

Cross-Classification Clustering: An Efficient Multi-Object Tracking Technique for 3-D Instance Segmentation in Connectomics

Yaron Meirovitch^{1,2*}, Lu Mi^{1*}, Hayk Saribekyan¹, Alexander Matveev³, David Rolnick¹, Nir Shavit^{1,4}
¹MIT, ²Harvard University, ³Neural Magic Inc., ⁴Tel-Aviv University

Abstract

Here we report on the supplementary materials for the paper “Cross-Classification Clustering: An Efficient Multi-Object Tracking Technique for 3-D Instance Segmentation in Connectomics”.

1. supplementary

We begin with a brief overview and then report on the following: **1.2. Neural Network Details** **1.3. Agglomeration** **1.4. Comparison with Baselines** **1.5. Performance of 3C on S1** **1.6. Video Results.**

1.1. Overview

Our paper presents a method for transferring instance segmentation from one image space to another using supervised learning (Instance Segmentation Transfer; IST). The IST method was utilized to generate 3-D instance segmentation of image stacks. This was achieved by firstly saturating the entire image space with 2-D seeds and then treating seed information as 2-D instance segmentation, and finally propagating seeds across images. We termed our IST solution *Cross-Classification Clustering* (3C) since it uses traditional classifications and supervised learning to cluster the space into an *a priori* unknown number of labels. We demonstrated our approach to 3-D instance segmentation on the difficult case of neural reconstruction in connectomics, a problem which requires swift and highly accurate inference.

1.2. Neural Network Details

Three types of network architecture were studied: **1)** 3C-Maxout, a light Maxout network [5] used for the large-scale experiment with the S1 dataset [3]. It has 6 output classes (1-4 3C colors, border prediction, new-object-prediction), and a $FoV=109\times109$. The input has 16 feature maps, encoding the l colors as equidistant space fractions between

0 and 1. We used $l = 4$. Each module includes a cascade of 64 4×4 convolutions, 2×2 maxpool and a maxout, as shown in Table 1. **2)** An even lighter 3C-Maxout network was used for the PNS dataset. It has 5 output classes (1-4 colors, not-axon), and $FoV=53\times53$, the input has 16 feature maps. The same structure is used as in 1), with half the features and a shallower network, as shown in Table 2. **3)** A module combination of convLSTM and UNET which we called 3C-LSTM-UNET. It has 4 classes (1-4 colors), and $FoV=512\times512$. Here we only report on the different types of layers included in this architecture, as shown in Table 4.

Our implementation of 3C-LSTM-UNET for the SNEMI3D dataset used Keras with Tensorflow backend. We ran all experiments on a server with a single Tesla V100-PCIE GPU. We trained our model by minimizing binary-entropy through back-propagation, using Adam with a learning rate of 10^{-4} . Our implementation of 3C-Maxout for the *PNS* and *S1* experiment used SGD solver in Caffe [2] with a learning rate of $7 \cdot 10^{-3}$.

We further compared the accuracy and number of parameters of 3C-LSTM-UNET with other commonly used network structures, such as UNET and Residual UNET, results are shown in Table 3.

1.3. Agglomeration

Our agglomeration step merges a 2-D seed with a seed propagated by the 3C algorithm, if the two segments overlap for at least 1/10 of the area of the originally placed seed. After the agglomeration step is terminated, some objects with biologically implausible properties, such as a small volume, improbable object flatness or impossible topology may occur. The problem of error detection in connectomics is not widely studied and is poorly understood, although recently several works suggested data-driven approaches to the problem, for example learning and detecting the morphology of implausible objects [11, 12]. Others applied a model-based approach for example for detecting biologically an implausible object topology of an X-junction [7] or by learning local and non-local biological constraints [4].

Orphans are objects that reside within the dataset but are either very small or do not reach the boundaries of the vol-

*These authors equally contributed to this work
{yaronm, lumi}@mit.edu

Type /layers	Number of kernels	Kernel size/Stride
conv1	128	4×4/1
maxpool	-	2×2/2
maxout 2:1	-	-
conv2	128	4×4/1
maxpool	-	2×2/2
maxout 2:1	-	-
conv3	128	4×4/1
maxpoo	-	2×2/2
maxout 2:1	-	-
conv4	128	4×4/1
maxpool	-	2×2/2
maxout 2:1	-	-
conv5	6	4×4/1

Table 1. Different types of layers with its corresponding parameters, number of kernels, and kernel size for the 3C-MaxOut architecture used for the S1 dataset.

