

# Neuron Segmentation based on CNN with Semi-supervised Regularization

Kun Xu<sup>†</sup>, Hang Su<sup>†</sup>, Jun Zhu<sup>†</sup>, Ji-Song Guan<sup>‡</sup> and Bo Zhang<sup>†\*</sup>

<sup>†</sup> Department of Computer Science & Technology; State Key Lab of Intelligent Technology and Systems

<sup>†</sup> Tsinghua National Lab for Information Science and Technology

<sup>‡</sup> School of Life Sciences; Tsinghua-Peking Center for Life Sciences; MOE Key Laboratory of Protein Sciences

<sup>†‡</sup> Center for Bio-Inspired Computing Research; Tsinghua University, Beijing, China

{xuk12@mails., suhangss@, dcszj@, jsguan@, dcszb@}tsinghua.edu.cn

## Abstract

*Neuron segmentation in two-photon microscopy images is a critical step to investigate neural network activities in vivo. However, it still remains as a challenging problem due to the image qualities, which largely results from the non-linear imaging mechanism and 3D imaging diffusion. To address these issues, we proposed a novel framework by incorporating the convolutional neural network (CNN) with a semi-supervised regularization term, which reduces the human efforts in labeling without sacrificing the performance. Specifically, we generate a putative label for each unlabeled sample regularized with a graph-smooth term, which are used as if they were true labels. A CNN model is therefore trained in a supervised fashion with labeled and unlabeled data simultaneously, which is used to detect neuron regions in 2D images. Afterwards, neuron segmentation in a 3D volume is conducted by associating the corresponding neuron regions in each image. Experiments on real-world datasets demonstrate that our approach outperforms neuron segmentation based on the graph-based semi-supervised learning, the supervised CNN and variants of the semi-supervised CNN.*

## 1. Introduction

The information coding and decoding in brain involves a large number of neurons. Thus, imaging activities in neuronal populations is essential in understanding the mechanisms of nervous system, since it allows to record activities of a neuronal soma and characterize the dynamic brain properties. Recently, two-photon microscopy emerges as a

powerful and attractive technique to interrogate local neural circuits [22] by using nonlinear light-matter interactions of multiple photons [8]. It enables neuroscientists to measure non-invasive neural network dynamics in live animals with spatial resolution sufficient to discern individual neurons, dendrites, and synapses [12]. Among the tasks of two-photon microscopy image analysis, neuron segmentation is a critical step in that it lays a foundation for the subsequent neuron activity observation. Besides, it is imperative to develop automatic neuron segmentation algorithms with dramatically increasing number of neurons (up to  $10^5$  to  $10^6$  everyday).

Currently, neuron segmentation in two-photon microscopy images remains as a challenging problem because the scattering of the laser and nonlinear imaging mechanism make it difficult to produce sharp images when a large magnification is needed. Besides, neuron segmentation is also challenged by the issue of 3D segmentation in the two-photon microscope volumes because of the diffusion when the lens moves up and down, as a sample image shown in Fig. 1.

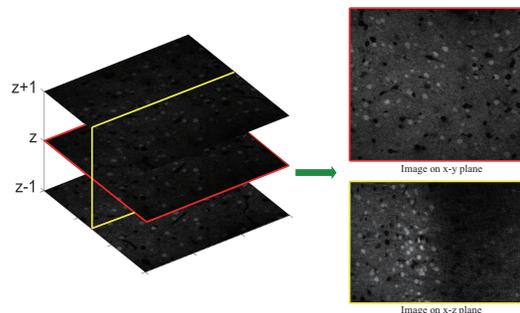


Figure 1: A sample of two-photon microscopy image. The images are obtained in vivo from V1 in 4-6 month old mice's cortex [27]. A mode-locked titanium/sapphire laser generated two-photon excitation at 960 nm whose power is maintained at 1.58W, and three photomultiplier collected emitted light in the range of 380 to 480, 500 to 540, and 560 to 650 nm. Power reaching the mouse brain ranges from 7.8-12.5 mW.

\*This research is partly supported by the National Basic Research Program (973 Program) of China (Nos. 2013CB329403, 2012CB316301, 2013CB835100), National Natural Science Foundation of China (Nos. 61571261, 61322308, 61332007), Brain inspired Computing Research, Tsinghua University (Nos. 20141080934) and China Postdoctoral Science Foundation (No. 2015M580099).

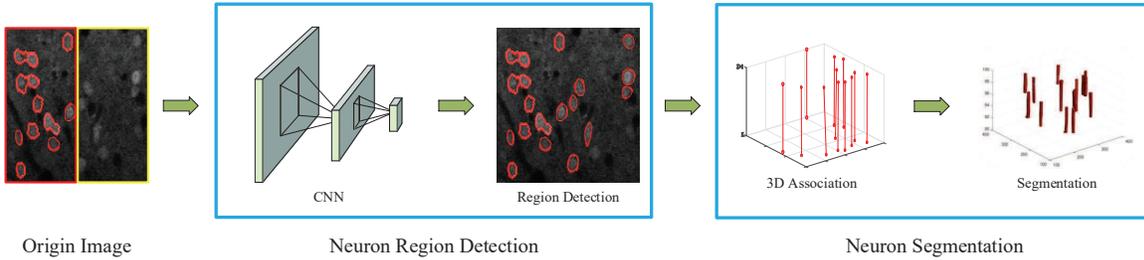


Figure 2: The work-flow of our algorithm, which implements neuron segmentation in 3D volume by associating the neuron regions image-by-image. In particular, we annotate a few neurons in a few images (neurons in the red block), and generate putative labels for the unlabeled ones (neurons in the yellow block). Both the labeled and unlabeled samples in the original images are fed to the semi-supervised CNN, and detect the neuron regions in each 2D image. Afterwards, we associate the corresponding neuron regions image-by-image with a global association method. High quality neuron segmentation is realized by separate the adjacent neurons with a shape prior.

