# Large Kernel Refine Fusion Net for Neuron Membrane Segmentation

Dongnan Liu[1] Donghao Zhang[1] Yang Song[1] Chaoyi Zhang[1] Heng Huang[2]
Mei Chen[3,4] Weidong Cai[1]

[1]School of Information Technologies, University of Sydney
[2]Department of Computer Science and Engineering, University of Pittsburgh
[3]Department of Electrical and Computer Engineering, State University of New York at Albany
[4]Robotics Institute, Carnegie Mellon University

## Abstract

*2D neuron membrane segmentation for Electron Microscopy (EM) images is a key step in the 3D neuron reconstruction task. Compared with the semantic segmentation tasks for general images, the boundary segmentation in EM images is more challenging. In EM segmentation tasks, we need not only to segment the ambiguous membrane boundaries from bubble-like noise in the images, but also to remove shadow-like intracellular structure. In order to address these problems, we propose a Large Kernel Refine Fusion Net, an encoder-decoder architecture with fusion of features at multiple resolution levels. We incorporate large convolutional blocks to ensure the valid receptive fields for the feature maps are large enough, which can reduce information loss. Our model can also process the background together with the membrane boundary by using residual cascade pooling blocks. In addition, the post-processing method in our work is simple but effective for a final refinement of the output probability map. Our method was evaluated and achieved competitive performances on two EM membrane segmentation tasks: ISBI2012 EM segmentation challenge and mouse piriform cortex segmentation task.*

## 1. Introduction

Human brains contain numerous interconnected neurons, which can be grouped into brain compartments [4]. The connectivity of the neurons thus affects the function of the whole tissue. However, among all the human organs, the structure-function relationship of the nerve system in the brain is relatively complicated [18]. The reason is that the axons and dendrites inside the neuron make it a high density neuropil [13]. In order to have further understanding of the relationship between the neuronal function and the connective structure, the neuronal circuit reconstruction for human brain becomes a necessary task in bioinformatics [29].

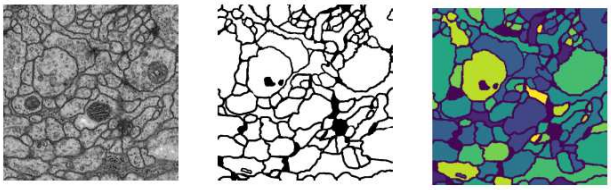

Figure 1. The original EM image (left), the binary boundary detection ground truth, with a value 0 for the membrane and 1 for the background (middle) and the neuron segmentation result (right)

The serial section transmission electron microscopy (ssTEM) is currently a widely used tool for imaging the neuron structure at a high resolution. In ssTEM, the original 3D volume is cut as a stack of consecutive 2D slices. Fig. 1 shows a slice of 2D image from ssTEM. It can be seen that the ssTEM images contain more detailed structure at high resolution such as some details inside a single cell. On the other hand, due to the slicing operation, the image resolution of the $z$ dimension is much lower than that of the $x$ and $y$ directions.

Due to the anisotropic characteristic of the data, the pipeline of the neuronal reconstruction task of ssTEM is as follows: (1) membrane segmentation for a probability map of each 2D slice, (2) neuron region segmentation based on the membrane probability map, (3) fusing the neuron segmentation result among multiple 2D slices into a 3D segmentation result, and (4) manually proofreading [14]. In this paper, we focus on the membrane segmentation task, also known as boundary detection, which is the first step in the reconstruction pipeline.

In this paper, we propose a Large Kernel Refine Fusion Net to perform the membrane segmentation task. When designing this network, we follow several principles: 1) The

segmentation task can be regarded as a per-pixel dense classification problem in the image. In this view, large kernels in the decoders are able to make dense connections between the feature maps from encoders and the following classifiers. Additionally, the large size of kernels renders large valid receptive fields for the feature maps. 2) Feature maps at different resolutions reveal different characteristics of the original image. Low-level resolution features encode the detailed features such as edges and circles while high-level resolution features represent some general features including category-level information. Our method fuses feature maps at different resolutions together to ensure all the information can be considered in the classifier. 3) In order to discriminate the boundaries from the background more effectively, we chose the chained residual pooling blocks proposed in RefineNet [19]. The block is constructed with pooling features in different window sizes with a residual connection. 4) A boundary refine block with a residual connection is proposed in [23] for boundary alignment. The full pre-activation residual block is proved to be useful in [12] as it improves the propagation of the gradient during the training. Inspired by these ideas, we add residual refine blocks behind each large kernel block to enhance the gradient flow in the whole network.

Our work has four main contributions. 1) We design an end-to-end network architecture in which feature maps from different resolutions of the encoders are fused. 2) Our network incorporates large kernels and residual cascade pooling blocks in the decoders which can enlarge the actual receptive field of the feature maps and process background together with the boundaries. 3) We use a simple post-processing method which can be directly applied on the probability map from the network. 4) Our method achieves competitive performances on the segmentation tasks for two public ssTEM datasets.

