Guided Proofreading of Automatic Segmentations for Connectomics

Anonymous CVPR submission

Paper ID 0947

Abstract

Automatic cell image segmentation methods in connectomics produce merge and split errors, which require correction through proofreading. Previous research has identified the visual search for these errors as the bottleneck in interactive proofreading. To aid error correction, we develop two classifiers to recommend candidate merge and split errors and their corrections to the user. These classifiers are informed by training a convolutional neural network with known errors in automatic segmentations against expert-labeled ground truth. Our classifiers detect potentially-erroneous regions by considering a large context region around a segmentation boundary. Corrections can then be performed as yes/no decisions resulting in faster correction times than previous methods. We evaluate our approach on connectomics datasets of different species and compare correction performance of novice and expert users against different existing systems. We report significant improvements compared to pure automatic and pure manual proofreading.

1. Introduction

In connectomics, neuroscientists annotate neurons and their connectivity within 3D volumes to gain insight into the functional structure of the brain. Rapid progress in automatic sample preparation and electron microscopy (EM) acquisition techniques has made it possible to image large volumes of brain tissue at nanometer resolution. A typical voxel size is $4 \times 4 \times 40$ nm^3. For a 1 mm^3 volume, this resolution results in 1 petabyte of data. With so much data, manual annotation is not feasible, and automatic annotation methods are needed [12, 22, 25, 17].

Automatic annotation by segmentation and classification of brain tissue is challenging [1] and all available methods make errors. This means we are left with large data which needs *proofreading* by humans. This crucial task serves two purposes: 1) to correct errors in the segmentation, and 2) to provide a large body of labeled data to train better automatic segmentation methods. Recent proofreading tools provide intuitive user interfaces to browse segmentation data in 2D and 3D and to identify and manually correct errors [30, 13, 19, 10]. Many kinds of errors exist, such as inaccurate boundaries, but the most common are *split errors*, where a single segment is labeled as two, and *merge errors*, where two segments are labeled as one (Fig. 1). With user interaction, split errors can be joined, and the missing boundary in a merge error can be defined with manually-seeded watersheds [10]. However, even with semi-automatic correction tools, the visual inspection to find errors takes the majority of the time [26].

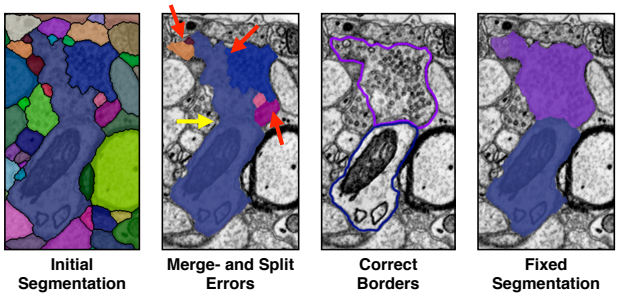

Figure 1. The most common proofreading corrections are fixing split errors (red arrows) and merge errors (yellow arrow). A fixed segmentation matches the cell borders.

Our goal is to add automatic detection of split and merge errors to proofreading tools. We design automatic classifiers that detect split and merge errors in segmentations so the user does not need to visually inspect the whole data volume to spot errors. A proofreading tool then recommends regions with a high probability of an error to the user, and suggest corrections to accept or reject. We call this process *guided proofreading*.

In this paper, we introduce classifiers to detect merge- and split errors based on a convolutional neural network (CNN). We believe that this is the first time that deep learning is applied to the task of proofreading. Our classifiers work on top of any existing automatic segmentation method to find potential errors and suggest corrections. Given a membrane segmentation from a fast automatic method, our classifiers operate on the boundaries of whole cell regions.

Compared to techniques that must analyze every input pixel, we reduce the data analysis to the boundaries. First, we train a CNN to detect only split errors. The output of this network is a probability whether a boundary between two segments is valid or not. We then reuse the same network to also detect merge errors by generating possible boundaries within a cell and inverting the split error score. We create corrections for both types of errors which can be accepted or rejected. This reduces the proofreading operation to simple yes/no decisions.

We further propose a greedy algorithm to perform proofreading. Possible erroneous regions are sorted by their score and the algorithm iteratively suggests a correction for each region. A user then works through this stream of regions and corrections. In a forced choice setting, the user either selects a correction or skips it to advance to the next region. This choice can be also performed automatically by running the algorithm until a configurable threshold is reached. In addition, if ground truth data is available, we can use a selection oracle to drive the forced choice selection. The oracle only accepts corrections which improve the automatic segmentation. This equals perfect proofreading.

We evaluate our method automatically by threshold and oracle on multiple real-world connectomics datasets. To evaluate the forced choice setting, we perform a quantitative user study. The study targets non-experts with no previous experience of proofreading electron microscopy data. We ask the participants to proofread a small segmentation volume in a fixed time frame by performing yes/no decisions. The user study is designed as a between-subjects experiment and compares guided proofreading against two other methods: a recently published fully interactive proofreading tool named *Dojo* by Haehn *et al*. [10] and the semi-automatic *focused proofreading* approach by Plaza [27]. We also asked four domain experts to use guided proofreading and focused proofreading for additional comparison.

