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 that automatically recommend candidate merge and splits to the user. These classifiers use a convolutional neural network (CNN) that has been trained with 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 by a user with yes/no decisions, which results in faster correction times than previous proofreading methods. We also present a fully-automatic mode that uses a probability threshold to make merge/split decisions. Extensive experiments using the automatic approach and comparing performance of novice and expert users demonstrate that our method performs favorably against state-of-the-art proofreading methods on different connectomics datasets.

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. With a voxel size of $4 \times 4 \times 40$ nm^3, a cubic millimeter volume is one petabyte of data. With so much data, manual annotation is not feasible, and automatic annotation methods are needed [13, 22, 25, 18].

Automatic annotation by segmentation and classification of brain tissue is challenging [1] and all available methods make errors, so the results must be *proofread* by humans. This crucial task serves two purposes: 1) to correct errors in the segmentation, and 2) to increase the body of labeled data from which to train better automatic segmentation methods.

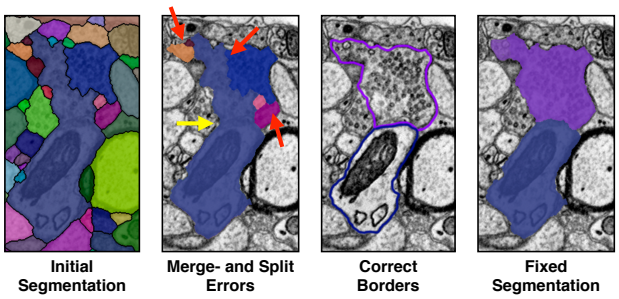

Figure 1: The most common proofreading corrections are fixing split errors (red arrows) and merge errors (yellow arrow). A fixed segmentation matches the cell borders.

Recent proofreading tools provide intuitive user interfaces to browse segmentation data in 2D and 3D and to identify and manually correct errors [31, 14, 20, 9]. 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 [9]. However, the visual inspection to find errors takes the majority of the time, even with semi-automatic correction tools [27].

Our goal is to automatically detect split and merge errors to reduce visual inspection time. Further, to reduce correction time, we propose corrections to the user to accept or reject. We call this process *guided proofreading*.

We learn a classifier for split errors with a convolutional neural network (CNN). This takes as input patches of membrane segmentation probabilities, cell segmentation masks, and boundary masks, and outputs a probability score. As we must process large data, this classifier only operates on cell boundaries, which reduces computation over methods that analyze every pixel. For merge errors, we invert and reuse the split classification network, instead asking it to rate a set of generated candidate boundaries that hypothesize a split.

We also propose a greedy algorithm for guided proofreading. Possible erroneous regions are sorted by their score, and

a candidate correction is generated for each region. Then, a user works through this list of regions and corrections. In a forced choice setting, the user either selects a correction or skips it to advance to the next region. In an automatic setting, errors with a high probability are automatically corrected first, given an appropriate probability threshold, after which the user would take over. Finally, to test the limits of performance, we create an oracle which only accepts corrections that improve the segmentation, based on knowledge of the ground truth. This is guided proofreading with a perfect user.

We evaluate these methods on multiple connectomics datasets. For the forced choice setting, we perform a quantitative user study with 20 novice users who have no previous experience of proofreading EM data. We ask participants to proofread a small segmentation volume in a fixed time frame. In a between-subjects design, we compare guided proofreading to the semi-automatic *focused proofreading* approach by Plaza [28]. In addition, we compare against the manual interactive proofreading tool *Dojo* by Haehn *et al*. [9]. We also asked four domain experts to use guided proofreading and focused proofreading for comparison.

This paper makes the following contributions. First, we present a CNN-based boundary classifier for split errors, plus a merge error classifier that inverts the split error classifier. This is used to propose merge error corrections, removing the need to manually draw the missing edge. These classifiers perform well without much training data, which is expensive to collect for connectomics data. Second, we developed a guided proofreading approach to correcting segmentation volumes, and an assessment scenario comparing forced-choice interaction with automatic and oracle proofreading. Third, we present the results of a quantitative user study assessing guided proofreading. Our method is able to reduce segmentation error faster than state-of-the-art semi-automatic tools for both novice and expert users.

