLight Instance Segmentation dataset with sample name (<*GAL4 line*>-<*slide code*>), number of annotated neurons, density category and split.

Sample name	Neurons	Cat.	Split
R14B11-20180905_65_D2	5	2	train
R14B11-20180905_65_D6	9	2	train
R24D12-20180921_65_J6	5	3	train
R38F04-20181005_63_G2	1	2	train
R38F04-20181005_63_G4	2	2	train
VT003236-20170602_62_G4	6	3	train
VT003236-20170602_62_G5	6	3	train
VT007080-20170517_61_A2	4	2	train
VT007080-20170517_61_A4	10	2	train
VT007080-20170517_61_A5	15	2	train
VT008135-20171122_61_C2	4	2	train
VT008647-20171222_63_D5	6	3	train
VT008647-20171222_63_D6	7	3	train
VT010264-20171222_63_H2	12	3	train
VT010264-20171222_63_H5	19	3	train
VT011049-20180918_66_I1	2	1	train
VT024641-20170615_62_D2	7	2	train
VT024641-20170615_62_D3	4	2	train
VT024641-20170615_62_D5	5	2	train
VT024641-20170615_62_D6	10	2	train
VT024641-20170615_62_E1	4	2	train
VT025523-20170915_64_I1	11	2	train
VT026776-20171017_62_J1	13	3	train
VT033051-20171128_61_E3	1	2	train
VT033296-20171010_62_B4	4	2	train
VT034391-20171128_61_G2	2	2	train
VT038149-20171103_62_F1	6	3	train
VT039484-20171020_64_C1	7	3	train
VT039484-20171020_64_C2	12	3	train
VT040430-20170919_63_C4	3	2	train
VT040433-20170919_63_E1	6	2	train
VT045568-20171020_66_C5	4	2	train
VT045568-20171020_66_D2	3	2	train
VT047848-20171020_66_I1	6	2	train
VT047848-20171020_66_I4	8	2	train
VT047848-20171020_66_J1	8	2	train
VT050217-20171110_61_D6	5	2	train
VT050217-20171110_61_E1	6	2	train
VT058568-20170926_64_E1	13	3	train
VT060731-20170517_63_F1	6	2	train
VT060731-20170517_63_F2	7	2	train
VT061467-20180911_62_E4	1	2	train
VT062059-20170727_61_D4	6	2	train
JRC_SS05008-20160318_24_B1	4		val
JRC_SS05008-20160318_24_B2	6		val
R22C03-20180918_66_J1	3	2	val
R9F03-20181030_62_B5	3	2	val
VT008194-20171222_63_A3	13	2	val
VT008194-20171222_63_A5	17	2	val
VT012403-20171128_61_B1	6	2	val
VT033614-20171124_64_H4	2	3	val
VT039350-20171020_64_A1	11	3	val
VT039350-20171020_64_A3	8	3	val
VT039350-20171020_64_A6	5	3	val
VT059775-20170630_63_D5	7	2	val
R54A09-20181019_64_H4	4	2	test
R54A09-20181019_64_H6	1	2	test
R73H08-20181030_62_G5	2	2	test
VT006202-20170511_63_C4	8	2	test
VT011145-20171222_63_I2	8	3	test
VT021537-20171003_61_C3	5	3	test
VT023747-20171017_61_F1	10	2	test
VT027175-20171031_62_H6	2	2	test
VT028606-20170721_65_A2	14	3	test
VT028606-20170721_65_A3	12	3	test
VT033453-20170721_65_D2	7	2	test
VT033453-20170721_65_D4	7	2	test
VT033453-20170721_65_D5	5	2	test
VT046838-20170922_62_A2	8	2	test
VT050157-20171110_61_C5	3	2	test
VT058571-20170926_64_G6	5	2	test

