

Two-Stage Generative Models of Simulating Training Data at The Voxel Level for Large-Scale Microscopy Bioimage Segmentation

Deli Wang^{1,2}, Ting Zhao³, Nenggan Zheng^{1,2*} and Zhefeng Gong⁴

¹Qiushi Academy for Advanced Studies, Zhejiang University

²School of Computer Science, Zhejiang University

³Janelia Research Campus, Howard Hughes Medical Institute

⁴School of Basic Medical Sciences, Zhejiang University

ephemera@zju.edu.cn, zhaot@janelia.hhmi.org, zng@cs.zju.edu.cn, zfgong@zju.edu.cn

Abstract

Bioimage Informatics is a growing area that aims to extract biological knowledge from microscope images of biomedical samples automatically. Its mission is vastly challenging, however, due to the complexity of diverse imaging modalities and big scales of multi-dimensional images. One major challenge is automatic image segmentation, an essential step towards high-level modeling and analysis. While progresses in deep learning have brought the goal of automation much closer to reality, creating training data for producing powerful neural networks is often laborious. To provide a shortcut for this costly step, we propose a novel two-stage generative model for simulating voxel level training data based on a specially designed objective function of preserving foreground labels. Using segmenting neurons from LM (Light Microscopy) image stacks as a testing example, we showed that segmentation networks trained by our synthetic data were able to produce satisfactory results. Unlike other simulation methods available in the field, our method can be easily extended to many other applications because it does not involve sophisticated cell models and imaging mechanisms.

1 Introduction

Since the digital age of imaging biological samples under microscopes, extracting useful information from microscope images using computers has become a major effort of computational biologists, leading to the growth of a new field called Bioimage Informatics [Peng, 2008]. The ultimate goal of the field is to convert images into biological knowledge automatically, which is expected to revolutionize how biologists make discoveries from their experiments [Meijering *et al.*, 2016]. One primary example is the reconstruction of wiring diagrams of the brain using automatic segmentation and object recognition [Takemura *et al.*, 2015; Takemura *et al.*, 2017]. Such computation-driven approaches, however, has not been extensively used in imaging pipelines of most biological re-

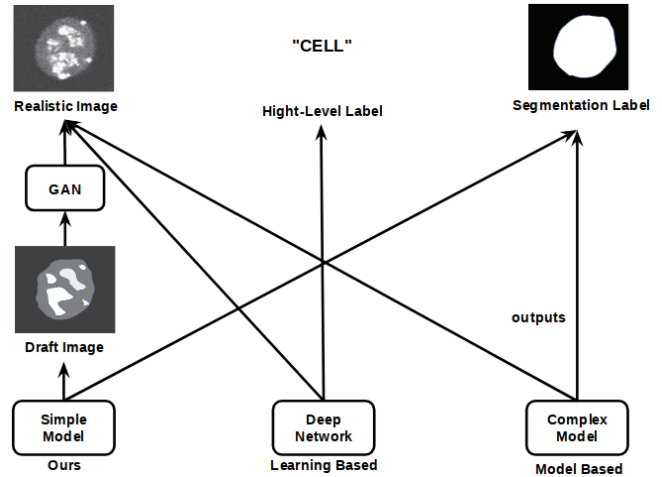


Figure 1: Illustration of image simulation approaches. Model-based methods can simulate images with segmentation labels, but it often needs expert knowledge design an accurate model, which cannot be easily generalized. Learning-based methods can simulate realistic images with less effort, but the existing methods cannot provide labels at the pixel resolution. Our proposed method uses a simple model to simulate draft images with segmentation labels and employs a GAN to refine the draft images while preserving the segmentation labels.

searches, due to the difficulty of developing effective image analysis for all varieties of bioimage data.

One major obstacle is segmentation, which labels 2D pixels or 3D voxels with numbers corresponding to biological objects [Meijering, 2012]. While a myriad of automatic segmentation methods have been developed and applied to various biological problems, none of them is flexible enough to handle a large variety of imaging systems, cell types, or unexpected imaging artifacts. In the current era of deep learning, it appears that the most promising way to tackle the challenge is by designing and training powerful segmentation networks [Long *et al.*, 2014; Ronneberger *et al.*, 2015; Yuan *et al.*, 2017]. This solution, however, relies on creating sufficient ground truth data, which itself is labor intensive, time consuming and error prone [Wiesmann *et al.*, 2017], let alone the difficulty of finding expert annotators in some research domains. The motivation of reducing the annotation

*Contact Author

hassle, and thus help bioimage informatics harness the power of deep neural networks, has led to the goal of our work: use computer simulation to replace the tedious manual annotation work. Specifically speaking, we want to find a method to simulate images of predefined labels and use them to train neural networks that can be applied to real images later.