## 2. Related Work

Traditionally, neuron membrane segmentation is conducted largely based on morphological processing techniques. For example, a multi-scale ridge detector [22] was proposed based on a simple neural network. In [16], radon-like features of the image are used for the connectom analysis. The method proposed in [26] detects the membranes in the EM images using Radon like features combined with neural networks. In the two-step classification and post-processing (TSC+PP) [31], the whole architecture contains three parts: image pre-processing methods such as adaptive threshold and dark blob elimination, the support vector machine (SVM), and post-processing. Although these methods can segment the basic contour of the neurons, there still remains some problems in the segmentation result. For example, in the result of [31], there remain many false merge and split errors. The multi-scale ridge detector [22] misclassifies some intracelluar mitochondrias as boundaries. Although TSC+PP [31] overcomes the two problems, there are still so many noises filled with the image, which make the membrane segmentation result ambiguous.

In recent years, deep architectures has shown competitive performance on segmentation, classification, and object detection tasks compared with the traditional machine learning methods. Deep learning based algorithms have been proposed on the neuronal membrane detection problems and have achieved impressive progress [21, 30, 24, 7, 32, 3, 27, 20]. In PyraMiD-LSTM [30], several units in the traditional recurrent architecture Long Short-Term Memory (LSTM) networks are aligned together as a multi-dimensional LSTM model. Then the topology connections are changed into a pyramidal style as a pyramidal LSTM. The model achieves high performance for the segmentation of the volumetric data. Different from the recurrent construction in PyraMiD-LSTM, Optree [32] proposes a conditional random field (CRF) based tree-structured segmentation model. In the testing part, the method updates the tree structure based on manually correction. Ciresan *et al* [7] proposed a deep fully connected network contains a series of convolutional and pooling layers as a per-pixel classifier. This method won the first place in ISBI2012 membrane segmentation challenge for EM images [1]. Similar with Ciresan, DIVE [10] is proposed as an optimized deep neural network with a merge-tree based watershed post-processing method. In the work of U-Net [24], considering that the information losses from low-level resolutions in the fully convolutional network due to the pooling and convolutional blocks, a skip connection is proposed in the architecture to obtain more detailed information such as edges for the decoders.

With the rapid development of deep convolutional neural network (CNN), many related methods are proposed for the semantic segmentation tasks on general images. SegNet [2] is a fully connected encoder-decoder network. In SegNet, the inputs are firstly downsampled by the encoders as low resolution feature maps. Then they are upsampled by the decoders to the input resolution. However, some feature information gets lost when passing through the pooling or stride convolutional layers in the encoders. In this way, some architectures which contain skip connections between encoders and decoders are proposed to fix the problem, such as U-Net [24] and LinkNet [5]. With each part of the encoders directly connected with the decoders, the feature information from the encoders can be kept. Even if the short connections prevent the information loss, the limit size of the convolutional kernels in the decoder makes the actual receptive field in the network very small which may be harmful for the segmentation. In this way, the Global Convolutional Network [23] (GCN) is proposed with large kernels to ensure the actual valid receptive field is large enough to

cover all the information in the feature maps.

Even though all these methods are proved to be useful in the segmentation tasks for general images, they lack the understanding of the EM images. EM images show many different characteristics from the general images. First, there remain some intracellular structures such as mitochondria, which is likely to be misclassified by some ordinary boundary detection algorithms. Second, when capturing the images, there is some noise due to the manual operation and the instrument itself, which makes the membrane boundries more ambiguous. Third, many vesciles shown as small bubbles appear all around the the neurons, which blur the boundaries and make them difficult to be separated from the background.

Most of the related methods for EM image segmentation are inspired by the deep architecture for the general image segmentation. However, there still remain some open questions such as how to make the short connections to prevent the loss of different levels of information and how to enlarge the actual receptive field to obtain more details in the feature maps. Our method is inspired by the architecture of RefineNet [19] and are able to solve the problems above. There are several significant differences between our work and the state-of-the-art methods: 1) Ulike the 4-cascade connection between the encoders and decoders proposed in RefineNet, U-Net, LinkNet, and GCN, we fuse the feature maps from the encoders of different resolutions together for the proceeding operation in the decoders. This is because in the original EM images, the ratio of the membrane to the whole image is much smaller than the ratio of background, which means there is not as much information in the feature maps. In the 4-cascade connection, two feature maps from high-level resolutions are fused together and passed through some convolutional blocks. Then the output is upscaled and fused with the feature maps from low level resolutions. If the feature maps with limited useful information pass through so many convolutional layers, the valid information remaining in the feature maps will be not enough for the classifier due to the information loss. 2) In order to ensure the actual receptive field is large enough, we add large kernel convolutional blocks directly after the feature maps from the encoders. 3) A full pre-activation residual connection is proposed to use instead of the traditional one to improve the gradient back propagation.