## 1.1. Related Work

There have witnessed a lot of work during the past years in cell segmentation [19] [10]. Active contour [16] [7] and level sets [3] [20] are the most popular algorithms in cell segmentation, but both of them are sensitive to the initialization because of the non-convex object function of the contour energy function. Recently, cell segmentation is implemented with a classifier trained to detect cell boundaries, which is applicable to different modalities and cell types [29]. To address the particular imaging mechanism of phase contrast microscope, Yin et al. [28] and Su et al. [24] studied the optical properties of phase contrast microscope, and developed an imaging models to approximate the image formation process of microscopy image [26]. Cell segmentation in phase contrast microscopy images is then implemented with a semi-supervised algorithm [25], which can be a prior of cell tracking by associating the corresponding cells in time lapsing image sequences [11]. Meijering makes a comprehensive review about evolution of cell segmentation during the past 50 years in [18].

With the rapid development of two-photon microscope, a lots of work has been made in neural cell segmentation, but it is still challenging in allocating the center of neural cells in 3D volume. In [9], cell boundaries are detected by incorporating the prior knowledge within a 2D image, and cell segmentation in a 3D volume is realized by inferring the posterior probabilities of the cell regions. To the best of the authors' knowledge, neuron segmentation in 3D volume is still challenging due to the properties of the images, e.g., neurons under the microscopy often lack distinctive textures and are with low signal-to-noise ratio.

Most of the aforementioned algorithms fall into the area of segmentation with *hand-crafted features*, which transforms the raw data into a suitable internal representation or feature vector. Recently, deep learning has dramatically improved the state-of-the-art in image recognition, speech recognition and other problems [5], since it allows computational models that learn representations of data *automat-*

*ically* with multiple levels of abstraction [15]. In this case, it does not require the considerable amount of engineering skills and domain expertise as the conventional options, and yields high quality results. Ever since the ImageNet competition in 2012 [21], the deep convolutional neural network(CNN) trained with millions of labeled natural images using supervised learning algorithms have achieved spectacular results in the detection, segmentation and recognition of objects and regions in images [13]. The success largely came from the abundant amount of data and efficient use of GPUs.

With the large volumes of data captured from modern biological experiments, it provides good probabilities to improve the performance of neuron segmentation with a deep architecture. However, learning a deep CNN is usually associated with the estimation of large number of parameters, which requires a large number of labeled image samples. *This bottleneck currently prevents the application of CNNs to many biological problems due to the limited amount of labeled training data.*

## 1.2. Our Proposal

In this paper, we address the problem of neuron segmentation in a 3D volume, which is obtained by stacking images corresponding to the vivo scan along the z-axis. In this paper, the *neuron segmentation* in a 3D volume is implemented by associating the corresponding *neuron regions* of the same neurons in successive 2D images. The work-flow is shown in Fig. 2.

Specifically, to address the complexity in nonlinear imaging mechanism of two-photon microscopy images, we apply the convolutional neural network (CNN) to extract the features with multiple level abstraction. With the convolutional layer, the CNN extracts features in the visual field with convolutional kernels optimized by back-propagation [13].

To *reduce the human efforts in annotation*, we propose to incorporate a semi-supervised regularization with the CNN models, such that both the labeled and unlabeled data

are used to train the network simultaneously. Specifically, we generate a *putative label* for each unlabeled sample, which are used as if they were true. In this case, the semi-supervised CNN can be trained as a general CNN. Human efforts in image annotation are reduced by not sacrificing the performance.

Besides, to address *neuron segmentation in 3D volume*, we propose to associate the neuron regions image-by-image with a global association method [30]. Neuron regions association is formulated in terms of finding the global maximum of a convex objective function, then solved efficiently through a linear programming method[2]. Compared with the greedy method which simply associates the closest neurons in contiguous images together [11], it avoids the erroneous associations by connecting one neuron region to more than one neuron regions in neighboring images.

In summary, our proposed algorithm is capable to learn a good representation of the microscopy images with the deep CNN, and reduces human efforts in annotation by using the abundant unlabeled samples in a semi-supervised paradigm. High quality neuron segmentation in a 3D volume is realized by associating the neuron regions in 2D images.

## 2. Neuron Region Detection based on SSL-CNN

In this section, we present how to detect neuron regions in two-photon microscope images with a CNN embedded a semi-supervised regularization term.

In order to reduce the spatial redundancy, we first partition a two-photon microscopy image into numerous over-segmented superpixels [1], which are fed to a deep CNN to learn a compact and meaningful representation for each superpixel. Neuron region detection is implemented by recognizing the identity of each superpixel based on the representation.

We first present the general framework of semi-supervised CNN in section 2.1, and then elaborate on the details of the semi-supervised graph-smooth regularization in section 2.2; finally, the problem is solved in a two-step optimization algorithm in section 2.3.

### 2.1. SSL-CNN

The convolutional neural network is a particular type of feed-forward artificial neural network with convolutional layers, which is inspired by the biological process [17]. Units in a convolutional layer are organized in feature maps, which only receives message from the neighborhood of the nodes in the previous layers rather than all of the nodes. The structure enables CNN to extract local features from the images. Besides, the parameters of each node in the same convolutional layer are shared to reduce the number of parameters in the convolutional neural network, since the local

statistics of images are often highly correlated. Parameters in the deep convolutional neural network can be trained with the back propagation algorithm [13].