Guided proofreading is applicable to all existing automatic segmentation methods that produce a label map. As such, we believe that our approach is a promising direction to proofread segmentations more efficiently and better tackle large volumes of connectomics imagery.

2. Related Work

Automatic Segmentation. Multi-terabyte EM brain volumes require automatic segmentation [13, 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 that learn from expert-segmented training data still exhibit high error rates.

Many works tackle this problem. NeuroProof [2] decreases error rates by learning an agglomeration on over-segmentations of images based on a random forest classifier. Vazquez-Reina *et al*. [33] consider whole EM volumes rather than a per-section approach, then solve a fusion problem with a global context. Kaynig *et al*. [17] propose a random forest classifier coupled with an anisotropic smoothing prior in a conditional random field framework with 3D segment fusion. Bogovic *et al*. [5] 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*. [29] use a contracting/expanding CNN path architecture to enable precise boundary localization with small amounts of training data. Lee *et al*. [21] 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*. [6, 14]. 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 [27] and AVIZO [31].

Recent papers have attacked the problem of proofreading massive datasets through crowdsourcing with novices [30, 4, 8]. One popular platform is EyeWire, by Kim *et al*. [19], where participants earn virtual rewards for merging over-segmented labelings to reconstruct retina cells.

Between expert systems and online games sit Mojo and Dojo, by Haehn *et al*. [9, 3], which use simple scribble interfaces for error correction. Dojo extends this to distributed proofreading via a minimalistic web-based user interface. 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, [9]). Dojo had the highest performance. In this paper, we use Dojo as a baseline for interactive proofreading.

All interactive proofreading solutions require the user to find potential errors manually, which takes the majority of the time [27, 9]. Recent papers propose computer-aided proofreading systems to reduce the time spent in this visual search task.

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

Karimov *et al*. [15] propose guided volume editing, which

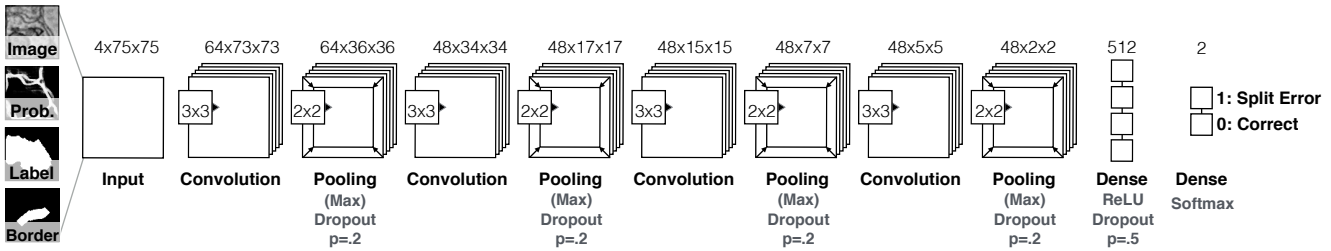

Figure 2: Our guided proofreading classifiers use a traditional CNN architecture of four convolutional layers, each followed by max pooling and dropout regularization. The 4-channel input patches are labelled either as correct splits or as split errors.

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 instead we 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 [28], who proposed *focused proofreading*. 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, we separate the backend affinity score calculation from the Raveler expert-level front end, and present our own novice-friendly interface (Sec. 4).

3. Method

3.1. Split Error Detection

We build a split error classifier with output p using a 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*. [5], we work with 2D slices rather than 3D volumes. This enables proofreading prior or in parallel to a computationally expensive stitching and 3D alignment of individual EM images.

Convolutional Neural Network Architecture. Split error detection of a given boundary is 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 itself. Intuitively, this yields a wider architecture with more filters rather than a deeper architecture with more layers. 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).