As shown in Fig. 1, currently there are two major categories of microscope image simulation approaches, including those based on models [Wiesmann *et al.*, 2017; Weigert *et al.*, 2018] and those based on learning [Johnson *et al.*, 2017; Yuan *et al.*, 2018]. The first category focuses on modeling the whole imaging process. To make the simulated images more realistic, numerous details, including object morphologies, intensities (or colors), sample properties, imaging optics, noises and so on, must be carefully considered, even though some parameters can be learned from real images [Zhao and Murphy, 2007; Peng and Murphy, 2011]. Simulating images in this way is a challenging task and has to be tuned by experts whenever there is a change of imaging condition. The second category takes advantages of the emerging deep learning technologies to generate visually realistic images. Generated from inputs with random configuration, these simulated images do not provide their pixel-level labels, limiting their application in segmentation. In order to simulate images that can be used to train segmentation networks, low level labels must be preserved. Unfortunately, to the best of our knowledge, there exists no generative network with such a property. Here, we propose a novel image simulation method based on two-stage generative models, which combines the advantages of model-based and learning-based methods, with a special objective function that preserves low level labels. Minimizing this objective function should lead to visually realistic images that are reliable for training while paired with predefined labels.

To show the effectiveness of our method, we test it on the problem of tracing neurons from light microscope (LM) image stacks, an important step in mapping brain circuits [Peng *et al.*, 2010; Feng *et al.*, 2015]. It is an open problem that has drawn much attention [Peng *et al.*, 2015], leading to the development of numerous algorithms with different strengths and weaknesses. Combining these methods can perform better than a single one. There are efforts of using machine learning to increase the generality of a single algorithm [Gu and Cheng, 2015], and recently, deep learning has been introduced into this area too [Zheng and Hong, 2016] [Liu *et al.*, 2017]. It has been shown that segmenting images using deep neural networks as a pre-processing step could significantly improve the performance of various neuron tracing algorithms [Li *et al.*, 2017; Liu *et al.*, 2018]. To unleash the power of machine learning, more high-quality ground-truth data are needed. It is usually a tedious task to trace neurons from LM images, and even with manually traced neurons, some heuristics has to be used to define the foreground for training [Liu *et al.*, 2017].

To overcome the challenge, we use our method to generate sufficient training data automatically. Specifically, we design a simple model, which can be done easily with a little prior knowledge, to simulate preliminary draft 3D LM image stacks (with segmentation labels) containing neurons of

basic tree morphologies, with only intensity histograms and blurring effects considered to simulate an arbitrary imaging process. Using a special morphology-preserving loss function, we build a GAN [Goodfellow *et al.*, 2014] that is capable of refining the draft stacks towards realistic intensity patterns at the voxel level while preserving their segmentation labels. The synthetic data is then used to train a well-verified deep segmentation network [Ronneberger *et al.*, 2015]. Experimental results show that the trained segmentation network functions satisfactorily on segmenting neurons from real LM neuron stacks, revealing a great potential of our proposed work.

We summarize our contributions as follows:

- Proposed a novel and general two-stage generative model, which combines the advantages of model-based and learning-based methods, to overcome the training data bottleneck in segmenting biological microscope images by neural networks.
- Built a morphology-preserving GAN with the ability of modifying images at the voxel level to refine simulated images.
- Showed the effectiveness of our method by applying it to the significant and challenging problem of neuron tracing from LM image stacks.

2 Method

2.1 Two-Stage Simulation Model

From coarse to fine, our model of simulating training data consists of two stages, which is illustrated in Fig. 2. At the first stage, we design a simple model M_γ by using our prior knowledge of LM neuron stacks, including neuron morphologies, intensity histograms, noises and blur effects, to generate draft stacks. In this model, neuron morphologies are simulated first to provide segmentation labels. At the second stage, we employ a GAN consisting of a refiner network R_θ and a discriminator network D_ϕ to adjust draft stacks. Specifically, we designed a morphology-preserving loss function to preserve segmentation labels during refinement (Equation 4). The synergy between the two stages results in a big advantage of our method: we do not have to model the imaging process at a meticulous level. The first stage only needs to provide a good start point for the second one, which is capable of tuning coarse simulations into realistic images. While paired with their segmentation labels from the first stage, the tuned images are expected to train a segmentation network to segment real images accurately.

From the statistics point of view, our generative model $G = (M_\gamma, R_\theta)$ is a sampler of a distribution p_G , which is an estimation of the real data distribution p_{data} . If $p_G \approx p_{data}$, we can simulate an arbitrary number of realistic neuron stacks with segmentation labels by sampling from p_G . The original GAN obtains the generative model G by optimizing the following minimax objective function:

$$\min_G \max_D \mathcal{V}(G, D) = \min_G \max_D \mathbb{E}_{\mathbf{x} \sim p_{data}} \log[D(\mathbf{x})] + \mathbb{E}_{\mathbf{x} \sim p_G} \log[1 - D(\mathbf{x})], \quad (1)$$