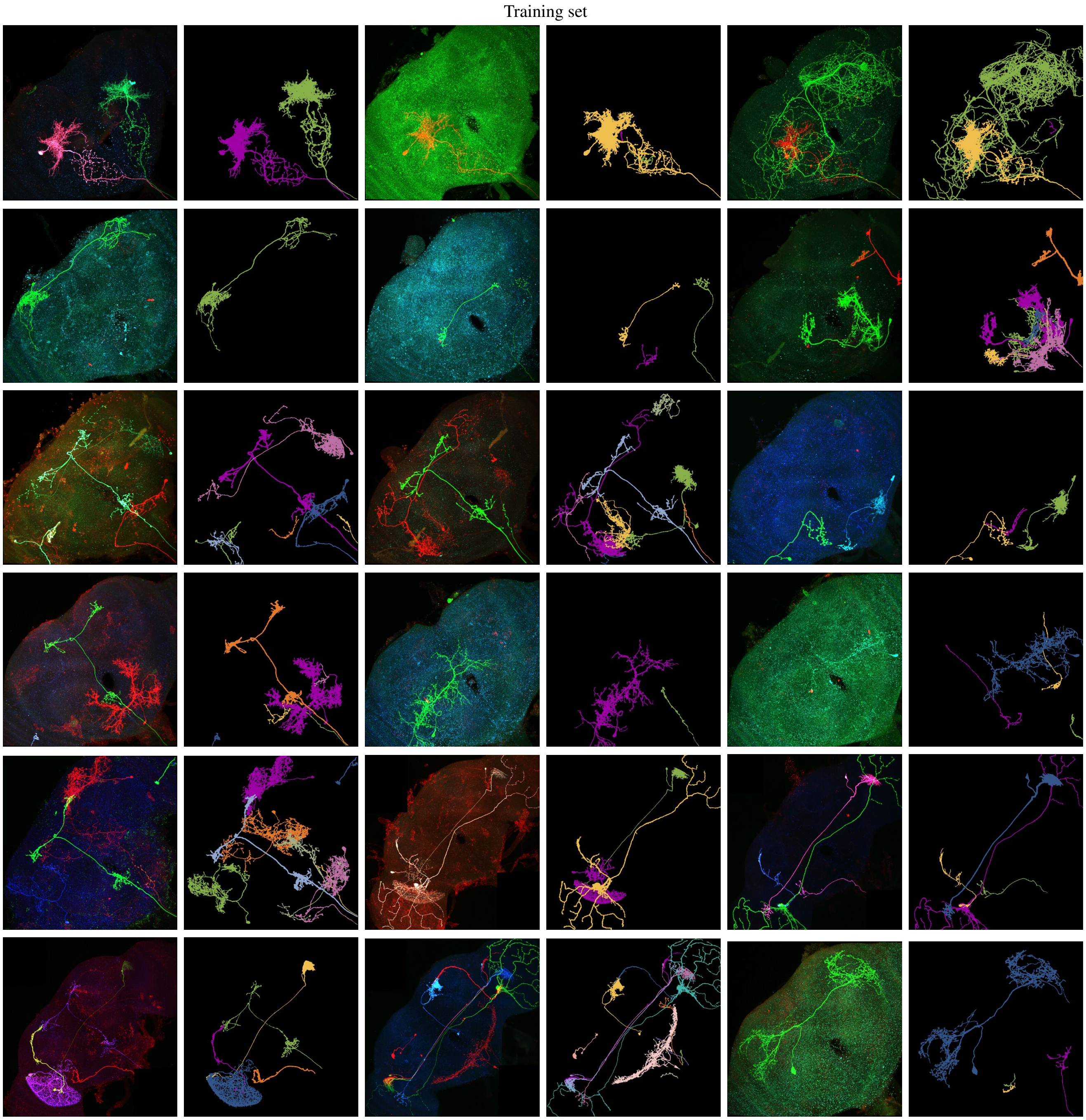