Our first contribution is a classifier for split error detection based on a convolutional neural network. The classifier performs well even when trained with little amounts of training data. This is important since generating ground truth labels in connectomics requires manually labeling pixels and is very time-consuming. Our second contribution is a mechanism to identify merge-errors by re-using the split error classifier. Merge errors are usually less common than split errors in the oversegmented automatic labelings. However, they require more interaction during correction since split lines need to be manually drawn. Our method reduces this to a single click by providing the potential correction. The split and merge error identification is executed as a greedy algorithm to correct segmentation volumes, the third contribution of this paper. The algorithm can be driven automatically with a threshold, by an oracle based on ground truth and interactively in a forced choice setting. Our final contribution is our quantitative user study. We present statistically significant results showing that novice and expert users of guided proofreading are able to proofread a given dataset better and faster than with existing interactive and semi-automatic proofreading tools. As a consequence, we are able to provide tools to proofread segmentations more efficiently, and so better tackle large volumes of connectomics imagery.

2. Related Work

Automatic Segmentation. Multi-terabyte EM brain volumes require automatic segmentation [12, 22, 24, 25], but can be hard to classify due to ambiguous intercellular space: the 2013 IEEE ISBI neurites 3D segmentation challenge [1] showed that existing algorithms which learn from expert-segmented training data still exhibit high error rates.

Many works tackle this problem. NeuroProof [2] decreases error rates by learning a agglomeration on over-segmentations of images, based on a random forest classifier. Vazquez-Reina *et al*. [33] take whole EM volumes into account rather than a per section approach, then solve a fusion problem with a global context. Kaynig *et al*. [16] propose a random forest classifier coupled with an anisotropic smoothing prior in a conditional random field framework with 3D segment fusion. Bogovic *et al*. [6] learn 3D features unsupervised, and show that they can be better than by-hand designs.

It is also possible to learn segmentation classification features directly from images with CNNs. Ronneberger *et al*. [28] use a contracting/expanding CNN path architecture to enable precise boundary localization with small amounts of training data. Lee *et al*. [20] recursively train very deep networks with 2D and 3D filters to detect boundaries.

All these approaches make good progress; however, in general, proofreading is still required to correct errors.

Interactive Proofreading. While proofreading is very time consuming, it is fairly easy for humans to perform corrections through splitting and merging segments. One expert tool is Raveler, introduced by Chklovskii *et al*. [7, 13]. Raveler is used today by professional proofreaders, and it offers many parameters for tweaking the process. Similar systems exist as products or plugins to visualization systems, *e.g*. V3D [26] or AVIZO [30].

Recent papers have attacked the problem of proofreading massive datasets through crowdsourcing with novices [29, 4, 9]. One popular platform is EyeWire, by Kim *et al*. [18], where participants earn virtual rewards for merging over-segmented labeling to reconstruct retina cells.

Between expert systems and online games sit Mojo and *Dojo*, by Haehn *et al*. [10, 3], which use simple scribble interfaces for error correction. Dojo extends this to distributed proofreading via a minimalistic web-based user interface.

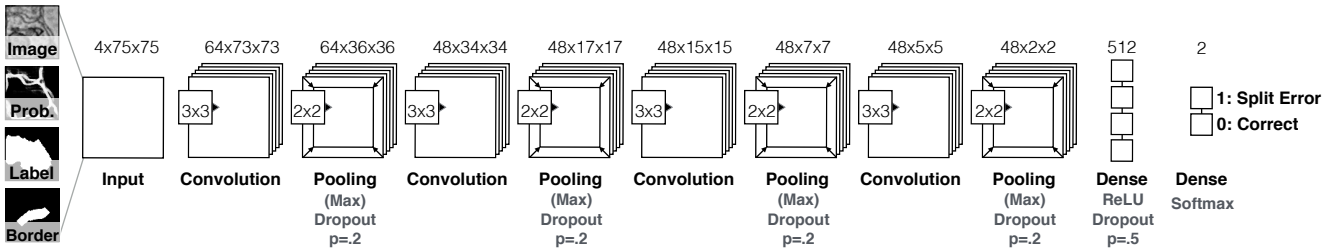

Figure 2. We build the guided proofreading classifiers using a traditional CNN architecture. The network is based on four convolutional layers, each followed by max pooling as well as dropout regularization. The 4-channel input patches are rated as either correct splits or as split errors.

The authors define requirements for general proofreading tools, and then evaluate the accuracy and speed of Raveler, Mojo, and Dojo through a quantitative user study (Sec. 3 and 4) [10]. Dojo had the highest performance. In this paper, we use Dojo as a baseline for interactive proofreading, and so we extend the Haehn *et al*. experiment.

All interactive proofreading solutions require the user to find potential errors manually, which takes the majority of the time [26, 10]. Recent works propose computer-aided proofreading systems which help reduce the time spent in this visual search task.

Computer-aided Proofreading. Uzunbas *et al*. showed that potential labeling errors can be found by considering the merge tree of an automatic segmentation method [32]. The authors track uncertainty throughout the automatic labeling by training a conditional random field. This segmentation technique produces uncertainty estimates, which can be used to present potential regions for proofreading to the user. While it works on isotropic volumes, more work is needed to apply it to anisotropic volumes, like most connectomics datasets.

Karimov *et al*. propose guided volume editing [14], which measures the difference in histogram distributions in image data to find potential split and merge errors in the corresponding segmentation. This lets expert users correct labeled computer-tomography datasets, using several interactions per correction. To correct merge errors, the authors create a large number of superpixels within a single segment and then successively group them based on dissimilarities. We were inspired by this approach but generate single watershed boundaries to handle the intracellular variance in high-resolution EM images (Sec. 3).

