DeepEM3D: Approaching human-level performance on 3D anisotropic EM image segmentation

Article *in* Bioinformatics · March 2017

DOI: 10.1093/bioinformatics/btx188

3 authors, including:

Tao Zeng
Washington State University
25 PUBLICATIONS **697** CITATIONS
SEE PROFILE

Bian Wu
Zhejiang University
22 PUBLICATIONS **266** CITATIONS
SEE PROFILE

Bioinformatics
doi.10.1093/bioinformatics/xxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

OXFORD

Subject Section

DeepEM3D: Achieving human-level performance on 3D anisotropic EM image segmentation

Tao Zeng [1], Bian Wu [1], and Shuiwang Ji [1,*]

[1] School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA 99164, USA

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Progress in 3D electron microscopy (EM) imaging has greatly facilitated neuroscience research in high-throughput data acquisition. Correspondingly, high-throughput automated image analysis methods are necessary to work in par with the speed of data being produced. One of such examples is the need for automated EM image segmentation for neurite reconstruction. However, the efficiency and reliability of current methods are still lagging far behind human performance.
Results: Here, we propose DeepEM3D, a deep learning method for segmenting 3D anisotropic brain electron microscopy images. In this method, the deep learning model can efficiently build feature representation and incorporate sufficient multi-scale contextual information. We propose to employ a combination of novel boundary map generation methods with optimized model ensemble to address the inherent challenges of segmenting anisotropic images. We evaluated our method by participating in the 3D segmentation of neurite in EM images (SNEMI3D) challenge. Our submission is ranked #1 on the current leaderboard. More importantly, our result is very close to the human-level performance in terms of the challenge evaluation metric; namely a Rand error of 0.06015 versus the human value of 0.05998.
Contact: sji@eecs.wsu.edu

1 Introduction

Brain functions are governed by information flow through the interconnected circuits of neurons. Investigating components and network structure of this circuit is one of the top priorities in neuroscience research (Lichtman and Denk, 2011; Peng *et al.*, 2015). The goal of EM-based connectomics is to map neuronal circuits at single-cell level. This consists of two parts; namely generating high-resolution 3D images of neural tissue and reconstruction of corresponding circuits using these images. While the progress in EM imaging allows high-throughput acquisition of high-resolution neural images at unprecedented scale, the speed of image analysis is lagging behind, forming a bottleneck to high-throughput connectomics (Lichtman and Denk, 2011). In early work of neuron circuits reconstruction that focused on organisms with simple neural system such as *C. elegans*, the labeling was mainly done manually. As the focus extends to more complex organisms, image resolution and size scales up, even semi-automated method becomes a burden for human intervention. Despite the attempts to fully automate this process, current methods are not able to achieve the required accuracy. As a result, reconstruction still relies heavily on human labor for proofreading of machine-segmented results. A breakthrough for automatic dense EM reconstruction is in urgent need, and better machine learning algorithms are considered as the major approach toward this goal (Jain *et al.*, 2010; Helmstaedter *et al.*, 2013).

A common approach to generate EM images is to do serial section imaging that slice and scan brain tissues and stack the images together as volume. The EM scanning can achieve very high resolution on surface (X-Y axes) of each slice, while slicing resolution in Z-direction is generally 10 times lower (Briggman and Bock, 2012), which results in anisotropic 3D images stack. Although recent advances in EM techniques have enabled isotropic 3D images, we focus on anisotropic 3D-EM images in this work, since majority of existing datasets of interest are anisotropic. Reconstructing neuronal circuit from such volume is very challenging. This is primarily because of the anisotropic nature of the data in which adjacent image slices are often misaligned due to imperfect sectioning on tissues or errors in alignment algorithms. In additional, processing large 3D volume is very time-consuming and computationally costly.

Typically, neurite reconstruction involves two steps; namely generation of probability map of membrane from EM images and segmentation

of neuronal process based on this map. Early studies applied methods such as random forest (Kaynig *et al.*, 2015; Liu *et al.*, 2013) to obtain probability maps. Recently, there has been increasing cases that use deep convolutional neural networks (DCNN) (Turaga *et al.*, 2010; Ciresan *et al.*, 2012; Berning *et al.*, 2015). For the segmentation step, previous methods usually first generate over-segmented regions to form a hierarchical region graph before some merging algorithms were applied to determine whether they should be merged. For anisotropic 3D EM data, the segmentation is usually done on 2D slice one by one and then these 2D regions are grouped across multiple sections into geometrically consistent 3D neuronal objects. Problems with such approaches are that, domain knowledge is required to extract region features, and building hierarchical region graph is computationally intensive.

It has been shown that putting more efforts to learn a better membrane predictor will simplify post-processing, thereby leading to dramatic increase in speed and improved applicability of methods. With the same spirit, we argue that algorithms providing highly accurate prediction of boundary probability map require only simple post-processing, thereby reducing both computation cost and requirement on domain knowledge.