Classifier Inputs. To train the CNN, we consider boundary context in the decision making process by using a 75×75 pixel patch at the center of an existing boundary. This size covers approximately 80% of all boundaries in the 6 nm Mouse S1 AC3 Open Connectome Project dataset. If the boundary length is not fully covered, we sample up to 10 non-overlapping patches along the boundary, and combine the resulting score by averaging weighted by the boundary length coverage per patch.

Similar to Bogovich *et al*. [5], we use grayscale image data, corresponding boundary probabilities, and a single binary mask combining the two neighboring labels as inputs to 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, we extract the border between two segments. Then, we dilate this border by 5 pixels to consider slight edge ambiguities and use this binary mask as an additional input. This creates a stacked 4-channel input patch. Fig. 3 shows examples of correct and erroneous input patches and their corresponding automatic segmentation and ground truth.

3.2. Merge Error Detection

Identification and correction of merge errors is more challenging than finding and fixing split errors, because we must

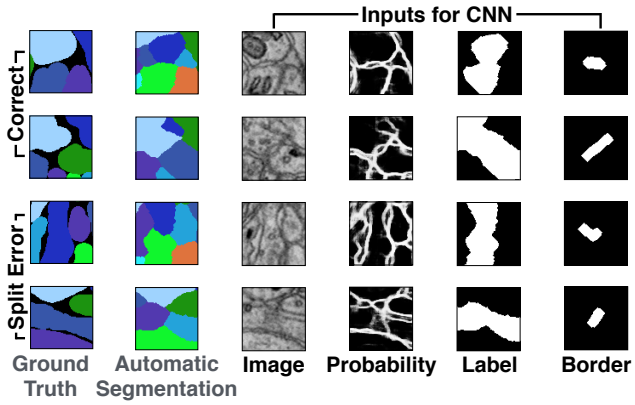

Figure 3: Example inputs for learning correct splits and split errors, as per a candidate segmentation versus the ground truth. Image, membrane probabilities, merged binary labels, and a dilated border mask provide 4-channel input patches.

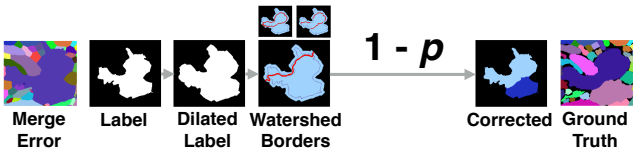

Figure 4: Merge errors are identified by generating randomly-seeded watershed borders within a dilated label segment. Then, each border is individually rated using the split error CNN by inverting the probability score. A confident rating for a correct split most likely indicates the missing border of the merge error, and can be used to correct the labeling.JT: Candidate to improve.

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*. [15], 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. We perform watershed on the inverted grayscale EM image. This yields 50 candidate splits.

Dilation of the segment prior to watershed is motivated by our observation that the generated splits tend to attach to real membrane boundaries. 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.

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. Then, we sort the merge and split error rankings separately from highest to lowest. For error correction, first we 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 truth data is available, the selection oracle *knows* whether a possible correction improves an automatic segmentation. This is realized by simply comparing the outcome of a correction using a defined measure. The oracle only accepts corrections which improve the automatic segmentation—others get discarded. This is guided proofreading with a perfect user, and allows us to assess the upper limit of performance.

Automatic selection with threshold. The decision whether to accept or reject a potential correction is taken 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 for a balanced classifier that has been trained on equal numbers of correct and error patches.

Forced choice setting. We present a user with the choice to accept or reject a correction. All potential split errors are seen. Inspecting all merge errors is not possible for users due to the sheer amount of generated borders. Therefore, we only present merge errors that 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 with respect to 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. Then, we 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 then forces a choice.

