Proofreading of Automatic Segmentations in Connectomics

Daniel Haehn*
Advisor: Hanspeter Pfister
Harvard University

1 Introduction

Our research targets the intersection of computer science and neuroscience - particularly in the field of *connectomics*. Connectomics aims to completely reconstruct the wiring diagram of the mammalian brain including billions of nerve cells and their connections [8, 11]. Scientists hope to better understand mental illnesses, learning disorders and neural pathologies by decoding this network [9]. To analyze neuronal connectivity at the level of individual synapses (i.e., connections between nerve cells), electron microscopy (EM) image stacks are processed. These images are acquired at nanometer resolution. This results in volume sizes of hundreds of terabytes which then get automatically labeled and classified. Severe noise and artifacts present in the data make this procedure difficult and lead to errors. Therefore, the automatic labeling requires *proofreading* by humans. In fact, over 120 manual corrections are required on average per cubic micron of tissue [7]. These are mainly *split errors*, where a single segment is labeled as two, and *merge errors*, where two segments are labeled as one (Fig. 1). Split errors can be corrected by joining labels and the missing boundary in merge errors can be identified [4].

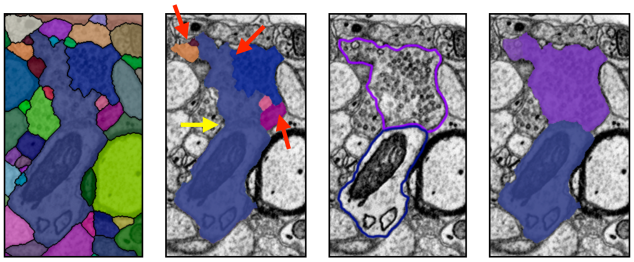

Figure 1: The most common proofreading corrections are fixing split errors (red arrows) and merge errors (yellow arrow).

To support this correction task, interactive proofreading tools and semi-automatic methods exist but are mostly geared towards domain experts [4, 12]. In connectomics, we instead need novice users to perform proofreading because of the large amounts of data - ideally in a distributed setting. Our studies have shown that this is quite challenging: In a controlled experiment, inexperienced users worsened the initial labeling instead of improving it [4]. We then discovered that simply finding errors is the most challenging step and takes the majority of time. This lead us to recently incorporate a machine learning based error suggestion method by training a convolutional neural network (CNN). The network detects and ranks split errors as well as merge errors. Likely errors and possible corrections are then suggested to the proofreader, correctable via point-and-click. Our initial results indicate that this enhances the proofreading performance. We now plan to incorporate a novel active learning approach to further increase the results, however, the ultimate solution as a combination of machine learning + user interface + visualization still has to be found.

*e-mail: haehn@seas.harvard.edu

2 Related Work

The bottleneck of editing an existing and imperfect automatic segmentation was identified by Peng et al. [10]. The authors developed software which supports three-dimensional proofreading. Also for 3D, Sicat et al. proposed a graph abstraction method to guide users to problematic regions indicating the need for corrections [12]. Janelia Farms then introduced Raveler [5], a software targeting expert users by offering many parameters for tweaking the proofreading process at the cost of a higher complexity. Raveler also includes a simple 3D renderer to validate corrections across slices.

In 2014, we developed two proofreading tools: Mojo and Dojo. Mojo is a stand-alone proofreading tool with a simple scribble interface for error correction. The need for distributed proofreading lead us to develop Dojo, a web-based proofreading framework with a minimalistic user interface. We defined requirements for proofreading tools in general and then evaluated the accuracy and speed of Raveler, Mojo and Dojo through a quantitative user study (Sec. 3 and 4) [4]. Al-Awami et al. then integrated Dojo into their proofreading management system Neuroblocks [1]. The system includes a scalable visualization for tracking the proofreading process among multiple users.

In 2015, Karimov et al. proposed a guided volume editing approach using histogram dissimilarity [6]. Measuring differences in histogram distributions helps their system to find potential errors and to suggest possible corrections. Their method shows promising results on Computed-Tomography Angiography datasets but is targeted towards expert users.

Recently, Uzunbas et al. showed that potential labeling errors can be found by taking the merge tree of an automatic segmentation method into account [13]. The authors keep track of uncertainties during the automatic labeling and then present them as regions for proofreading. This requires access to the merge tree of the segmentation stage which is not always available.