Most closely related to our approach is the work of Plaza, who proposed *focused proofreading* [27]. This method generates affinity scores by analyzing a region adjacency graph across slices, then finds the largest affinities based on a defined impact score. This yields edges of potential split errors which can be presented to the proofreader. Plaza reports that additional manual work is required to find and correct merge errors. Focused proofreading builds upon NeuroProof [2] as its agglomerator, and is open source with integration into Raveler. As the closest related work, we wish to use this method as a baseline to evaluate our approach (Sec. 4). However, as Haehn *et al*. showed that Raveler is less performant than Dojo for novice users, we separate the backend affinity score calculation from the expert-level front end, and present our own interface (Sec. 4).

3. Method

3.1. Split Error Detection

We build a split error classifier with output p using a convolutional neural network (CNN) to check whether an edge within an existing automatic segmentation is valid ($p = 0$) or not ($p = 1$). Rather than analyzing every input pixel, the classifier operates only on segment boundaries which requires less pixel context and is faster. In contrast to Bogovic *et al*. [6], we work with 2D slices rather than 3D volumes. This enables proofreading prior or in parallel to an expensive alignment of individual EM images.

Convolutional Neural Network Architecture. Split error detection of a given boundary is really a binary classification task since the boundary is either correct or erroneous. However, in reality the score p is between 0 and 1. The classification complexity arises from hundreds of different cell types in connectomics data rather than from the classification decision. Intuitively, this yields a wider (meaning more filters) rather than a deeper (meaning more layers) architecture. We explored different architectural configurations - including residual networks [11] - by performing a brute force parameter search and comparing precision and recall (see supplementary materials). Our final CNN configuration for split error detection is composed of four convolutional layers, each followed by max pooling as well as dropout regularization to prevent overfitting due to limited training data. Fig. 2 shows the CNN architecture for split error detection.

Classifier Inputs. To train the CNN for split error detection, we take boundary context information into

consideration for the decision making process. For this, we use a 75×75 pixel patch at the center of an existing boundary. This covers approximately 80% of all boundaries in real-world connectomics data with nanometer resolution. If the boundary edge is not fully covered, we sample up to 10 non-overlapping patches along the boundary and combine the resulting score by weighted averaging based on boundary length coverage per patch. Similar to Bogovich *et al*. [6], we use grayscale image data, corresponding boundary probabilities, and a single binary mask combining the two neighboring labels as features for our CNN. However, we observed that the boundary probability information generated from EM images is often misleading due to noise or artifacts in the data. This can result in merge errors within the automatic segmentation. To better direct our classifier to train on the true boundary edge, we extract the border between two segments. We then dilate this border by 5 pixels to consider slight edge ambiguities and use this binary mask as an additional feature to create a stacked 4-channel input patch. Fig. 3 shows examples of correct and erroneous feature patches and their corresponding automatic segmentation and ground truth.

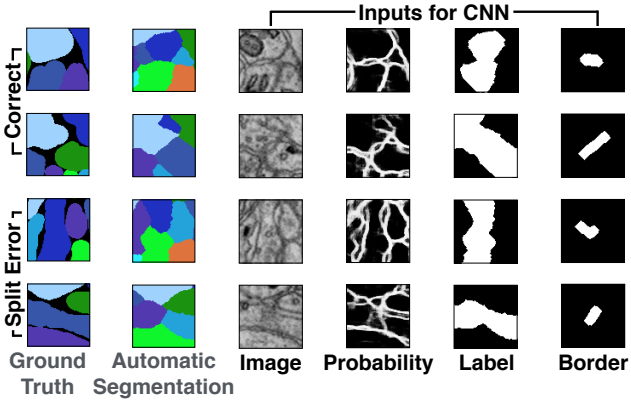

Figure 3. Example inputs for learning correct splits and split errors as reflected in the segmentation relative to the ground truth. Image, membrane probabilities, merged binary labels, and a dilated border mask are combined to 4-channel input patches.

3.2. Merge Error Detection

Identification and correction of merge errors is more challenging than finding and fixing split errors, because we must look inside segmentation regions for missing or incomplete boundaries and then propose the correct boundary. However, we can reuse the same trained CNN for this task. Similar to guided volume editing by Karimov *et al*. [14] we generate potential borders within a segment. For each segmentation label, we dilate the label by 20 pixel and generate 50 potential boundaries through the region by randomly placing watershed seed points at opposite sides of the label boundary. For watershed, we use the inverted gray scale EM image as features. This yields 50 candidate splits.

Dilation of the segment prior to watershed is motivated by our observation that the generated split likely attaches to the real membrane boundary. These boundaries are then individually rated using our split error classifier. For this, we invert the probability score such that a correct split (previously encoded as $p = 0$) is most likely a candidate for a merge error (now encoded as $p = 1$). In other words, if a generated boundary is ranked as correct, it probably should be in the segmentation. Fig. 4 illustrates this procedure.

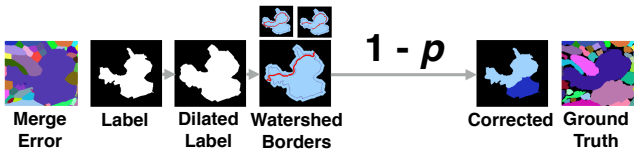

Figure 4. Merge errors are identified by generating randomly seeded watershed borders within a dilated label segment. These borders then are individually rated using the split error CNN by inverting the probability score. This way, a confident rating for a correct split most likely indicates the missing border of the merge error and can be used for correcting the labeling.