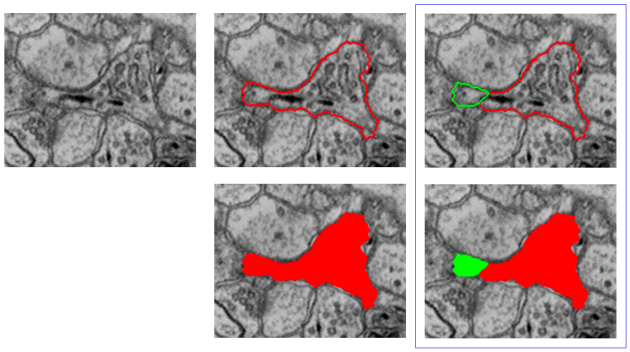

Figure 5: Our guided proofreading user interface. A candidate error region is shown on the left. The user must choose between the region being a split error which needs correcting (center) or not (right). Hovering highlights the current selection with a blue border, and a mouse click confirms the choice and advances to the next potential error.

3.4. User Interface

We integrate guided proofreading into an existing large data connectomics workflow. The web-based system is designed with a novice-friendly user interface (Fig. 5). We show the current labeling of a cell boundary outline and its proposed correction overlayed on EM image data. The user is not possible to distinguish the current labeling from 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.

4. Evaluation

We evaluate guided proofreading on multiple different real-world connectomics datasets of different species. All datasets were acquired using either serial section electron microscopy (ssEM) or serial section transmission electron microscopy (ssTEM). We perform experiments with the selection oracle, with automatic selection with threshold, and in the forced choice setting via a between-subjects user study with both novice and expert participants.

4.1. Datasets

L. Cylinder. We use the left part of the 3-cylinder mouse cortex volume of Kasthuri *et al*. [16] ($2048 \times 2048 \times 300$ voxels). The tissue is dense mammalian neuropil from layers 4 and 5 of the S1 primary somatosensory cortex of a healthy mouse, and was acquired using ssEM. The dataset resolution is $3 \times 3\,\text{nm}^2$ per pixel, with section thickness of 30 nm. Image data and a manually-labeled expert 'ground truth' segmentation is available publicly for the entire dataset[1].

AC4 subvolume. This is part of a publicly-available dataset of mouse cortex that was published for the ISBI 2013 challenge "SNEMI3D: 3D Segmentation of neurites in EM images". The dataset resolution is $6 \times 6 \times 30$ nm^3/voxel and it was acquired using ssEM. Haehn *et al*. [9] found the most representative subvolume ($400 \times 400 \times 10$ voxels) of this dataset with respect to the distribution of object sizes, and used it for their interactive connectomics proofreading tool experiments. We use their publicly available data, labeled ground truth, and study findings[2].

CREMI A/B/C. As part of the MICCAI 2016 challenge on circuit reconstruction from electron microscopy images (CREMI), six ssTEM datasets were made publicly available[3], each $1250 \times 1250 \times 125$ voxels. Since only three datasets include manually-labeled 'ground truth', we use these three volumes for our experiments. The volumes are part of an adult fruit fly (Drosophila melanogaster) brain. The resolution of all three datasets is $4 \times 4 \times 40$ nm^3/voxel.

Automatic segmentation pipeline. We use a state-of-the-art method to create a dense automatic segmentation of the data. Membrane probabilities are generated using a CNN based on the U-net architecture [29]. The probabilities are used to seed watershed and generate an oversegmentation using superpixels. Agglomeration is then performed by the GALA active learning classifier [26].

4.2. Classifier Training

We train our split error classifier on the L. Cylinder dataset. We use the first 250 sections of the data for training and validation. For n-fold cross validation, we select one quarter of this data and re-select after each epoch. We minimize cross-entropy loss and update using stochastic gradient descent with Nesterov momentum [23]. 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 is in our dense automatic segmentation. From these regions, we sample 112,760 correct and 112,760 split error patches with 4-channels (Sec. 3.1). The patches are normalized. To augment our training data, we rotate patches within each mini-batch by $k * 90$ degrees with randomly chosen integer 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, and a limited brute force parameter search (see supplementary material).