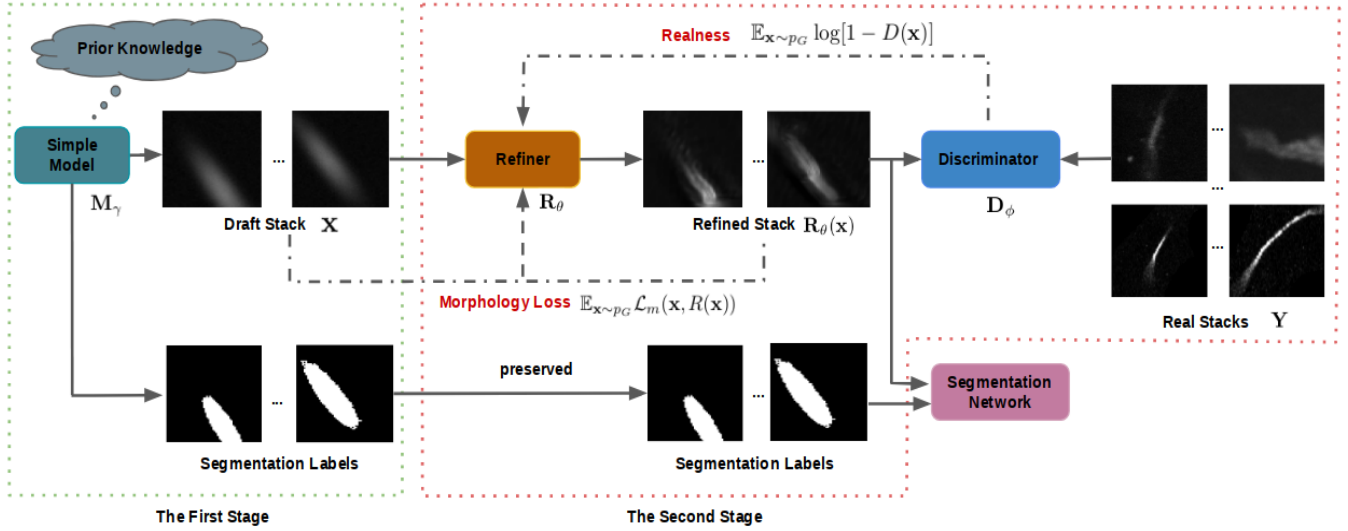


Figure 2: Architecture of our proposed two-stage LM neuron stack simulation method. The first stage simulates draft stacks and segmentation labels with a simple model; and the second stage refines draft stacks by learning features from real stacks while preserving segmentation labels.

where D is the discriminator. It is essentially minimizing the JS divergence of p_G and p_{data} :

$$JS(p_{data}||p_G) = \frac{1}{2}KL(p_{data}||\frac{p_{data} + p_G}{2}) + \frac{1}{2}KL(p_G||\frac{p_{data} + p_G}{2}), \quad (2)$$

where

$$KL(P(x)||Q(x)) = \int_x P(x) \log\left(\frac{P(x)}{Q(x)}\right). \quad (3)$$

If we start with some random p_G , optimizing Equation 1 probably fails because Equation 2 is almost a constant ($\log 2$) when p_G and p_{data} has little overlap. Unfortunately, this is often the case given the high dimensionality of p_{data} that reflects the complexity of neuron stacks. By incorporating some prior knowledge into M_γ , we start with a distribution much closer to p_{data} , or having much more significant overlap with p_{data} , than a random one does.

In order to fine tune the draft stacks and preserve their segmentation labels simultaneously, we define the following morphology-preserving objective function:

$$\min_R \max_D \mathcal{V}(G(M, R), D) = \min_R \max_D \mathbb{E}_{\mathbf{x} \sim p_{data}} \log[D(\mathbf{x})] + \mathbb{E}_{\mathbf{x} \sim p_G} \log[1 - D(\mathbf{x})] + \lambda \mathbb{E}_{\mathbf{x} \sim p_G} \mathcal{L}_m(\mathbf{x}, R(\mathbf{x})), \quad (4)$$

where \mathcal{L}_m is the morphology loss, which we will discuss in detail in Subsection 2.3.

2.2 Incorporate Prior Knowledge into M_γ

Generally, the simple model M_γ is where we incorporate prior knowledge. By sketching the objects to be modeled, we expect M_γ to provide a basic but reasonable estimation p_G^0 of p_{data} . Moreover, M_γ gives us the ability of controlling

the simulation process so that we have the chance to produce segmentation labels.

In the case of simulating LM neuron stacks, we model the neuron morphology and the imaging effects, including intensity histograms, noises and blur effects. To imitate neuron morphology, we use a series of nodes with different radii to simulate typical neuronal structures. Mathematically, the neuron morphology is simulated by a tree structure: $\{\mathbf{n}_i = (x_i, y_i, z_i, r_i, \mathbf{n}_j) | i = 1, \dots, N, j = 0, \dots, N, i \neq j, x_i, y_i, z_i, r_i \in \mathbb{R}\}$, where node \mathbf{n}_i is a sphere with center (x_i, y_i, z_i) and radius r_i . \mathbf{n}_j is the parent node of \mathbf{n}_i , and \mathbf{n}_0 is defined as an empty node to represent the root.