In our prior work (Fakhry *et al.*, 2016a), we have developed deep learning methods for segmenting 2D EM images (Arganda-Carreras *et al.*, 2015). Accurate segmentation of 3D anisotropic EM images not only requires accurate segmentation of 2D slices, but also needs to make consistent predictions across slices. This is further complicated by the fact that alignment of 2D slices is a challenging task, and thus segmentation methods have to deal with misalignment problems. In this work, we have developed a set of methods for accurate segmentation of 3D anisotropic EM images. Specifically, we proposed a pipeline for segmenting 3D EM image with minimum computation cost for post-processing, yet producing very promising results. Our approach in 3D EM neurite reconstruction archived human-level accuracy. To the best of our knowledge, this is the first time automatic neurite reconstruction reaches this level of performance. The main contributions of this work are: (1) We proposed a novel deep network model for 3D neurite boundary detection. (2) We developed a pipeline for segmenting image stacks which requires far less computation during post-processing and can be easily generalized to other dataset. (3) We developed an ensemble strategy to deal with anisotropy and misalignment problems which are commonly seen in 3D EM data. (4) Our method represents the first computational approach that achieved human-level accuracy in 3D-EM neurite segmentation.

2 Deep learning for anisotropic 3D EM images

Here we present a deep learning method for anisotropic 3D EM image segmentation. Our method is able to produce highly accurate 3D neurite boundary probability map, thereby requiring only simple watershed method to do segmentation. We applied our methods to the anisotropic 3D images from the 3D segmentation of neurites in EM images (SNEMI3D) (`http://brainiac2.mit.edu/SNEMI3D/`) challenge. We created new 3D boundaries of neurite from segmentation labels in training data. We then used these labels and raw image stacks to train our deep models. To tackle problems resulted from the anisotropic resolution, we derived a series of architecture variants differing in the number of input slices, which allows us to observe the effects of anisotropy in various scales. Upon inference, we used an ensemble strategy to effectively fuse the outputs of variants in order to integrate useful cross-slice information, while suppressing the cross-slice noise introduced by anisotropy. Finally, simple 3D watershed was used to produce final 3D segmentation.

2.1 Overview of key technical innovations

Traditionally, classifiers trained on handcrafted features were used to merge over-segmented regions during post-processing. This process requires a lot of computation and makes the whole system not end-to-end trainable. In order to simplify the post-processing in a way that relies only on watershed algorithm for final segmentation, two issues need to be addressed.

First, watershed algorithm is expected to connect geometrically non-separated areas. Thus a tiny broken point on the boundary between two regions can lead to a false merge. In other word, post-processing by watershed is sensitive to false negative prediction on boundary. On the other hand, this method is robust to false positive boundary. As long as no enclosed region is formed, it would not result in over-segmented regions.

The anisotropic property of 3D EM data is another challenge for 3D segmentation that uses simple post-processing. Comparing to X-Y directions, the boundary locations do not change smoothly along Z-axis, leading to high probability of broken boundary in Z-direction. In addition, 3D EM image volume is comprised of stacked section images in which tissues are sectioned slice by slice across Z-direction. Such image acquisition method has inherent problem of misalignment or deformation.

Taken together, our proposed method requires probability maps to be very accurate in terms of precision while allowing some degree of tolerance in recall metric. Hence, the efforts to reduce false broken boundary while attempting to keep false boundary rate unchanged is the key to high quality segmentation. To this end, we applied several techniques at multiple steps in the pipeline aiming at increasing the boundary precision metric while still minimizing the negative affect caused by possible lower recall metric.

First, we applied boundary enhancement technique to the training stack. This generated several boundaries varied in thickness, which were used as labels to train multiple deep models. The idea is that the wide boundary helps maintain its continuity but may tend to reduce the size of small regions or even remove them. By contrast, thin boundary is able to keep the size of small regions but is at risk of resulting in broken boundary in the stack that contain misaligned or deformed slices.

Second, we designed a novel deep architecture that incorporates inception and residual learning techniques in the object abstraction path, and skip connection and pyramid context growing techniques in the resolution recovery path. We constructed several variants of models in order to have them trained on different sizes of slices in Z-direction with labels of various thickness. Careful combination of probability maps produced by these models can enhance predicted boundary that takes advantage of 3D information while reducing anisotropic effects, resulting in high quality boundary maps. In following sections, we describe these techniques in details and show the improvement made by these techniques.

2.2 SNEMI3D dataset

The SNEMI3D challenge data consist of two image stacks used for training and test, respectively. Each stack contains 100 1024×1024 slices. The serial section scanning electron microscopy (ssSEM) technique was used to generate these image stacks by sectioning tissue into slices across the Z-dimension. This yields anisotropic 3D images with high-resolution in the X-Y plane and low resolution along the Z-dimension. The image resolution is $6 \times 6 \times 30$ nm/voxel that covers a micro-cube of approximately $6 \times 6 \times 3$ microns (Kasthuri *et al.*, 2015). There are 400 neurites in the training stack that have been labeled consistently across the 100 slices. Some neurites are split into several segments in some slices, but consistent labeling across the z-direction is still required. This increases the complexity of 3D segmentation significantly. The labels of the test stack are not available to challenge participants, and segmentations submitted by participants are evaluated by challenge organizers. Adapted rand index (ARI), computed as ARI=1- F-score, was used for evaluation. The ground truth labels were based on the consensus between two human expert raters, and

the difference between them was listed as human values on the challenge leader board.