[1] https://software.rc.fas.harvard.edu/lichtman/vast/
[2] http://rhoana.org/dojo/
[3] http://www.cremi.org

Table 1: Training parameters, cost, and results of our guided proofreading classifier versus focused proofreading by Plaza [28]. Both methods were trained on the same mouse brain dataset using the same hardware (Tesla X GPU).

Guided Proofreading	
Parameters	*Results—Test Set*
Filter size: 3x3	Cost [m]: 383
No. Filters 1: 64	Val. loss: 0.0845
No. Filters 2–4: 48	Val. acc.: 0.969
Dense units: 512	Test. acc.: 0.94
Learning rate: 0.03–0.00001	Prec./Recall: 0.94/0.94
Momentum: 0.9–0.999	F1 Score: 0.94
Mini-Batchsize: 128	
Focused Proofreading	
Parameters	*Results—Test Set*
Iterations: 3	cost [m]: 217
Learning strategy: 2	Test. acc.: 0.99
Mito agglomeration: Off	Prec./Recall: HP: add?/?
Threshold: 0.0	F1 Score: HP: add?

Table 1 lists the final parameters. Our CNN configuration results in 171,474 learnable parameters. We assume that training has converged if the validation loss does not decrease for 50 epochs. We test the CNN by generating a balanced set of 8,780 correct and 8,780 error patches using unseen data of the left cylinder dataset.

4.3. Baseline Comparisons

Interactive proofreading. Haehn *et al.*'s comparison of interactive proofreading tools concludes that novices perform best when using Dojo [9]. We use the publicly available findings of their user study as a baseline comparison. We use the data of all Dojo users in aggregate as a baseline.

Computer-aided proofreading. We compare against focused proofreading by Plaza [28]. Focused proofreading performs graph analysis on the output from NeuroProof [2], instead of our GALA approach. Therefore, for training our focused proofreading baseline, we replace GALA in our automatic segmentation pipeline with NeuroProof but use exactly the same input data including membrane probabilities. We obtained the best possible parameters for NeuroProof by consulting the developers (Table 1). Rather than using Raveler as the frontend, we use our own interface (Fig. 5) to compare only the classifier part of Plaza's approach.

4.4. Experiments

Selection oracle evaluation. We use the selection oracle as described in Sec. 3.3 for the decision whether to accept or reject a correction. The purpose of this experiment is to investigate how many corrections are required to reach the best possible outcome. This is a direct comparison of the guided proofreading and focused proofreading classifiers but can only be performed if ground truth data is available. We perform this experiment on all datasets listed above.

Automatic method evaluation. For this experiment, we accept all suggested corrections if the rankings are above a configured threshold $p_t = .95$ (Sec. 3.3). We observed this value as stable in previous experiments with the guided proofreading classifiers (see supplementary material). We compare against the focused proofreading classifier and perform this experiment on all reported datasets.

Forced choice user experiments. We conducted a quantitative user study to evaluate the forced choice setting (Sec. 3.3). In particular, we evaluated how participants perform while correcting an automatic segmentation using the guided proofreading and focused proofreading tools. We designed a single factor between-subject experiment with the factor *proofreading classifier*, and asked participants to proofread the AC4 subvolume in a fixed time frame of 30 minutes. To enable comparison against the interactive proofreading study by Haehn *et al.* [9], we use the exact same study conditions, dataset, and time limit. The experiment was performed on a machine with standard off-the-shelf hardware. All participants received monetary compensation.

Novice study design. We recruited participants with no experience in electron microscopy data or proofreading, through flyers, mailing lists, and personal interaction. Based on sample size calculation theory, we estimated the study needed ten users per proofreading tool including four potential dropouts [7, 12]. All twenty participants completed the study ($N = 20$, 10 female; 19-65 years old, M=30).