In general, the more precisely we model the imaging process of the target neuron image stacks, the closer p_G^0 and p_{data} will be, which means less work by the refiner. On the other hand, it requires more expert knowledge and intellectual work to design a more precise model, countering our goal of reducing human work. Therefore, our philosophy is to make M_γ as simple as possible, with the constraint that the draft stacks can be tuned by the refiner towards realistic appearance matching their underlying labels.

More formally speaking, if p_G is able to converge to p_{data} , for any positive number ϵ , there exists positive numbers δ and N , so that when the JS divergence of p_G^0 and p_{data} satisfies $JS(p_G^0, p_{data}) < \delta$, we will have $JS(p_G^N, p_{data}) < \epsilon$. That means we can obtain a satisfactory p_G within reasonable iteration steps as long as p_G^0 falls in some neighborhood of p_{data} . In our experiments, we find that the simulated intensity distributions, noises and blur effects are not necessary to be as precise as real ones to get such a good p_G^0 . Approximations as simple as Gaussian distributions and Gaussian kernels work well. Specifically, we

- use Gaussian distributions to approximate neuron intensity distribution and noises;
- convolve the neuron stack with a Gaussian kernel to sim-

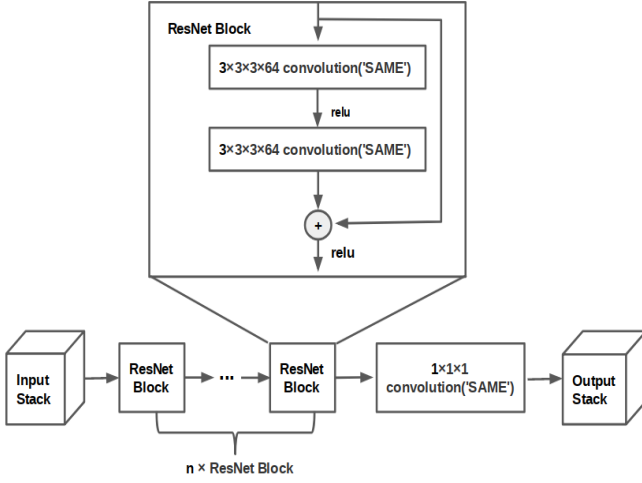


Figure 3: Structure of the refiner.

ulate blur effects.

2.3 Iterative Learning with MP Loss

In order to refine the draft stacks produced by M_γ to make them more realistically looking while preserving their segmentation labels, we employ the GAN, which consists of R_θ and D_ϕ , to learn features iteratively from real stacks.

While training R_θ , we keep the parameters of D_ϕ fixed, and minimize the following MP (Morphology Preserving) loss:

$$\mathcal{L}_{R_\theta} = \frac{1}{|X|} \sum_{\mathbf{x} \in X} \log[1 - D(\mathbf{x})] + \lambda \frac{1}{|X|} \sum_{\mathbf{x} \in X} \mathcal{L}_m(\mathbf{x}, R_\theta(\mathbf{x})), \quad (5)$$

where X is the set of draft stacks, λ defines a trade-off between preserving morphologies and being realistic, and

$$\mathcal{L}_m(\mathbf{x}, R_\theta(\mathbf{x})) = \frac{|\mathbf{x}_F \cup R_\theta(\mathbf{x})_F| - |\mathbf{x}_F \cap R_\theta(\mathbf{x})_F|}{|\mathbf{x}_F \cup R_\theta(\mathbf{x})_F|}, \quad (6)$$

where \mathbf{x}_F and $R_\theta(\mathbf{x})_F$ represent foreground voxels of \mathbf{x} and $R_\theta(\mathbf{x})$ respectively. $\mathcal{L}_m(\mathbf{x}, R_\theta(\mathbf{x}))$ measures the difference between foreground (neuron) voxels of the draft stack \mathbf{x}_F and the refined stack $R_\theta(\mathbf{x})$.

We compute $|\mathbf{x}_F \cup R_\theta(\mathbf{x})_F|$ and $|\mathbf{x}_F \cap R_\theta(\mathbf{x})_F|$ using the following approximations:

$$|\mathbf{x}_F \cup R_\theta(\mathbf{x})_F| \approx \frac{1}{2} \sum_i [g(\mathbf{x}_i) + g(R_\theta(\mathbf{x})_i) + |g(\mathbf{x}_i) - g(R_\theta(\mathbf{x})_i)|], \quad (7)$$

$$|\mathbf{x}_F \cap R_\theta(\mathbf{x})_F| \approx \frac{1}{2} \sum_i [g(\mathbf{x}_i) + g(R_\theta(\mathbf{x})_i) - |g(\mathbf{x}_i) - g(R_\theta(\mathbf{x})_i)|], \quad (8)$$

where

$$g(x) = \frac{1}{1 + e^{-K(x-\mu)}}, \quad (9)$$

Layers	Configurations
input	$64 \times 64 \times 16$
convolution	kernel size 5, stride 2, channels 64
max pooling	kernel size 2, stride 2
resnet block	$\times 3$
convolution	kernel size 3, stride 1, channels 128
resnet block	$\times 3$
convolution	kernel size 3, stride 1, channels 256
resnet block	$\times 3$
average pooling	kernel size 2, stride 2
flatten	
fully connected	1024 units
fully connected	2 units