Type /layers	Number of kernels	Kernel size/Stride
conv1	64	4×4/1
maxpool	-	2×2/2
maxout 2:1	-	-
conv2	64	4×4/1
maxpool	-	2×2/2
maxout 2:1	-	-
conv3	64	4×4/1
maxpool	-	2×2/2
maxout 2:1	-	-
conv4	5	4×4/1

Table 2. Different types of layers with its corresponding parameters, number of kernels, and kernel size for the 3C-MaxOut architecture used for the PNS dataset.

Network Structure	Number of Parameters	Training/Validation Accuracy
3C-LSTM-UNET	4M	0.963/0.961
3C-UNET-LSTM	4M	0.861/0.864
3C-UNET	31M	0.881/0.880
3C-Residual UNET	17M	0.915/0.863

Table 3. Comparison of 3C-LSTM-UNET, 3C-UNET-LSTM, 3C-UNET, 3C-Residual UNET on SNEMI3D data for number of parameters, training accuracy and validation accuracy.

ume. For teravoxel mammalian-cortex connectomics, all final objects (neuronal compartments) must be large enough to reach the datasets boundary. Thus, all orphans are incorrect split objects which if possible needs to be merged to other split or orphan objects. In this paper we used a simple heuristic to avoid the existence of any orphan in the

Module	Type /Layer	Number of kernels	Kernel size/Stride
ConvLSTM	convLSTM	20	3 × 3/1
	BN	-	-
UNET	conv1	32	3×3/1
	conv2	64	3×3/1
	conv3	128	3×3/1
	conv4	256	3×3/1
	conv5	512	3×3/1
	conv6	1	1×1/1
	BN	-	-
	maxpool	-	2×2/2
	upsample	-	2×2/1

Table 4. Different types of layers with its corresponding parameters, number of kernels and kernel size occur in module ConvLSTM and UNET for construction of 3C-LSTM-UNET architecture.

dataset. We marked all orphans and then iteratively merged the strongest edge between an orphan and a non-orphan object, where weighted edges are defined by the 3C algorithm. In the end, if the dataset still contained orphans, we iteratively connected the closest pair of an orphan and a non-orphan objects, by measuring distance on the affinity graph with the weighted edges defined by the border probability map.

1.4. Comparison with Baselines

Here we supplement the comparison from the main text for the SNEMI3D experiment. We provide additional 2-D results for sections $Z=5, 25, 45, 65, 85$, generated from the 3C-LSTM-UNET network, with the same ground truth and the two baselines reported in the main text, Neuroproof [8, 9, 10] and Watershed [5]. The results are shown in Figure 1. Each segmented image is provided with the corresponding value of Variation of Information (VI) [6].

1.5. Performance of 3C on S1 [3]

We implemented a light-weight version of 3C in order to swiftly reconstruct the large scale S1 dataset [3] (described in **Neural Network Details**) (implemented in C and Cilk with XNN backend [5]). This system generates borders on-the-fly and computes and updates seeds by the 3-D connected components of low border probability regions. Then 3C is used to propagate and correct seed information without any post-processing or agglomeration steps. The results of this experiment are shown in the main text as well as in the next section. Here we report on the performance on a single multi-core machine.

In this experiment, we executed on the same Intel infrastructure reported by Matveev et al.’s multi-core swift pipeline using their fast Cilk-Based, Cache-aware CPU im-