## 2.1. Methods

## 2.2. Large Kernel Refine Fusion Network

In this section, we introduce our deep Large Kernel Refine Fusion Network. Our architecture is an encoder-decoder structure shown in Fig. 2. In this work, we use the ResNet blocks from ResNet50 [11] which is pretrained on the ImageNet [25] for the encoders. The numbers of the channels of four side outputs at different resolutions are $256$, $512$, $1024$ and $2048$ respectively. The output feature maps of each ResNet layer in the encoder pass through a large kernel convolutional block with a kernel size of 7 and a filter number of $64$ for a large receptive field. In order to refine the boundaries from the feature maps after the large kernel convolutional blocks, the outputs are sent to a residual refine block with a filter number of $64$ in the next step. Then we fuse the side outputs from four resolution levels together. Before fusing them together, we use bilinear upsample function to upscale the size of the feature maps the same as the input size. For fusion, we use concatenation instead of summation. Even though the studies in [5] and [23] show that summation can speed up the training and reduce the parameters in the architecture, there would be some information loss when the features of different resolutions are summed. After we fuse the feature maps together, we propose a residual cascade pooling layer with a size of $64$ for background and membrane detection. At the end of the network, the final convolutional layers contain two parts. First we make a final refinement with a residual refine block whose filter number is $64$. Second we use a ReLU layer and a batch normalization layer before each $1 \times 1$ convolutional layer.

### 2.2.1 Large kernel convolutional block

In the experiment of [34], we can find out the sizes of the actual receptive fields are always smaller than the theoretical receptive fields, especially in the deeper layers. For a traditional convolutional block with a small kernel size, it can cover the whole object in the feature maps at the low level resolutions. At the high resolution layers, even though the receptive field is large enough to cover the whole image, the information from the receptive field only contains a limited part of the whole feature map due to the limit of the actual receptive field size. Inspired by the large kernel analysis in [23], we propose to use large kernel convolutional blocks in our network right after the outputs from the encoders with different resolutions. However, if we apply a convolutional block with a large kernel size directly to the network, there will be a large amount of parameters which increase the computational burden. In our method, we employ a simulation of a $K \times K$ convolutional kernel comprising a combination of one $K \times 1$ convolutional kernel and one $1 \times K$ convolutional kernel shown in Fig. 2.

### 2.2.2 Residual refine block

After the large kernel convolutional block, we define a residual block. Our design is inspired by the boundary refinement block proposed in [23]. However, the design in [23] does not contain a non-linear activation function before the first internal convolutional layer, which could easily result in vanishing gradients during the back propagation.