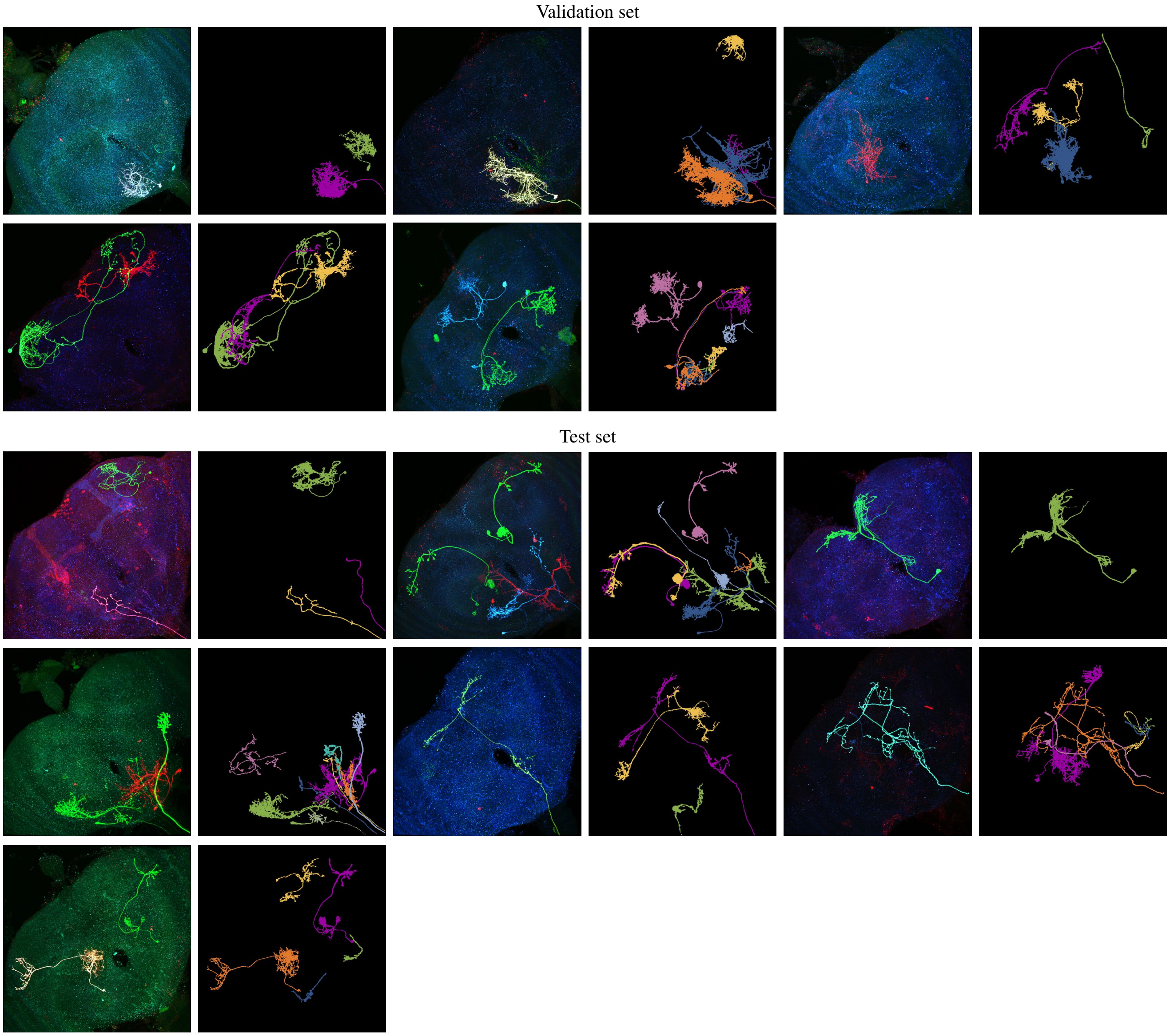

