DESIGNING DEEP NEURAL NETWORKS TO AUTOMATE SEGMENTATION FOR SERIAL BLOCK-FACE ELECTRON MICROSCOPY

Matthew Guay[1] *Zeyad Emam*[2] *Adam Anderson*[3] *Richard Leapman*[1]

[1] National Institute of Biomedical Imaging and Bioengineering, Bethesda, MD 20892
[2] Applied Mathematics, University of Maryland, College Park, MD 20742
[3] Computer Science, University of Maryland, College Park, MD 20742

ABSTRACT

Today, serial block-face scanning electron microscopy (SBF-SEM) is capable of producing teravoxel-scale 3D images of biological structures at nanometer-scale resolutions. Image segmentation is fundamental to data analysis workflows in biological electron microscopy (EM), but SBF-SEM datasets can greatly exceed the manual segmentation capacity of a laboratory. Fast automated segmentation algorithms would alleviate this problem, but practical solutions remain unavailable for many biological problems of interest. Segmentation algorithms using deep neural networks have recently demonstrated significant performance gains, but designing high-performing networks that effectively solve targeted problems remains challenging. We are developing *genenet*, a Python package to rapidly discover, train, and deploy high-performing neural network architectures for SBF-SEM segmentation with little user intervention. Here, we demonstrate how to use *genenet* to train an ensemble of segmentation networks for a human platelet tissue sample. Initial results indicate this approach is viable for accelerating the segmentation process.

Index Terms— Deep learning, automated segmentation, electron microscopy, serial block-face imaging

1. INTRODUCTION

Electron microscopy (EM) allows biomedical researchers to investigate the structure of biological systems at spatial resolutions of 1-50 nm. Instrument limitations historically restricted EM sample thicknesses to 0.1-1 μm, but the development of serial block-face electron microscopy (SBF-SEM) has enabled 3D imaging of samples with thicknesses of hundreds of micrometers [1]. This data affords biologists new insight into the organization of large, complex tissue structures, but analyzing SBF-SEM datasets requires data analysis workflows capable of handling terabytes of data. Currently, one serious analysis bottleneck is *image segmentation*. Segmentation partitions an image volume into labeled regions corresponding to image content, and it is fundamental to many applications in biological EM - separating cells from background, or organelles from cytoplasm within cells. This is a slow, tedious process when done by hand, and interest in automating segmentation algorithmically has existed for decades. However, a segmentation algorithm is only practical if the time required to manually correct the algorithm's output is less than the time required to manually segment the algorithm's input, and for some problems this target remains elusive. Axon tracing in connectomics is one problem that has garnered attention [2], but analysis of other biological tissue faces similar challenges. The difficulty of automating a segmentation task depends on image complexity and the characteristics of objects in each segmentation class. It is common for biological research to analyze samples where features of interest exist at multiple spatial scales down to the microscope resolution limit, image contrast alone isn't enough to distinguish between classes, and extremely high accuracy is required for the result to be useful.

In the past decade, deep neural networks have demonstrated substantial performance improvements over previous methods in a number of computer vision problems, including EM segmentation. In 2012, Ciresan et al. [3] used a convolutional neural network to segment neural membranes with a sliding-window strategy - a 2D portion of an image volume is input to the network, the output is a classification of the center pixel of the image. This approach significantly outperformed competitors in the ISBI 2012 neural membrane segmentation challenge [4], but classifying an image volume one voxel at a time is slow and involves much redundant computation.

In 2015, Ronneberger et al. [5] demonstrated a network capable of simultaneously segmenting all pixels in a large image window. Their u-net was one of the first examples of a *convolutional encoder-decoder network* for biomedical segmentation, and many variants have followed [6, 7, 8]. In a convolutional encoder-decoder network, an encoding path uses convolutions and spatial pooling to decompose an input image into a multiscale collection of features, while a decoding path synthesizes an output image by upsampling and convolving combinations of encoder features. Convolutional

This work utilized the computational resources of the NIH HPC Biowulf cluster. (https://hpc.nih.gov)

encoder-decoder networks for image segmentation typically contain dozens or hundreds of layers, and many design decisions are required to specify a network architecture. Much remains unknown about optimal design patterns for many EM segmentation problems, and we aim to explore the design of encoder-decoder networks for the segmentation problems encountered in our lab.