Table 1: Configuration of the discriminator.

where K is a big number and

$$\mu = \arg \max_{\theta} \int_{\theta}^{255} p_f(x) dx \geq 0.95, \quad (10)$$

where p_f is the foreground intensity distribution of the draft stacks. As an offset term of $g(x)$, μ defines a threshold between foreground and background voxels by pushing $g(x)$ to 1 for $x > \mu$ and to 0 for $x < \mu$. With a large K value, $\frac{1}{2}[g(\mathbf{x}_i) + g(R_\theta(\mathbf{x})_i) - |g(\mathbf{x}_i) - g(R_\theta(\mathbf{x})_i)|] \approx 1$ if and only if \mathbf{x}_i and $R_\theta(\mathbf{x})_i$ are both foreground voxels; otherwise it is close to 0. Similarly, $\frac{1}{2}(g(\mathbf{x}_i) + g(R_\theta(\mathbf{x})_i) + |g(\mathbf{x}_i) - g(R_\theta(\mathbf{x})_i)|)$ is close to 1 iff either \mathbf{x}_i or $R_\theta(\mathbf{x})_i$ is a foreground voxel, and 0 otherwise. Converting $\mathcal{L}_m(\mathbf{x}, R_\theta(\mathbf{x}))$ to a differentiable form using $g(x)$ enables back-propagation in neural network training.

While training D_ϕ , we keep the parameters of R_θ fixed, and minimize:

$$\mathcal{L}_{D_\phi} = -\frac{1}{|X|} \sum_{\mathbf{x} \in X} \log(1 - D_\phi(R(\mathbf{x}))) - \frac{1}{|Y|} \sum_{\mathbf{x} \in Y} \log(D_\phi(\mathbf{x})), \quad (11)$$

where Y is the set of real neuron stacks.

In each training iteration, we train one step for D_ϕ and three steps for R_θ because we find that D_ϕ is much faster to converge than R_θ .

As illustrated in Fig. 3, we implement R_θ as a fully convolutional network, which consists of n resnet blocks [He *et al.*, 2016] and a $1 \times 1 \times 1$ convolution layer. The absence of striding and pooling enabled voxel-level modification of the draft stacks.

3 Experiments

3.1 Data

We used neuron stacks from BigNeuron¹ to evaluate our method. Launched as a community effort to define and advance state-of-the-art of LM neuron reconstruction, the BigNeuron project has collected a large set of neuron image stacks of different species and nervous system regions acquired by various light microscopy protocols. From this