2.3 Boundary label generation

In order to train classifiers for identifying neurite boundary, a common approach is to use the provided membrane labels. However, in the SNEMI3D dataset, such membrane labels do not form closed boundaries for all neurite. Since we rely heavily on accurate boundaries to simplify the post-processing step, the quality of deep model is of critical importance. The false broken boundary in the training labels is harmful, as it may lead to trained networks that predict more false negative. This would consequently result in the false merge of regions, as the simplified post-processing is unable to correct the broken section. In this spirit, instead of directly using the provided membrane label, we generated our own boundary labels from the segmentation labels of training dataset. Suppose B is the binary boundary label map we want to generate, L is the provided segmentation map, we have

$$B(x,y,z) = \begin{cases} 0, \text{if } S_{x,y,z} = 1 \\ 1, \text{if } S_{x,y,z} > 1, \end{cases}$$

$$\text{where } S_{x,y,z} = |L \cap K_{x,y,z}|.$$

Here $B = 1$ denotes boundary and 0 denotes neurite, $K_{x,y,z}$ is a 3D kernel centered at location (x, y, z), $S_{x,y,z}$ is the number of distinct labels in the 3D space formed by the intersection of segmentation label map and kernel, $|\cdot|$ denotes set cardinality. This basically says if pixels in the 3D kernel centered at (x, y, z) contains more than one distinct segmentation labels, pixel at (x, y, z) of boundary label map is marked 1. In this study, three different kernels were used: 5^3 cube, 3^3 cube, and 6-connected neighborhood along each axis (up-down, left-right, front-back). These are denoted as $c5$, $c3$, and $l3$ respectively. We used sliding window search to mark all pixels as either boundary or not.

2.4 Convolution neural networks for dense prediction

Convolution neural networks (CNNs) have been successful in solving many visual tasks in recent years (LeCun *et al.*, 1998; Krizhevsky *et al.*, 2012). CNNs in these tasks build hierarchical representations of the input in a bottom-up fashion. This is mainly achieved by applying convolutional layers followed by pooling in alteration to increase contextual information of each unit while reducing its spatial resolution. The increase of contextual information is achieved by increased receptive field size as well as increased number of feature maps. All together, they compute object level representations at higher layers.

In dense prediction, the task is to label each pixel in the image. It thus needs to preserve spatial resolution. Therefore, such task implicitly requires to solve the conflicting goals of preserving resolution and augmenting context at the same time. Fully convolutional networks (FCN), which extends the architecture used in image classification tasks by adding a resolution recovery path, was proposed to efficiently generate such dense predictions (Long *et al.*, 2015). The resolution recovery path in FCN uses bilinear kernels or learned transpose-convolution (deconvolution) kernels to recover the spatial resolution from multiple intermediate network layers, which were then aggregated to yield final prediction. Such aggregation is also referred to as skip connections in later studies (Pinheiro *et al.*, 2016). FCN-based models have been widely used in EM image analysis tasks (Fakhry *et al.*, 2016b), yielding significant improvements over traditional methods.

For the purpose of 3D EM image segmentation, it is natural to extend the 2D FCN architecture to 3D so the 3D contextual information can be incorporated. The example in Figure 3 clearly shows the advantage of using 3D contextual information. One major obstacle to a fully 3D FCN is the high computational and memory cost related to 3D convolution. More importantly, the anisotropic property in many 3D EM images may cause problem if we simply treat all three dimensions the same in 3D convolution. In light of these challenges, we tackled both problems jointly by designing a 3D-2D hybrid architecture for FCN-based model (Lee *et al.*, 2015).

2.5 Inception and residual module structure

In deep learning, the network depth is an important factor. However, more layers may also lead to overfitting and gradient vanishing problems. Size of receptive field, or size of viewing field that the network can take information from, is another important factor that affects network performance. Larger receptive field includes more information. But as the receptive field grows larger, it also takes in more unrelated information (noise) if the object of interest does not fully occupies the field. Considering the fact that objects in a real task may vary in size, a fix-sized receptive field may not be sufficient.

Inception networks (Szegedy *et al.*, 2015) try to solve the above problem by using stacked inception modules. The inception module allows information to pass through kernels of multiple sizes (which correspond to multiple receptive field sizes) in a parallel manner. The downside of this approach is that use of multiple kernels increases the number of network parameters and computation cost. Inception networks alleviate this problem by applying 1x1 convolution to reduce the number of input feature maps prior to convolution with large kernels. This effectively reduces model complexity yet achieves superior performance.

Another recent advance in deep learning is residual learning (He *et al.*, 2016), which introduces shortcut path that skip one or serval processing layers. The shortcut connections are equivalent to performing simple identity mapping, requiring minimum additional computation. The processing layers in the block parallel to the shortcut are considered to learn residual between input and output feature maps, instead of the projection to output directly. Such residual learning is considered to be much easier than learning the original functions and is able to mitigate the gradient vanishing problem. As a result, residual learning enables the training of very deep networks.