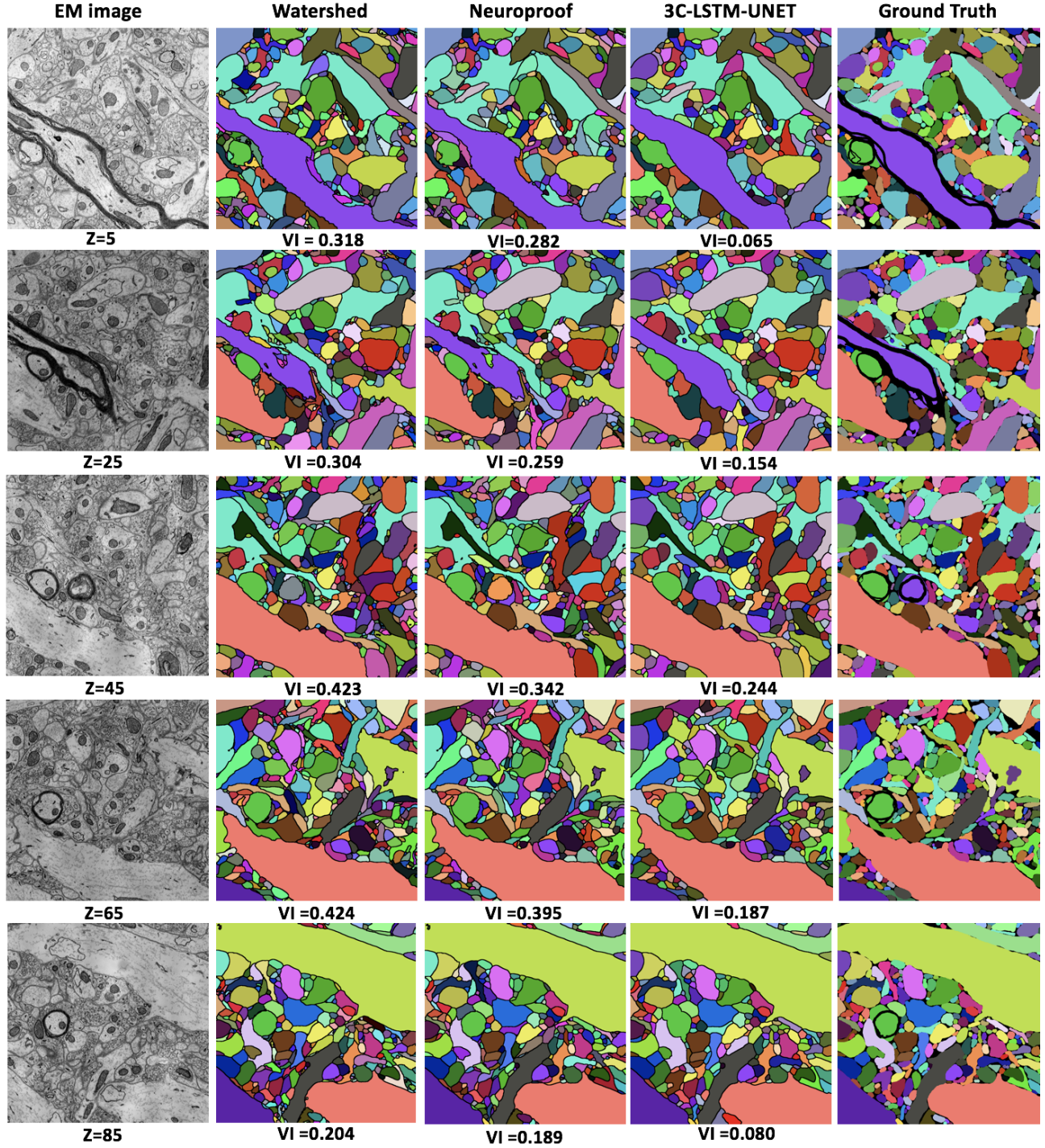


Figure 1. SNEMI3D: The 3C-LSTM-UNET Results compared with baseline techniques: Watershed, Neuroproof, and ground truth for sections Z=5, 25, 45, 65, 85.

plementation [5]. Although this pipeline is no longer considered state of the art, we compared to it here because it is the fastest reconstruction pipeline known to us (reconstruction S1 on a single machine in about 8 hours). The S1 mouse somatosensory cortex dataset of Kasthuri et al. [3]

contains 1840 electron microscopy images sized 180GB in total (90 gigavoxels of EM). The generation of a full segmentation of the dataset by the 3C algorithm took 129 hours, of which 40 hours were spent on the algorithm logic and I/O. The rest of the computation time (79 hours) was

spent on the CNN 4-socket CPU computation on an Intel (18x4)-core machine. Together with 8 hours of border generation, this amounts to a rate of 0.2MB/s. This is only an order of magnitude slower than the blitz pipeline of Matveev et al. [5] which processes 2.5MB of data per second, while our pipeline is delivering improved accuracy, both qualitatively, as seen in our video and images, and in terms of local accuracy as demonstrated in the main text in the Neuroproof test. These numbers indicate an order of magnitude improvement compared with the MaskExtend single-object reconstruction algorithm [7], aligned with the simulations presented in the main text, carried out against MaskExtend [7] and FFN [1].