3.3. Error Correction

We use the proposed classifiers in combination to perform corrections of split and merge errors in automatic segmentations. For this, we first perform merge error detection for all existing segments in a dataset and store the inverted rankings $1 - p$ as well as potential corrections. After that, we perform split error detection and store the ranking p for all neighboring segments in the segmentation. We then sort the merge and split error rankings individually from highest to lowest. For error correction, we first loop through the potential merge error regions and then through the potential split error regions. During this process, each error is now subject to a yes/no decision which can be provided in different ways:

Selection oracle. If ground data is available, the selection oracle *knows* whether a possible correction improves an automatic segmentation. This is realized by simply applying the correction and comparing the outcome using a defined measure. The oracle only accepts corrections which improve the automatic segmentation - others get discarded. This equals perfect proofreading.

Automatic selection with threshold. The decision whether to accept or reject a potential correction is done by comparing rankings to a threshold p_t. If the inverted score $1 - p$ of a merge error is higher than a threshold $1 - p_t$, the correction is accepted. Similarly, a correction is accepted for a split error if the ranking p is higher than p_t. Our experiments have shown that the threshold p_t is the

same for merge and split errors which makes sense for a balanced classifier.

Forced choice setting. A user is presented with the choices of either accepting or rejecting a correction. This way, all potential split errors are looked at. Inspecting all merge errors is not possible for users due to the sheer amount of generated borders. We therefor only present merge errors which satisfy $1 - p_t$.

In all cases, a decision has to be made to advance to the next possible erroneous region. If a merge error correction was accepted, the newly found boundary is added to the segmentation data. This partially updates the merge error and split error ranking in respect of the new segment. If a split error correction was accepted, two segments are merged in the segmentation data and the disappearing segment is removed from all error rankings. We then perform merge error detection on the now larger segment and update the ranking. We also update the split error rankings to include all new neighbors and re-sort. The error with the next highest ranking is now subject to the yes/no decision.

3.4. User Interface

Guided proofreading is integrated into an existing workflow for large connectomics data. The system is web-based and is designed with a minimalistic user interface showing three components. We show the outline of the current labeling of a cell boundary and its proposed correction on top of the EM image data. For the user, it is not possible to distinguish the current labeling and the proposed correction to avoid selection bias. We also show a solid overlay of the current and the proposed labeling. In addition, we show the image without overlays to provide an unoccluded view. User interaction is simple and involves one mouse click on either the current labeling or the correction. After interaction, the next potential error is shown. Figure 5 shows the user interface.

4. Evaluation

We evaluate guided proofreading (GP) on multiple real-world connectomics datasets of different species. In particular, we evaluate GP on two datasets of mouse brain and three datasets of fruitfly brain (drosophila). For comparison, we choose the fully interactive proofreading software *Dojo* by Haehn *et al*. [10] as well as the aided proofreading framework *focused proofreading* (FP) by Plaza [27]. We first describe the evalation on mouse brain data and then the evaluation on drosophila brain.

Training. To initially train our network, we use the blue 3-cylinder mouse cortex volume of Kasthuri *et al*. [15] ($2048 \times 2048 \times 300$ voxels). The tissue is dense mammalian neuropil from layers 4 and 5 of the S1 primary somatosensory cortex of a healthy mouse. The resolution of our dataset is $3\,nm$ per pixel, and the section thickness is $30\,nm$. The image data and a manually-labeled expert segmentation is publicly available as ground truth for the entire dataset[1]. We use the first 250 sections of the data for training and validation and the last 50 for testing. We use a state-of-the-art method to create a dense automatic segmentation of the data. To generate training data, we identify correct regions and split errors in the automatic segmentation by intersection with ground truth regions. This is required since extracellular space is not labeled in the ground truth but in our dense automatic segmentation. From these regions, we sample 120,000 correct and 120,000 split error patches with 4-channels as described above. The patches are normalized and to further augment our training data, we rotate patches within each mini-batch by $k * 90$ degrees with randomly chosen k. The training parameters such as filter size, number of filters, learning rate, and momentum are the result of intuition and experience, studying recent machine learning research as well as a brute force parameter search within a limited range (see supplementary material). The final parameters and training results are listed in table 1. For baseline comparison, we also list the parameters and training results of focused proofreading in this table but elaborate on these further in section 4. Our CNN configuration results in approximately 170,000 learnable parameters. We assume that training has converged if the validation loss does not decrease for 50 epochs.

For performance comparison on data of a different species, in particular on fruitfly brain (drosophila), we re-train our network. The training procedure is according

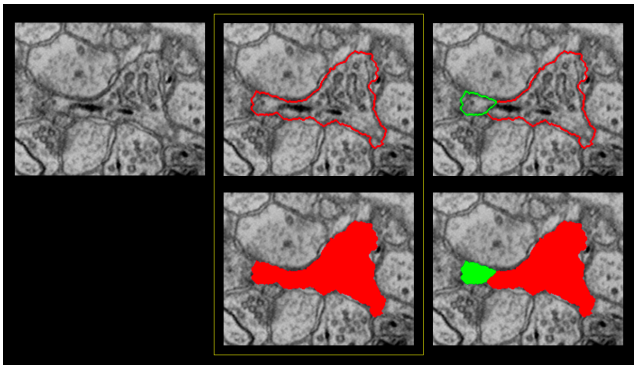

Figure 5. Segmentation correction with the guided proofreading user interface. An image without overlays is shown on the left. A split error (right) and its correction (center) are the possible choices for the user. Hovering highlights the current selection with a yellow border and a mouse click confirms it and advances to the next potential error.