Mathematically, we define the set of annotated superpixels as  $\mathcal{L} \triangleq \{(\mathbf{x}_l, \mathbf{y}_l)\}_{l=1}^{N_l}$  with  $N_l$  being the number of annotated superpixels;  $\mathbf{x}_l$  is the image of the  $l$ th sample which is fed to the CNN; the indicator  $\mathbf{y}_l \in \{\mathcal{C}_1, \dots, \mathcal{C}_L\}$  corresponds to the identity of the  $l$ th sample. The set of unlabeled superpixels is defined as  $\mathcal{U} \triangleq \{\mathbf{x}_u\}_{u=1}^{N_u}$  with  $N_u$  being the number of unlabeled superpixels; and  $\mathbf{x}_u$  is the corresponding image of an unlabeled superpixel. Thereby, the detection of neuron regions can be conducted by statistically inferring the identities of the unlabeled superpixels.

Softmax loss is one of the most commonly used loss function for CNN as

$$L(\mathbf{X}_l, \mathbf{Y}_l) = -\frac{1}{N_l} \sum_{l=1}^{N_l} \left( \sum_{c=1}^L \mathbb{I}(\mathbf{y}_l = \mathcal{C}_j) \log(f_c(\mathbf{x}_l)) \right), \quad (1)$$

where  $\mathbf{X}_l = \{\mathbf{x}_l\}_{l=1}^{N_l}$  denotes the set of the labeled superpixels and  $\mathbf{Y}_l$  denotes labels of them.  $\mathbb{I}(\cdot)$  is an indicator function and  $f_c(\mathbf{x}_l)$  is the output of the softmax layer in CNN which denotes the probability of  $\mathbf{x}_l$  belongs to the  $c$ -th category. In order to take the unlabeled samples into consideration, we introduce a regularization term to use the information of unlabeled data as

$$\tilde{L}(\mathbf{X}_l, \mathbf{Y}_l, \mathbf{X}_u) = L(\mathbf{X}_l, \mathbf{Y}_l) + \gamma R(\mathbf{X}_l, \mathbf{Y}_l, \mathbf{X}_u). \quad (2)$$

The  $R(\mathbf{X}_l, \mathbf{Y}_l, \mathbf{X}_u)$  is the regularization term, in which  $\mathbf{X}_u$  denotes the set of unlabeled samples  $\mathcal{U}$  and  $\gamma$  is the balanced parameter. The structure of SSL-CNN is illustrated in Fig. 3, which takes both the labeled data and the unlabeled data as the input. We collect the positive samples (neuron regions) with the image patches where the corresponding superpixel lies in the center of it; the negative samples are acquired by simply drawing patches from the background or introducing center offset for the positive samples. Note that we ignore the pooling layer in our CNN because of the low-resolution of a specific neuron (usually no more than 150 pixels).

### 2.2. Semi-supervised Graph-smooth Regularization

In general, the regularization term encourages the instances with similar features to share the same label. In this paper, we use the graph smoothness [4] as the semi-supervised regularization on both the labeled and unlabeled samples, which is appealing due to its close form optimal solution. In this case, information carried by the unlabeled samples is incorporated and offers advantages to identify the corresponding labels.

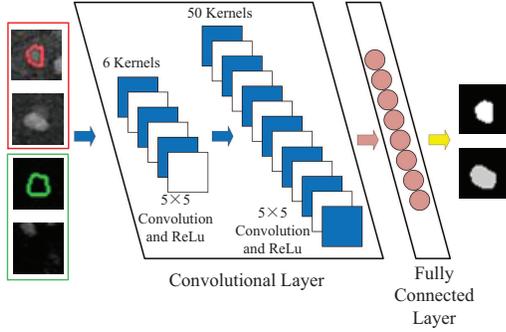


Figure 3: The structure of CNN for neuron region detection. The samples in red and green blocks are corresponding to the positive (neuron regions) and negative (background regions) samples, respectively. The blue arrows stand for the convolution operation, the pink arrow stands for the inner product and the yellow arrow stands for the softmax operation. We train the CNN network with both the labeled (with markers) and unlabeled (without markers) samples simultaneously.

In this paper, we construct an  $\epsilon$ -nearest neighboring graph to characterize the pairwise similarity between samples, with its edge being the RBF kernel function as

$$w_{ij} = \begin{cases} \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2), & \|\mathbf{x}_i - \mathbf{x}_j\|^2 < \epsilon \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

By collecting all  $w_{ij}$  together, the adjacency matrix of the weight graph is denoted as  $\mathbf{W} = [w_{ij}]$ , which is an  $N \times N$  nonnegative and symmetric matrix with  $N = N_l + N_u$  being the number of total superpixels. Afterwards, we can calculate the Laplacian matrix  $\mathbf{L}$  that corresponds to  $\mathbf{W}$ , i.e.,  $\mathbf{L} = \mathbf{D} - \mathbf{W}$  with  $\mathbf{D}$  being the diagonal degree matrix.

Hereby, the regularization term in Eq. (2) can be formulated as

$$\begin{aligned} R(\mathbf{X}_l, \mathbf{Y}_l, \mathbf{X}_u) &= \sum_{i,j=1}^N w_{ij} \|\mathbf{y}_i - \mathbf{y}_j\|_2^2 \\ &= \text{tr}(\mathbf{f}^T \mathbf{L} \mathbf{f}), \end{aligned} \quad (4)$$

where  $\mathbf{f} = [\mathbf{y}_i]^N$  with  $\mathbf{y}_i$  being the labels of samples; and  $\mathbf{y}_i = [f_1(\mathbf{x}_i), \dots, f_L(\mathbf{x}_i)]$  if  $\mathbf{x}_i$  is corresponding to the unlabeled sample.