3 Preliminary Methods

We explored methods for fully interactive proofreading and compared different available software products. Since correcting segmentations is especially difficult for novice users, we then started looking into ways on how to use machine learning to support proofreading.

3.1 Interactive Proofreading

A powerful proofreading tool is crucial for enhancing segmentations efficiently and effectively. Particularly, communicating the three dimensional property of EM stacks is essential for identifying segmentation errors in 3D and for confirming the correctness of changes in the automatic segmentation. Existing software solutions tend to be geared towards domain experts and often have a steep learning curve. One example is Raveler [5], a stand-alone proofreading tool developed at the Janelia Farms Research Campus. The

huge scale of the data that needs to be segmented and proofread, however, requires that proofreading is crowdsourced. This means that proofreading really has to be performed by non-domain-experts and novice users.

In collaboration with neuroscientists at the Harvard Center for Brain Science we have developed two software tools to increase the efficiency, accuracy and usability for proofreading large and complex EM data. First, we have developed Mojo - a powerful standalone software application for experienced as well as non-expert users with advanced semi-automatic proofreading features. Building on the experience we gained from Mojo, we developed Dojo ("Distributed Mojo") - a web-based, distributed proofreading application (Fig. 2). Dojo is geared towards non-expert users and offers easy access through a web browser. Additional features include 3D volume rendering and parameter free tools for collaborative proofreading.

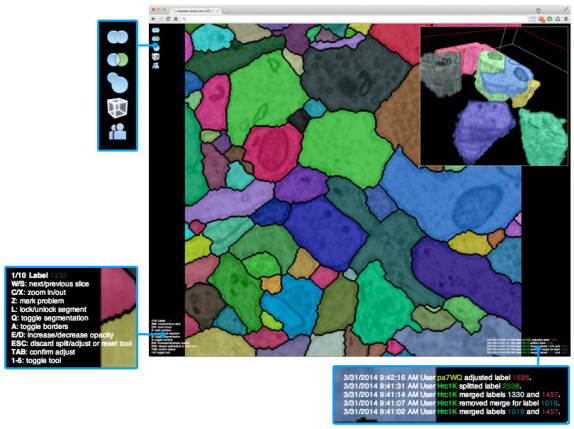

Figure 2: The user interface of Dojo consists of the 2D slice view (center), the 3D volume renderer (top right corner), the toolbox (top left corner), additional textual information (bottom left corner) and an activity log including all connected users (bottom right corner).

In a paper published at IEEE Vis 2014, we defined a set of requirements and design guidelines for visual proofreading applications [4]. We think that proofreading tools should be as minimalistic and as easy-to-use as possible. In addition, 3D visualization is especially important for novice users to grasp an understanding of the volumetric EM data. We evaluated our claims by performing a comparative user study between Mojo, Dojo and Raveler. 30 participants with no former experience in proofreading EM data were recruited and randomly assigned to each tool. The participants were asked to proofread a small representative subvolume of connectomics data for 30 minutes. Our results showed that novice users have problems performing this task (Sec. 4).

3.2 Machine Learning Supported Proofreading

With simple user interaction, split and merge errors can be corrected. However, even with semi-automatic correction tools, the visual inspection of the data to find the errors in the first place takes the majority of the time. Our goal was to add automatic detection of split and merge errors to proofreading tools. Instead of the user visually inspecting the whole image volume, we designed automatic classifiers that detect split and merge errors in 2D segmentations.

Inspired by work proposed by Bogovic et al. for label agglomeration [2], we built a split error classifier using a convolutional neural network. Our CNN checks the boundaries of an existing automatic segmentation. We use multiple channels of context information of the boundary for the decision making process such as the gray scale EM image, a probability mask, a label binary mask and a border binary mask (Fig. 3). Using this information, the CNN provides a probability for a split error for each boundary in automatically labeled data. Two variations of the CNN were trained: *RGBANet* with a combined 4-channel input and *MergeNet* with four separate input channels which then get concatenated. We sample up to 10 decision points for each boundary. The probabilities of the decision points then get weighted by the length of the boundary and averaged. A greedy algorithm finally merges neighboring regions sequentially, starting with the highest probability score.

