

Three Dimensional Fluorescence Microscopy Image Synthesis and Segmentation

Chichen Fu
Purdue University
West Lafayette, Indiana

Soonam Lee
Purdue University
West Lafayette, Indiana

David Joon Ho
Purdue University
West Lafayette, Indiana

Shuo Han
Purdue University
West Lafayette, Indiana

Paul Salama
Indiana University-Purdue University
Indianapolis, Indiana

Kenneth W. Dunn
Indiana University
Indianapolis, Indiana

Edward J. Delp
Purdue University
West Lafayette, Indiana

Abstract

Advances in fluorescence microscopy enable acquisition of 3D image volumes with better image quality and deeper penetration into tissue. Segmentation is a required step to characterize and analyze biological structures in the images and recent 3D segmentation using deep learning has achieved promising results. One issue is that deep learning techniques require a large set of groundtruth data which is impractical to annotate manually for large 3D microscopy volumes. This paper describes a 3D deep learning nuclei segmentation method using synthetic 3D volumes for training. A set of synthetic volumes and the corresponding groundtruth are generated using spatially constrained cycle-consistent adversarial networks. Segmentation results demonstrate that our proposed method is capable of segmenting nuclei successfully for various data sets.

1. Introduction

Fluorescence microscopy is a type of an optical microscopy that uses fluorescence to image 3D subcellular structures [1, 2]. Three dimensional segmentation is needed to quantify and characterize cells, nuclei or other biological structures.

Various nuclei segmentation methods have been investigated in the last few decades. Active contours [3, 4] which minimizes an energy functional to fit desired shapes has been one of the most successful methods in microscopy image analysis. Since active contours uses the image gradient to evolve a contour to the boundary of an object, this method can be sensitive to noise and highly dependent on initial

contour placement. In [5] an external energy term which convolves a controllable vector field kernel with an image edge map was introduced to address these problems. In [6] 2D region-based active contours using image intensity to identify a region of interest was described. This achieves better performance on noisy image and is relatively independent of the initial curve placement. Extending this to 3D, [7] described 3D segmentation of a rat kidney structure. This technique was further extended to address the problem of 3D intensity inhomogeneity [8]. However, these energy functional based methods cannot distinguish various structures. Alternatively, [9, 10] described a method known as Squassh to solve the energy minimization problem from a generalized linear model to couple image restoration and segmentation. In addition, [11] described multidimensional segmentation using random seeds combined with multi-resolution, multi-scale, and region-growing technique.

Convolutional neural network (CNN) has been used to address problems in segmentation and object identification [12]. Various approaches, based on CNNs, have been used in the biomedical area [13]. U-Net [14] is a 2D CNN which uses an encoder-decoder architecture with skip connections to segment cells in light microscopy images. In [15] a multi-input multi-output CNN for cell segmentation in fluorescence microscopy images to segment various size and intensity cells was described. Since these approaches [14, 15] are 2D segmentation methods, they may fail to produce reasonable segmentation in 3D. More specifically, stacking these 2D segmentation images into 3D volume may result in misalignment in the depth direction [7]. Also, in [16] a method that trained three networks from different directions in a volume and combined these three results to produce a form

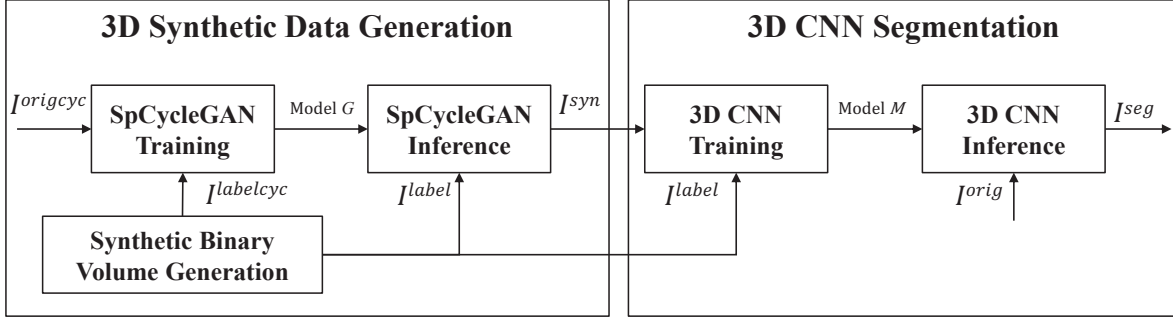


Figure 1. Block diagram of the proposed approach for 3D nuclei segmentation

of 3D segmentation was described. A 3D U-Net [17] was introduced to identify 3D structures by extending the architecture of [14] to 3D. However, this approach requires manually annotated groundtruth to train the network. Generating groundtruth for 3D volumes is tedious and is generally just done on 2D slices, obtaining true 3D groundtruth volumes are impractical. One way to address this is to use synthetic ground truth data [18, 19]. A method that segments nuclei by training a 3D CNN with synthetic microscopy volumes was described in [20]. Here, the synthetic microscopy volumes were generated by blurring and noise operations.

Generating realistic synthetic microscopy image volumes remains a challenging problem since various types of noise and biological structures with different shapes are present and need to be modeled. Recently, in [21] a generative adversarial network (GAN) was described to address image-to-image translation problems using two adversarial networks, a generative network and a discriminative network. In particular, the discriminative network learns a loss function to distinguish whether the output image is real or fake whereas the generative network tries to minimize this loss function. One of the extensions of GANs is Pix2Pix [22] which uses conditional GANs to learn the relationship between the input image and output image that can generate realistic images. One issue with Pix2Pix [22] is that it still requires paired training data to train the networks. In [23] coupled GANs (CoGAN) for learning the joint distribution of multi-domain images without having the corresponding groundtruth images was introduced. Later, cycle-consistent adversarial networks (CycleGAN) [24] employed a cycle consistent term in the adversarial loss function for image generation without using paired training data. More recently, a segmentation method using concatenating segmentation network to CycleGAN to learn the style of CT segmentation and MRI segmentation was described in [25].