[1]The Kasthuri 3-cylinder mouse cortex volume is available at https://software.rc.fas.harvard.edu/lichtman/vast/

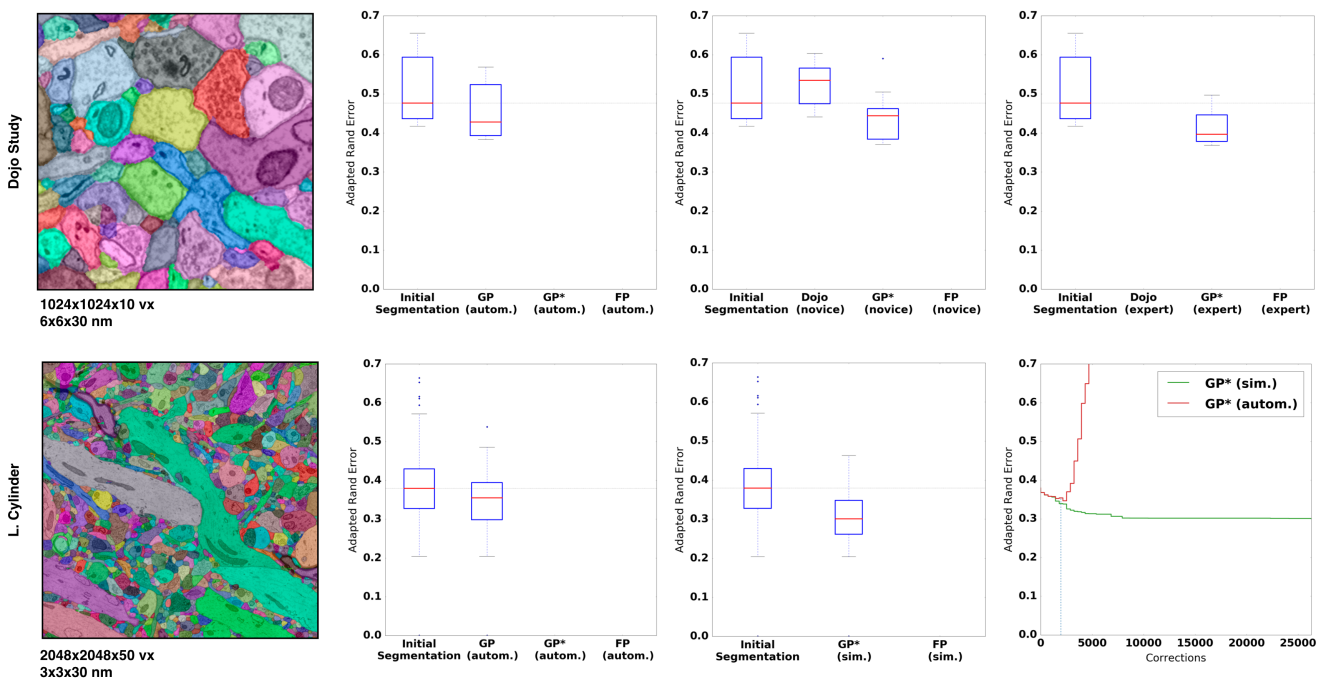

Figure 6. Performance evaluation of the classifiers on two mouse brain datasets measured as adapted Rand error (lower scores are better). We compare guided proofreading (GP), guided proofreading with active label suggestion (GP*) and focused proofreading. Proofreading is performed automatically (autom., with probability threshold $p_t = .95$), simulated as a perfect user (sim.), or by novice and expert users as indicated. The first row of images shows the results of a user study and includes comparisons to the interactive proofreading software Dojo by Haehn *et al.* [10]. GP* is able to correct the segmentation further than other methods. The second row shows the results of the simulated user compared to automatic GP* and FP performance. The bottom right graph compares automatic GP* and simulated GP* per individual correction. The blue dashed line here indicates the moment the probability threshold p_t is reached. The simulated user is able to correct the initial segmentation beyond this threshold while automatic GP* then introduces errors.

	cost [m]	Val. loss	Val. acc.	Test acc.	Prec./Recall	F1 Score
Guided Proofreading Filter size: 3x3 No. Filters 1: 64 No. Filters 2-4: 48 Dense units: 512 Learning rate: 0.03-0.00001 Momentum: 0.9-0.999 Mini-Batchsize: 128	383	0.0845	0.969	0.94	0.94/0.94	0.94
Focused Proofreading Iterations: 3 Learning strategy: 2 Mito agglomeration: Off Threshold: 0.2	43	?	?	0.839	?/?	?

Table 1. Training parameters, cost and results of our guided proofreading classifier versus focused proofreading by Plaza [27]. Both methods were trained on the same mouse brain dataset using the same hardware (Tesla K40 graphics card). While the training of our classifier is more expensive, testing accuracy is superior.

to our initial training and network architecture as well as parameters are not changed. We further elaborate on the drosophila datasets in section 4. Fig. 7 displays receiver operating characteristics (ROC) for guided proofreading trained on mouse and drosophila data, as well as our comparison baseline focused proofreading trained on these datasets respectively.

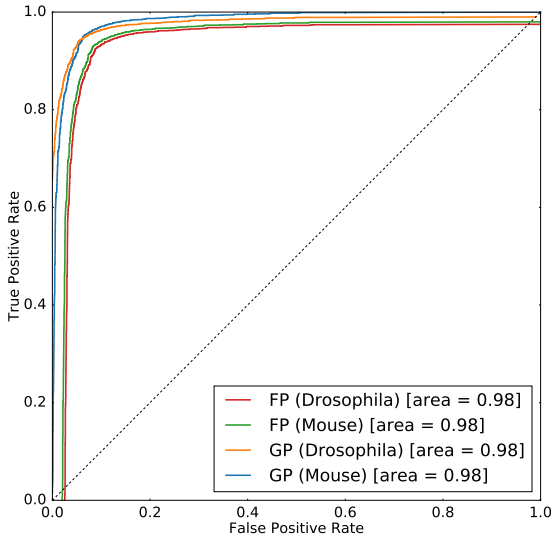

Figure 7. ROC performance of guided proofreading (GP) and focused proofreading (FP) trained separately on mouse and drosophila brain images. The area under the curve indicates better performance for GP.