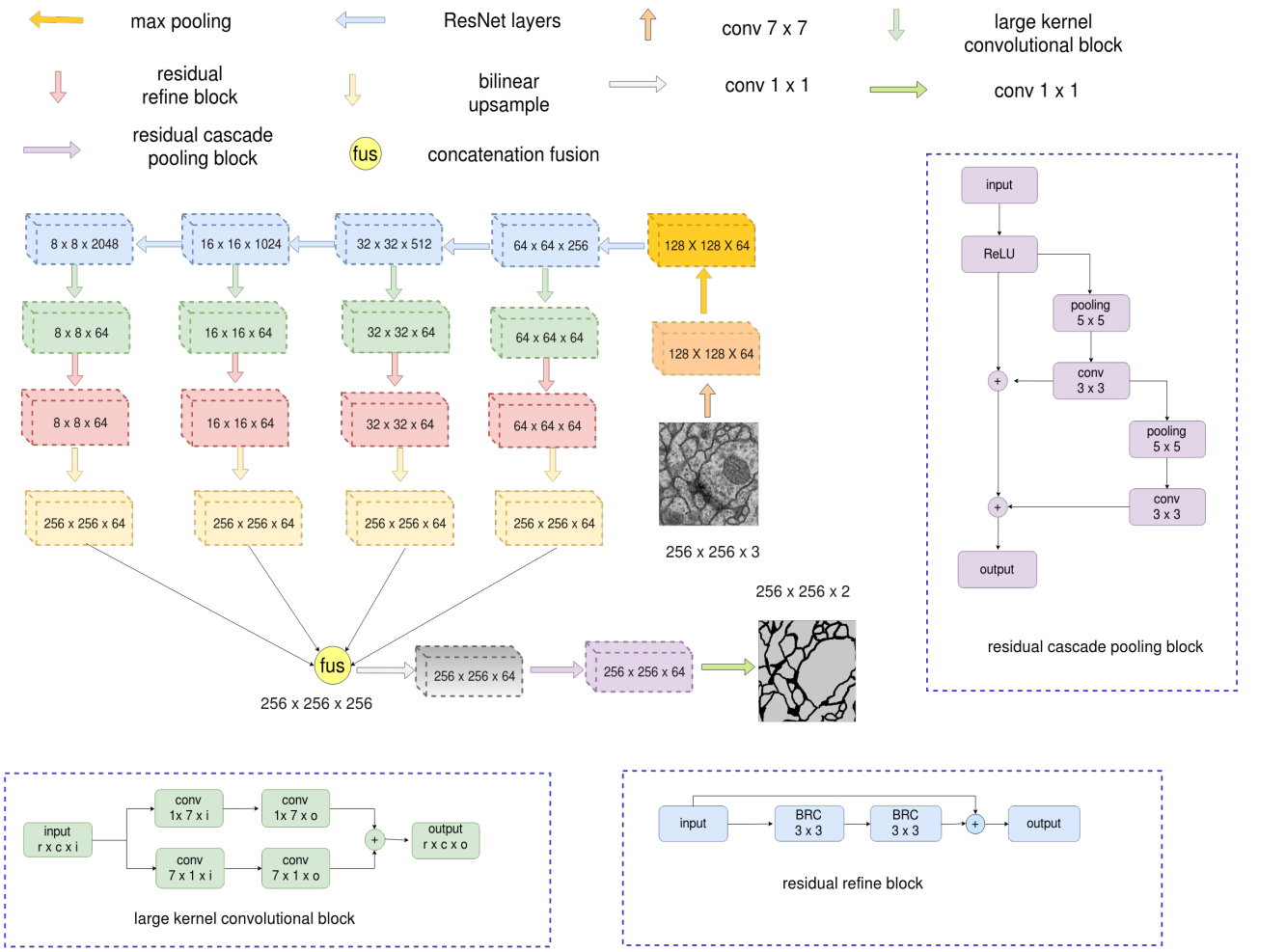


Figure 2. The architecture of our proposed framework. In first maxpooling layer, we set the kernel size as 3, the stride as 2 and the padding size as 1. In the two $1 \times 1$ and one $7 \times 7$ conv blocks, there is a BN-ReLU connection connected before. In the large kernel convolutional block, $r$ and $c$ mean the length and width of the feature map. $i$ and $o$ represent the number of input and output channel respectively. The $BRC$ block in residual refine block represent a BN-ReLU-Conv connection. The type of the pooling layers in the residual cascade pooling block is maxpooling.

To solve this problem, we construct the residual connection with a full pre-activation connection proposed in [12] which makes the back propagation of the gradient more effective. In our residual refine block, non-linear ReLU layers are placed on the side branches of the main information path together with the convolutional blocks. In our experiment, the filter number of each block is set to 64.

#### 2.2.3 Residual cascade pooling block

The output feature maps from the residual refine blocks then pass through a residual cascade pooling layer, which is proposed to be used for analysis of the background context in the image. As shown in Fig. 2, a residual cascade pooling block contains two layers of pooling and convolutions. Then the outputs from the two layers are summed together with the original input. The architecture forms a residual connection to process the image background. In this block, after the input passes through one pooling layer, the side output feature map contains information at a higher resolution. When fusing the features of different resolutions together, the final output contains context information from different resolutions, which produces more useful context information for the final layer.

### 2.3. Boundary Refinement Post-processing

Although the output probability maps from our Large Kernel Refine Fusion Net can show the details well, there is still some noise which make some membrane boundaries ambiguous. Inspired by the morphological boundary refinement method proposed in [6], we propose a post-processing method based on thresholding and distance transform. The

effect of our method is shown in Fig. 3, where some noise which makes the boundaries ambiguous is removed after the post-processing.

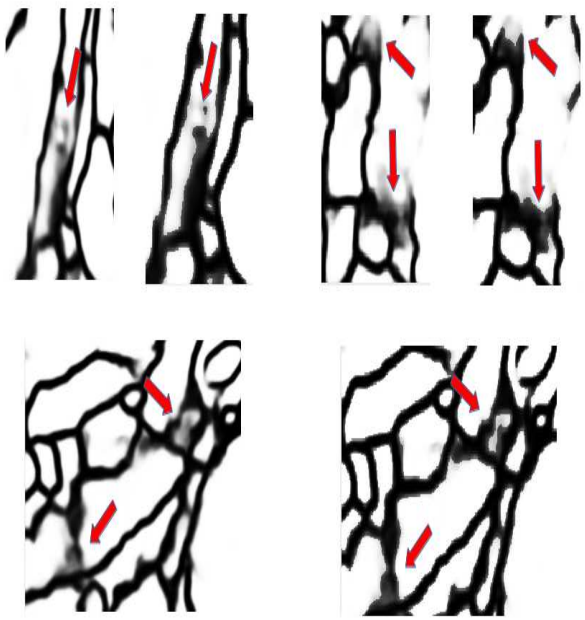

Figure 3. Some examples to show the effect of our Boundary Refine Post-processing. Left: part of the original probability map, right: the corresponding result after post-processing