In this paper, we present a 3D segmentation method to identify and segment nuclei in fluorescence microscopy volumes without the need of manual segmented groundtruth volumes. Three dimensional synthetic training data is generated using spatially constrained CycleGAN. A 3D CNN

network is then trained using 3D synthetic data to segment nuclei structures. Our method is evaluated using hand segmented groundtruth volumes of real fluorescence microscopy data from a rat kidney. Our data are collected using two-photon microscopy with nuclei labeled with Hoechst 33342 staining.

2. Proposed Method

Figure 1 shows a block diagram of our method. We denote I as a 3D image volume of size $X \times Y \times Z$. Note that I_{z_p} is a p^{th} focal plane image, of size $X \times Y$, along the z -direction in a volume, where $p \in \{1, \dots, Z\}$. Note also that I^{orig} and I^{seg} is the original fluorescence microscopy volume and segmented volume, respectively. In addition, let $I_{(q_i:q_f, r_i:r_f, p_i:p_f)}$ be a subvolume of I , whose x -coordinate is $q_i \leq x \leq q_f$, y -coordinate is $r_i \leq y \leq r_f$, z -coordinate is $p_i \leq z \leq p_f$, where $q_i, q_f \in \{1, \dots, X\}$, $r_i, r_f \in \{1, \dots, Y\}$, $p_i, p_f \in \{1, \dots, Z\}$, $q_i \leq q_f$, $r_i \leq r_f$, and $p_i \leq p_f$. For example, $I_{(241:272, 241:272, 131:162)}^{\text{seg}}$ is a subvolume of a segmented volume, I^{seg} , where the subvolume is cropped between 241st slice and 272nd slice in x -direction, between 241st slice and 272nd slice in y -direction, and between 131st slice and 162nd slice in z -direction.

As shown in Figure 1, our proposed method consists of two steps: 3D synthetic data generation and 3D CNN segmentation. We first generate synthetic binary volumes, I^{labelcyc} , and then use them with a subvolume of the original image volumes, I^{origcyc} , to train a spatially constrained CycleGAN (SpCycleGAN) and obtain a generative model denoted as model G . This model G is used with another set of synthetic binary volume, I^{label} , to generate corresponding synthetic 3D volumes, I^{syn} . For 3D CNN segmentation, we can utilize these paired I^{syn} and I^{label} to train a 3D CNN and obtain model M . Finally, the 3D CNN model M is used to segment nuclei in I^{orig} to produce I^{seg} .

2.1. 3D Synthetic Data Generation

Three dimensional synthetic data generation consists of synthetic binary volume generation, SpCycleGAN training, and SpCycleGAN inferences. In synthetic binary volume

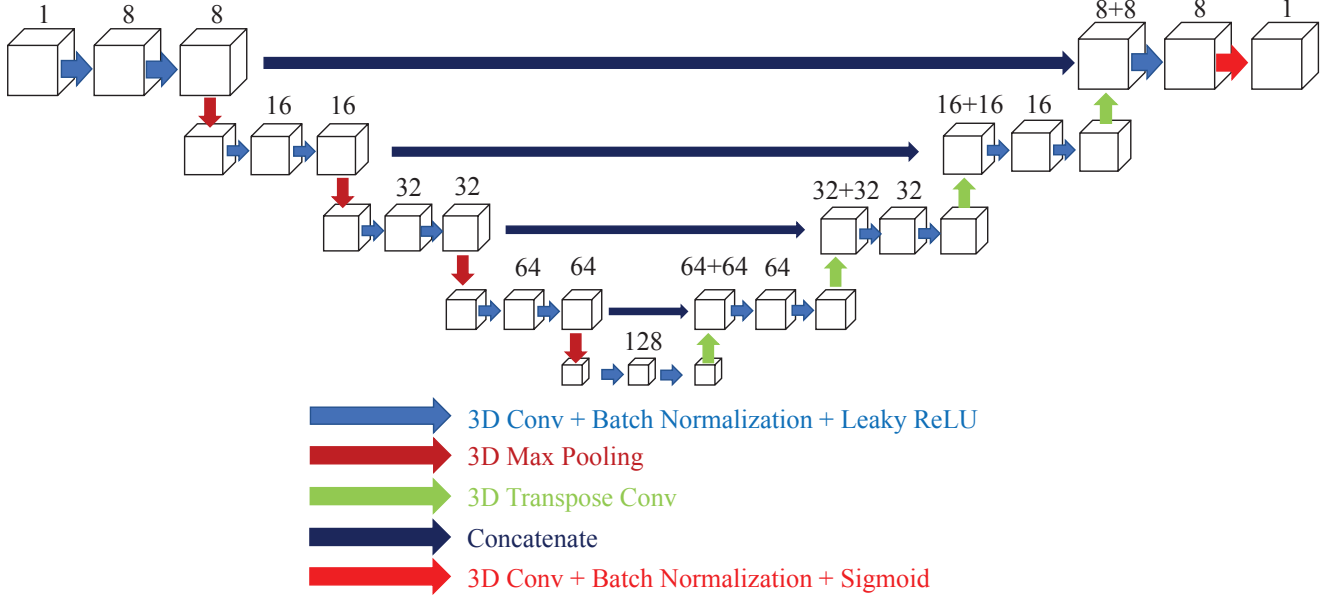


Figure 2. Architecture of our modified 3D U-Net

generation, nuclei are assumed to have an ellipsoidal shape, multiple nuclei are randomly generated in different orientations and locations in a volume [20]. The original CycleGAN and our SpCycleGAN were trained to generate a set of synthetic volumes.