¹<https://alleninstitute.org/bigneuron/data>

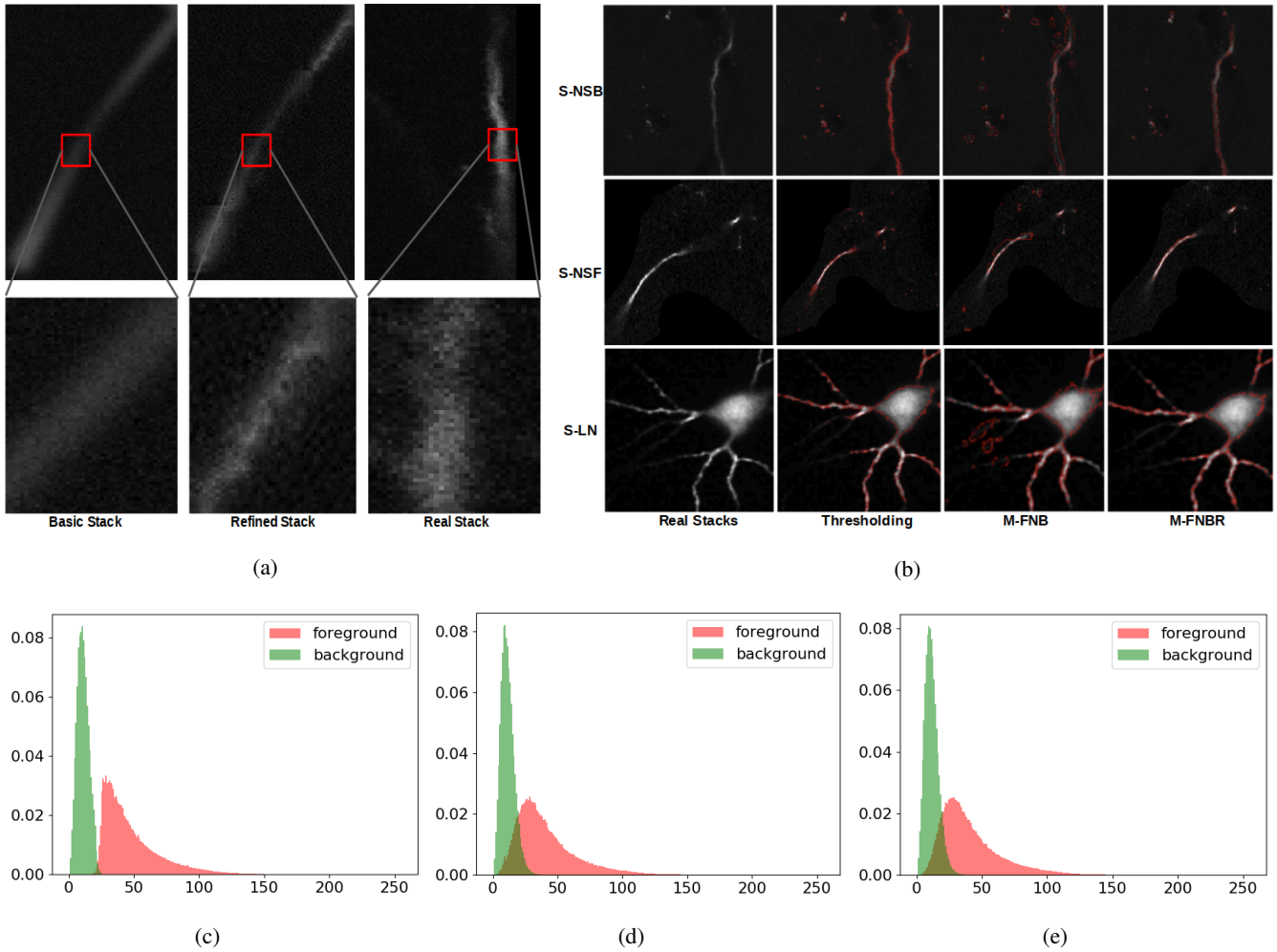


Figure 4: (a) Effects of the refiner; (b) Segmentation results. (c)~(e) Intensity distributions of draft stacks, refined stacks and real stacks.

data set, we selected three groups of stacks with different intensity characteristics to create our test data set:

- S-NSF: stacks with noises surrounding the foreground;
- S-NSB: stacks with noises spreading all over the background;
- S-LN: stacks with little noise.

Due to the big size of each single stack, which can be as big as $2000 \times 2000 \times 100$, it is not practical to input the whole stack into the discriminator and segmentation networks. Instead, we use small patches with the size of $64 \times 64 \times 16$. The test patches (about 5K) are selected from one stack and the training patches (about 50K) are selected from others. To exclude patches containing few foreground voxels, we use a $64 \times 64 \times 16$ sliding window with strides (16, 16, 4) to traverse each stack. Only if the average intensity of voxels in the window is greater than a predefined threshold, it will be selected. The segmentation ground truth was generated by tracing neurons semi-automatically in neuTube [Feng *et al.*, 2015], open-source software widely used for digital neuron reconstruction.

3.2 Implementation Details

Discriminator. The configuration of the discriminator is shown in Table 1.

Parameters. The number of resnet blocks in the refiner is set to 12. The coefficient λ in Equation 5 is set to 0.5, which gives a balance of preserving morphology and learning features from real stacks in our experiments.

3.3 Results

Qualitative Results

Fig. 4(a) shows the refining effects of our refiner. In the global view, it keeps the morphology, which means the shape of the simulated neuron is preserved after refinement; and in the local view, it learns realistic voxel patterns, such as incorporating certain randomness into the typical intensity distribution, which generally has brighter values for voxels closer to the central skeleton of the neuron. As shown Fig. 4(c), the distributions of foreground and background voxels in a draft stack are barely mixed. After being tuned by the refiner, the

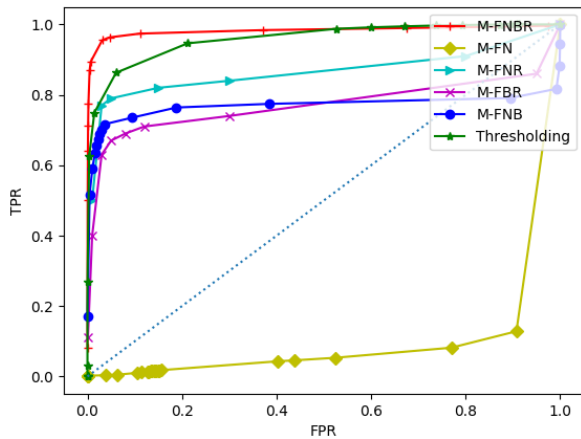


Figure 5: Roc curves made by applying segmentation thresholds. The AUC of M-FNBR is the biggest, which means M-FNBR has the best generalization ability, indicating the effectiveness of our synthetic data.

two distributions have a much more significant overlap (Fig. 4(d)), resembling real distributions quite well (Fig. 4(e)).