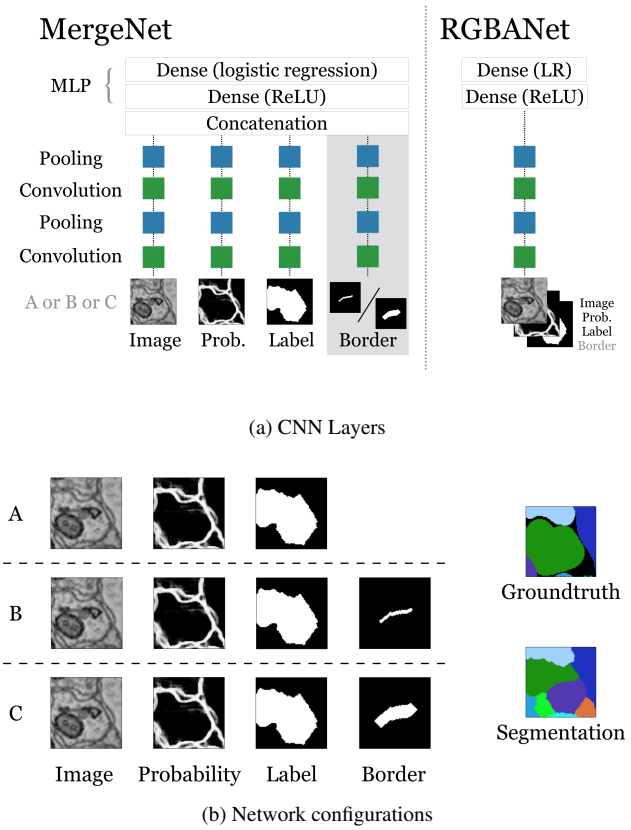

Figure 3: (a) Our network architecture with up to four input channels, each with two convolutional and two pooling layers. MergeNet includes four separate input channels and RGBANet uses a combined 4-channel input. (b) We trained three different network configurations with three and four inputs: A) image, boundary map probability, and merged binary mask; B) A configuration extended with a small border mask, to focus on the specific boundary in question; C) A configuration extended with a large border mask.

For merge errors, we generate randomly seeded watershed splits in a cell and then let our split error network rate their borders. If the CNN detects a boundary with a very low split score, then the boundary should have been in the segmentation and the region is a candidate for a merge error. This way, we were able to use one network for both split errors and merge errors.

We trained our CNN with 500k+ erroneous and correct patches of a manually labeled large connectomics volume (2048 x 2048 x 250 voxels). The training resulted in validation loss of 0.07 and test accuracy of 90.8%. These numbers lead to our initial hope that our network in combination with a greedy algorithm would be able to perform proofreading fully automatically. Unfortunately, this was not the case and it turned out that the user still has to be in the

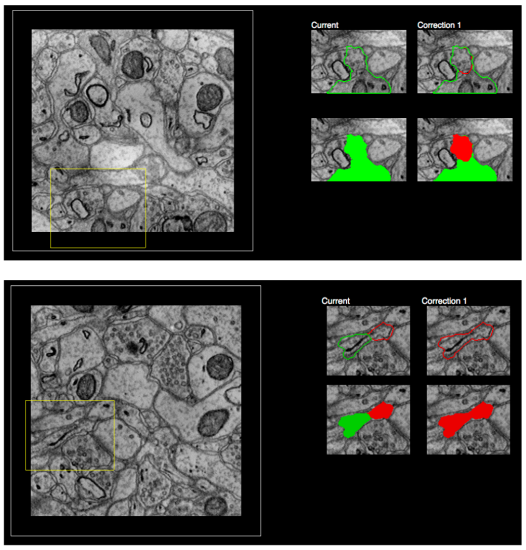

Figure 4: The recommendation user interface showing the current labeling of a cell boundary and the proposed correction as well as an overview of the whole image. A merge error is shown in the top and a split error is shown in the bottom.

loop (Sec. 4). For this purpose, we designed a very simple user interface which shows a proofreading error and a potential correction - one at a time (Fig. 4). The CNN then recommends likely errors and their corrections for proofreading within the interface. We call this approach *Guided Proofreading*.

4 Preliminary Results

First, we compared the interactive proofreading tools Mojo, Dojo and Raveler through a quantitative user study. Second, we used our machine learning based classifier to recommend likely errors for correction and compared the results against Dojo users.

4.1 Interactive Proofreading

We conducted a quantitative user study to evaluate the performance and usability of three proofreading tools. In particular, we evaluated how nearly untrained non-experts can correct an automatic segmentation using Mojo, Dojo and Raveler. We designed a between-subjects experiment and asked the participants to proofread a small dataset in a fixed time frame of 30 minutes. We used the most representative sub-volume of a larger freely available real-world data set where expert ground truth is available. We recruited participants from all fields with no experience in electron microscopy data or proofreading of such. As baseline, we asked two domain experts to label the same sub-volume from scratch using manual segmentation in the same fixed time frame.