2.1.1 CycleGAN

The CycleGAN is trained to generate a synthetic microscopy volume. CycleGAN uses a combination of discriminative networks and generative networks to solve a minimax problem by adding cycle consistency loss to the original GAN loss function as [21, 24]:

$$\begin{aligned} \mathcal{L}(G, F, D_1, D_2) = & \mathcal{L}_{\text{GAN}}(G, D_1, I^{\text{labelcyc}}, I^{\text{origcyc}}) \\ & + \mathcal{L}_{\text{GAN}}(F, D_2, I^{\text{origcyc}}, I^{\text{labelcyc}}) \\ & + \lambda \mathcal{L}_{\text{cyc}}(G, F, I^{\text{origcyc}}, I^{\text{labelcyc}}) \quad (1) \end{aligned}$$

where

$$\begin{aligned} \mathcal{L}_{\text{GAN}}(G, D_1, I^{\text{labelcyc}}, I^{\text{origcyc}}) = & \mathbb{E}_{I^{\text{origcyc}}} [\log(D_1(I^{\text{origcyc}}))] \\ & + \mathbb{E}_{I^{\text{labelcyc}}} [\log(1 - D_1(G(I^{\text{labelcyc}})))] \end{aligned}$$

$$\begin{aligned} \mathcal{L}_{\text{GAN}}(F, D_2, I^{\text{origcyc}}, I^{\text{labelcyc}}) = & \mathbb{E}_{I^{\text{labelcyc}}} [\log(D_2(I^{\text{labelcyc}}))] \\ & + \mathbb{E}_{I^{\text{origcyc}}} [\log(1 - D_2(F(I^{\text{origcyc}})))] \end{aligned}$$

$$\begin{aligned} \mathcal{L}_{\text{cyc}}(G, F, I^{\text{origcyc}}, I^{\text{labelcyc}}) = & \mathbb{E}_{I^{\text{labelcyc}}} [\|F(G(I^{\text{labelcyc}})) - I^{\text{labelcyc}}\|_1] \\ & + \mathbb{E}_{I^{\text{origcyc}}} [\|G(F(I^{\text{origcyc}})) - I^{\text{origcyc}}\|_1]. \end{aligned}$$

Here, λ is a weight coefficient and $\|\cdot\|_1$ is L_1 norm. Note that Model G maps I^{labelcyc} to I^{origcyc} while Model F maps I^{origcyc} to I^{labelcyc} . Also, D_1 distinguishes between I^{origcyc} and $G(I^{\text{labelcyc}})$ while D_2 distinguishes between

I^{labelcyc} and $F(I^{\text{origcyc}})$. $G(I^{\text{labelcyc}})$ is an original like microscopy volume generated by model G and $F(I^{\text{origcyc}})$ is generated by model F that looks similar to a synthetic binary volume. Here, I^{origcyc} and I^{labelcyc} are unpaired set of images. In CycleGAN inference, I^{syn} is generated using the model G on I^{label} . As previously indicated I^{syn} and I^{label} are a paired set of images. Here, I^{label} is served as a groundtruth volume corresponding to I^{syn} .

2.1.2 Spatially Constrained CycleGAN

Although the CycleGAN uses cycle consistency loss to constrain the similarity of the distribution of I^{origcyc} and I^{syn} , CycleGAN does not provide enough spatial constraints on the locations of the nuclei. CycleGAN generates realistic synthetic microscopy images but a spatial shifting on the location of the nuclei in I^{syn} and I^{label} was observed. To create a spatial constraint on the location of the nuclei, a network H is added to the CycleGAN and takes $G(I^{\text{labelcyc}})$ as an input to generate a binary mask, $H(G(I^{\text{labelcyc}}))$. Here, the architecture of H is the same as the architecture of G . Network H minimizes a L_2 loss, $\mathcal{L}_{\text{Spatial}}$, between $H(G(I^{\text{labelcyc}}))$ and I^{labelcyc} . $\mathcal{L}_{\text{Spatial}}$ serves as a spatial regulation term in the total loss function. The network H is trained together with G . The loss function of the SpCycleGAN is defined as:

$$\begin{aligned} \mathcal{L}(G, F, H, D_1, D_2) = & \mathcal{L}_{\text{GAN}}(G, D_1, I^{\text{labelcyc}}, I^{\text{origcyc}}) \\ & + \mathcal{L}_{\text{GAN}}(F, D_2, I^{\text{origcyc}}, I^{\text{labelcyc}}) \\ & + \lambda_1 \mathcal{L}_{\text{cyc}}(G, F, I^{\text{origcyc}}, I^{\text{labelcyc}}) \\ & + \lambda_2 \mathcal{L}_{\text{spatial}}(G, H, I^{\text{origcyc}}, I^{\text{labelcyc}}) \quad (2) \end{aligned}$$

where λ_1 and λ_2 are the weight coefficients for \mathcal{L}_{cyc} and $\mathcal{L}_{spatial}$, respectively. Note that first three terms are the same and already defined in Equation (1). Here, $\mathcal{L}_{spatial}$ can be expressed as

$$\mathcal{L}_{spatial}(G, H, I^{origyc}, I^{labelcyc}) = \mathbb{E}_{I^{labelcyc}}[||H(G(I^{labelcyc})) - I^{labelcyc}||_2].$$