Training set

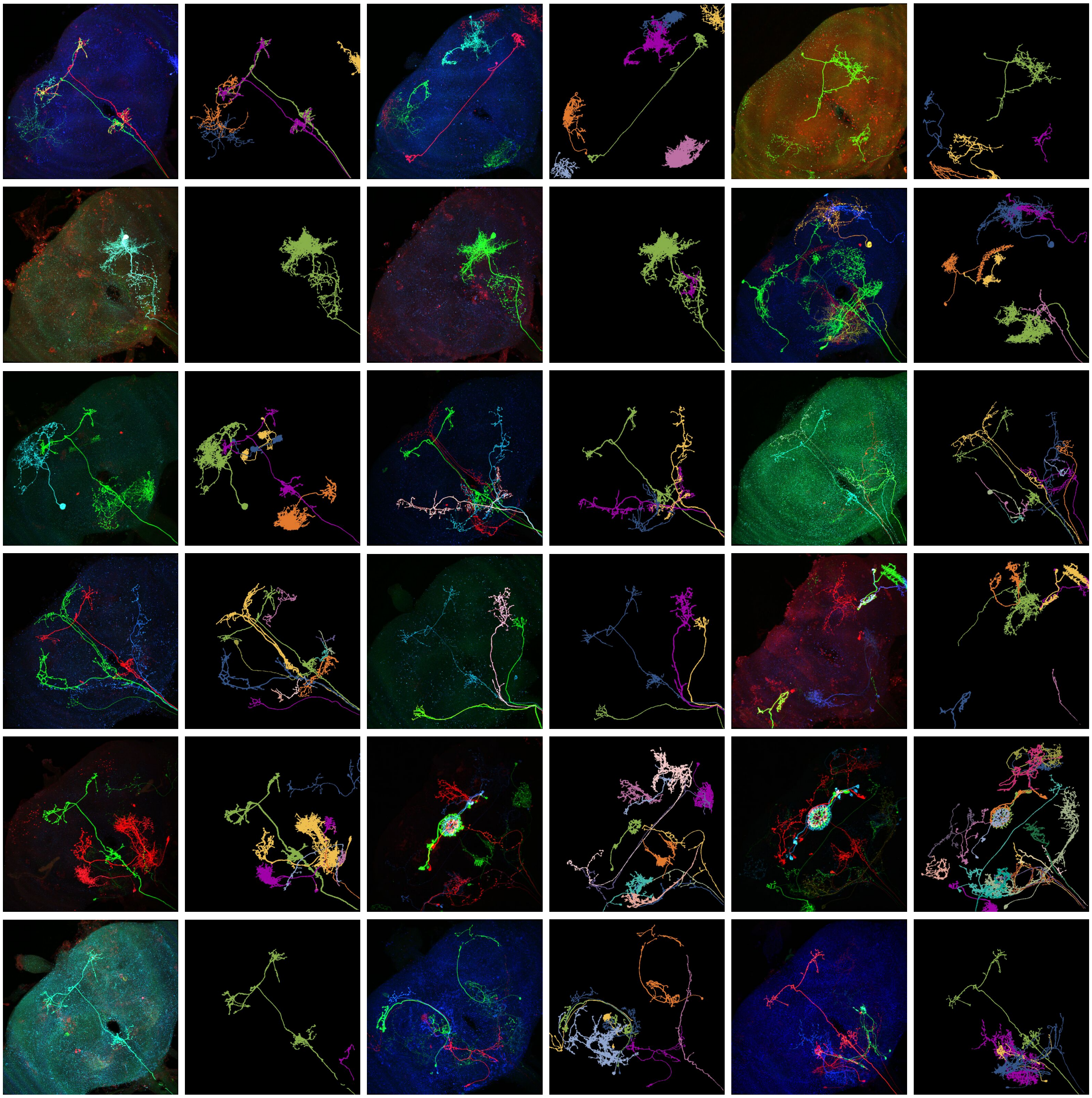
Figure 10. Maximum intensity projections (MIP) of 3d light microscopy samples and ground truth (gt) instance segmentations of all samples in the completely labeled set. MIP and gt are depicted next to each other in alternating order. Images are scaled to same width, some images are center cropped. Figure continued on next page.