We measured proofreading performance using the similarity measure variation of information (VI) between the ground truth and the proofreading output. The participants performed slightly better with Dojo (Fig. 5). Interestingly, the majority of participants made the initial segmentation worse rather than correcting it.

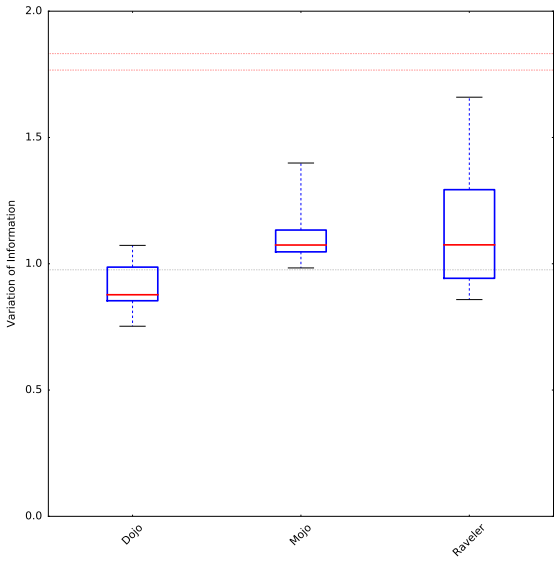

Figure 5: We compared the three interactive proofreading tools Dojo, Mojo and Raveler. Here we show the VI similarity measure for each tool after proofreading for 30 minutes (lower is better). Participants using Dojo achieved a lower VI than subjects with the other tools. These results are statistically significant. The gray line shows the VI of the automatic segmentation before proofreading. The red lines show the VI of the experts' manual segmentation after 30 minutes.

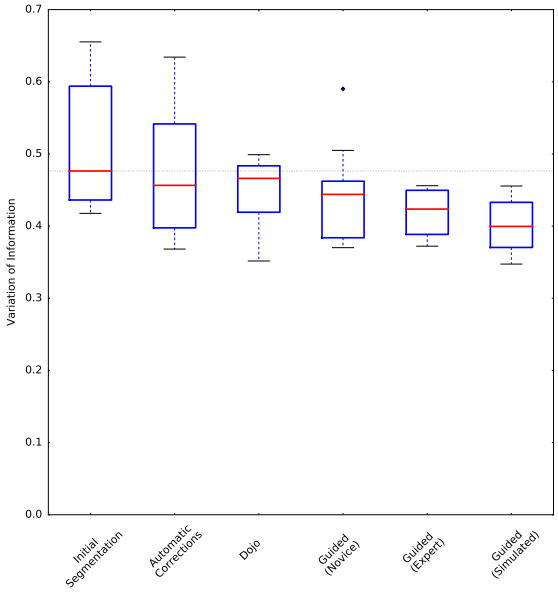

Figure 6: Guided Proofreading vs. Interactive Proofreading: We compare distributions of VI measures across 10 sections for the initial automatic segmentation, a fully-automatic correction of recommended errors based on a threshold of acceptance, the best participant in our Dojo user study, a novice user and two experts using our recommendation user interface and our simulated user. Lower scores are better.

This let us conclude the following: a) proofreading for novices is hard without further guidance, and b) a minimalistic and easy-to-use software tool enables better corrections. The detailed results are available in our paper [4] and the developed tools are available as open source software[1].

4.2 Machine Learning Supported Proofreading

We repeated the experiment described above but now let our classifier recommend likely errors for proofreading. Since the participants performed best using Dojo, we compared against the average and the best Dojo user. One novice user and two expert users were asked to perform proofreading using the recommendation interface (Fig. 4). The experimental conditions were exactly the same as in our interactive proofreading study.

Again, we measured proofreading performance using VI between the ground truth and the proofreading output. The recommendation interface enabled both novice and expert users to perform better than the best Dojo user (Fig. 6). This was not surprising since the classifier suggested where to proofread first.

To further evaluate our classifier, we performed the experiment fully automatic and with a simulated perfect user (error rate 0) and random recommendations. The fully automatic approach did not yield good results due to the complexity of the data - our greedy algorithm made things worse after a probability threshold was reached. The performance of random recommendations showed that the ranking by our classifier indeed is valuable.