2.2. 3D U-Net

Figure 2 shows the architecture of our modified 3D U-Net. The filter size of each 3D convolution is $3 \times 3 \times 3$. To maintain the same size of volume during 3D convolution, a voxel padding of $1 \times 1 \times 1$ is used in each convolution. A 3D batch normalization [26] and a leaky rectified-linear unit activation function are employed after each 3D convolution. In the downsampling path, a 3D max pooling uses $2 \times 2 \times 2$ with stride of 2 is used. In the upsampling path, feature information is retrieved using 3D transpose convolutions. Our modified 3D U-Net is one layer deeper than conventional U-Net as can be seen in Figure 2. Our training loss function can be expressed as a linear combination of the Dice loss (\mathcal{L}_{Dice}) and the binary cross-entropy loss (\mathcal{L}_{BCE}) such that

$$\mathcal{L}_{seg}(T, S) = \mu_1 \mathcal{L}_{Dice}(T, S) + \mu_2 \mathcal{L}_{BCE}(T, S) \quad (3)$$

where

$$\mathcal{L}_{Dice}(T, S) = \frac{2(\sum_{i=1}^N t_i s_i)}{\sum_{i=1}^N t_i^2 + \sum_{i=1}^N s_i^2}$$

$$\mathcal{L}_{BCE}(T, S) = -\frac{1}{N} \sum_{i=1}^N t_i \log(s_i) + (1 - t_i) \log(1 - s_i),$$

respectively [27]. Note that T is the set of the targeted groundtruth values and $t_i \in T$ is a targeted groundtruth value at i^{th} voxel location. Similarly, S is a probability map of binary volumetric segmentation and $s_i \in S$ is a probability map at i^{th} voxel location. Lastly, N is the number of entire voxels and μ_1, μ_2 serve as the weight coefficient between to loss terms in Equation (3). The network takes a grayscale input volume with size of $64 \times 64 \times 64$ and produces an voxelwise classified 3D volume with the same size of the input volume. To train our model M , V pairs of synthetic microscopy volumes, I^{syn} , and synthetic binary volumes, I^{label} , are used.

2.2.1 Inference

For the inference step we first zero-padded I^{orig} by 16 voxels on the boundaries. A 3D window with size of $64 \times 64 \times 64$ is used to segment nuclei. Since the zero padded I^{orig} is bigger than the 3D window, the 3D windows is slid to x, y , and z -directions by 32 voxels on zero-padded I^{orig} [20]. Nuclei partially observed on boundaries

of the 3D window may not be segmented correctly. Hence, only the central subvolume of the output of the 3D window with size of $32 \times 32 \times 32$ is used to generate the corresponding subvolume of I^{seg} with size of $32 \times 32 \times 32$. This process is done until the 3D window maps an entire volume.

3. Experimental Results

We tested our proposed method on two different rat kidney data sets. These data sets contain grayscale images of size $X = 512 \times Y = 512$. Data-I consists of $Z = 512$ images, Data-II consist of $Z = 64$.

Our SpCycleGAN is implemented in Pytorch using the Adam optimizer [28] with default parameters given by CycleGAN [24]. In addition, we used $\lambda_1 = \lambda_2 = 10$ in the SpCycleGAN loss function shown in Equation (2). We trained the CycleGAN and SpCycleGAN to generate synthetic volumes for Data-I and Data-II, respectively. A $128 \times 128 \times 128$ synthetic binary volume for Data-I denoted as $I^{labelcyc_{Data-I}}$ and a $128 \times 128 \times 300$ subvolume of original microscopy volume of Data-I denoted as $I^{origcyc_{Data-I}}$ were used to train model G^{Data-I} . Similarly, a $128 \times 128 \times 128$ synthetic binary volume for Data-II denoted as $I^{labelcyc_{Data-II}}$ and a $128 \times 128 \times 32$ subvolume of original microscopy volume of Data-II denoted as $I^{origcyc_{Data-II}}$ were used to train model $G^{Data-II}$.

We generated 200 sets of $128 \times 128 \times 128$ synthetic binary volumes, $I^{label_{Data-I}}$ and $I^{label_{Data-II}}$ where $I^{label_{Data-I}}$ and $I^{label_{Data-II}}$ are generated according to different size of nuclei in Data-I and Data-II, respectively. By using the model G^{Data-I} on $I^{label_{Data-I}}$, 200 pairs of synthetic binary volumes, $I^{label_{Data-I}}$, and corresponding synthetic microscopy volumes, $I^{syn_{Data-I}}$, of size of $128 \times 128 \times 128$ were obtained. Similarly, by using model $G^{Data-II}$ on $I^{label_{Data-II}}$, 200 pairs of $I^{label_{Data-II}}$ and corresponding $I^{syn_{Data-II}}$, of size of $128 \times 128 \times 128$ were obtained. Since our modified 3D U-Net architecture takes volumes of size of $64 \times 64 \times 64$, we divided $I^{label_{Data-I}}$, $I^{syn_{Data-I}}$, $I^{label_{Data-II}}$, and $I^{syn_{Data-II}}$ into adjacent non overlapping $64 \times 64 \times 64$. Thus, we have 1600 pairs of synthetic binary volumes and corresponded synthetic microscopy volumes per each data to train our modified 3D U-Net. Note that these 1600 synthetic binary volumes per each data are used as groundtruth volumes to be paired with corresponding synthetic microscopy volumes. Model M^{Data-I} and $M^{Data-II}$ are then generated.

Our modified 3D U-Net is implemented in Pytorch using the Adam optimizer [28] with learning rate 0.001. For the evaluation purpose, we use different settings of using 3D synthetic data generation methods (CycleGAN or SpCycleGAN), different number of pairs of synthetic training volume V ($V = 80$ or $V = 1600$) among 1600 pairs of synthetic binary volume corresponding synthetic microscopy volume. Also, we use different loss functions with different

settings of the μ_1 and μ_2 . Moreover, we also compared our modified 3D U-Net with 3D encoder-decoder architecture [20]. Lastly, small objects which are less than 100 voxels were removed using 3D connected components.