2.6 Multi-scale context aggregation

In natural images, objects in view can vary in size depending on its physical size as well as viewing distance to it. Similar to the benefit of using multi-scale receptive fields at the module level, providing multi-scale context information at whole network level can improve semantic segmentation. Recently, dilated convolution has been developed to allow larger receptive field without increasing the number of parameters. Yu and Koltun (2016) applied a series of dilated convolution filters with progressively larger dilation ratio from bottom to top layers, giving rise to very large receptive field at top layers. Chen *et al.* (2016) proposed a spatial pyramid multi-dilated convolutional module to extract features from various receptive fields in parallel. These results are then combined together element-wise as outputs. We employed parallel operations similar to pyramid dilated convolution. Instead of convolution, we applied multiple dilated deconvolutions to aggregate multi-scale context at each upsampling layers, which were then combined to generate output feature maps.

2.7 CNN architecture design

In design of dense CNN architecture for 3D EM neurite segmentation, the following objectives were considered to improve the quality of probability map. (1) High efficiency in terms of both training and test time. (2) Aggregation of multi-scale context information needed for classification. (3) Use Z-directional information while being able to minimize the adversarial affect of noise. As illustrated in Figure 1, at each resolution level, we incorporate residual techniques into inception module by connecting identity projection from bottom layer to top layer similar to Inception-V4

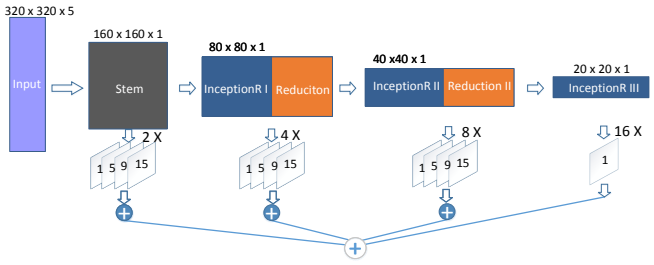


Fig. 1: The architecture of DeepEM3D-Net. Details for each module are given in the supplementary figure.

(Szegedy *et al.*, 2016). As stated in 2.5, use of inception-residual benefits training time and allows multi-scale information to be processed at multiple layers, leading to very efficient hierarchical feature representation. In the top-down spatial recovery path, we applied skip connections similar to FCN but with added pyramid dilated deconvolution to further enhance the representation power of multi-scale information. We name this dense CNN architecture as DeepEM3D-Net for the rest of this paper. To address the anisotropy problem in 3D, the networks was designed to have only two 3D convolution layers to integrate 3D information in the early stage and perform 2D convolution for the rest of following layers. This ensures the minimum computation cost while taking advantage of 3D context information. Such early 3D fusion design allows the network to incorporate information in Z-direction but mainly handling information in X-Y directions. In addition, we also derived a few variants of architecture to accept different numbers of image slices in Z-direction. We expected that combining the outputs of these variants can help minimize the adversary effect of noise caused by misalignment and anisotropy in such 3D data.

3 The proposed segmentation pipeline

3.1 Data augmentation

Data augmentation can increase the effective size of training data, alleviating the overfitting problem. It is commonly used in training CNN for image classification tasks. We augmented both training and test data by applying several spatial transformations on the input image. The transformations were combinations of horizontal and vertical mirroring, rotations by +90, -90, and 180 degrees in X-Y plane, and flip/horizontal mirroring in Z-direction, resulting in a total of 16 variants. For training, all 16 transformations were used. Upon testing, after given the 16 transformed data to the networks, we applied reverse transformations to each probability map. We took the average of predictions from each variation as the final output of boundary probability map.

3.2 Model training and ensemble

In our study, we trained 5 dense CNN models with various boundary labels that were described in Section 2.3. These models differ only in the first two layers in order to take in different number of slices in Z-direction. Two models take $h \times w \times 5$ input and were trained with boundary labels $c5$ and $c3$, where h and w denote input size in X and Y-direction, respectively. We used 380×380 for training and 1024×1024 at test time. The other three models take $h \times w \times 1$, $h \times w \times 3$, $h \times w \times 5$ inputs, respectively, and were trained with boundary label $l3$. The models that accepts 3 and 5 slices perform 3D convolution in the first two layers and 2D convolution in the remaining layers. Models that take only 1 slice perform 2D convolution in all layers. During training, we randomly cropped $380 \times 380 \times d$ patches from 16 variants of raw image volumes, where the sizes of d are 1, 3 or 5 depending on the corresponding models. To tackle the problem of imbalanced class between the boundary and non-boundary (neurite), we introduced class weights into the softmax loss layer by adding weights that are inversely proportional to class ratio. Upon testing, we combined these models to improve the final segmentation. In Section 4, we show that optimization of the combination of boundary probability maps from multiple models gives improved 3D EM segmentation.