In our experiment, we firstly set a threshold value of $0.5$ on the original probability map to get a binary image $I_o$. Then the exact Euclidean distance transformation is applied on $I_o(x)$, where $x$ is defined as the input image. In this distance map, we set a distance value 1 as a threshold to generate a distance binary image $I_b(x)$. The final probability map $I_f(x)$ can then be represented by a linear combination of $I_o(x)$ and $I_b(x)$:

$$I_f(x) = \gamma I_b(x) + (1-\gamma) I_o(x) \tag{1}$$

where $\gamma$ is the ratio of the binary distance transform image to the final probability. Based on our empirical studies, we set $\gamma$ as $0.43$ in the ISBI2012 EM segmentation challenge [1].

# 3. Experiments and Results

## 3.1. Experiment Setting

We evaluate our method on two EM membrane segmentation tasks, namely the Mouse Piriform Cortex segmentation [17] and ISBI2012 EM segmentation challenge [1]. In the ISBI2012 challenge, we use ADAM [15] as the optimizer for the network with $\beta_1 = 0.9$, $\beta_2 = 0.99$, $\epsilon = 10^{-8}$ and a weight decay of $0.0005$. We set the initial learning rate as $0.0001$ for the first $100$ epochs. Then we use the "poly" learning rate decay for the next $50$ epochs following the equation:

$$lr_{iter+1} = lr_{iter} \times (1 - \frac{iter}{iter_{max}})^{power} \tag{2}$$

where we set the $power$ as $0.9$ and $iter_{max}$ as $50 \times iterations/epoch$. In the binary ground truth of the EM images, the ratio of the background is always larger than that of the membrane, which causes a class imbalance problem. In our experiment, we propose to use a cross-entropy loss with weight parameters assigned to each class to solve the problem. In ISBI2012 challenge, the average ratio of the membrane and background of the training ground truth is $1:4$ and we assign the weight of the membrane and the background in the cross entropy loss as $5$ and $1.25$ respectively. In the Mouse Piriform Cortex segmentation task, in order to evaluate the effectiveness of our proposed Large Kernel Refine Fusion Net, we only use the ADAM optimizer with the same parameters as in the ISBI2012 challenge. Other settings including weighted cross-entropy loss, learning rate decay and post-processing are not used in this experiment. All the networks in the experiment are implemented using Pytorch (`http://pytorch.org`).

## 3.2. Experiment Evaluation

To evaluate the performance of our method, we follow the Rand F-score mentioned in [1] on the two EM segmentation tasks. Suppose that $S$ is the predicted segmentation result and $T$ is the ground truth. We define a $p_{ij}$ as the probability of a randomly selected pixel belongs to the $i$-th segment in $S$ and the $j$-th segment in $T$. The $V_{split}^{Rand}$ score and $V_{merge}^{Rand}$ score in range $[0, 1]$ are defined as follows:

$$\begin{aligned} V_{split}^{Rand} &= \frac{\sum_{ij} p_{ij}^2}{\sum_k t_k^2} \\ V_{merge}^{Rand} &= \frac{\sum_{ij} p_{ij}^2}{\sum_k s_k^2} \end{aligned} \tag{3}$$

where $s_i = \sum_j p_{ij}$ and $t_j = \sum_i p_{ij}$. $V_{split}^{Rand}$ score and $V_{merge}^{Rand}$ score are defined as the probability of two randomly selected voxels are from the same segment in $S$ when given they belong to the same segment in $T$ and the probability of two randomly selected voxels are from the same segment in $T$ when given they belong to the same segment in $S$ respectively. The $V_{split}^{Rand}$ score and $V_{merge}^{Rand}$ score become higher when there are less split and merge errors respectively. In order to combine the two scores together, the weighted harmonic mean is used:

$$V_{\alpha}^{Rand} = \frac{\sum_{ij} p_{ij}^2}{\alpha \sum_k s_k^2 + (1-\alpha) \sum_k t_k^2} \tag{4}$$

Table 1. The whole process of our experiment which shows the effect of different data augmentations, optimisers, learning rate schedule and weight assignment for cross entropy loss

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Lr Schedule? | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Class-weighted Loss? | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Gaussian Blur? | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Affine Transformation? | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ |
| Elastic Transformation? | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ |
| Image Dropout? | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ |
| Post-processing? | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ |
| $V^{Rand}_{thinning}$ score | 0.9131 | 0.9619 | 0.9661 | 0.9677 | 0.9704 | 0.9756 | 0.9705 | 0.9766 |