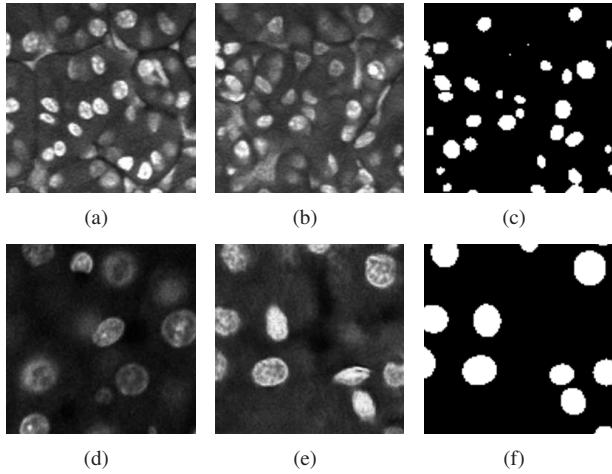


Figure 3. Slices of the original volume, the synthetic microscopy volume, and the corresponding synthetic binary volume for Data-I and Data-II (a) original image of Data-I, (b) synthetic microscopy image of Data-I, (c) synthetic binary image of Data-I, (d) original image of Data-II, (e) synthetic microscopy image of Data-II, (f) synthetic binary image of Data-II

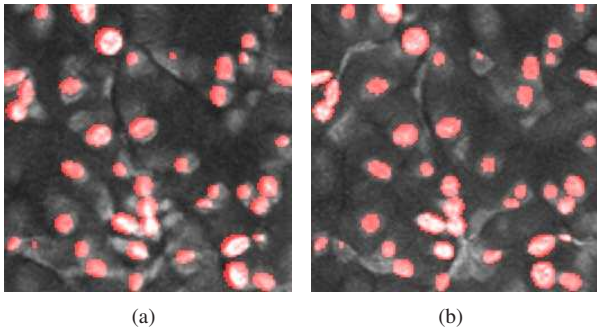


Figure 4. A comparison between two synthetic data generation methods overlaid on the corresponding synthetic binary image (a) CycleGAN, (b) SpCycleGAN

Figure 3 shows the synthetic images generated by our proposed method. The left column indicates original images whereas middle column shows synthetic images artificially generated from corresponding synthetic binary images provided in right column. As can be seen from Figure 3, the synthetic images reflect characteristics of the original microscopy images such as background noise, nuclei shape, orientation and intensity.

Additionally, two synthetic data generation methods between CycleGAN and SpCycleGAN from the same synthetic binary image are compared in Figure 4. Here, the synthetic binary image is overlaid on the synthetic microscopy image and labeled in red. It is observed that our spatial constraint loss reduces the location shift of nuclei between a

synthetic microscopy image and its synthetic binary image. Our realistic synthetic microscopy volumes from SpCycleGAN can be used to train our modified 3D U-Net.

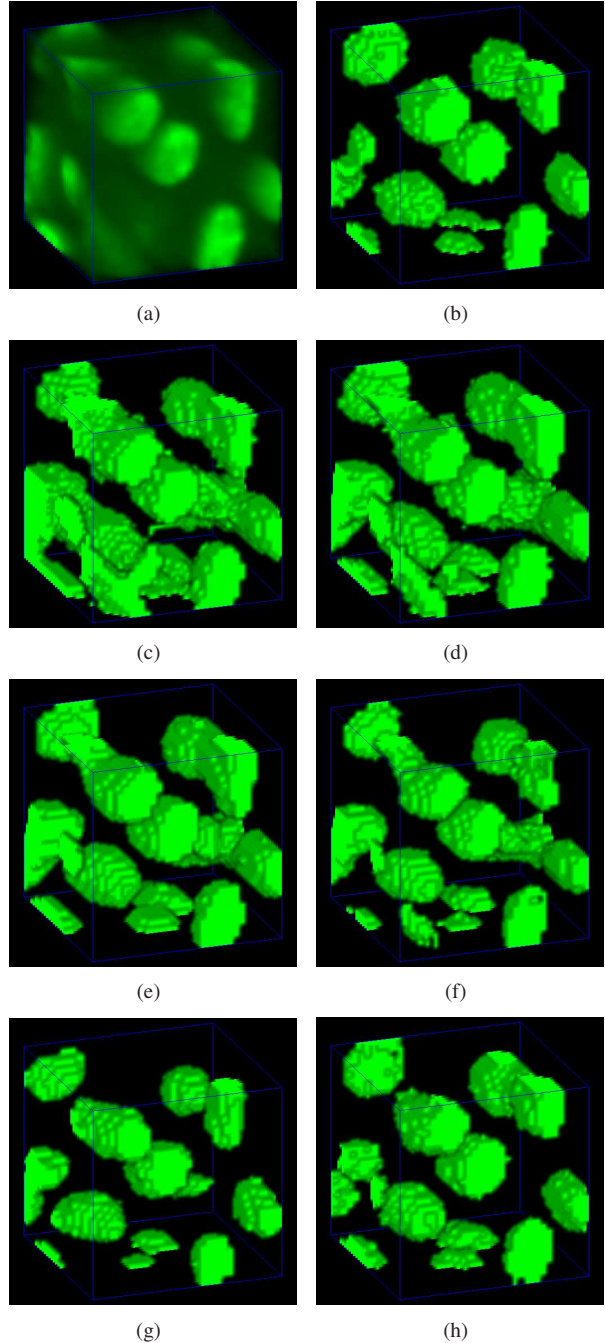


Figure 5. 3D visualization of subvolume 1 of Data-I using Voxx [29] (a) original volume, (b) 3D ground truth volume, (c) 3D active surfaces from [7], (d) 3D active surfaces with inhomogeneity correction from [8], (e) 3D Squassh from [9, 10], (f) 3D encoder-decoder architecture from [20], (g) 3D encoder-decoder architecture with CycleGAN, (h) 3D U-Net architecture with SpCycleGAN (Proposed method)