Quantitative Results

For each group of stacks, we trained five U-Net models [Ronneberger *et al.*, 2015] to segment real LM stacks. All parameters and training environments of these models are the same except for the training data. The model names and their training data details are shown in Table 2. The segmentation results in Table 3 and Fig. 5 show that the model trained by our synthetic data (M-FNBR) has the best overall performance in the experiments. When compared to thresholding, M-FNBR separates foreground and background more precisely and consistently (Fig. 4(b)).

Table 3 also shows that M-FN performs poorly on real LM stacks segmentation task, which suggests that simulating intensity histograms and noises only is almost useless to generate data for training neural networks. M-FNR performs much better than M-FN, but it is still not comparable to M-FNBR, indicating that modeling blur effects, even in an oversimplified way, is helpful. For stacks in the S-LN group, M-FBR gives good results, but it performs poorly on S-NSF and S-NSB, which means that adding background noises is necessary when simulating stacks containing notable noises. Last but most interestingly, we can tell how useful the refiner is by comparing M-FNB and M-FNBR. As shown in Table 3 and Fig. 4(b), the precision values of M-FNB are much lower than those from M-FNBR, indicating a dramatic increase of false positives caused by the absence of the refiner. It implies that the refiner is able to produce data that can teach neural networks to distinguish complicated foreground and background intensity patterns.

4 Conclusion

Extracting useful knowledge from LM stacks often relies on segmenting biological objects from noisy background. Automatic segmentation has been widely considered as a major bottleneck of automating the discovery process, despite

Model	Foreground	Noises	Blur	Refined
M-FNBR	✓	✓	✓	✓
M-FN	✓	✓		
M-FNR	✓	✓		✓
M-FBR	✓		✓	✓
M-FNB	✓	✓	✓	

Table 2: Model names and training data details

Stack	Model	Precision	Recall	F1
S-NSF	Thresholding	0.283	0.749	0.411
	M-FNBR	0.704	0.859	0.774
	M-FN	0.906	0.001	0.002
	M-FNR	0.653	0.821	0.727
	M-FBR	0.214	0.836	0.340
	M-FNB	0.606	0.901	0.725
S-NSB	Thresholding	0.582	0.727	0.647
	M-FNBR	0.586	0.865	0.698
	M-FN	0.864	0.103	0.184
	M-FNR	0.537	0.694	0.605
	M-FBR	0.368	0.736	0.490
	M-FNB	0.308	0.705	0.429
S-LN	Thresholding	0.770	0.625	0.689
	M-FNBR	0.796	0.601	0.685
	M-FN	0.883	0.007	0.013
	M-FNR	0.633	0.471	0.540
	M-FBR	0.538	0.912	0.676
	M-FNB	0.272	0.632	0.381

Table 3: Segmentation results.

decades of efforts from computational biologists. While a new hope has sprung from the advancement of deep segmentation networks, the progress of applying powerful networks are slowed down by a lack of sufficient training data. To tackle this challenge, we proposed a two-stage training data simulation method. At the first stage, we incorporate prior knowledge into a simple model to generate draft neuron stacks with voxel-level labels. At the second stage, we employ a GAN to adjust draft stacks. In the refiner design, we proposed a morphology-preserving loss and its approximation for practical computing. By minimizing the morphology-preserving loss, we can make the simulated stacks look like real stacks without messing up the underlying labels, which is crucial for producing high-quality training data. Experimental results from the application on neuron tracing show that U-Net trained by our synthetic data functions satisfactorily on real tasks, suggesting a potentially significant impact of our proposed work on bioimage informatics.

Acknowledgments

This work is supported by the Zhejiang Provincial Natural Science Foundation (LR19F020005), National Natural Science Foundation of China (61572433) and thanks for a gift grant from Baidu inc. Also partially supported by the Fundamental Research Funds for the Central Universities.