Each study session began with a five minute standardized explanation of the task. Then, the participants were asked to perform a 3 minute proofreading task on separate but representative data using focused proofreading. The participants were allowed to ask questions during this time. The classifier did not matter in this case since the user interface was the same. The experimenter then loaded the AC4 subvolume with initial pre-computed classifications by either guided proofreading or focused proofreading depending on assignment. After 30 minutes, the participants completed the raw NASA-TLX standard questions for task evaluation [10].

Expert study design. We recruited 4 domain experts to evaluate the performance of both guided and focused proofreading. We obtained study consent and randomly assigned 2 experts to proofread using each classifier. The experts performed the 3 minute test run on different data prior to proof-

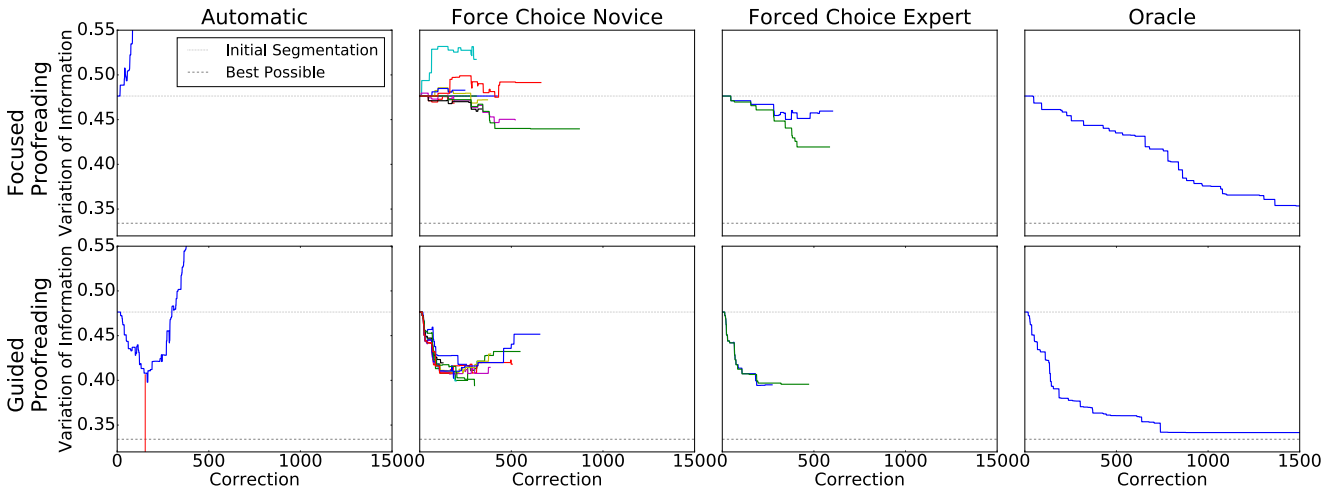

Figure 6: Performance comparison of focused proofreading by Plaza and our guided proofreading methods on the AC4 subvolume. All measurements are reported as median VI, the lower the better. We compare different approaches of accepting or rejecting corrections for each method: automatic selection with threshold (indicated by the red line), forced choice by ten novice users, forced choice by two domain experts, and the selection oracle. In all cases, guided proofreading yields better results with less corrections.

reading for 30 minutes. After the task ended, the experts were asked to complete the raw NASA-TLX questionnaire.

Evaluation metric. For each experiment, we measure the similarity between proofread segmentations and the manual 'ground truth' labelings using *variation of information* (VI). VI is a measure of the distance between two clusterings, closely related to mutual information, where lower means better. We treat VI as a continuous variable during analysis.

5. Results and Discussion

5.1. Precision and Recall

We compare the guided proofreading and focused proofreading classifiers and report precision and recall as well as f1 scores across datasets.

L. Cylinder. Evaluation was performed on previously unseen sections of mouse cortex volume of Kasthuri *et al*. [16]. We generated an unbalanced dataset of 81,184 correct and 8,780 split error patches in respect to the ground truth labeling. We then ranked each patch using focused proofreading and guided proofreading and compared precision/recall (Table 2. Our method exhibits higher precision and recall for correct and error patches.