By substituting the regularization term in Eq. (4) as the regularization term to Eq. (2), the softmax loss of SSL-CNN is

$$\tilde{L}(\mathbf{X}_l, \mathbf{Y}_l, \mathbf{X}_u) = L(\mathbf{X}_l, \mathbf{Y}_l) + \gamma \text{tr}(\mathbf{f}^T \mathbf{L} \mathbf{f}), \quad (5)$$

where  $f$  maps the input to the label space, which is related to the CNN parameters;  $L(\mathbf{X}_l, \mathbf{Y}_l)$  is the standard loss function in Eq. (1).

### 2.3. Optimization Algorithm

Obviously, the loss function of SSL-CNN in Eq. (5) achieves its minimum when both of its terms, i.e., the stan-

dard loss of CNN and the regularization term, are the minimized simultaneously. The minimum of the first term is achieved when the CNN completely fits the labeled data where  $f(\mathbf{x}_i) = \mathbf{y}_i$  for each  $\mathbf{x}_i \in \mathbf{X}_l$ . The second term is minimized when the labeling is consistent with the geometry of the data induced by the graph structure, i.e., if two samples are close in the feature space, then the corresponding labels should also be consistent.

In this case, we propose a two-step optimization algorithm to solve the problem, i.e., we obtain putative labels for the samples in  $\mathcal{U}$  by minimizing the graph-smooth regularization, and then solve the SSL-CNN problem in a supervised fashion as if the putative labels are true labels.

In order to minimize the regularization term, we propose to propagate the human annotation to the unlabeled samples via label propagation. To this end, we rearrange the Laplacian matrix in (4) by splitting it into labeled and unlabeled sub-matrices as

$$\mathbf{L} \leftarrow \begin{bmatrix} \mathbf{L}_{ll} & \mathbf{L}_{lu} \\ \mathbf{L}_{ul} & \mathbf{L}_{uu} \end{bmatrix}, \quad (6)$$

where  $\mathbf{L}_{ll}$  is the Laplacian sub-matrix that characterizes the relationship between labeled superpixels;  $\mathbf{L}_{uu}$  denotes the sub-matrix corresponding to the unlabeled superpixels;  $\mathbf{L}_{lu}$  is a sub-matrix which interrelates the labeled and unlabeled superpixels; and  $\mathbf{L}_{ul} = \mathbf{L}_{lu}^T$ .

Hence, the minimum of the graph-smooth regularization is achieved by

$$\min \text{tr}([\mathbf{Y}_l^T, \mathbf{Y}_u^T] \begin{bmatrix} \mathbf{L}_{ll} & \mathbf{L}_{lu} \\ \mathbf{L}_{ul} & \mathbf{L}_{uu} \end{bmatrix} \begin{bmatrix} \mathbf{Y}_l \\ \mathbf{Y}_u \end{bmatrix}), \quad (7)$$

where  $\mathbf{Y}_l$  and  $\mathbf{Y}_u$  are indicator matrices corresponding to the labeled and unlabeled atoms, respectively. Specifically, the indicator matrices  $\mathbf{Y}_l$  and  $\mathbf{Y}_u$  are constructed by stacking up the binary indicators  $\{\mathbf{y}_n\}_{n=1}^N$  in rows correspondingly. The problem can be simplified to

$$\mathbf{Y}_u^* = \arg \min_{\mathbf{Y}_u} \text{tr}(\mathbf{Y}_u^T \mathbf{L}_{uu} \mathbf{Y}_u) + 2\text{tr}(\mathbf{Y}_u^T \mathbf{L}_{ul} \mathbf{Y}_l). \quad (8)$$

Based on the work in [31], label propagation can be solved by Gaussian fields harmonic Functions (GFHF) as

$$\mathbf{Y}_u^* = -\mathbf{L}_{uu}^{-1} \mathbf{L}_{ul} \mathbf{Y}_l, \quad (9)$$

where  $-\mathbf{L}_{uu}^{-1} \mathbf{L}_{ul}$  is regarded as a label propagation matrix on the graph; and the regularization term is minimized when the labels of the samples in  $\mathcal{U}$  are equal to  $\mathbf{Y}_u^*$ , which serves as *putative labels* in this paper.

Ideally, the regularization term would be minimized if the output of CNN fits the putative labels  $\mathbf{Y}_u^*$ , and therefore we reformulate the regularization minimization problem as

$$\min \|f(\mathbf{X}_u) - \mathbf{Y}_u^*\|, \text{ with } \mathbf{Y}_u^* = -\mathbf{L}_{uu}^{-1} \mathbf{L}_{ul} \mathbf{Y}_l. \quad (10)$$

Hence, the optimization problem of then loss function in Eq. (5) can be reformulated as

$$\min_f -\frac{1}{N_l} \sum_{l=1}^{N_l} \sum_{j=1}^L \mathbb{I}(y_l = C_j) \log(f_j(\mathbf{x}_l)) + \lambda \|f(\mathbf{X}_u) - \mathbf{Y}_u^*\|. \quad (11)$$

In this paper, we treat the putative labels *as if they were true labels*. A CNN model is therefore trained in a supervised fashion with labeled and unlabeled data simultaneously by rewritten the objective loss in Eq. (5) with a softmax objective as

$$\min_f -\frac{1}{N_l} \sum_{l=1}^{N_l} \sum_{j=1}^L \mathbb{I}(\mathbf{y}_l = C_j) \log(f_j(\mathbf{x}_l)) - \frac{\lambda}{N_u} \sum_{l=1}^{N_u} \sum_{j=1}^L \mathbb{I}(\mathbf{y}_u^* = C_j) \log(f_j(\mathbf{x}_u)). \quad (12)$$

In this case, the problem can be solved with the back propagation method in a supervised fashion. The two-step optimization algorithm is described in Algorithm. 1.