Table 2. The ablation experiment for our proposed architecture

| | | | | | | |
|---|---|---|---|---|---|---|
| residual refine? | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ |
| large kernels? | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ |
| residual cascade pooling? | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ |
| 4-cascade $\rightarrow$ single connection? | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ |
| Dropout layer? | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ |
| $V^{Rand}_{thinning}$ score | 0.9595 | 0.9610 | 0.9639 | 0.9711 | 0.9756 | 0.9686 |

When $\alpha$ is $0.5$, it represents the Rand F-score, which is equal to:

$$V_F^{Rand} = \frac{2V_{split}^{Rand}V_{merge}^{Rand}}{V_{split}^{Rand} + V_{merge}^{Rand}} \tag{5}$$

As there remains the thickness variations of the predicted borders, we use the $V_F^{Rand}$ after border thinning to enhance its robustness. The evaluation of our experiment is based on a script in IMAGEJ [8], which calculates the best $V_F^{Rand}$ score after thinning over threshold in different values for the probability map (`http://imagej.net/Segmentation_evaluation_after_border_thinning_-_Script`).

### 3.3. ISBI2012 EM Segmentation Challenge

In this challenge, the EM dataset contains a training stack with its corresponding segmentation ground truth annotated by experts and a testing stack with its ground truth kept by the hosts of the challenge. There are 30 slices of image with size $512 \times 512$ in both the training and testing stacks. All the EM images are collected from the first instar larva ventral nerve cord of the Drosophila [1]. In this challenge, the $V_F^{Rand}$ score after border thinning is obtained by submitting the predicted probability map to the challenge website (`http://brainiac2.mit.edu/isbi_challenge/`).

#### 3.3.1 Data augmentation

There are only 30 EM images slices in the training dataset, thus data augmentation is needed to train our complex deep architecture. In our experiment, besides some basic augmentations such as crop, flip and rotate, we also try to add gaussian filter, affine transform, elastic transform, and image dropout. Table 1 shows that among all the augmentation methods, only image dropout has a negative effect on the result. This is because the useful information in the whole image (membrane) is only in about one-fifth of the whole image. If we further drop some information during training, there will be a worse classification result due to the lack of useful information. Gaussian filter can remove the noise in the images while elastic and affine transform can produce a lot of distorted images, which can prevent the overfitting problem under the huge amount of the parameters in the network.

#### 3.3.2 Ablation experiment

Table 2 shows the performance of the model when removing parts of the convolutional blocks proposed in our method. We can see that when adding the large kernel blocks, residual refine blocks, and residual cascade pooling blocks, the segmentation results become better than the model without them. Even though the experiments proposed in [19] proves that the 4-cascade connection achieves better performance than the single connection for general images segmentation, we can get a better segmentation result by using the single connection in our experiment. This is because in the 4-cascade connection, the feature maps pass through too many convolutional blocks before being merged together, which causes information loss. In addition, dropout layers are proved to solve the overfitting problems in [33, 24]. In our experiments, we try to add dropout layers before the final convolutional layer. We have experimented with a range of dropout rates of $0.1$, $0.3$, and $0.5$ but none of them im-

prove the segmentation accuracy.

#### 3.3.3 Result comparison

Fig. 4 is one slice of the test dataset and its corresponding segmentation result. From the highlighted parts, we can see our method can not only remove the intracellular shadows, but also segment some ambiguous boundaries clearly. In addition, the vesicles all around the image are diminished to a large extent.

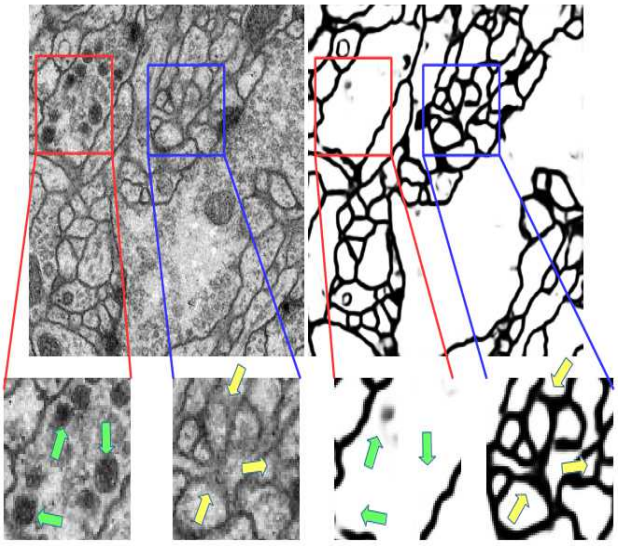

Figure 4. Left: the 1/30 slice of the testing dataset and some highlight details. Right: the corresponding segmentation result.