Our work on machine learning supported proofreading is currently being extended by incorporating active learning.

5 Proposed Work

Proofreading will always be necessary since every automatic method makes errors. In addition, the human will always be the driving force behind every proofreading attempt. Based on our observations so far, we think that the ultimate proofreading tool is a combination of machine learning, user interface design and (information-)visualization.

We currently extend our trained CNN by adding an *active learning* component inspired by generative adversarial nets [3]. To improve our network and identify a possible pattern in wrong classifications, we use a second CNN (Fig. 7). This network is trained to distinguish inputs from our classifier and from human labeled annotations. Ideally, both inputs should be non-distinguishable - meaning that if the new network does not classify well, our original classifier does. However, outputs identified as originating from our original classifier will be re-evaluated by an oracle and re-added as training data to our original classifier.

Our hope is that the new network identifies a certain classification pattern in our original trained network. By re-adding these patches as training data, our original classifier will be fine tuned. Finally, we think that additional gain in proofreading performance can be made by adding visualizations to the user interface. We want to explore different visualizations to a) reduce the correction time for each error and b) aid the correction decisions (especially for novice users). Finally, our future proofreading solution will be used by hundreds of high school students in the Boston area to help process the large amounts of connectomics data.

6 Discussion

The task of automatic cell boundary segmentation is difficult, and trying to improve such segmentations automatically as a post-process through split and 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.

However, there is value in being able to recommend to users possible regions for correction, as the time cost of proofreading is dominated by the visual search for errors. Besides optimizing our CNN performance by leveraging an active learning approach, we think that adding (information-)visualization components can improve the results and/or reduce the time spent analyzing errors. The visualizations have to be simple and reduce rather than add complexity. Hopefully, this will ultimately lead to expert-comparable proofreading performance for novice users.

[1]Our interactive proofreading software and a live demo are available at http://rhoana.org/dojo/.

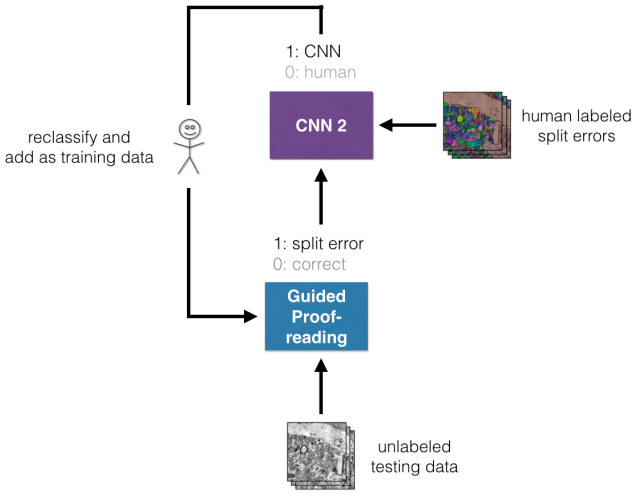

Figure 7: Our proposed active learning approach: *CNN 2* (purple) distinguishes whether a split error was identified by a human or by our guided proofreading classifier (blue). Errors detected as identified by our classifier are then re-evaluated by a human. If *CNN 2* classifies well, there is most likely a pattern of wrong classifications which then gets re-used as training data for our classifier.

7 Conclusion

Large amounts of images, such as in connectomics, require a lot of manpower to correct the automatic labeling. The proofreading task itself can not be done automatically and specifically requires the help of non-experts. Currently existing proofreading tools do not let novice users proofread efficiently - even with semi-automatic features. The perfect proofreading solution most likely requires a combination of machine learning, user interface design and visualization.

References

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

[3] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680, 2014.

[4] 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 (Proceedings IEEE SciVis 2014)*, 20(12):2466–2475, 2014.

[5] Janelia Farm. Raveler. https://openwiki.janelia.org/wiki/display/flyem/Raveler, 2014. Accessed on 31/03/2016.

[6] 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.

[7] 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.

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

[9] J. W. Lichtman, J. Livet, and J. R. Sanes. A technicolour approach to the connectome. *Nature reviews. Neuroscience*, 2008.

[10] 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.

[11] S. Seung. *Connectome: How the Brain's Wiring Makes Us Who We Are*. Houghton Mifflin Harcourt, Feb. 2012.

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

[13] 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.