4.1. Mouse Brain

Mouse brain is a common target for connectomics research because the structural proportions are similar to human brains [21]. For our first experiment we recruited novice and expert participants as part of a quantitative user study. Our second experiment is performed on a larger dataset and we evaluate a simulated user.

User study. Recently, Haehn *et al*. evaluated the interactive proofreading tools Raveler, Mojo, and Dojo as part of an experiment with novice users [10]. The participants corrected an automatic segmentation with merge and split errors. The dataset was the most representative sub-volume (based on object size histograms) of a larger connectomics dataset and $400x400x10$ voxels in size. The participants were given a fixed time frame of 30 minutes to perform the correction interactively. While participants clearly struggled with the proofreading task, the best performing tool in their evaluation was Dojo. The dataset including manually labeled ground truth and the results of Haehn *et al*. are publicly available. This means we are able to use their findings as a baseline for comparison of GP for novices. In particular, we use the best performing user of Dojo who was truly an outlier as reported by Haehn *et al*.

Since interactive proofreading most likely yields lower performance than aided proofreading, we also compare against FP by Plaza [27] which is integrated in Raveler and freely available. For FP we consulted an expert to obtain the best possible parameters as shown in table 1. Besides performance by novices, we are also interested in expert proofreading performance. Therefore, we design between-subjects experiments for 20 novice users and separately, for 6 expert users using the exact same conditions as Haehn *et al*. The recruiting, consent and debriefing process is further described in the supplementary material. We randomly assign 10 novices to GP with active label suggestion (GP*) and 10 novices to FP. For the expert experiment, we assign accordingly. In addition to human performance, we also evaluate automatic GP, automatic GP with active label suggestion (GP*) and automatic FP. Due to the automatic nature, we do not enforce the 30 minute time limit but we stop once our probability threshold of $p_t = .95$ is reached. This value was observed as stable in previous experiments using automatic GP (see supplementary material). To measure proofreading performance in comparison to ground truth, we use the adapted Rand error (aRE) metric [31]. aRE is a measure of dissimilarity, related to introduced errors, meaning lower scores are better.

The results of our comparisons are shown in the first row of Fig. 6. In all cases, GP* is able to correct the segmentation further than other methods (aRE measures: automatic GP XX, GP* XX, FP XX, novice Dojo XX, GP* XX, FP XX, expert Dojo XX, GP* XX, FP XX). This is not surprising since guided proofreading works for both merge and split errors while FP does not and in interactive Dojo the majority of time is spent finding errors which is minimized for aided proofreading solutions. In fact, the average correction time for novices is for GP* 3.6 (expert X), for FP Y (expert YY), and for Dojo 30 (expert ZZ) seconds.

Simulated experiment. For our second experiment with mouse brain data, we proofread the last 50 slices of the blue 3-cylinder mouse cortex volume of Kasthuri *et al*. [15] which we also used for testing in section 3. The data was not seen by the network before and includes $2048x2048x50$ voxels with a total number of 17,560 labeled objects. Since an interactive evaluation of such a large dataset would consume a significant amount of time, we restrict our experiment to a simulated (perfect) user and to automatic corrections, both with GP, GP* and FP. Similar to our comparison study, the simulated user assess a stream of errors by comparing the adapted Rand error measure before and after each performed correction. The simulated user is designed to be perfect and only accepts corrections if the measure is reduced. This time, we do not enforce a time limit to see the lower bound of possible corrections. For automatic GP and GP*, we use our defined probability threshold $p_t = .95$.

The results of this experiment are shown in the second row of Fig. 6. GP* is again able to correct the segmentation further than other methods (aRE measures: automatic GP XX, GP* XX, FP XX, simulated GP* XX, FP XX). Again, the results are not surprising since GP* can correct merge and split errors.

4.2. Drosophila Brain

The drosophila brain is analyzed by connectomics researchers because of its small size and hence, a reasonable target to obtain a complete wiring diagram. Despite the size, fruit flies exhibit complex behaviors and are in general well studied. We evaluate the performance of our guided proofreading classifiers on three different datasets of adult fly brain. The datasets are publicly available as part of the MICCAI 2016 challenge on circuit reconstruction from electron microscopy images (CREMI)[2]. Each dataset consists of $1250x1250x125$ voxels of training data (A,B,C) as well as testing data (A+,B+,C+) of the same dimensions. Manually labeled ground truth is also available for A,B, and C but not for the testing data.

Since drosophila brain exhibits different cell structures than mouse brain, we retrain the guided proofreading classifiers (and our automatic segmentation pipeline) as well as focused proofreading combined on the three training

[2]The MICCAI CREMI challenge data is available at http://www.cremi.org

datasets. We use 300 slices of the A,B,C samples for training and validation, and 75 slices for testing. This results in YYY correct and ZZZ split error patches (respectively, XXX and YYY for testing). The architecture and all parameters of our classifiers stay the same. The trained GP classifier exhibits a reasonable performance on the testing data as seen in Fig. 7.