Table 1. Accuracy, Type-I and Type-II errors for known methods and our method on subvolume 1, subvolume 2 and subvolume 3 of Data-I

Method	Subvolume 1			Subvolume 2			Subvolume 3		
	Accuracy	Type-I	Type-II	Accuracy	Type-I	Type-II	Accuracy	Type-I	Type-II
Method [7]	84.09%	15.68%	0.23%	79.25%	20.71%	0.04%	76.44%	23.55%	0.01%
Method [8]	87.36%	12.44%	0.20%	86.78%	13.12%	0.10%	83.47%	16.53%	0.00%
Method [9, 10]	90.14%	9.07%	0.79%	88.26%	11.67%	0.07%	87.29%	12.61%	0.10%
Method [20]	92.20%	5.38%	2.42%	92.32%	6.81%	0.87%	94.26%	5.19%	0.55%
3D Encoder-Decoder + CycleGAN + BCE ($\mu_1 = 0, \mu_2 = 1, V = 80$)	93.05%	3.09%	3.87%	91.30%	5.64%	3.06%	94.17%	3.96%	1.88%
3D Encoder-Decoder + SpCycleGAN + BCE ($\mu_1 = 0, \mu_2 = 1, V = 80$)	94.78%	3.42%	1.79%	92.45%	6.62%	0.92%	93.57%	6.10%	0.33%
3D U-Net + SpCycleGAN + BCE ($\mu_1 = 0, \mu_2 = 1, V = 80$)	95.07%	2.94%	1.99%	93.01%	6.27%	0.72%	94.04%	5.84%	0.11%
3D U-Net + SpCycleGAN + DICE ($\mu_1 = 1, \mu_2 = 0, V = 80$)	94.76%	3.00%	2.24%	93.03%	6.03%	0.95%	94.30%	5.22%	0.40%
3D U-Net + SpCycleGAN + DICE and BCE ($\mu_1 = 1, \mu_2 = 10, V = 80$)	95.44%	2.79%	1.76%	93.63%	5.73%	0.64%	93.90%	5.92%	0.18%
3D U-Net + SpCycleGAN + DICE and BCE ($\mu_1 = 1, \mu_2 = 10, V = 1600$)	95.37%	2.77%	1.86%	93.63%	5.69%	0.68%	94.37%	5.27%	0.36%
3D U-Net + SpCycleGAN + DICE and BCE + PP ($\mu_1 = 1, \mu_2 = 10, V = 1600$) (Proposed method)	95.56%	2.57%	1.86%	93.67%	5.65%	0.68%	94.54%	5.10%	0.36%

Our proposed method was compared to other 3D segmentation methods including 3D active surface [7], 3D active surface with inhomogeneity correction [8], 3D Squash [9, 10], 3D encoder-decoder architecture [20], 3D encoder-decoder architecture with CycleGAN. Three original 3D subvolumes of Data-I were selected to evaluate the performance of our proposed method. We denote the original volume as subvolume 1 ($I_{(241:272,241:272,31:62)}^{orig}$), subvolume 2 ($I_{(241:272,241:272,131:162)}^{orig}$), and subvolume 3 ($I_{(241:272,241:272,231:262)}^{orig}$), respectively. Corresponding groundtruth of each subvolume was hand segmented. Vox [29] was used to visualize the segmentation results in 3D and compared to the manually annotated volumes. In Figure 5, 3D visualizations of the hand segmented subvolume 1 and the corresponding segmentation results for various methods were presented. As seen from the 3D visualization in Figure 5, our proposed method shows the best performance among presented methods visually compared to hand segmented groundtruth volume. In general, our proposed method captures only nuclei structure whereas other presented methods falsely detect non-nuclei structures as nuclei. Note that segmentation results in Figure 5(g) yields

smaller segmentation mask and suffered from location shift. Our proposed method shown in Figure 5(h) outperforms Figure 5(g) since our proposed method uses spatially constrained CycleGAN and takes consideration of the Dice loss and the binary cross-entropy loss.

All segmentation results were evaluated quantitatively based on voxel accuracy, Type-I error and Type-II error metrics, using 3D hand segmented volumes. Here, accuracy = $\frac{n_{TP} + n_{TN}}{n_{total}}$, Type-I error = $\frac{n_{FP}}{n_{total}}$, Type-II error = $\frac{n_{FN}}{n_{total}}$, where n_{TP} , n_{TN} , n_{FP} , n_{FN} , n_{total} are defined to be the number of true-positives (voxels segmented as nuclei correctly), true-negatives (voxels segmented as background correctly), false-positives (voxels falsely segmented as nuclei), false-negatives (voxels falsely segmented as background), and the total number of voxels in a volume, respectively.

The quantitatively evaluations for the subvolumes are shown in Table 1. Our proposed method outperforms other compared methods. The smaller Type-I error shows our proposed method successfully rejects non-nuclei structures during segmentation. Also, our proposed method has reasonably low Type-II errors compared to other segmentation methods. Moreover, in this table, we show that our pro-