---

**Algorithm 1** Two-Step Optimization Algorithm

---

**Require:** Data points  $\{\mathbf{x}_i\}_{i=1}^N$ , Label  $\mathbf{Y}_l$ , Mini-batch size  $n$

Calculate the Laplacian matrix  $\mathbf{L}$  as

$$\mathbf{L} = \mathbf{D} - \mathbf{W},$$

with  $\mathbf{W}$  and  $\mathbf{D}$  being the adjacency matrix and the diagonal degree matrix, respectively.

Calculate the putative labels  $\mathbf{Y}_u^*$  using Eq. (9) as

$$\mathbf{Y}_u^* = -\mathbf{L}_{uu}^{-1} \mathbf{L}_{ul} \mathbf{Y}_l.$$

Reformulate the loss function as

$$\min_f -\frac{1}{N_l} \sum_{l=1}^{N_l} \sum_{j=1}^L \mathbb{I}(\mathbf{y}_l = C_j) \log(f_j(\mathbf{x}_l)) - \frac{\lambda}{N_u} \sum_{l=1}^{N_u} \sum_{j=1}^L \mathbb{I}(\mathbf{y}_u^* = C_j) \log(f_j(\mathbf{x}_u)).$$

**repeat**

Sample a mini-batch from  $\tilde{\mathbf{X}} = \{\mathbf{x}_i\}^n$

Feed forward the CNN and calculate the loss based on  $\tilde{\mathbf{Y}}_l$  for labeled data and  $\tilde{\mathbf{Y}}_u^*$  for unlabeled data in the mini-batch

Back propagate the loss and update the weights of the CNN

**until** The weights of CNN converge

---

### 3. Neuron Segmentation in 3D Volume

In this section, we implement neuron segmentation in 3D volume by associating the neuron region image-by-image. To address this problem, we propose a global association method, inspired by Zhang et al. [30], by modeling it as an assignment problem.

Specifically, the global association can be modeled as

$$\begin{aligned} \min_{a_{i,j}} & \sum_{i=1}^{n_z} \sum_{j=1}^{n_{z+1}} a_{i,j} d(i, j) \\ \text{s.t.} & \sum_{i=1}^{n_z} a_{i,j} \leq 1, \sum_{j=1}^{n_{z+1}} a_{i,j} \leq 1 \\ & \sum_{i=1}^{n_z} \sum_{j=1}^{n_{z+1}} a_{i,j} = \min(n_z, n_{z+1}), \\ & a_{i,j} \in \{0, 1\}, \forall i, j \end{aligned} \quad (13)$$

where  $n_z$  and  $n_{z+1}$  denote the number of neurons in the  $z$ -th image and  $(z+1)$ -th image, respectively; and  $a_{i,j}$  is a binary variable of which 1 stands for the  $i$ -th neuron on  $z$ -th image are associated with the  $j$ -th neuron on  $(z+1)$ -th image and vice versa. The  $d(i, j)$  denotes the distance between  $i$ -th neuron in  $z$ -th image and the  $j$ -th neuron in  $(z+1)$ -th image, which is defined as

$$d(i, j) = \begin{cases} \|C_{i,z} - C_{j,z+1}\|_2^2 & \text{Cell } i, j \text{ are overlapped} \\ \infty & \text{no overlap} \end{cases}$$

where  $C_{i,z}$  stands for the centroid of the  $i$ -th neuron in the  $z$ -th image. In this paper, we assume that there must be overlaps if two neuron regions in successive images belong to the same neuron when projecting them on the  $x$ - $y$  plane, which is reasonable in inspecting the brain images. Neuron segmentation is conducted by solving problem in Eq. (13).

In Eq. (13), the first two constraints ensure that each region cannot associate to more than one regions in the next image to make the result consistent. The third constrains prevent the problem from degradation where  $a_{i,j} = 0$  for all  $i$  and  $j$ . The Kuhn-Munkres algorithm [14] is applied to solve the optimization which swaps the association to minimize the sum of distance in each iteration. The algorithm converges within  $n$  iterations with time complexity  $O(\min(n_z, n_{z+1})^4)$ . Thus, the neuron  $i$  and neuron  $j$  are associated if  $a_{i,j} = 1$  and  $d(i, j) \neq \infty$  by solving the problem in Eq. (13).

We show a sample result of association between a few neuron regions in two successive images in Fig. 4. It demonstrates that the corresponding neuron regions that belong to the same neurons are associated correctly, which provides a good prior for neuron segmentation.

However, it is a common case that two neurons are adjacent in  $z$ -axis, which leads to the fact that two neurons are associated together as one. In this paper, we separate the adjacent neurons by exploring the morphology of neurons. According to the visual inspection, neurons always exhibit as an ellipsoid in 3D volume, i.e., the sectional areas are larger in the middle of the neurons while small on

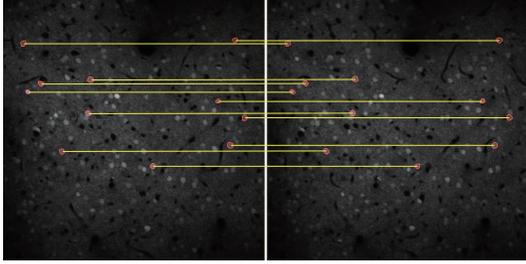


Figure 4: The result of image-by-image association.

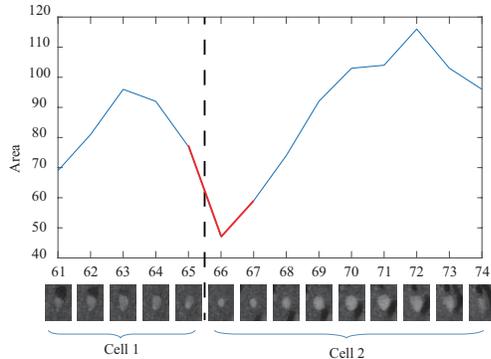


Figure 5: Separate adjacent neurons in 3D volume.

both ends. It inspires us to separate neurons by examining the sectional areas of neuron regions along the z-axis. In Fig. 5, We demonstrate the adjacent neuron separation, in which two adjacent neurons are separated at the minimum of the sectional areas. Specifically, we assign the neuron at the local-minimum to the side with a smaller gradient, as is shown in Fig. 5.