AC4 subvolume. We generated 3,488 correct and 332 error patches (10 merge errors, 322 split errors). Again, guided proofreading achieves better precision/recall scores (Table 3).

Table 2: Classifier comparison on an unbalanced test set of the L. Cylinder volume.

	precision	recall	f1 score	#
Focused Proofreading				
correct	0.93	0.31	0.47	81,184
split error	0.11	0.78	0.19	8,780
Guided Proofreading				
correct	1.00	0.93	0.96	81,184
split error	0.61	0.96	0.74	8,780

Table 3: Classifier comparison on correct and split error patches of the AC4 subvolume.

	precision	recall	f1 score	#
Focused Proofreading				
correct	0.94	0.69	0.80	3,488
split error	0.14	0.51	0.21	332
Guided Proofreading				
correct	1.00	0.92	0.96	3,488
split error	0.54	0.95	0.69	332

5.2. Forced Choice User Experiment

We performed a user study to evaluate the forced choice error correction method among novices and experts. To be comparable to Haehn's *et al*. Dojo user study, participants were asked to proofread the AC4 subvolume for 30 minutes. We counted 10 merge errors and 322 split errors by computing the maximum overlap of the initial segmentation in respect to the ground truth labeling (both provided by Haehn *et al*.). For evaluation, we measure the performance

of proofreading quantitatively by comparing VI scores of segmentations to ground truth. The median VI of the initial segmentation was 0.476 (SD=0.089). Most novices and all experts were able to improve upon this score with both, focused proofreading and guided proofreading (Fig. 6).

Novice performance. Participants using focused proofreading were able to reduce the median VI of the automatic segmentation to 0.469 ($SD = 0.87$). On average, users viewed 423.4 corrections and accepted 45.8 of them with an average time of 4.9 seconds spent per correction. Participants using guided proofreading were able to reduce the median VI to 0.424 ($SD = 0.037$). Here, users viewed on average 353.4 corrections and accepted 106.9 in 6.2 seconds. While three users of focused proofreading made the initial segmentation worse, all participants using guided proofreading were able to improve it. In comparison to the results of Haehn *et al*., focused and guided proofreading outperform interactive proofreading with Dojo (median VI 0.535, $SD = 0.055$). The slope of VI score per correction in Fig. 6 shows that guided proofreading enables improvements with fewer corrections than the other tools. Interestingly, novice performance decreases after approximately 300 corrections. There are two explanations for this: user fatigue and increasing uncertainty during error suggestion from the classifier. Regarding fatigue, we suggest for future experiments to add short breaks after every ten minutes.

Expert performance. Domain experts were able to improve the initial segmentation in all cases. With focused proofreading, the median VI of the automatic segmentation was 0.439 ($SD = 0.084$). With guided proofreading, the median VI was 0.396 ($SD = 0.032$) as shown in Fig. 7.

Subjective responses. All subjective responses were recorded using the NASA-TLX workload index. Mental, physical, and temporal demands were reported slightly higher for participants using focused proofreading. However, these differences were not statistically significant. This is not surprising since the user interface was the same for both control groups.

5.3. Automatic Error Correction

Selection oracle. Accepting and rejecting proofreading corrections with the selection oracle yields the best performance on all datasets because all detected merge and split are corrected as long as the segmentation is improved. Fig. 6 shows slope of VI reduction using the selection oracle on the AC4 subvolume (initial median VI 0.476, $SD = 0.089$). With focused proofreading, the selection oracle reaches a median VI of 0.353 ($SD = 0.037$) after 1605 corrections. With guided proofreading, the oracle results in a median VI