Validation set

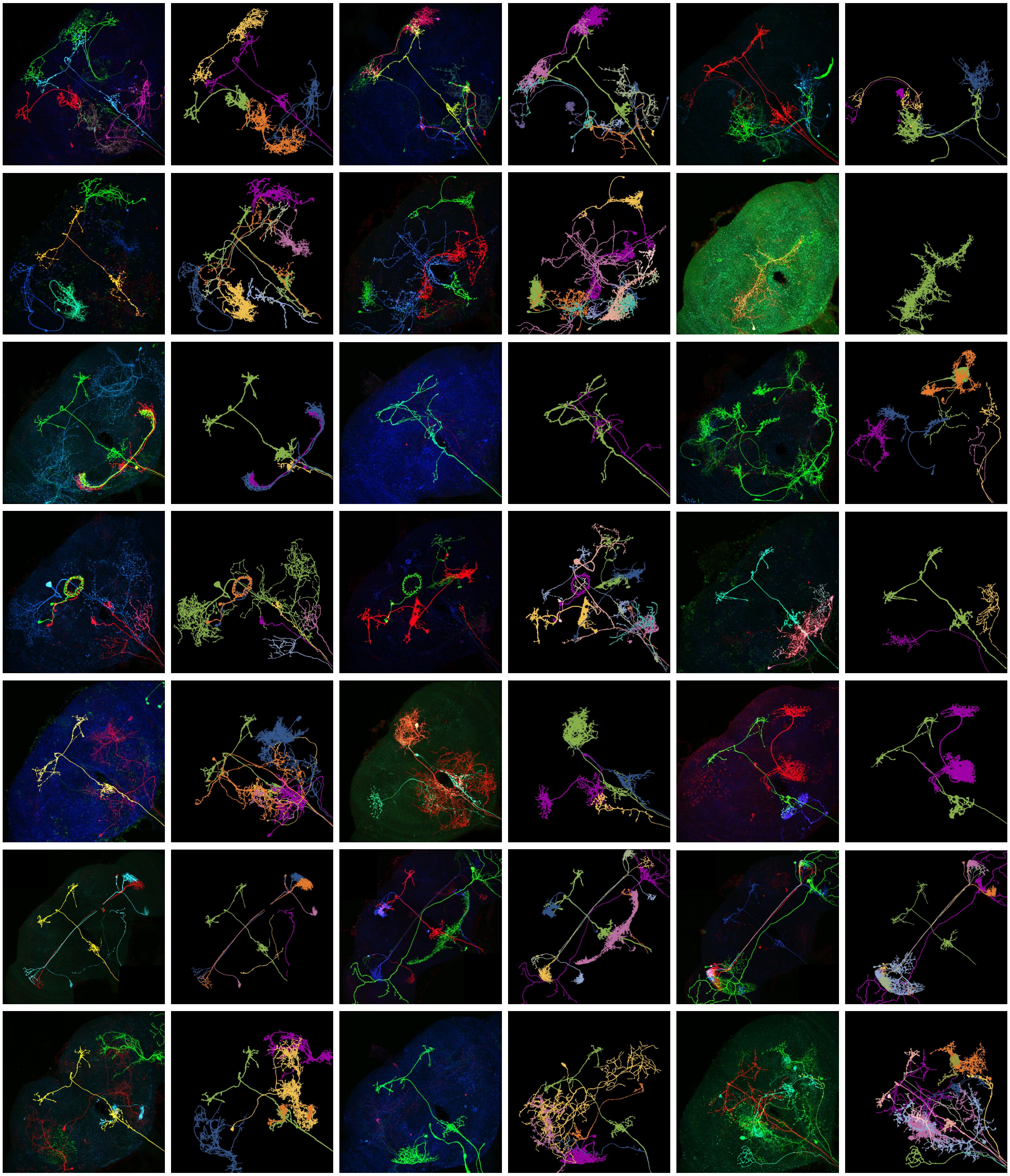
Test set

Figure 11. Maximum intensity projections (MIP) of 3d light microscopy samples and ground truth (gt) instance segmentations of all samples in the completely labeled set. MIP and gt are depicted next to each other in alternating order. Images are scaled to same width, some images are center cropped. Figure continued from previous page.



Training set

Figure 12. Maximum intensity projections (MIP) of 3d light microscopy samples and ground truth (gt) instance segmentations of all samples in the partly labeled set. MIP and gt are depicted next to each other in alternating order. Images are scaled to same width, some images are center cropped. Figure continued on next page.



Training set (continued from previous page)

Figure 13. Maximum intensity projections (MIP) of 3d light microscopy samples and ground truth (gt) instance segmentations of all samples in the partly labeled set. MIP and gt are depicted next to each other in alternating order. Images are scaled to same width, some images are center cropped. Figure continued on next page.