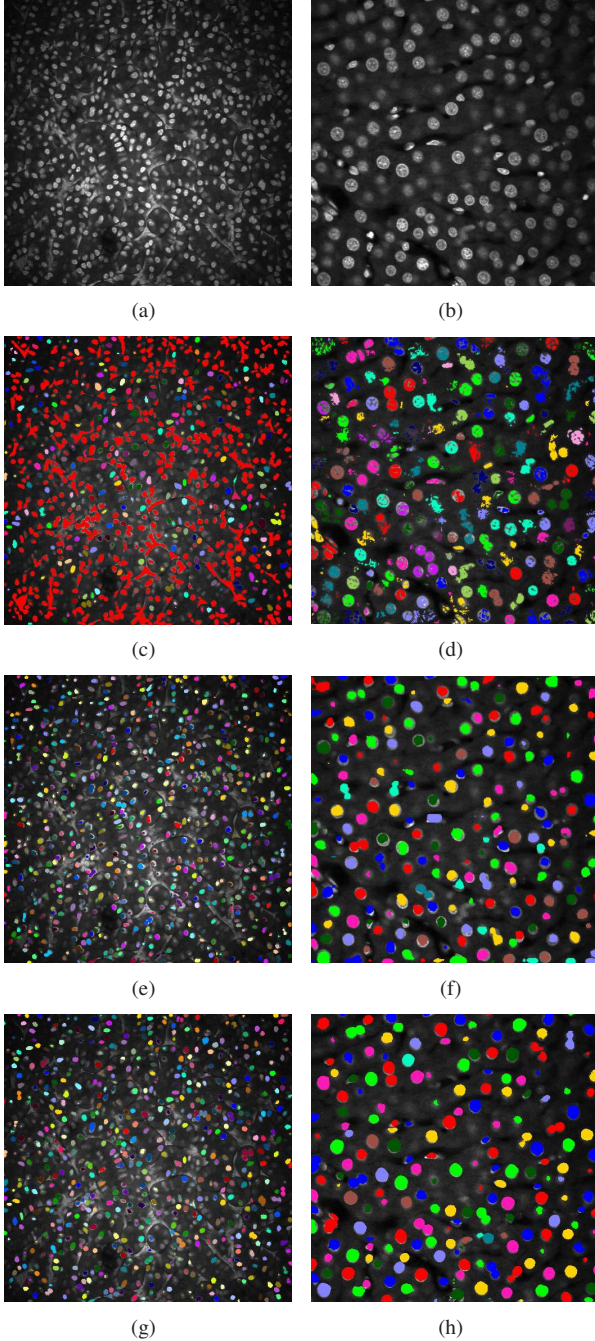


Figure 6. Original images and their color coded segmentation results of Data-I and Data-II (a) Data-I $I_{z_{66}}^{orig}$, (b) Data-II $I_{z_{31}}^{orig}$, (c) Data-I $I_{z_{66}}^{seg}$ using [20], (d) Data-II $I_{z_{31}}^{seg}$ using [20], (e) Data-I $I_{z_{66}}^{seg}$ using 3D encoder-decoder architecture with CycleGAN, (f) Data-II $I_{z_{31}}^{seg}$ using 3D encoder-decoder architecture with CycleGAN, (g) Data-I $I_{z_{66}}^{seg}$ using 3D U-Net architecture with SpCycleGAN (Proposed method), (h) Data-II $I_{z_{31}}^{seg}$ using 3D U-Net architecture with SpCycleGAN (Proposed method)

posed SpCycleGAN creates better paired synthetic volumes which reflects in segmentation accuracy. Instead of 3D encoder-decoder structure, we use 3D U-Net which leads

to better results since 3D U-Net has skip connections that can preserve spatial information. In addition, the combination of two loss functions such as the Dice loss and the BCE loss turns out to be better for the segmentation task in our application. In particular, the Dice loss constrains the shape of the nuclei segmentation whereas the BCE loss regulates voxelwise binary prediction. It is observed that training with more synthetic volumes can generalize our method to achieve better segmentation accuracy. Finally, the post-processing (PP) that eliminates small components helps to improve segmentation performance.

To make this clear, segmentation results were color coded using 3D connected component labeling and overlaid on the original volumes. The method from [20] cannot distinguish between nuclei and non-nuclei structures including noise. This is especially recognizable from segmentation results of Data-I in which multiple nuclei and non-nuclei structures are colored with the same color. As can be observed from Figure 6(e) and 6(f), segmentation masks are smaller than nuclei size and suffered from location shifts. Conversely, our proposed method shown in Figure 6(g) and 6(h) segments nuclei with the right shape at the correct locations.

4. Conclusion

In this paper we presented a modified 3D U-Net nuclei segmentation method using paired synthetic volumes. The training was done using synthetic volumes generated from a spatially constrained CycleGAN. The combination of the Dice loss and the binary cross-entropy loss functions are optimized during training. We compared our proposed method to various segmentation methods and with manually annotated 3D groundtruth from real data. The experimental results indicate that our method can successfully distinguish between non-nuclei and nuclei structure and capture nuclei regions well from various microscopy volumes. One drawback of our proposed segmentation method is that our method cannot separate nuclei if they are physically touching to each other. In the future, we plan to develop nuclei localization method to identify overlapping nuclei to individuals.

5. Acknowledgments

This work was partially supported by a George M. O'Brien Award from the National Institutes of Health under grant NIH/NIDDK P30 DK079312 and the endowment of the Charles William Harrison Distinguished Professorship at Purdue University.

Data-I was provided by Malgorzata Kamocka of Indiana University and was collected at the Indiana Center for Biological Microscopy.