## 4. Experiment

In this section, we report the performance of neuron region detection in 2D images and neuron segmentation in 3D volume in section 4.2 and section 4.3, respectively. As a comparison, we also implement neuron region detection based on the graph-smooth SSL and supervised CNN and conduct the neuron segmentation by associating the corresponding regions with the greedy association algorithm [11].

### 4.1. Data set

We valid our approach with the dataset of neural morphology images generated by two-photon microscope, as is shown Fig. 1. For evaluation, we manually annotated 2000 neuron regions in the images, with half of them used for training and the remaining used for testing. We further collect the the negative samples, i.e., the non-neuron regions, by randomly sampling from the original images, since most of the microscopy image regions are non-neuron. To keep

balance, the number of negative samples is approximately triple to the positive ones. Moreover, we use the unlabeled data as auxiliary samples when training the SSL-CNN network, which includes 70,000 samples with approximately 4000 positive ones.

### 4.2. Result of Neuron Region Detection

In this section, we implement neuron region detection with the SSL-CNN and then evaluate it qualitatively. In particular, we design a SSL-CNN network, which has two convolutional layers with the first and second layer being 6 kernels and 50 kernels, respectively. The output of the second convolutional layer is connected to fully connected layers to identify the superpixels, as is shown Fig. 3. As comparisons, we also implement neuron region detection based on the graph-based semi-supervised learning [24] and the supervised CNN [13], respectively. The sample results of neuron region detection are demonstrated in Fig. 6.

As the results demonstrated in column (d) and column (e) of Fig. 6, neuron regions are detected with high qualities based on our semi-supervised CNN, since the inherent structures are well explored with our network and facilitate the subsequent neuron region detection. However, there missed some neuron regions for the graph-supervised learning algorithm, as is shown in column (b) of Fig. 6. The main reason is because neuron detection is implemented with the microscopy image intensities, which degenerates the performance especially with poor image qualities or low contrast between neuron regions and background. Our algorithm also outperforms the supervised CNN when there are not enough labeled samples for network training, which is illustrated in column (c) of Fig. 6. The main reason is that our SSL-CNN can leverage the information from the unlabeled samples.

In Table 1, we report the qualitative evaluation of our proposed algorithm and the alternative graph-based semi-supervised learning and supervised CNN. We evaluate the performance with different label size in terms of precision, recall and  $F_1$ -score, i.e.,

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}},$$

which is a widely used measurement for classifiers. Each number is averaged over 5 trails. We control the experimental condition to preserve a high precision to preserve a high accuracy of the putative labels and reduce the probability of error propagation over the network.

From Table 1, the performance of SSL-CNN converges to the optimal solution much faster than the supervised CNN. Especially for the recall, the SSL-CNN converges to its optimal with 200 labeled samples whereas the supervised CNN needs 600 labeled samples. Moreover, the F-score of SSL-CNN also coverages earlier compared to the

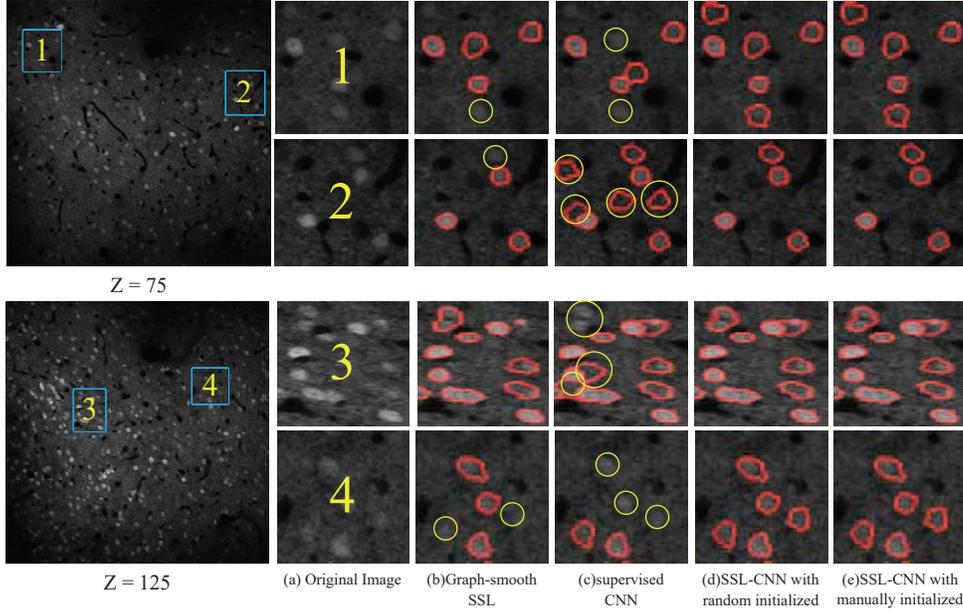


Figure 6: Sample result of neuron region detection of the 75-th image(Up) and 125-th image(Bottom) of the image set based on different algorithms. The error detections are marked in yellow circles. (a)Origin image. (b) graph-smooth SSL (c) supervised CNN (d) randomly initialized SSL-CNN (e) Manually initialized SSL-CNN.