3.3 Probability map generation

Full image stack was processed in a sliding scheme during test as shown in Figure 2. Input moves 1 slice along the Z-dimension each time and feeds one or several full size neighboring images (1024×1024) to each of the models as described before. The number of slices fed includes 1, 3 or 5 based on the corresponding model configuration. Each model yields 1 slice of probability map spatially aligned with the center input slice. As a result, the entire stack of probability map is produced one by one. In this sliding process, each output slice takes only about 5 seconds to produce on GPU. The entire stack takes less then 10 minutes to be produced. For each CNN model, all 16 variants of raw image stack were given to produce

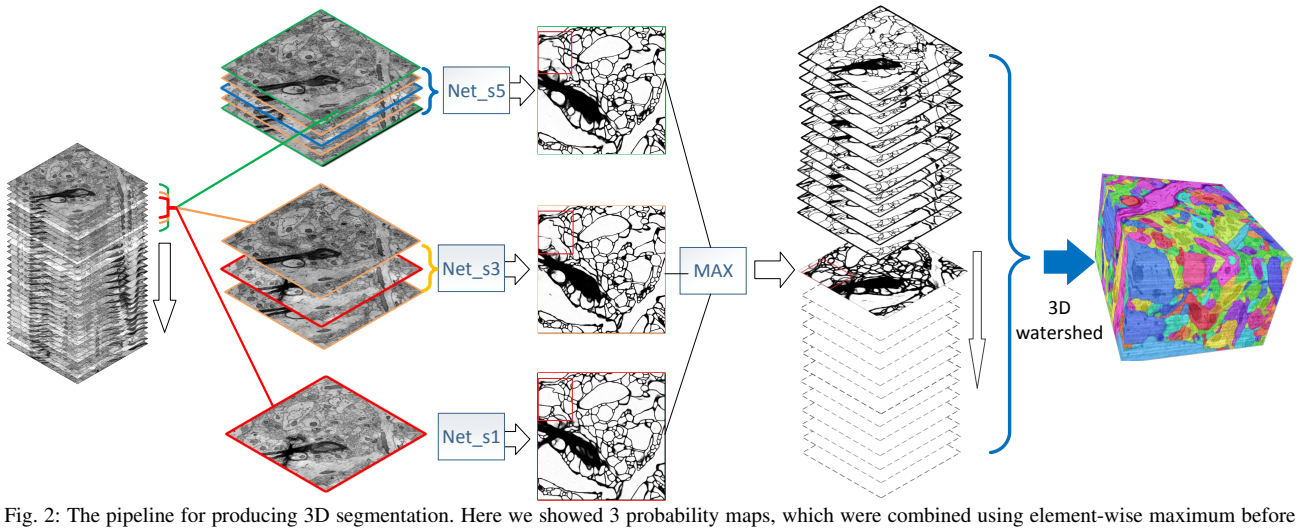


Fig. 2: The pipeline for producing 3D segmentation. Here we showed 3 probability maps, which were combined using element-wise maximum before the watershed algorithm was applied to generate final segmentation. The actual combinations of using multiple outputs in the experiment were described in Section 4.

Table 1. Segmentation performance archived by different deep models. Note that pDeepEM3D-Net uses only the original deconvolution for upsampling.

Architecture	Rand errors
DeepEM3D-Net	0.0912
pDeepEM3D-Net	0.0928
RDN	0.0940

16 prediction stacks, which were reverse-transformed back to raw spatial order for voxel-wise averaging.

3.4 Post-processing

The 5 averaged output probability maps of each network model were further combined together. Since our post-processing is relatively sensitive to false negative error while robust to false positive error of predicted boundary probability map, we combined the 5 probability maps by taking maximum voxel value instead of average as in most ensemble methods. This ensures boundary continuity to the maximum degree and minimizes the probability of broken boundaries.

After obtaining the combined probability map stack, we smoothed the probability maps with a Gaussian kernel of size 6x6 and standard deviation of 1 on each 2D slice. Then, the 3D watershed algorithm was directly applied to the entire smoothed stack to generate the final segmentations. Since we trained the models using boundary labels that are connected in 3 perpendicular directions (Section 2.3), we used 6-connected neighborhood (up-down, left-right, front-back in 3D space) for catchment basin search. Watershed algorithm constructed regions from adjacent catchment basins determined by local minima, which typically lead to over-segmentation. To address this issue, H-minima transform technique was applied to suppress undesired minima to effectively reduce over-segmentation. The H-minima transform is controlled by a designated depth threshold, where small threshold leads to more over-segmentation while large threshold leads to more undesirable merges. To compute an optimal depth threshold, we performed a grid search for this parameter with the training data and empirically fine-tuned it on the test data.

In addition to the benefit of efficiency, the 3D watershed method needs only 1 parameter to be tuned and does not rely on any hand-crafted features. Our models and ensemble approaches are the keys to generate high quality probability maps such that simple post-prossing methods is able to yield high quality segmentation directly without relying on any computationally intensive post-processing methods.

4 Experimental results

4.1 Deep model selection

In order to identify CNN architectures suitable for 3D EM segmentation, we start by evaluating three CNN models. First, we designed the preliminary version of DeepEM3D-Net, termed pDeepEM3D-Net, which applied deconvolution in the upsampling layers, instead of the pyramid dilated deconvolution kernels. We then employed multiple dilated kernels, and this leads to DeepEM3D-Net. These two models were compared with residual deconvolutional networks (RDN) (Fakhry *et al.*, 2016b), which were also designed for EM image segmentation. Ensemble models usually outperform single models. However, fully training of multiple deep models is very time-consuming. So we performed an early stop during training for 15K iterations and evaluated these three models' performance on the test dataset. We used identical post-processing parameters for all models to produce the respective final segmentations. Results in Table 1 show that pDeepEM3D-Net yields better segmentation than RDN. During experiment, we also noticed that the validation dataset accuracy by RDN started to drop after 14K iterations, which is an indication of overfitting. In contrast, pDeepEM3D-Net was able to improve accuracy after 15K iterations. Note that by adding pyramid dilated deconvolution kernels to pDeepEM3D-Net, the DeepEM3D-Net further improved performance. Taken together, we thereby chose DeepEM3D-Net as our base architecture for further experiments. To handle the anisotropic problem, we constructed several variants of DeepEM3D-Net that were slightly modified in order to accept different numbers of slices.