2. METHODS

2.1. Genenet

The *genenet* library is designed to allow humans and algorithms to easily specify an encoder-decoder network architecture. Algorithmic architecture design allows biomedical researchers to discover high-performing networks for target applications with little manual intervention. In our work, networks consist of sequences of convolutions (*stacks*) at different spatial scales, connected by upsampling and downsampling operations. Design choices, such as the number of spatial scales or the number of convolution layers per stack, can be encoded as numeric hyperparameters, along with optimization hyperparameters such as learning rate and regularization weights. Before instantiation in TensorFlow, a network architecture is represented as a *Gene graph*.

A Gene is an object that is responsible for building a portion of a neural network's computation graph. A tree of Genes recursively builds the graph; leaf Genes build small subgraphs, and their constructions are assembled by a hierarchy of parent Genes to produce a full TensorFlow Graph. For each hyperparameter for each Gene, the hyperparameter value is a sum of the values stored in the Gene and each of its ancestors. This schema allows computation graph hyperparameters to be easily modified at a number of levels, from a single computation layer to the entire graph. An example of a Gene graph can be seen in Figure 1.

Gene graphs can be used to algorithmically construct neural networks in TensorFlow. The *genenet* library is capable of randomly sampling from spaces of encoder-decoder network architectures when supplied with a feasible region for all hyperparameters, as a rudimentary form of automated neural network architecture design. These networks can then be trained and their performance evaluated. For training, networks use the ADAM optimization method, minimizing a combination of class frequency-balanced prediction cross-entropy and ℓ^1 and ℓ^2 regularization terms applied to convolution layers.

These networks exhibit a diversity of architectures, a benefit for network ensembles due to the relationship between ensemble generalization error and ensemble ambiguity, a measure of disagreement between ensemble members [9]. An ensemble of networks can then be used to produce an image segmentation. Each network produces a class prediction map, a probability distribution over possible classes for each voxel in an image. The class prediction maps are averaged to produce an ensemble class prediction map. The final segmentation is created by choosing the most-probable class for each voxel from the class prediction map.

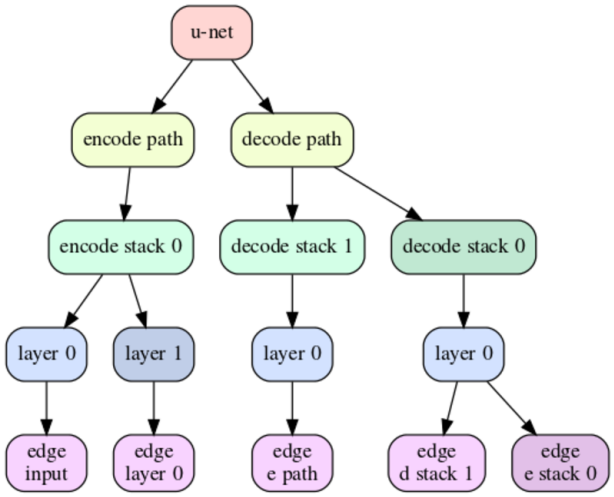

Fig. 1. The Gene graph of a small u-net. The root UNetGene has an EncodePathGene and DecodePathGene as children. Each path contains StackGenes specifying stacks of convolutions. Each convolution layer is built by a ComputationGene, whose children are EdgeGenes that specify which Genes' outputs to use as ComputationGene inputs.

Upon public release, *genenet* will be available from `https://github.com/LeapmanLab/genenet`.

2.2. Platelet tissue segmentation

We used *genenet* to train encoder-decoder networks to segment an SBF-SEM dataset derived from human platelet cells. The resin-embedded tissue sample was imaged using a Gatan 3View® to produce an image volume with dimensions $250 \times 2000 \times 2000$. We sought to segment this data into seven classes: background, cell cytoplasm, and five organelle classes - mitochondria, canalicular system, alpha granules, dense granules, and dense granule cores. A subvolume with dimensions $50 \times 800 \times 800$ was manually segmented to serve as training data. The 32000000 voxels in the training data consisted of 11541800 background voxels (36.1%), 17687776 cytoplasm voxels (55.3%), 196963 mitochondria voxels (0.62%), 1486295 alpha granule voxels (4.6%), 996845 canalicular voxels (3.1%), 75134 dense granule voxels (0.23%), and 15187 dense granule core voxels (0.047%).