Data Size		40	80	120	160	200	300	400	500	600	700	800	900	1000
Recall	SSL	0.70	0.79	0.80	0.86	0.86	0.87	0.88	0.91	0.90	0.91	0.91	0.92	0.92
	CNN	0.88	0.91	0.89	0.94	0.92	0.93	0.93	0.94	0.94	0.95	0.95	0.95	0.95
	SSL-CNN	0.92	0.94	0.94	0.95	0.95	0.95	0.96	0.96	0.96	0.97	0.97	0.97	0.96
Precision	SSL	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.99	0.99	0.99	0.99	0.99
	CNN	0.99	0.98	0.98	0.98	0.98	0.99	0.99	0.99	0.99	0.99	0.99	1.00	0.99
	SSL-CNN	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.98	0.98	0.98	0.99
F1-score	SSL	0.82	0.88	0.89	0.93	0.92	0.93	0.93	0.95	0.95	0.95	0.95	0.95	0.96
	CNN	0.93	0.94	0.94	0.96	0.95	0.96	0.96	0.97	0.97	0.97	0.97	0.97	0.97
	SSL-CNN	0.96	0.96	0.97	0.97	0.97	0.97	0.97	0.98	0.97	0.98	0.97	0.97	0.97

Table 1: The performance of each classifier with different size of data.

graph-based SSL and supervised CNN. It means that human efforts in annotation is reduced significantly, namely to no more than 30% compared to the supervised CNN.

#### 4.2.1 Discussion

After inspecting the visual pattern of the two two-photon images, the neurons in the image plane exhibit point-spread patterns due to its particular imaging mechanism [6]. It inspires us to design a convolution kernel by taking the particular properties into consideration, rather than simply initializing the kernels randomly. Specifically, the convolutional

kernels are initialized as a point-spread function [23] as

$$\text{psf}(v_x, v_y) = \left[ \frac{2J_1(\sqrt{v_x^2 + v_y^2})}{\sqrt{v_x^2 + v_y^2}} \right]^6,$$

where  $\text{psf}(v_x, v_y)$  denotes the point spread function at  $(v_x, v_y)$  and  $J_1(\cdot)$  denotes the first-order Bessel function of the first kind. A sample result of kernels generated by the point spread function is illustrated in Fig. 7.

With the kernels generated by point spread function to initialize the CNN, the loss value averaged over 100 trails is illustrated in Fig. 8. The loss value of the manually initialized CNN converges faster than the randomly initialized CNN, where manually initialized CNN converges at after about 30 iterations whereas randomly initialized CNN con-

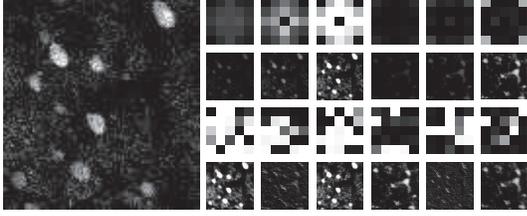


Figure 7: The result of Kernel initialization. The left image is the origin microscopy image. The first row and the second row is corresponding to the manually designed kernel and its convolution on the origin image, respectively. The third row is the kernel generated by randomly initialized CNN and the forth row is corresponding to its convolution on the original image.

verges after 50 iterations. Meanwhile, the loss value of manually initialized CNN is almost the same as the randomly initialized CNN when they both converge and from the column (e) in Fig. 6, the accuracy is also comparable. It demonstrates that the manually designed kernels can accelerate the convergence speed by taking into the imaging mechanism of the two-photon microscope.

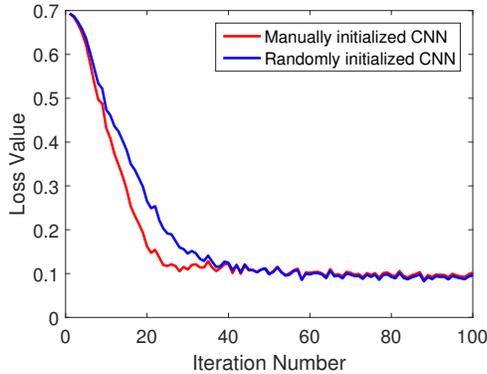


Figure 8: Average loss. Red line for manually initialized CNN and blue for randomly initialized CNN

### 4.3. 3D Association

Finally, the quantitative evaluation of neuron region detection will be illustrated and an example of neuron region detection is demonstrated in Fig. 9. We only sample a subset from the whole set to make the figure uncluttered.

The performance of neuron segmentation is illustrated in Tab. 2. We use 200 positive samples to train the CNN and generate putative labels. From Tab. 2, SSL-CNN outperforms CNN, since it is benefit from the vast amount of unlabeled samples. It is noted that the performance of neuron segmentation in volume is inferior to the neuron region detection due to the 3D diffusion in the region association where false associating two neuron into one neuron will miss detect two neurons.

Besides, the result of neuron segmentation based on the

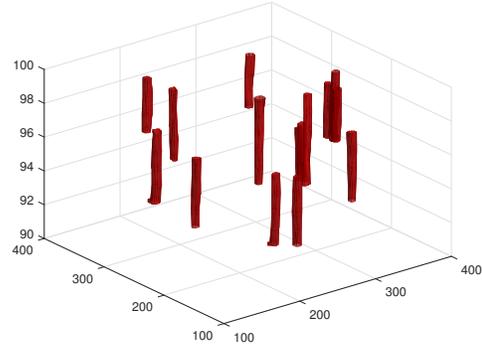


Figure 9: Final Result of neuron segmentation in 3D Visualization

	Precision	Recall	F-score
SSL	0.99	0.62	0.76
CNN	0.75	0.82	0.81
SSL-CNN	0.90	0.83	0.86

Table 2: Performance of neuron segmentation with 200 positive samples

greedy association is also reported in Tab. 3. The recall rate is significantly inferior to the algorithm based on global association because of the inconsistent association. The association result of greedy association is the subset of the global association because if two neuron regions both associate to one neuron region, the association with a larger distance will be dropped to make the result consistent, which leads to the low recall rates.