References

- [Feng *et al.*, 2015] Linqing Feng, Ting Zhao, and Jinhyun Kim. neutube 1.0: A new design for efficient neuron reconstruction software based on the swc format. *Eneuro*, 2(1), 2015.
- [Goodfellow *et al.*, 2014] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Xu Bing, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *International Conference on Neural Information Processing Systems*, 2014.
- [Gu and Cheng, 2015] Lin Gu and Li Cheng. Learning to boost filamentary structure segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 639–647, 2015.
- [He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Sun Jian. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2016.
- [Johnson *et al.*, 2017] Gregory R Johnson, Rory M Donovan-Maiye, and Mary M Maleckar. Generative modeling with conditional autoencoders: Building an integrated cell. *arXiv preprint arXiv:1705.00092*, 2017.
- [Li *et al.*, 2017] Rongjian Li, Tao Zeng, Hanchuan Peng, and Shuiwang Ji. Deep learning segmentation of optical microscopy images improves 3-d neuron reconstruction. *IEEE Transactions on Medical Imaging*, 36(7):1533–1541, 2017.
- [Liu *et al.*, 2017] Siqi Liu, Donghao Zhang, Yang Song, Hanchuan Peng, and Weidong Cai. Triple-crossing 2.5 d convolutional neural network for detecting neuronal arbours in 3d microscopic images. In *International Workshop on Machine Learning in Medical Imaging*, pages 185–193. Springer, 2017.
- [Liu *et al.*, 2018] Min Liu, Huiqiong Luo, Yinghui Tan, Xueping Wang, and Weixun Chen. Improved v-net based image segmentation for 3d neuron reconstruction. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 443–448. IEEE, 2018.
- [Long *et al.*, 2014] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 39(4):640–651, 2014.
- [Meijering *et al.*, 2016] E Meijering, A. E. Carpenter, H. Peng, F. A. Hamprecht, and J. C. Olivo-Marin. Imagining the future of bioimage analysis. *Nature Biotechnology*, 34(12):1250, 2016.
- [Meijering, 2012] Erik Meijering. Cell segmentation: 50 years down the road [life sciences]. *IEEE Signal Processing Magazine*, 29(5):140–145, 2012.
- [Peng and Murphy, 2011] T. Peng and R. F. Murphy. Image-derived, three-dimensional generative models of cellular organization. *Cytometry Part A the Journal of the International Society for Analytical Cytology*, 79A(5):383–391, 2011.
- [Peng *et al.*, 2010] H. Peng, Z Ruan, F. Long, J. H. Simpson, and E. W. Myers. V3d enables real-time 3d visualization and quantitative analysis of large-scale biological image data sets. *Nature Biotechnology*, 28(4):348–353, 2010.
- [Peng *et al.*, 2015] Hanchuan Peng, Erik Meijering, and Giorgio A Ascoli. From diadem to bigneuron, 2015.
- [Peng, 2008] Hanchuan Peng. Bioimage informatics: a new area of engineering biology. *Bioinformatics*, 24(17):1827–36, 2008.
- [Ronneberger *et al.*, 2015] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing & Computer-assisted Intervention*, 2015.
- [Takemura *et al.*, 2015] Shin-ya Takemura, C Shan Xu, Zhiyuan Lu, Patricia K Rivlin, Toufiq Parag, Donald J Olbris, Stephen Plaza, Ting Zhao, William T Katz, Lowell Umayam, et al. Synaptic circuits and their variations within different columns in the visual system of drosophila. *Proceedings of the National Academy of Sciences*, 112(44):13711–13716, 2015.
- [Takemura *et al.*, 2017] Shin-ya Takemura, Yoshinori Aso, Toshihide Hige, Allan Wong, Zhiyuan Lu, C Shan Xu, Patricia K Rivlin, Harald Hess, Ting Zhao, Toufiq Parag, et al. A connectome of a learning and memory center in the adult drosophila brain. *Elife*, 6:e26975, 2017.
- [Weigert *et al.*, 2018] Martin Weigert, Kaushikaram Subramanian, Sebastian T Bundschuh, Eugene W Myers, and Moritz Kreysing. Biobeam—multiplexed wave-optical simulations of light-sheet microscopy. *PLoS Computational Biology*, 14(4):e1006079, 2018.
- [Wiesmann *et al.*, 2017] Veit Wiesmann, Matthias Bergler, Ralf Palmisano, Martin Prinzen, Daniela Franz, and Thomas Wittenberg. Using simulated fluorescence cell micrographs for the evaluation of cell image segmentation algorithms. *BMC Bioinformatics*, 18(1):176, 2017.
- [Yuan *et al.*, 2017] Zehuan Yuan, Lu Tong, Yirui Wu, Zehuan Yuan, Lu Tong, Yirui Wu, Zehuan Yuan, Lu Tong, and Yirui Wu. Deep-dense conditional random fields for object co-segmentation. In *Twenty-sixth International Joint Conference on Artificial Intelligence*, 2017.
- [Yuan *et al.*, 2018] Hao Yuan, Lei Cai, Zhengyang Wang, Xia Hu, Shaoting Zhang, and Shuiwang Ji. Computational modeling of cellular structures using conditional deep generative networks. *Bioinformatics*, 2018.
- [Zhao and Murphy, 2007] Ting Zhao and Robert F Murphy. Automated learning of generative models for subcellular location: building blocks for systems biology. *Cytometry Part A*, 71(12):978–990, 2007.
- [Zheng and Hong, 2016] Zhihao Zheng and Pengyu Hong. Incorporate deep-transfer-learning into automatic 3d neuron tracing. In *The First International Conference on Neuroscience and Cognitive Brain Information*, 2016.