We trained 80 randomly-generated networks over the course of 24 hours using the NIH's Biowulf computing cluster. Mutable hyperparameters were input size, number of features per convolution layer, number of spatial scales, number of convolution layers per stack, the (base-10) log of the

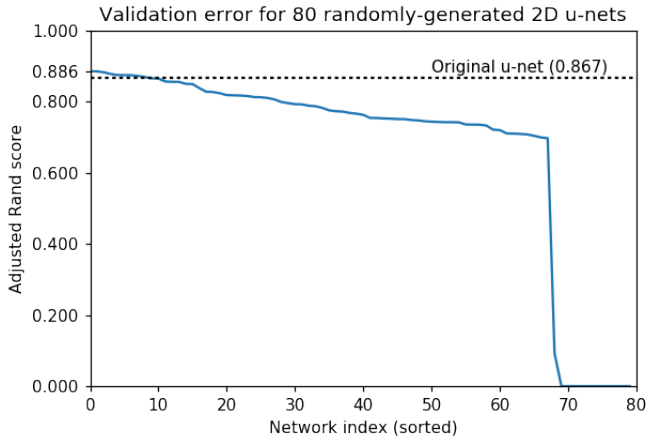

Fig. 2. Network validation adjusted Rand scores, sorted from best performance to worst. Nine networks outperformed the original 2D u-net in this metric.

learning rate, and the log of several regularization parameters - ℓ^1 and ℓ^2 penalty terms on convolution layer dictionary elements and convolution layer activations. As a baseline for comparison, we also trained a copy of the original u-net from [5], modifying the final 1×1 convolution to output 7 features instead of 2 to accommodate our problem's 7 segmentation classes. All networks were trained for 100000 iterations on a $40 \times 800 \times 800$ subset of the training data, with the remaining $10 \times 800 \times 800$ volume reserved for validation. Data augmentation via elastic deformation was used to expand the set of training data [5]. After training, networks were evaluated by computing adjusted Rand scores [10] on the validation data. We then computed validation adjusted Rand scores for ensembles of the n best networks for $n \in [1, 15]$ and found that $n = 4$ was optimal. The ensemble of the best 4 networks was used to segment a $10 \times 800 \times 800$ portion of the platelet volume. This testing volume lies directly above the training subvolume in the platelet dataset. Two lab members then corrected the ensemble output, tracking the time required for each 800×800 z-slice of the segmentation. This time was compared with the time required for each of the two lab members to manually segment comparable 800×800 portions of the image volume, in order to determine if the algorithm accelerates the segmentation workflow.

3. RESULTS

3.1. Random network generation

For each of the 80 randomly-generated networks, the final validation adjusted Rand scores are plotted in decreasing order in Figure 2. The top 9 networks have adjusted Rand scores higher than the original u-net evaluated on the same data, demonstrating that the random-sampling strategy is capable of producing high-performing network architectures. Gene graphs of those architectures are too large to include here, but a sense of the variation can be gleaned by comparing parameter counts, number of spatial scales, and numbers of convolution layers per stack. The latter two can be represented as a sequence of layer per stack counts, divided into layers per encoding stack (LPES) and layers per decoding stack (LPDS). Table 1 offers a comparison.

Net ID	Error	# Params	LPES	LPDS
13	**0.886**	39.3M	1,1,3,1	3,1,5,3,3
3	0.885	33.9M	1,2,2	6,6,3,4
45	0.883	20.1M	3,1,2,2	1,1,1,1,1
38	0.878	27.2M	6,6,6	4,5,5,5
71	0.875	33.5M	2,2,4,2	3,3,6,6,4
27	0.874	12.7M	4,5,4,3	3,1,3,1,2
57	0.874	4.7M	2,3	4,2,1
49	0.872	23.4M	1,2,1	2,1,1,2
32	0.869	28.1M	1,1,3	2,1,2,2
Original	0.867	31.0M	2,2,2,2	2,2,2,2,2

Table 1. A comparison of certain architectural parameters and validation error (adjusted Rand score) for the nine best randomly-generated networks and the original 2D u-net [5]. LPES: (convolution) layers per encoding stack. LPDS: layers per decoding stack.

3.2. Ensemble performance

The top-4 network ensemble achieved a validation adjusted Rand score of 0.901. A qualitative comparison of validation data, human segmentation, ensemble segmentation, and segmentation using the original u-net can be found in Figure 3. Note that both algorithmic outputs fail to properly classify the red and dark red objects (dense granules and dense granule cores), the least-common classes in the training data. The segmentation networks also find "phantom" organelles not identified by the human.

Manual correction of the top-4 ensemble output proved significantly faster than manual segmentation. For each 800×800 image z-slice, lab member 1 averaged 22.3 min per segmentation vs. 10.9 min per correction, a $2.04\times$ speedup. Lab member 2 averaged 18.6 min per segmentation vs. 7.9 min per correction, a $2.35\times$ speedup. Running time of the segmentation algorithm was comparatively negligible, taking no more than 2 seconds per z-slice.