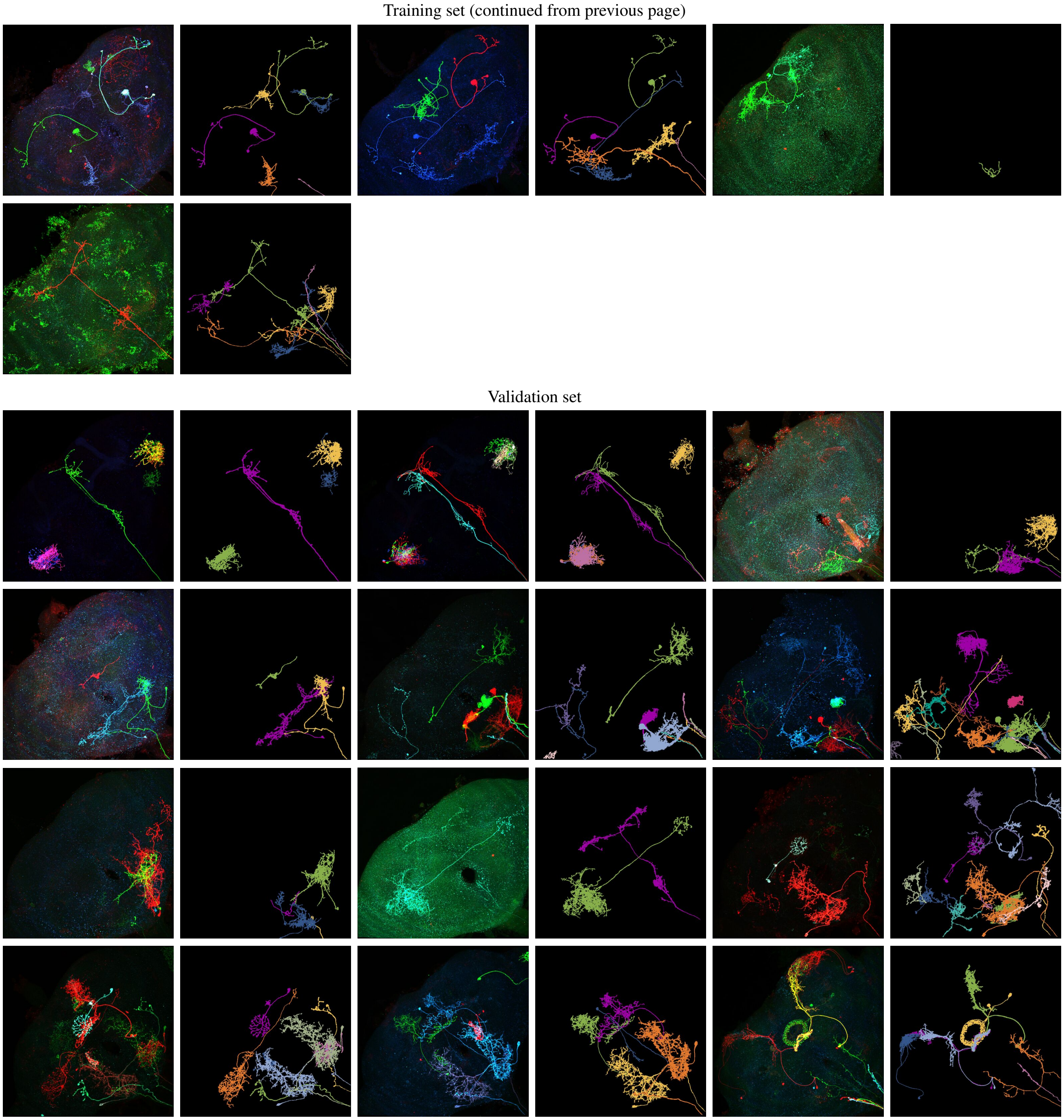


Figure 14. Maximum intensity projections (MIP) of 3d light microscopy samples and ground truth (gt) instance segmentations of all samples in the partly labeled set. MIP and gt are depicted next to each other in alternating order. Images are scaled to same width, some images are center cropped. Figure continued on next page.