Table 2. Segmentation performance achieved by different combinations of networks. $s\#$ indicates the number of slices that network accepts; $k\#$ denotes the type of boundary labels used for training. "Global boundary" indicates that the boundaries were used for the entire test stack, while "Additional local boundary" indicates the application to partial slices in test stack. More details are given in the texts.

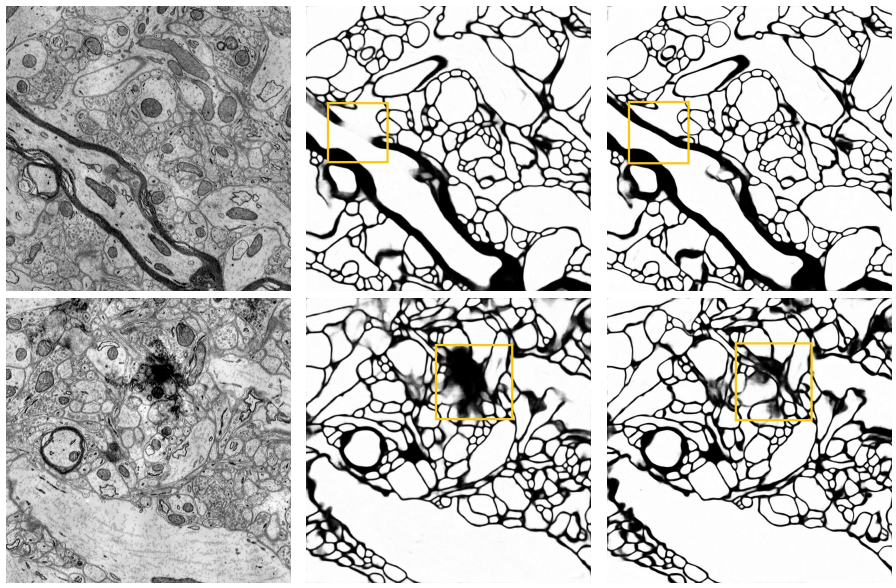
Global boundary	Additional local boundary	Rand error
$s1l3$, $s1c3$, $s1c5$	$s5c3$	0.0601
$s1l3$, $s1c3$, $s1c5$	$s5c5$	0.0648
$s1l3$	$s5c5$	0.0791
$s5c5$, $s1l3$	NA	0.0814
$s5c5$	NA	0.0906

4.2 Approaches to archive human-level performance

To fully evaluate the effects of boundary thickness and usefulness of information provided in Z-direction, we trained 5 variants of DeepEM3D-Net described in Section 2.7. Each DeepEM3D-Net model was trained for 50K iterations, which took about 4-5 days on a single GPU. After close inspection of the dataset, we noticed that the training and test stacks are very different in terms of neurite morphology. Thus fine-tuned thresholds for H-minima and size of gaussian kernel that performed well on training stack do not produce similar performance on the test stack. Therefore, instead of reporting the results on validation data, we report test data results in the following.

The performance of different ensemble strategies is summarize in Table 2. We first used the $s5c5$ model, which accepts a patch of 5 slices and trained with label $c5$, to produce one probability map. This yielded a rand index of 0.094 on the test image stack. We noticed that the predicted probability maps produced by $s5c5$ contain some broken boundaries. From both the raw image stack and the produced probability map, we can see that boundaries in the slices above and below are well-aligned with each other. The middle slice, however, shifted dramatically and is inconsistent with any of its neighboring slices. This indicates the occurrence of misalignment in stacking. Such misalignment appears from time to time, and one example can be seen in Figure 4, which is in contrast to Figure 3. This kind of problem was discussed in Section 3.2. CNN architectures designed to take the advantage of information from multiple slices are also prone to the noise brought about by those adjacent slices. To address this broken boundary problem, we considered using probability maps produced by the $s1l3$ model, which accepts single slice input and produces probability maps that are independent of neighboring slices. Given that the properties of $s1l3$ and $s5c5$ are complimentary in dealing with anisotropy 3D data, we combined these two probability maps by taking voxel-wise maximum. This combination was able to reduce the rand index error to 0.0814.

Despite the fact that thick boundary is beneficial for segmentation in large region as it enhances the boundary continuity, it hampers segmentation in small regions. Given that the above $s5c5$ model was trained on



Raw image $s1l3$ $s1c5$

Fig. 3: Examples of predicted boundary in well-aligned image stack. The first column represents raw image in test stack. The second and third colums are the predictions of $s1l3$ and $s1c5$ CNN models, which take 1 and 5 slice(s) as input, respectively. Note that by using Z-direction context information, $s1c5$ model yields better boundaries than $s1l3$, which uses only X-Y plane context information (yellow box).