4. CONCLUSION

Our work demonstrates that the random architecture generation process enabled by *genenet* is viable for creating high-performing encoder-decoder networks. The automated nature of this process means it can be used in settings where access

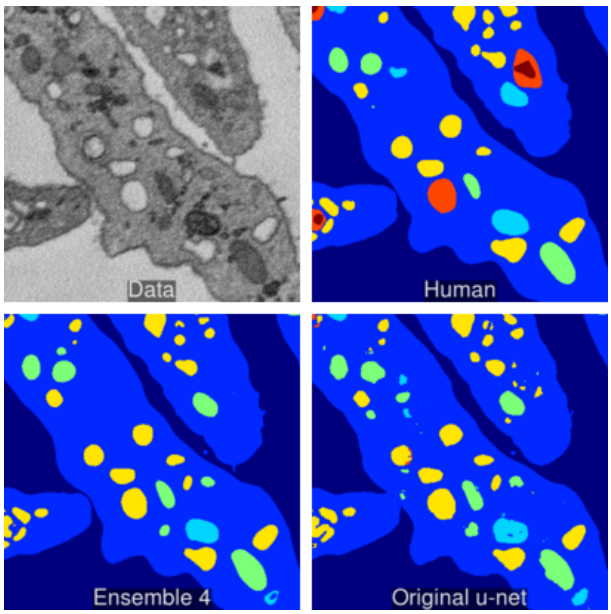

Fig. 3. A visualization of segmentation algorithm output for a validation data sample. The output of the top-4 ensemble and the original u-net [5] are compared with a human-created ground truth segmentation.

to a machine learning expert capable of making informed network design choices is not available.

We have also demonstrated that those networks can be combined to form ensembles which effectively accelerate the segmentation of a SBF-SEM dataset. The $10 \times 800 \times 800$ subvolume used to test this process is roughly 0.66% of the unprocessed platelet data, and further performance improvements are required to enable the efficient segmentation of the full image volume. Our next step is to produce a correction-training feedback loop, using corrected labels to produce new training data for the ensemble networks. The ensemble can then be used to produce new segmentations. If the addition of training data decreases correction times sufficiently, this loop can be used to segment the entirety of a large EM image volume.

Other improvements to the current ensemble algorithm are also possible. We trained a small number of 2D u-nets for a short time, and the random sampling strategy explored only a subset of the encoder-decoder architectures that can be realized with *genenet*. The errors which caused the greatest difficulty for correction involved misclassifying an existing organelle, and misclassifying non-organelle cellular material as an organelle. Even for humans, 3D context is required to avoid making those mistakes, and 3D encoder-decoder networks should benefit similarly. By leveraging these additional capabilities, we intend to use *genenet* to construct practical segmentation algorithms to fully exploit the wealth of data SBF-SEM can provide to biologists.

5. REFERENCES

[1] Winfried Denk and Heinz Horstmann, "Serial block-face scanning electron microscopy to reconstruct three-dimensional tissue nanostructure," *PLoS biology*, vol. 2, no. 11, pp. e329, 2004.

[2] Moritz Helmstaedter, Kevin L Briggman, Srinivas C Turaga, Viren Jain, H Sebastian Seung, and Winfried Denk, "Connectomic reconstruction of the inner plexiform layer in the mouse retina," *Nature*, vol. 500, no. 7461, pp. 168, 2013.

[3] Dan Ciresan, Alessandro Giusti, Luca M Gambardella, and Jürgen Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," in *Advances in neural information processing systems*, 2012, pp. 2843–2851.

[4] Ignacio Arganda-Carreras, Srinivas C Turaga, Daniel R Berger, Dan Cireşan, Alessandro Giusti, Luca M Gambardella, Jürgen Schmidhuber, Dmitry Laptev, Sarvesh Dwivedi, Joachim M Buhmann, et al., "Crowdsourcing the creation of image segmentation algorithms for connectomics," *Frontiers in neuroanatomy*, vol. 9, 2015.

[5] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 234–241.

[6] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger, "3d u-net: learning dense volumetric segmentation from sparse annotation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2016, pp. 424–432.

[7] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *3D Vision (3DV), 2016 Fourth International Conference on*. IEEE, 2016, pp. 565–571.

[8] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *arXiv preprint arXiv:1511.00561*, 2015.

[9] Anders Krogh and Jesper Vedelsby, "Neural network ensembles, cross validation, and active learning," in *Advances in neural information processing systems*, 1995, pp. 231–238.

[10] Lawrence Hubert and Phipps Arabie, "Comparing partitions," *Journal of classification*, vol. 2, no. 1, pp. 193–218, 1985.