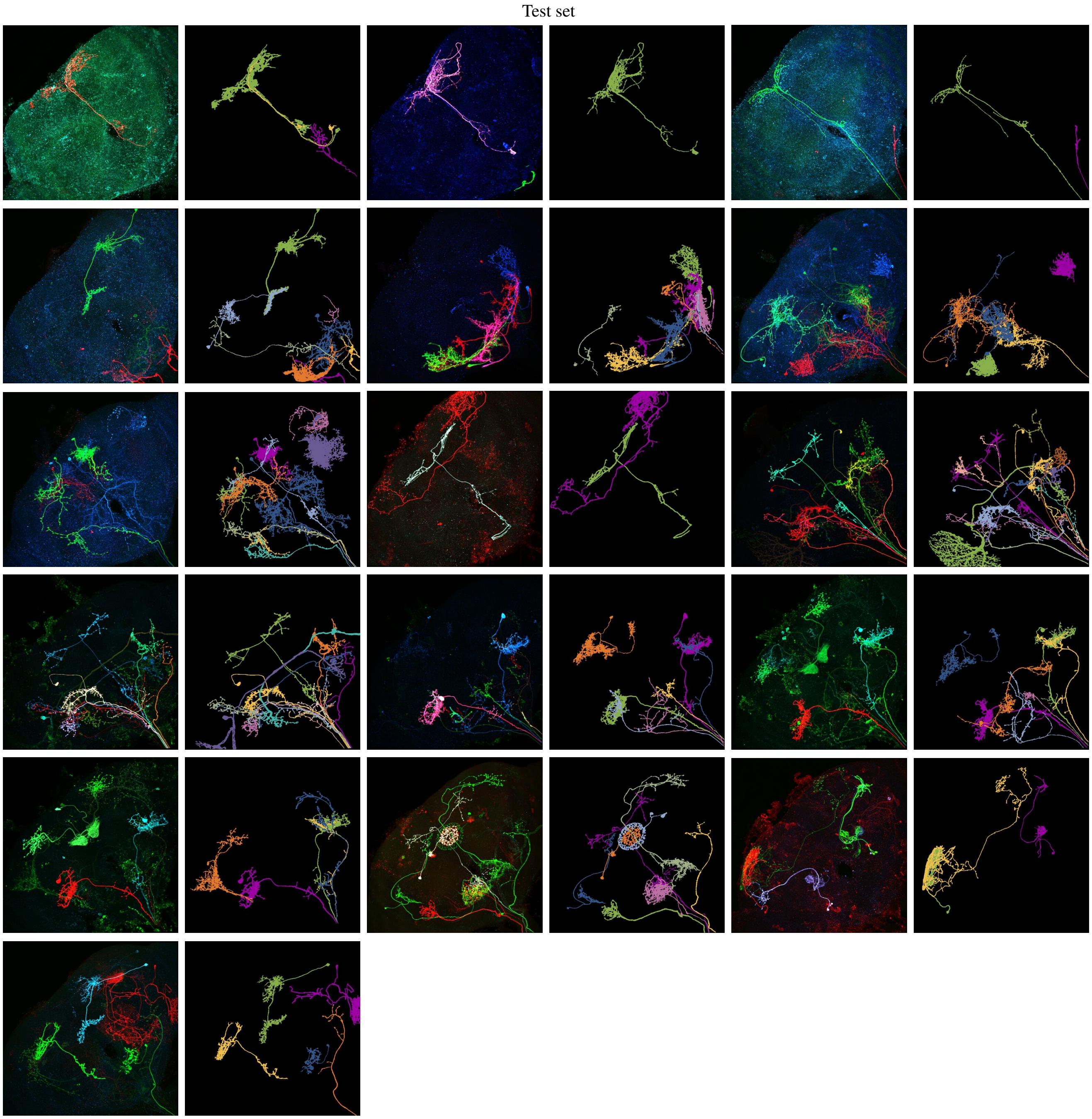


Figure 15. Maximum intensity projections (MIP) of 3d light microscopy samples and ground truth (gt) instance segmentations of all samples in the partly labeled set. MIP and gt are depicted next to each other in alternating order. Images are scaled to same width, some images are center cropped. Figure continued from previous page.

References

[1] Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: Learning dense volumetric segmentation from sparse annotation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, pages 424–432, Cham, 2016. Springer International Publishing. 7

[2] Bin Duan, Logan A Walker, Douglas H Roossien, Fred Y Shen, Dawen Cai, and Yan Yan. Unsupervised neural tracing in densely labeled multispectral brainbow images. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 1122–1126, 2021. 7

[3] Erica Ehrhardt, Samuel C Whitehead, Shigehiro Namiki, Ryo Minegishi, Igor Siwanowicz, Kai Feng, Hideo Otsuna, FlyLight Project Team, Geoffrey W Meissner, David Stern, Jim Truman, David Shepherd, Michael H. Dickinson, Kei Ito, Barry J Dickson, Itai Cohen, Gwyneth M Card, and Wyatt Korff. Single-cell type analysis of wing premotor circuits in the ventral nerve cord of drosophila melanogaster. *bioRxiv*, 2023. 9

[4] Shahar Frechter, Alexander Shakeel Bates, Sina Tootoonian, Michael-John Dolan, James Manton, Arian Rokkum Jamasb, Johannes Kohl, Davi Bock, and Gregory Jefferis. Functional and anatomical specificity in a higher olfactory centre. *eLife*, 8:e44590, 2019. 9

[5] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna M. Wallach, Hal Daumé III, and Kate Crawford. Datasheets for datasets. *CoRR*, abs/1803.09010, 2018. 1

[6] gunpowder contributors. gunpowder: a library to facilitate machine learning on large, multi-dimensional images, 2019. 7

[7] Michal Januszewski, Jeremy Maitin-Shepard, Peter Li, Jörgen Kornfeld, Winfried Denk, and Viren Jain. Flood-filling networks. *CoRR*, abs/1611.00421, 2016. 7

[8] Michał Januszewski, Jörgen Kornfeld, Peter H. Li, Art Pope, Tim Blakely, Larry Lindsey, Jeremy Maitin-Shepard, Mike Tyka, Winfried Denk, and Viren Jain. High-precision automated reconstruction of neurons with flood-filling networks. *Nature Methods*, 15(8):605–610, 2018. 7

[9] Arnim Jenett, Gerald M. Rubin, Teri-T B. Ngo, David Shepherd, Christine Murphy, Heather Dionne, Barret D. Pfeiffer, Amanda Cavallaro, Donald Hall, Jennifer Jeter, Nirmala Iyer, Dona Fetter, Joanna H. Hausenfluck, Hanchuan Peng, Eric T. Trautman, Robert R. Svirskas, Eugene W. Myers, Zbigniew R. Iwinski, Yoshinori Aso, Gina M. DePasquale, Adrianne Enos, Phuson Hulamm, Shing Chun Benny Lam, Hsing-Hsi Li, Todd R. Laverty, Fuhui Long, Lei Qu, Sean D. Murphy, Konrad Rokicki, Todd Safford, Kshiti Shaw, Julie H. Simpson, Allison Sowell, Susana Tae, Yang Yu, and Christopher T. Zugates. A gal4-driver line resource for drosophila neurobiology. *Cell reports*, 2(4):991–1001, 2012. 2

[10] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 7

[11] Jean Livet, Tamily A. Weissman, Hyuno Kang, Ryan W. Draft, Ju Lu, Robyn A. Bennis, Joshua R. Sanes, and Jeff W. Lichtman. Transgenic strategies for combinatorial expression of fluorescent proteins in the nervous system. *Nature*, 450(7166):56–62, 2007. 7