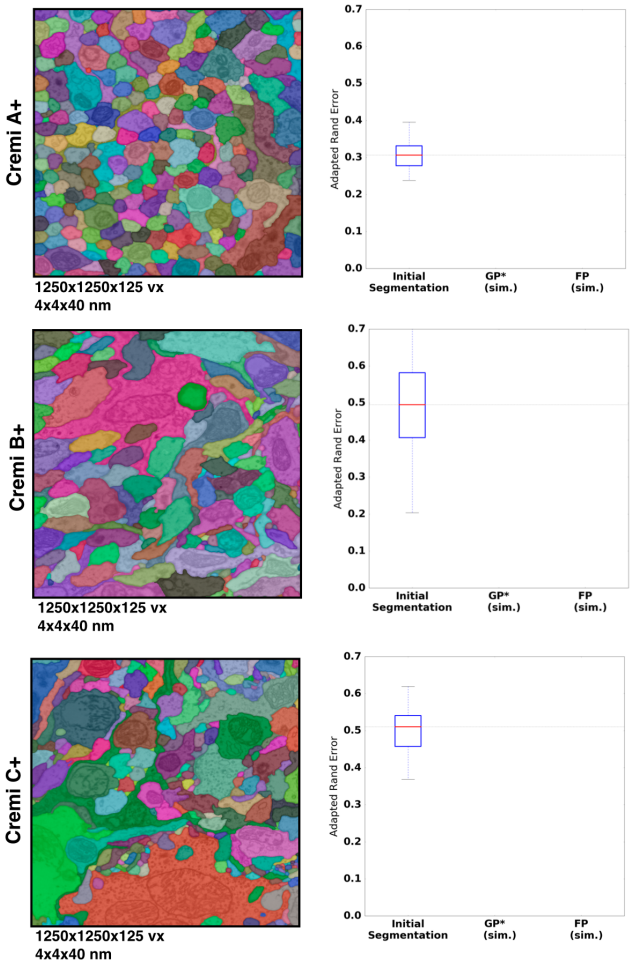

Figure 8. Results of guided proofreading with active label suggestion (GP*) and focused proofreading performed automatically on three drosophila datasets. The datasets are part of the MICCAI 2016 CREMI challenge and publicly available. We measure performance as adapted Rand error (the lower, the better). GP* is able to correct the initial segmentation further than FP. Our GP* scores places us XXnd on the CREMI leaderboard.

We then use the trained GP* and FP classifiers to evaluate proofreading automatically. Since ground truth labeling is not available, the evaluation is performed by submitting our results to the CREMI leaderboard. Again, we use adapted Rand error to quantify the performance. Fig. 8 shows the results for each of the A+,B+, and C+ datasets. The performance of GP* is significantly better than FP and places us XXnd on the CREMI leaderboard.

5. Quantitative Results

6. Conclusions

The task of automatic cell boundary segmentation is difficult, and trying to improve such segmentations automatically as a post-process through merge and split error correction is, in principle, no different than trying to improve the underlying cell boundary segmentation. Due to the task difficulty, manual proofreading of connectomics segmentations is necessary, but it is a time consuming and error-prone task. Humans are the bottleneck and minimizing the manual labor is the goal. We have addressed this problem through training a convolutional neural network to detect ambiguous regions from labeled data—in effect, by finding a non-linear mapping between image and segmentation data. This allows us to identify merge and split errors with better performance than existing systems. Our experiments have shown that guided proofreading has the potential to reduce the bottleneck in the analysis of large connectomics datasets. To encourage testing of our proposed architecture and replicate our experiments, we provide our framework and data as free and open research at (link omitted for review).

References

[1] IEEE ISBI challenge: SNEMI3D - 3D segmentation of neurites in EM images. http://brainiac2.mit.edu/SNEMI3D, 2013. Accessed on 11/01/2016. 1, 2

[2] Neuroproof: Flyem tool, hhmi / janelia farm research campus. https://github.com/janelia-flyem/NeuroProof, 2013. Accessed on 03/15/2106. 2, 3

[3] A. K. Al-Awami, J. Beyer, D. Haehn, N. Kasthuri, J. W. Lichtman, H. Pfister, and M. Hadwiger. Neuroblocks - visual tracking of segmentation and proofreading for large connectomics projects. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):738–746, Jan 2016. 2

[4] J. Anderson, S. Mohammed, B. Grimm, B. Jones, P. Koshevoy, T. Tasdizen, R. Whitaker, and R. Marc. The Viking Viewer for connectomics: Scalable multi-user annotation and summarization of large volume data sets. *Journal of Microscopy*, 241(1):13–28, 2011. 2

[5] ANON. Anon. *ANON*, 2016.

[6] J. A. Bogovic, G. B. Huang, and V. Jain. Learned versus hand-designed feature representations for 3d agglomeration. *CoRR*, abs/1312.6159, 2013. 2, 3, 4

[7] D. B. Chklovskii, S. Vitaladevuni, and L. K. Scheffer. Semi-automated reconstruction of neural circuits using electron microscopy. *Current Opinion in Neurobiology*, 20(5):667 – 675, 2010. Neuronal and glial cell biology New technologies. 2

[8] D. C. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In *NIPS*, 2012.