Address all correspondence to Edward J. Delp,

References

- [1] C. Vonesch, F. Aguet, J. Vonesch, and M. Unser, "The colored revolution of bioimaging," *IEEE Signal Processing Magazine*, vol. 23, no. 3, pp. 20–31, May 2006. [1](#)
- [2] K. W. Dunn, R. M. Sandoval, K. J. Kelly, P. C. Dagher, G. A. Tanner, S. J. Atkinson, R. L. Bacallao, and B. A. Mollitoris, "Functional studies of the kidney of living animals using multicolor two-photon microscopy," *American Journal of Physiology-Cell Physiology*, vol. 283, no. 3, pp. C905–C916, September 2002. [1](#)
- [3] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, January 1988. [1](#)
- [4] R. Delgado-Gonzalo, V. Uhlmann, D. Schmitter, and M. Unser, "Snakes on a plane: A perfect snap for bioimage analysis," *IEEE Signal Processing Magazine*, vol. 32, no. 1, pp. 41–48, January 2015. [1](#)
- [5] B. Li and S. T. Acton, "Active contour external force using vector field convolution for image segmentation," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2096–2106, August 2007. [1](#)
- [6] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, February 2001. [1](#)
- [7] K. Lorenz, P. Salama, K. Dunn, and E. Delp, "Three dimensional segmentation of fluorescence microscopy images using active surfaces," *Proceedings of the IEEE International Conference on Image Processing*, pp. 1153–1157, September 2013, Melbourne, Australia. [1](#), [5](#), [6](#)
- [8] S. Lee, P. Salama, K. W. Dunn, and E. J. Delp, "Segmentation of fluorescence microscopy images using three dimensional active contours with inhomogeneity correction," *Proceedings of the IEEE International Symposium on Biomedical Imaging*, pp. 709–713, April 2017, Melbourne, Australia. [1](#), [5](#), [6](#)
- [9] G. Paul, J. Cardinale, and I. F. Sbalzarini, "Coupling image restoration and segmentation: A generalized linear model/Bregman perspective," *International Journal of Computer Vision*, vol. 104, no. 1, pp. 69–93, March 2013. [1](#), [5](#), [6](#)
- [10] A. Rizk, G. Paul, P. Incardona, M. Bugarski, M. Mansouri, A. Niemann, U. Ziegler, P. Berger, and I. F. Sbalzarini, "Segmentation and quantification of subcellular structures in fluorescence microscopy images using Squash," *Nature Protocols*, vol. 9, no. 3, pp. 586–596, February 2014. [1](#), [5](#), [6](#)
- [11] G. Srinivasa, M. C. Fickus, Y. Guo, A. D. Linstead, and J. Kovacevic, "Active mask segmentation of fluorescence microscope images," *IEEE Transactions on Image Processing*, vol. 18, no. 8, pp. 1817–1829, August 2009. [1](#)
- [12] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, June 2015, Boston, MA. [1](#)
- [13] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. van Ginneken, and C. I. Sanchez, "A survey on deep learning in medical image analysis," *arXiv preprint arXiv:1702.05747*, February 2017. [1](#)
- [14] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," *Proceedings of the Medical Image Computing and Computer-Assisted Intervention*, pp. 231–241, October 2015, Munich, Germany. [1](#), [2](#)
- [15] S. E. A. Raza, L. Cheung, D. Epstein, S. Pelengaris, M. Khan, and N. Rajpoot, "MIMO-Net: A multi-input multi-output convolutional neural network for cell segmentation in fluorescence microscopy images," *Proceedings of the IEEE International Symposium on Biomedical Imaging*, pp. 337–340, April 2017, Melbourne, Australia. [1](#)
- [16] A. Prason, K. Petersen, C. Igel, F. Lauze, E. Dam, and M. Nielsen, "Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network," *Proceedings of the Medical Image Computing and Computer-Assisted Intervention*, pp. 246–253, September 2013, Nagoya, Japan. [1](#)
- [17] O. Cicek, A. Abdulkadir, S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: Learning dense volumetric segmentation from sparse annotation," *Proceedings of the Medical Image Computing and Computer-Assisted Intervention*, pp. 424–432, October 2016, Athens, Greece. [2](#)
- [18] X. Zhang, Y. Fu, A. Zang, L. Sigal, and G. Agam, "Learning classifiers from synthetic data using a multichannel autoencoder," *arXiv preprint arXiv:1503.03163*, pp. 1–11, March 2015. [2](#)
- [19] I. B. Barbosa, M. Cristani, B. Caputo, A. Rognhaugen, and T. Theoharis, "Looking beyond appearances: Synthetic training data for deep CNNs in re-identification," *arXiv preprint arXiv:1701.03153*, pp. 1–14, January 2017. [2](#)
- [20] D. J. Ho, C. Fu, P. Salama, K. Dunn, and E. Delp, "Nuclei segmentation of fluorescence microscopy images using three dimensional convolutional neural networks," *Proceedings of the Computer Vision for Microscopy Image Analysis workshop at Computer Vision and Pattern Recognition*, pp. 834–842, July 2017, Honolulu, HI. [2](#), [3](#), [4](#), [5](#), [6](#), [7](#)
- [21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Proceedings of the Advances in Neural Information Processing Systems*, pp. 2672–2680, December 2014, Montreal, Canada. [2](#), [3](#)
- [22] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5967–5976, July 2017, Honolulu, HI. [2](#)
- [23] M. Y. Liu and O. Tuzel, "Coupled generative adversarial networks," *Proceedings of the Advances in Neural Information Processing Systems*, pp. 469–477, December 2016, Barcelona, Spain. [2](#)

- [24] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” *arXiv preprint arXiv:1703.10593*, pp. 1–16, March 2017. [2](#), [3](#), [4](#)
- [25] Y. Huo, Z. Xu, S. Bao, A. Assad, R. G. Abramson, and B. A. Landman, “Adversarial synthesis learning enables segmentation without target modality ground truth,” *arXiv preprint arXiv:1712.07695*, pp. 1–4, December 2017. [2](#)
- [26] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv preprint arXiv:1502.03167*, March 2015. [4](#)
- [27] F. Milletari, N. Navab, and S. A. Ahmadi, “V-Net: Fully convolutional neural networks for volumetric medical image segmentation,” *Proceedings of the IEEE 2016 Fourth International Conference on 3D Vision*, pp. 565–571, October 2016, Stanford, CA. [4](#)
- [28] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, pp. 1–15, December 2014. [4](#)
- [29] J. L. Clendenon, C. L. Phillips, R. M. Sandoval, S. Fang, and K. W. Dunn, “Voxx: A PC-based, near real-time volume rendering system for biological microscopy,” *American Journal of Physiology-Cell Physiology*, vol. 282, no. 1, pp. C213–C218, January 2002. [5](#), [6](#)