[12] Matteo Maggioni, Vladimir Katkovnik, Karen Egiazarian, and Alessandro Foi. Nonlocal transform-domain filter for volumetric data denoising and reconstruction. *IEEE Transactions on Image Processing*, 22(1):119–133, 2013. 7

[13] Lisa Mais, Peter Hirsch, and Dagmar Kainmueller. Patchperpix for instance segmentation. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV*, page 288–304, Berlin, Heidelberg, 2020. Springer-Verlag. 4, 7

[14] Lisa Mais, Peter Hirsch, Claire Managan, Kaiyu Wang, Konrad Rokicki, Robert R Svirskas, Barry J Dickson, Wyatt Korff, Gerald M Rubin, Gudrun Ihrke, et al. Patchperpixmatch for automated 3d search of neuronal morphologies in light microscopy. *bioRxiv*, pages 2021–07, 2021. 4

[15] Geoffrey W Meissner, Aljoscha Nern, Zachary Dorman, Gina M DePasquale, Kaitlyn Forster, Theresa Gibney, Joanna H Hausenfluck, Yisheng He, Nirmala A Iyer, Jennifer Jeter, Lauren Johnson, Rebecca M Johnston, Kelley Lee, Brian Melton, Brianna Yarbrough, Christopher T Zugates, Jody Clements, Cristian Goina, Hideo Otsuna, Konrad Rokicki, Robert R Svirskas, Yoshinori Aso, Gwyneth M Card, Barry J Dickson, Erica Ehrhardt, Jens Goldammer, Masayoshi Ito, Dagmar Kainmueller, Wyatt Korff, Lisa Mais, Ryo Minegishi, Shigehiro Namiki, Gerald M Rubin, Gabriella R Sterne, Tanya Wolff, Oz Malkesman, and FlyLight Project Team. A searchable image resource of *Drosophila* gal4 driver expression patterns with single neuron resolution. *eLife*, 12:e80660, 2023. 1, 2, 3, 4, 9

[16] Mai M Morimoto, Aljoscha Nern, Arthur Zhao, Edward M Rogers, Allan M Wong, Mathew D Isaacson, Davi D Bock, Gerald M Rubin, and Michael B Reiser. Spatial readout of visual looming in the central brain of *Drosophila*. *eLife*, 9: e57685, 2020. 9

[17] napari contributors. napari: a multi-dimensional image viewer for python, 2019. 5

[18] Aljoscha Nern, Barret D. Pfeiffer, and Gerald M. Rubin. Optimized tools for multicolor stochastic labeling reveal diverse stereotyped cell arrangements in the fly visual system. *Proceedings of the National Academy of Sciences*, 112(22): E2967–E2976, 2015. 1

[19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 7

[20] Barret D Pfeiffer, Teri-T B Ngo, Karen L Hibbard, Christine Murphy, Arnim Jenett, James W Truman, and Gerald M Rubin. Refinement of tools for targeted gene expression in drosophila. *Genetics*, 186(2):735–755, 2010. 9

[21] Konrad Rokicki, Christopher M. Bruns, Cristian Goina, David Schauder, Donald J. Olbris, Eric T. Trautman, Rob

Svirskas, Jody Clements, David Ackerman, Antje Kazimiers, Leslie L. Foster, Tom Dolafi, Mark Bolstad, Hideo Otsuna, Yang Yu, Todd Safford, and Sean D. Murphy. Janelia Workstation Codebase, 2019. 4

[22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 7

[23] Louis K Scheffer, C Shan Xu, Michal Januszewski, Zhiyuan Lu, Shin-ya Takemura, Kenneth J Hayworth, Gary B Huang, Kazunori Shinomiya, Jeremy Maitlin-Shepard, Stuart Berg, et al. A connectome and analysis of the adult drosophila central brain. *Elife*, 9:e57443, 2020. 4

[24] Uygar Sümbül, Douglas Roossien, Dawen Cai, Fei Chen, Nicholas Barry, John P Cunningham, Edward Boyden, and Liam Paninski. Automated scalable segmentation of neurons from multispectral images. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2016. 7

[25] Fei Wang, Kaiyu Wang, Nora Forknall, Christopher Patrick, Tansy Yang, Ruchi Parekh, Davi Bock, and Barry J. Dickson. Neural circuitry linking mating and egg laying in drosophila females. *Nature*, 579(7797):101–105, 2020. 9

[26] Yang Yu and Hanchuan Peng. Automated high speed stitching of large 3d microscopic images. In *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 238–241, 2011. 4