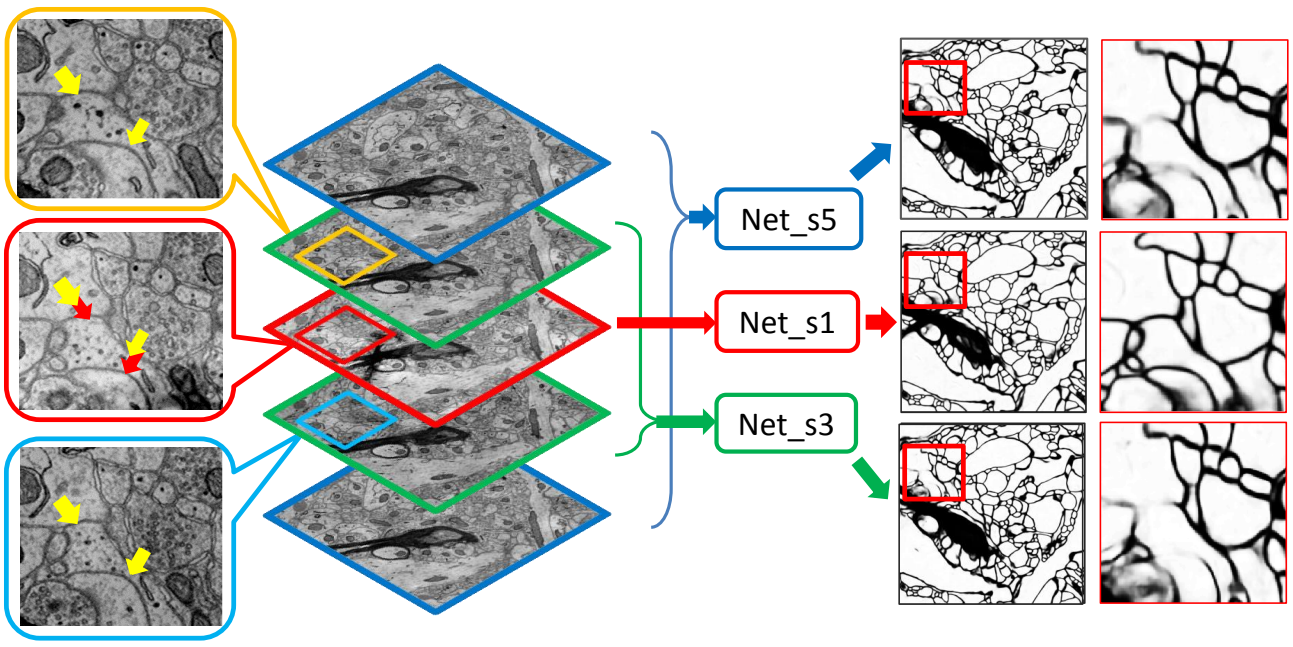


Fig. 4: Examples of boundary predictions in misaligned image stack. First column: zoomed-in raw images corresponding to original image slices in the second column. Red arrows in the middle slice point to shifted boundary from supposedly aligned location indicated by the yellow arrows. Second column: Example of consecutive raw image stacks. Third column: 3 Networks that accept 5, 3, 1 slice(s) as input and produce one slice of prediction corresponding to the central slice. Fourth column: Predicted slice from 3 networks respectively. Fifth column: zoomed-in image from prediction in the fourth column. Note that the network (Net-$s1$) using only one input slice yields better boundary than those use multiple slices.

thick boundaries and consequently produced thick boundaries, we examined the effect of such factor on the test dataset. Specifically, instead of combining $s5c5$ with $s1l3$, we used $s1l3$ throughout the entire stack, but we combined the probability maps of $s5c5$ and $s1l3$ by taking their maximum value in slices that were noticeably misaligned. Such local enhancement is based on the fact that misaligned slices can have large spatial displacement such that even successful boundary prediction in 2D slice can still result in broken boundary in Z-direction. We can see that such combination improved the segmentation and yielded a rand index error of 0.0791.

Next, we added probability maps of $s1c3$ and $s1c5$ globally into the above ensemble configuration. These two probability maps were obtained by using information in the Z-direction slices and trained with thin boundary labels $l3$. This further improved the rand index error to 0.0648. Finally, we managed to improve the performance in small regions while maintaining boundary connectivity in large regions. This was achieved by

replacing thick $c5$ boundary with slightly thinner $c3$ on those misaligned slices. As a result, we achieved a rand index error of 0.06015, which is very close to the human value of 0.05998. The current SNEMI3D challenge leader board as of October 15th, 2016 is given in Table 3.

Table 3. The SNEMI3D challenge leader board as of October 15th, 2016.