	Precision	Recall	F-score
CNN	0.75	0.72	0.73
SSL-CNN	0.88	0.72	0.79

Table 3: Neuron segmentation on greedy association.

## 5. Conclusion

In this paper we propose a CNN with a semi-supervised regularization to address the neuron segmentation in 3D volume. With unlimited unlabeled data, we introduce a regularization term to the loss function of the CNN, such that the performance is improved by incorporating the information embedded in the unlabeled data. Human efforts in annotation are therefore reduced significantly by not sacrificing the performance. With neuron region properly harnessed in 2D image, neuron segmentation in 3D volume is achieved by optimizing a global association problem. Experimental results demonstrate that the high quality neuron segmentation is realized with our algorithm.

## References

- [1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11):2274–2282, 2012.
- [2] O. Al-Kofahi, R. J. Radke, S. K. Goderie, Q. Shen, S. Temple, and B. Roysam. Automated cell lineage construction: a rapid method to analyze clonal development established with murine neural progenitor cells. *Cell cycle*, 5(3):327–335, 2006.
- [3] M. Ambühl, C. Brepsant, J.-J. MEISTER, A. Verkhovsky, and I. Sbalzarini. High-resolution cell outline segmentation and tracking from phase-contrast microscopy images. *Journal of microscopy*, 245(2):161–170, 2012.
- [4] M. Belkin, P. Niyogi, and V. Sindhwani. On manifold regularization. In *AISTATS*, 2005.
- [5] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1798–1828, 2013.
- [6] W. Denk, J. H. Strickler, and W. W. Webb. Two-photon laser scanning fluorescence microscopy. *Science*, 248(4951):73–76, 1990.
- [7] H. Grimm, A. Verkhovsky, A. Mogilner, and J.-J. Meister. Analysis of actin dynamics at the leading edge of crawling cells: implications for the shape of keratocyte lamellipodia. *European Biophysics Journal*, 32(6):563–577, 2003.
- [8] F. Helmchen and W. Denk. Deep tissue two-photon microscopy. *Nature methods*, 2(12):932–940, 2005.
- [9] H. Hu, G. Chen, Y. Liu, and P. Wang. Center location and segmentation of cell two-photon microscopic images based on random sample consensus. In *Image and Signal Processing (CISP), 2013 6th International Congress on*, volume 2, pages 588–592. IEEE, 2013.
- [10] N. N. Kachouie, P. Fieguth, and E. Jervis. Watershed deconvolution for cell segmentation. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pages 375–378. IEEE, 2008.
- [11] T. Kanade, Z. Yin, R. Bise, S. Huh, S. Eom, M. F. Sandbothe, and M. Chen. Cell image analysis: Algorithms, system and applications. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pages 374–381. IEEE, 2011.
- [12] J. N. Kerr and W. Denk. Imaging in vivo: watching the brain in action. *Nature Reviews Neuroscience*, 9(3):195–205, 2008.
- [13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012.
- [14] H. W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.
- [15] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
- [16] F. Li, X. Zhou, H. Zhao, and S. T. Wong. Cell segmentation using front vector flow guided active contours. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2009*, pages 609–616. Springer, 2009.
- [17] M. Matsugu, K. Mori, Y. Mitari, and Y. Kaneda. Subject independent facial expression recognition with robust face detection using a convolutional neural network. *Neural Networks*, 16(5):555–559, 2003.
- [18] E. Meijering. Cell segmentation: 50 years down the road [life sciences]. *Signal Processing Magazine, IEEE*, 29(5):140–145, 2012.
- [19] N. Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.
- [20] D. Padfield, J. Rittscher, N. Thomas, and B. Roysam. Spatio-temporal cell cycle phase analysis using level sets and fast marching methods. *Medical image analysis*, 13(1):143–155, 2009.
- [21] O. Russakovsky, J. Deng, H. Su, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.
- [22] P. T. So, C. Y. Dong, B. R. Masters, and K. M. Berland. Two-photon excitation fluorescence microscopy. *Annual review of biomedical engineering*, 2(1):399–429, 2000.
- [23] E. H. K. S. Stefan W Hell. Fundamental improvement of resolution with a 4pi-confocal fluorescence microscope using two-photon excitation. *Optics Communications*, 93(5-6):277–282, 1992.
- [24] H. Su, Z. Yin, S. Huh, and T. Kanade. Cell segmentation in phase contrast microscopy images via semi-supervised classification over optics-related features. *Medical image analysis*, 17(7):746–765, 2013.
- [25] H. Su, Z. Yin, S. Huh, T. Kanade, and J. Zhu. Interactive cell segmentation based on active and semi-supervised learning. *IEEE transactions on medical imaging*, 2015.
- [26] H. Su, Z. Yin, T. Kanade, and S. Huh. Phase contrast image restoration via dictionary representation of diffraction patterns. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2012*, pages 615–622. Springer, 2012.
- [27] H. Xie, Y. Liu, Y. Zhu, X. Ding, Y. Yang, and J.-S. Guan. In vivo imaging of immediate early gene expression reveals layer-specific memory traces in the mammalian brain. *Proceedings of the National Academy of Sciences*, 111(7):2788–2793, 2014.
- [28] Z. Yin, T. Kanade, and M. Chen. Understanding the phase contrast optics to restore artifact-free microscopy images for segmentation. *Medical image analysis*, 16(5):1047–1062, 2012.
- [29] C. Zhang, J. Yarkony, and F. A. Hamprecht. Cell detection and segmentation using correlation clustering. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2014*, pages 9–16. Springer, 2014.
- [30] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 1–8, 2008.
- [31] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Twentieth International Conference on Machine Learning (ICML)*, pages 912–919, 2003.