[9] R. J. Giuly, K.-Y. Kim, and M. H. Ellisman. DP2: Distributed 3D image segmentation using micro-labor workforce. *Bioinformatics*, 29(10):1359–1360, 2013. 2

[10] D. Haehn, S. Knowles-Barley, M. Roberts, J. Beyer, N. Kasthuri, J. Lichtman, and H. Pfister. Design and evaluation of interactive proofreading tools for connectomics. *IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE SciVis 2014)*, 20(12):2466–2475, 2014. 1, 2, 3, 5, 6, 7

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 3

[12] V. Jain, B. Bollmann, M. Richardson, D. Berger, M. Helmstädter, K. Briggman, W. Denk, J. Bowden, J. Mendenhall, W. Abraham, K. Harris, N. Kasthuri, K. Hayworth, R. Schalek, J. Tapia, J. Lichtman, and S. Seung. Boundary learning by optimization with topological constraints. In *Proc. IEEE CVPR 2010*, pages 2488–2495, 2010. 1, 2

[13] Janelia Farm. Raveler. https://openwiki.janelia.org/wiki/display/flyem/Raveler, 2014. Accessed on 11/01/2016. 1, 2

[14] A. Karimov, G. Mistelbauer, T. Auzinger, and S. Bruckner. Guided volume editing based on histogram dissimilarity. *Computer Graphics Forum*, 34(3):91–100, May 2015. 3, 4

[15] N. Kasthuri, K. J. Hayworth, D. R. Berger, R. L. Schalek, J. A. Conchello, S. Knowles-Barley, D. Lee, A. Vázquez-Reina, V. Kaynig, T. R. Jones, et al. Saturated reconstruction of a volume of neocortex. *Cell*, 162(3):648–661, 2015. 5, 7

[16] V. Kaynig, T. Fuchs, and J. Buhmann. Neuron geometry extraction by perceptual grouping in sstem images. In *Proc. IEEE CVPR*, pages 2902–2909, 2010. 2

[17] V. Kaynig, A. Vazquez-Reina, S. Knowles-Barley, M. Roberts, T. R. Jones, N. Kasthuri, E. Miller, J. Lichtman, and H. Pfister. Large-scale automatic reconstruction of neuronal processes from electron microscopy images. *Medical image analysis*, 22(1):77–88, 2015. 1

[18] J. S. Kim, M. J. Greene, A. Zlateski, K. Lee, M. Richardson, S. C. Turaga, M. Purcaro, M. Balkam, A. Robinson, B. F. Behabadi, M. Campos, W. Denk, H. S. Seung, and EyeWirers. Space-time wiring specificity supports direction selectivity in the retina. *Nature*, 509(7500):331336, May 2014. 2

[19] S. Knowles-Barley, M. Roberts, N. Kasthuri, D. Lee, H. Pfister, and J. W. Lichtman. Mojo 2.0: Connectome annotation tool. *Frontiers in Neuroinformatics*, (60), 2013. 1

[20] K. Lee, A. Zlateski, A. Vishwanathan, and H. S. Seung. Recursive training of 2d-3d convolutional networks for neuronal boundary detection. *arXiv preprint arXiv:1508.04843*, 2015. 2

[21] J. W. Lichtman and W. Denk. The big and the small: Challenges of imaging the brain's circuits. *Science*, 334(6056):618–623, 2011. 7

[22] T. Liu, C. Jones, M. Seyedhosseini, and T. Tasdizen. A modular hierarchical approach to 3D electron microscopy image segmentation. *Journal of Neuroscience Methods*, 226(0):88 – 102, 2014. 1, 2

[23] J. Masci, A. Giusti, D. C. Ciresan, G. Fricout, and J. Schmidhuber. A fast learning algorithm for image segmentation with max-pooling convolutional networks. In *ICIP*, 2013.

[24] J. Nunez-Iglesias, R. Kennedy, T. Parag, J. Shi, and D. B. Chklovskii. Machine learning of hierarchical clustering to segment 2D and 3D images. *PLoS ONE*, 8(8):e71715+, 2013. 2

[25] J. Nunez-Iglesias, R. Kennedy, S. M. Plaza, A. Chakraborty, and W. T. Katz. Graph-based active learning of agglomeration (GALA): A python library to segment 2D and 3D neuroimages. *Frontiers in Neuroinformatics*, 8(34), 2014. 1, 2

[26] H. Peng, F. Long, T. Zhao, and E. Myers. Proof-editing is the bottleneck of 3D neuron reconstruction: The problem and solutions. *Neuroinformatics*, 9(2-3):103–105, 2011. 1, 2, 3

[27] S. M. Plaza. Focused Proofreading: Efficiently Extracting Connectomes from Segmented EM Images, Sept. 2014. 2, 3, 5, 6, 7

[28] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. 2

[29] S. Saalfeld, A. Cardona, V. Hartenstein, and P. Tomančák. CATMAID: collaborative annotation toolkit for massive amounts of image data. *Bioinformatics*, 25(15):1984–1986, 2009. 2

[30] R. Sicat, M. Hadwiger, and N. J. Mitra. Graph abstraction for simplified proofreading of slice-based volume segmentation. In *EUROGRAPHICS Short Paper*, 2013. 1, 2

[31] R. Unnikrishnan, C. Pantofaru, and M. Hebert. A measure for objective evaluation of image segmentation algorithms. pages 34–, 2005. 7

[32] M. G. Uzunbas, C. Chen, and D. Metaxas. An efficient conditional random field approach for automatic and interactive neuron segmentation. *Medical Image Analysis*, 27:31 – 44, 2016. Discrete Graphical Models in Biomedical Image Analysis. 3

[33] A. Vázquez-Reina, M. Gelbart, D. Huang, J. Lichtman, E. Miller, and H. Pfister. Segmentation fusion for connectomics. In *Proc. IEEE ICCV*, pages 177–184, Nov 2011. 2