Group	Rand Error
human values	0.05998
DIVE (our team)	0.06015
IAL	0.06561
IAL_deprecated	0.08039
Team Gala	0.10041
SCI	0.10829
MIT	0.11361
Princeton-MIT	0.11501
FlyEM	0.12504
rll	0.13111

5 Conclusions

3D EM neurite reconstruction plays an important role in connectome studies. Its key is to accurately segment raw neuronal EM images so each single neurite is delineated by a unique label in 3D space. Generating boundary probability map followed by post-processing is a widely used approach for this reconstruction task. Most traditional post-processing methods are time-consuming and require domain knowledge, which forms a bottleneck for both speed and accuracy in the pipeline. We proposed a novel deep architecture for obtaining very accurate boundary probability maps, thus enabling a simple post-processing paradigm for final segmentation. Our model uses the power of inception and residual structure in the bottom-up path to efficiently integrate image information, and combines techniques of skip connection with pyramid multi-scale contexture aggregation in the top-down path to produce dense prediction. Also, to maximally exploit the advantage of this model, we trained multiple variants of the deep models that accept different numbers of input slices and predict boundary of different thickness. The ensemble strategy for boundary enhancement of probability maps produced by these models is the key to obtain high quality segmentation, as well as to suppress noise in Z-direction alignment.

Altogether, the results are exciting. To the best of our knowledge, our approach is the first automatic 3D neurite reconstruction method that achieved human-level performance. In addition, we believe that the simplified post-prossing approach can generalize well to other similar dense prediction problems. In the future, we plan to apply similar end-to-end learning approaches, such as recurrent convolutional models, to 3D space for such tasks. We expect to completely remove the need for post-processing and further increase the generality of method in 3D neurite reconstruction.

Acknowledgements

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research. This work was supported in part by National Science Foundation grant DBI-1641223.

References

Arganda-Carreras, I., Turaga, S. C., Berger, D. R., Cireşan, D., Giusti, A., Gambardella, L. M., Schmidhuber, J., Laptev, D., Dwivedi, S., Buhmann, J. M., *et al.* (2015). Crowdsourcing the creation of image segmentation algorithms for connectomics. *Frontiers in Neuroanatomy*, **9**.

Berning, M., Boergens, K. M., and Helmstaedter, M. (2015). SegEM: Efficient image analysis for high-resolution connectomics. *Neuron*, **87**(6), 1193–1206.

Briggman, K. L. and Bock, D. D. (2012). Volume electron microscopy for neuronal circuit reconstruction. *Current Opinion in Neurobiology*, **22**(1), 154–161.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2016). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*.

Ciresan, D., Giusti, A., Gambardella, L. M., and Schmidhuber, J. (2012). Deep neural networks segment neuronal membranes in electron microscopy images. In *Advances in neural information processing systems*, pages 2843–2851.

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

Fakhry, A., Zeng, T., and Ji, S. (2016b). Residual deconvolutional networks for brain electron microscopy image segmentation. *IEEE Transactions on Medical Imaging*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Helmstaedter, M., Briggman, K. L., Turaga, S. C., Jain, V., Seung, H. S., and Denk, W. (2013). Connectomic reconstruction of the inner plexiform layer in the mouse retina. *Nature*, **500**(7461), 168–174.

Jain, V., Seung, H. S., and Turaga, S. C. (2010). Machines that learn to segment images: a crucial technology for connectomics. *Current Opinion in Neurobiology*, **20**(5), 653–666.

Kasthuri, N., Hayworth, K. J., Berger, D. R., Schalek, R. L., Conchello, J. A., Knowles-Barley, S., Lee, D., Vázquez-Reina, A., Kaynig, V., Jones, T. R., *et al.* (2015). Saturated reconstruction of a volume of neocortex. *Cell*, **162**(3), 648–661.

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

Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1106–1114.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, **86**(11), 2278–2324.

Lee, K., Zlateski, A., Ashwin, V., and Seung, H. S. (2015). Recursive training of 2d-3d convolutional networks for neuronal boundary prediction. In *Advances in Neural Information Processing Systems*, pages 3573–3581.

Lichtman, J. W. and Denk, W. (2011). The big and the small: challenges of imaging the brain's circuits. *Science*, **334**(6056), 618–623.

Liu, T., Seyedhosseini, M., Ellisman, M., and Tasdizen, T. (2013). Watershed merge forest classification for electron microscopy image stack segmentation. In *2013 IEEE International Conference on Image Processing*, pages 4069–4073. IEEE.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440.

Peng, H., Hawrylycz, M., Roskams, J., Hill, S., Spruston, N., Meijering, E., and Ascoli, G. A. (2015). BigNeuron: large-scale 3D neuron reconstruction from optical microscopy images. *Neuron*, **87**(2), 252–256.

Pinheiro, P. O., Lin, T.-Y., Collobert, R., and Dollár, P. (2016). Learning to refine object segments. *arXiv preprint arXiv:1603.08695*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9.

Szegedy, C., Ioffe, S., and Vanhoucke, V. (2016). Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261*.

Turaga, S. C., Murray, J. F., Jain, V., Roth, F., Helmstaedter, M., Briggman, K., Denk, W., and Seung, H. S. (2010). Convolutional networks can learn to generate affinity graphs for image segmentation. *Neural Computation*, **22**(2), 511–538.

Yu, F. and Koltun, V. (2016). Multi-scale context aggregation by dilated convolutions. In *Proceedings of the International Conference on Learning Representations*.