Table 3 shows the comparison of our method with the state-of-the-art approaches on the ISBI2012 challenge. As our method is about deep neural network, we choose to compare with some deep learning based approaches among the 120 groups attending in this challenge. For more details about the ranking, please refer to the leaderboard on the official website (`http://brainiac2.mit.edu/isbi_challenge/leaders-board-new`).

Table 3 shows that our proposed architecture works better than PolyMtl [9], RotEqNet [21], IDSIA [7], and U-Net [24] without any further processing. Although Pyramid-LSTM [30] achieves high performance on 3D image processing, the anisotropic characteristic of this EM dataset limits its capability. DIVE-SCI [10] and Optree [32] process the dataset with complicated post-processing methods. However, our method outperforms them with a simple post-processing method, which is effective and easy to reproduce together with any other architecture. Although the result from CUMedVision [6] is slightly better than ours, the result is obtained by averaging the results from several models. Compared with their complicated training process, our method is more efficient.

Table 3. Evalution on ISBI2012 challenge.

| Method | $V_{Rand}$ |
|---|---|
| CUMedVision [6] | 0.9768 |
| DIVE-SCI [10] | 0.9762 |
| IDSIA [7] | 0.9730 |
| U-Net [24] | 0.9727 |
| RotEqNet [21] | 0.9712 |
| Optree [32] | 0.9712 |
| PolyMtl [9] | 0.9690 |
| Pyramid-LSTM [30] | 0.9676 |
| Ours (with post-processing) | 0.9766 |
| Ours (without post-processing) | 0.9756 |

### 3.4. Mouse Piriform Cortex Segmentation

In this experiment, we use the piriform cortex EM dataset of the mouses from [17], which is captured from adult mouse piriform cortex by ssTEM. The whole dataset contains four stacks of EM images which are in different image sizes. Unlike the experiment settings in [17] and [28], we use stack1 and stack2 with the slice image sizes of $256 \times 256$ and $512 \times 512$ respectively for training, and stack3 and stack4 with the slice image sizes of $512 \times 512$ and $256 \times 256$ respectively for testing. In order to show the effect of our proposed architecture, we do not use our proposed post-processing method in the comparison experiment. The data augmentations of this experiment include vertical, horizontal flipping, and rotating with $90°$, $180°$, and $270°$.

Table 4. Comparison with other deep networks for semantic segmentation on Mouse Piriform Cortex dataset.

| Method | Stack3 | Stack4 |
|---|---|---|
| SegNet [2] | 0.8164 | 0.6399 |
| LinkNet [5] | 0.8201 | 0.8100 |
| GCN [23] | 0.8286 | 0.8583 |
| PSPNet [33] | 0.7367 | 0.8404 |
| Proposed | 0.8534 | 0.8685 |

Table 4 is the comparison between our proposed architecture and some state-of-the-art methods. Our method outperform them under the same experiment setting. Compared with SegNet [2], PSPNet [33] contains a pyramid pooling layer which is able to increase the segmentation accuracy by collecting more representative information than the ordinary pooling layers. However, neither of them can prevent the low-level feature information diminishing due to the number of convolutional layers in the architectures. LinkNet [5] overcomes this problem by using the skip connections between the encoders and decoders. Although

LinkNet is able to achieve better performance in segmentation, the traditional convolutional layers with small kernel size make the actual receptive field of the feature maps too small. In this way, GCN [23] is proposed with large convolutional kernels to enlarge the actual receptive field for the feature maps to retain more information from all resolutions. In our proposed network, we optimise the boundary refine block proposed in [23] with the idea of pre-activation. In addition, a residual cascade pooling block is proposed to use for processing the membrane more effective. As shown in Table 4, the $V^{Rand}_{thinning}$ is higher than the others on the two testing set. Such improvement is important for the final result in 3D reconstruction.

## 4. Conclusion

In this work, we present a Large Kernel Refine Fusion Net for the neuronal membrane segmentation of EM images. By fusing the feature maps directly in the decoders and upscaling for the next residual cascade pooling blocks, the network keeps the detailed information in the feature maps. The large kernels in the decoder enlarge the actual receptive field for the features from the encoder. The residual refine block refines the boundaries and improves the gradient flow during the training. Additionally, the background information is also processed by a residual cascade pooling block to further enhance the segmentation performance. Evaluated on two EM segmentation datasets, our proposed method is shown to outperform some state-of-the-art algorithms. In the future work, our proposed method will be implemented on other ssTEM datasets to evaluate its effectiveness and robustness.

## References


[1] I. Arganda-Carreras, S. Turaga, D. Berger, D. Cireşan, et al. Crowdsourcing the creation of image segmentation algorithms for connectomics. *Frontiers in neuroanatomy*, 9:142, 2015.

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[3] T. Beier, B. Andres, U. Köthe, and F. A. Hamprecht. An efficient fusion move algorithm for the minimum cost lifted multicut problem. In *ECCV*, pages 715–730, 2016.