1.6. Video Results

Finally, we provide here the list of videos illustrating the results of the 3C technique in three different reconstruction scenarios. These videos are given as supplementary material and will accompany the publication:

1. A video surveying the benchmark SNEMI3D reconstruction.
2. A video surveying the large-scale *SI* reconstruction.
3. A video surveying the large-scale *PNS* reconstruction.

References

- [1] Michał Januszewski, Jörgen Kornfeld, Peter H Li, Art Pope, Tim Blakely, Larry Lindsey, Jeremy Maitin-Shepard, Mike Tyka, Winfried Denk, and Viren Jain. High-precision automated reconstruction of neurons with flood-filling networks. *Nature methods*, 15(8):605, 2018.
- [2] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.
- [3] Narayanan Kasthuri, Kenneth Jeffrey Hayworth, Daniel Raimund Berger, Richard Lee Schalek, José Angel Conchello, Seymour Knowles-Barley, Dongil Lee, Amelio Vázquez-Reina, Verena Kaynig, Thouis Raymond Jones, et al. Saturated reconstruction of a volume of neocortex. *Cell*, 162(3):648–661, 2015.
- [4] Brian Matejek, Daniel Haehn, Haidong Zhu, Donglai Wei, Toufiq Parag, and Hanspeter Pfister. Biologically-constrained graphs for global connectomics reconstruction. *CVPR*, 2019.
- [5] Alexander Matveev, Yaron Meirovitch, Hayk Saribekyan, Wiktor Jakubiuk, Tim Kaler, Gergely Odor, David Budden, Aleksandar Zlateski, and Nir Shavit. A multicore path to connectomics-on-demand. In *Proceedings of the 22Nd ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, PPoPP ’17, pages 267–281, New York, NY, USA, 2017. ACM.
- [6] Marina Meilă. Comparing clusterings. *Journal of multivariate analysis*, 98(5):873–895, 2007.
- [7] Yaron Meirovitch, Alexander Matveev, Hayk Saribekyan, David Budden, David Rolnick, Gergely Odor, Seymour Knowles-Barley Thouis Raymond Jones, Hanspeter Pfister, Jeff William Lichtman, and Nir Shavit. A multi-pass approach to large-scale connectomics. *arXiv preprint arXiv:1612.02120*, 2016.
- [8] Juan Nunez-Iglesias, Ryan Kennedy, Toufiq Parag, Jianbo Shi, and Dmitri B Chklovskii. Machine learning of hierarchical clustering to segment 2D and 3D images. *PloS one*, 8(8):e71715, 2013.
- [9] Toufiq Parag, Anirban Chakraborty, Stephen Plaza, and Louis Scheffer. A context-aware delayed agglomeration framework for electron microscopy segmentation. *PloS one*, 10(5):e0125825, 2015.
- [10] Toufiq Parag, Fabian Tschopp, William Grisaitis, Srinivas C Turaga, Xuewen Zhang, Brian Matejek, Lee Kamentsky, Jeff W Lichtman, and Hanspeter Pfister. Anisotropic EM segmentation by 3D affinity learning and agglomeration. *arXiv preprint arXiv:1707.08935*, 2017.
- [11] David Rolnick, Yaron Meirovitch, Toufiq Parag, Hanspeter Pfister, Viren Jain, Jeff W Lichtman, Edward S Boyden, and Nir Shavit. Morphological error detection in 3d segmentations. *arXiv preprint arXiv:1705.10882*, 2017.
- [12] Jonathan Zung, Ignacio Tartavull, Kisuk Lee, and H Sebastian Seung. An error detection and correction framework for connectomics. In *Advances in Neural Information Processing Systems*, pages 6818–6829, 2017.