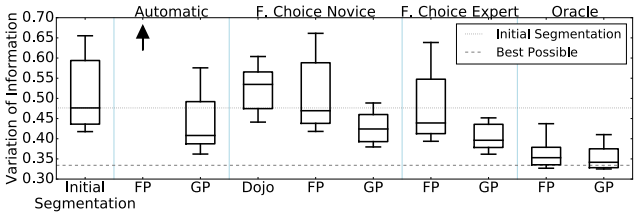

Figure 7: VI distributions of guided proofreading (GP), focused proofreading (FP) and Dojo output across the AC4 subvolume with different error correction approaches. The performance of focused proofreading with automatic selection is an order of magnitude higher (as indicated by the arrow) and is reported in the text.

of 0.342 ($SD = 0.03$) after approximately 800 corrections and then stagnates until all regions are proofread. Both results are very close to the best possible median VI of 0.334 (calculated by computing maximum overlap with the ground truth). The slope of the trails in Fig. 6 shows that guided proofreading requires less corrections to reach a reasonable reduction in VI. Fig. 7 shows the VI distribution across methods. The results on the L. Cylinder dataset (initial VI 0.379, $SD = 0.118$) are similar: Focused proofreading reduces the median VI to 0.298 ($SD = 0.075$) after $26,170$ corrections ($2,419$ accepted). Guided proofreading reduces the median VI to 0.2996 ($SD = 0.073$) after $27,491$ corrections ($2,696$ accepted).

Automatic selection with threshold. Focused proofreading was not designed to run automatically. This explains very poor performance on the AC4 subvolume (VI of 1.9, $SD = 0.496$) and on the L. Cylinder dataset (VI of 2.75, $SD = 0.789$). For guided proofreading we define the threshold $p_t = 0.95$ for both datasets. This reduces median VI in the AC4 subvolume to 0.398 ($SD = 0.068$). This is an impressive result and comparable to expert performance. Guided proofreading also reduces VI in the L. Cylinder data to 0.352 ($SD = 0.087$).

Merge Error Detection Guided proofreading performs merge error detection prior to split error detection. The classifier automatically found 10 merge errors in the AC4 subvolume of which 4 reduce VI . Automatic selection with $p_t = 0.95$ corrected all of these 4 true errors and in addition 2 false ones (precision/recall 0.67/1.00, f1-score 0.80). In 50 sections of the L. Cylinder dataset 151 merge errors were automatically found of which 17 reduce VI. Automatic selection with $p_t = 0.95$ corrected 6 true errors and 30 false ones (precision/recall 0.17/0.35, f1-score 0.23). DH: Not good. The split error correction cleans up but still not good, might be bug. hard to track down on the large data.

5.4. Limitations

Guided proofreading works on 2D image sections. This enables error correction without a computationally expensive alignment process. However, the output requires an additional (block-)merging step prior to 3D analysis.

6. Conclusions

In this paper, we have trained classifiers to find merge and split errors in automatic segmentations of connectomics data. We then reduce the correction of these errors to simple yes/no decisions. This allows proofreading with better performance than existing systems. Our experiments also show that automatic proofreading works. This has the potential to further reduce the manual labor and will be the target of future research. 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, 6

[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] 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

[6] 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

[7] L. Faulkner. Beyond the five-user assumption: Benefits of increased sample sizes in usability testing. *Behavior Research Methods, Instruments, & Computers*, 35(3):379–383, 2003. 6

[8] 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

[9] 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, 5, 6

[10] S. G. Hart and L. E. Staveland. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Human Mental Workload*, volume 52 of *Advances in Psychology*, pages 139 – 183. 1988. 6

[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] W. Hwang and G. Salvendy. Number of people required for usability evaluation: The 10+-2 rule. *Commun. ACM*, 53(5):130–133, May 2010. 6

[13] 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

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

[15] 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. 2, 4

[16] 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

[17] 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

[18] 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

[19] 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

[20] 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

[21] 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

[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] Y. Nesterov. A method for unconstrained convex minimization problem with the rate of convergence o (1/k2). In *Doklady an SSSR*, volume 269, pages 543–547, 1983. 5

[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] 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, 2014. 5

[27] 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

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

[29] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. 2, 5

[30] 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

[31] 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

[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. 2

[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