[4] A. Cardona, S. Saalfeld, S. Preibisch, B. Schmid, A. Cheng, J. Pulokas, P. Tomancak, and V. Hartenstein. An integrated micro-and macroarchitectural analysis of the drosophila brain by computer-assisted serial section electron microscopy. *PLoS biology*, 8(10):e1000502, 2010.

[5] A. Chaurasia and E. Culurciello. Linknet: Exploiting encoder representations for efficient semantic segmentation. *arXiv preprint arXiv:1707.03718*, 2017.

[6] H. Chen, X. Qi, J. Cheng, P. Heng, et al. Deep contextual networks for neuronal structure segmentation. In *AAAI*, pages 1167–1173, 2016.

[7] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In *NIPS*, pages 2843–2851, 2012.

[8] T. J. Collins et al. Imagej for microscopy. *Biotechniques*, 43(1 Suppl):25–30, 2007.

[9] M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, et al. The importance of skip connections in biomedical image segmentation. In *MICCAI Workshops*, pages 179–187, 2016.

[10] A. Fakhry, H. Peng, and S. Ji. Deep models for brain em image segmentation: novel insights and improved performance. *Bioinformatics*, 32(15):2352–2358, 2016.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *ECCV*, pages 630–645, 2016.

[13] M. Helmstaedter. Cellular-resolution connectomics: challenges of dense neural circuit reconstruction. *Nature methods*, 10(6):501, 2013.

[14] V. Kaynig, A. Vazquez-Reina, S. Knowles-Barley, M. Roberts, T. R. Jones, N. Kasthuri, E. Miller, J. Lichtman, and H. Pfister. Large-scale automatic reconstruction of neuronal processes from electron microscopy images. *Medical image analysis*, 22(1):77–88, 2015.

[15] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.

[16] R. Kumar, A. Vázquez-Reina, and H. Pfister. Radon-like features and their application to connectomics. In *CVPR Workshops*, pages 186–193, 2010.

[17] K. Lee, A. Zlateski, V. Ashwin, and H. S. Seung. Recursive training of 2d-3d convolutional networks for neuronal boundary prediction. In *NIPS*, pages 3573–3581, 2015.

[18] J. W. Lichtman and W. Denk. The big and the small: challenges of imaging the brains circuits. *Science*, 334(6056):618–623, 2011.

[19] G. Lin, A. Milan, C. Shen, and I. Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *CVPR*, pages 5168–5177, 2017.

[20] T. Liu, C. Jones, M. Seyedhosseini, and T. Tasdizen. A modular hierarchical approach to 3d electron microscopy image segmentation. *Journal of neuroscience methods*, 226:88–102, 2014.

[21] D. Marcos, M. Volpi, N. Komodakis, and D. Tuia. Rotation equivariant vector field networks. In *ICCV*, pages 5058–5067, 2017.

[22] Y. Mishchenko. Automation of 3d reconstruction of neural tissue from large volume of conventional serial section transmission electron micrographs. *Journal of neuroscience methods*, 176(2):276–289, 2009.

[23] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun. Large kernel mattersimprove semantic segmentation by global convolutional network. In *CVPR*, pages 4353–4361, 2017.

[24] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015.

[25] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[26] M. Seyedhosseini, R. Kumar, E. Jurrus, R. Giuly, M. Ellisman, H. Pfister, and T. Tasdizen. Detection of neuron membranes in electron microscopy images using multi-scale context and radon-like features. In *MICCAI*, pages 670–677, 2011.

[27] M. Seyedhosseini, M. Sajjadi, and T. Tasdizen. Image segmentation with cascaded hierarchical models and logistic disjunctive normal networks. In *ICCV*, pages 2168–2175, 2013.

[28] W. Shen, B. Wang, Y. Jiang, Y. Wang, et al. Multi-stage multi-recursive-input fully convolutional networks for neuronal boundary detection. In *ICCV*, 2017.

[29] O. Sporns, G. Tononi, and R. Kötter. The human connectome: a structural description of the human brain. *PLoS Computational Biology*, 1(4):e42, 2005.

[30] M. F. Stollenga, W. Byeon, M. Liwicki, and J. Schmidhuber. Parallel multi-dimensional lstm, with application to fast biomedical volumetric image segmentation. In *NIPS*, pages 2998–3006, 2015.

[31] X. Tan and C. Sun. Membrane extraction using two-step classification and post-processing. In *Proc. of ISBI*, 2012.

[32] M. Uzunbaş, C. Chen, and D. Metaxsas. Optree: a learning-based adaptive watershed algorithm for neuron segmentation. In *MICCAI*, pages 97–105, 2014.

[33] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *CVPR*, pages 2881–2890, 2017.

[34] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. *ICLR*, 2015.