Residual Deconvolutional Networks for Brain Electron Microscopy Image Segmentation

Ahmed Fakhry, Tao Zeng, and Shuiwang Ji, *Senior Member, IEEE*

Abstract—Accurate reconstruction of anatomical connections between neurons in the brain using electron microscopy (EM) images is considered to be the gold standard for circuit mapping. A key step in obtaining the reconstruction is the ability to automatically segment neurons with a precision close to human-level performance. Despite the recent technical advances in EM image segmentation, most of them rely on hand-crafted features to some extent that are specific to the data, limiting their ability to generalize. Here, we propose a simple yet powerful technique for EM image segmentation that is trained end-to-end and does not rely on prior knowledge of the data. Our proposed residual deconvolutional network consists of two information pathways that capture full-resolution features and contextual information, respectively. We showed that the proposed model is very effective in achieving the conflicting goals in dense output prediction; namely preserving full-resolution predictions and including sufficient contextual information. We applied our method to the ongoing open challenge of 3D neurite segmentation in EM images. Our method achieved one of the top results on this open challenge. We demonstrated the generality of our technique by evaluating it on the 2D neurite segmentation challenge dataset where consistently high performance was obtained. We thus expect our method to generalize well to other dense output prediction problems.

Index Terms—Residual learning, deconvolutional networks, deep learning, image segmentation, electron microscopy, brain circuit reconstruction.

1 Introduction

The automated 3D reconstruction of neurites in brain EM image stacks remains one of the most challenging problems in neuroscience [1], [2], [3]. In such problems, neurons spanning multiple adjacent image slices are expected to be consistently identified and reconstructed. Conventionally, this problem has been approached as a 2D prediction task, where each image slice is segmented individually. Then, a post-processing step was performed to generate 3D segmentation. The post-processing step usually involves heuristic off-the-shelf classifiers that were trained to link similar segments together across the entire image stack. These classifiers usually rely on hand-crafted features which incorporate prior knowledge and understanding of the data. Thus, classifiers that worked well on some problems/datasets are not guaranteed to perform similarly in different scenarios. It is thus desirable to design a fully trainable system with minimal post-processing to perform the 3D segmentation task in an end-to-end fashion.

Currently, deep convolutional neural networks (CNNs) [4] are one of the main tools used for semantic segmentation. These models are very powerful and capable of extracting hierarchical features from raw image data. They are characterized by their ability to learn features directly from the raw images without relying on prior knowledge. CNNs have achieved success in different areas of machine learning and computer vision. Improved performance has been achieved in image classification [5], [6], [7], [8], [9] and object detection tasks [10]. Recently, this success has been extended to dense output prediction problems such as semantic segmentation [11], [12], [13], [14], [15]. These problems find applications in neuroscience of neuronal membrane segmentation in electron microscopy (EM) images [16], [17], [18] and multi-modality infant brain image segmentation [19]. Although deep models are rapidly approaching human-level performance on object recognition tasks, their performance on dense output prediction problems is still far behind human expert performance, especially in brain connectomics involving high-resolution EM image analysis [20], [21], [22], [23], [24], [25], [26], [27], [28].

Several other computational methods have been used to tackle the membrane detection problem from EM images in addition to deep learning techniques. These include hierarchical contextual models [29], cascades of classifiers [30], [31], random forests [32] and biological priors [33]. However, deep learning techniques have been proved to outperform these techniques on several computer vision tasks. Consequently, it is believed that continuously improving the deep learning models is a promising direction for achieving better performance on challenging tasks such as dense output prediction.

In dense output prediction tasks such as EM image segmentation, CNNs are expected to generate pixel-level predictions. That is, each pixel in the input image is given a prediction, resulting in a probability map whose size equals to that of the input image. A common approach to achieve dense prediction is to extract a fixed-sized patch centered on each pixel and employ a regular CNN as used in image classification to determine the label of the center pixel [16], [17]. However, such approaches only incorporate limited contextual information contained in the patch. Contextual information can be increased by enlarging the patch size, but excessively large patches tend to compromise the full-

- *A. Fakhry was with the Department of Computer Science, Old Dominion University, Norfolk, VA 23529. E-mail: afakhry@cs.odu.edu*
- *T. Zeng and S. Ji are with the School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA 99164. E-mails: {tzeng; sji}@eecs.wsu.edu*

Manuscript received ; revised .

resolution, pixel-level predictions. Thus, dense output prediction problems face the conflicting goals of full-resolution prediction and incorporation of sufficient contextual information [34].

In this work, we proposed a simple yet powerful model known as residual deconvolutional network (RDN) to address this challenge. Our proposed model naturally balances the tradeoff between increasing contextual window required for multi-scale reasoning and the ability to preserve pixel-level resolution and accuracy expected for dense output prediction. We achieved these goals by adding multiple residual shortcut paths to a fully deconvolutional network with minimum additional computations. This allows for the training of very deep deconvolutional networks that incorporate sufficient contextual information, and the multi-scale full-resolution features are extracted and provided through the residual paths. The final dense predictions are made by integrating features computed through both pathways, thereby achieving the conflicting goals in dense output prediction in the same framework.

We evaluated our method on the challenging problem of neurite segmentation from 3D EM images, which is a key step in dense brain circuit reconstruction. We participated in the open challenge on 3D EM image segmentation [35], and we achieved the second place among many teams. Note that most of the challenge participants rely on the given probability maps generated by a CNN as inputs to their techniques and focus on creating heuristic post-processing techniques to generate final segmentations. In contrast, we used end-to-end trainable models with minimum post-processing to achieve the top results. Our technique does not rely on prior knowledge of the data. We thus expect our method to generalize well to other dense output prediction problems. We demonstrated this by extending our experiments to the 2D EM image segmentation challenge dataset [36], where consistently high performance was achieved. The results generated by our method can be coupled with any post-processing technique used in the challenge, leading to improved performance.

2 Residual Deconvolutional Networks

Most of the dense prediction methods do not explicitly address the problem of losing pixel-level resolution. This is mainly because most of the CNNs that were used for dense output prediction are variations of the ones that achieved excellent performance on classification and recognition tasks. In those tasks, it is a common approach to reduce the feature map sizes using pooling layers to increase the receptive fields of the resulting feature maps, thereby increasing the contextual window used to generate the single prediction for a given image. When those networks are tailored towards dense prediction, the attempts to reconstruct a full-resolution prediction is hampered by the loss of pixel-specific resolution information.

Fully convolutional networks (FCNs) [13], [37] are efficient approaches to generate dense predictions for image segmentation. The idea is to reconstruct the full-sized input by performing several deconvolution operations at multiple scales through aggregated bilinear interpolation. The segmentation performance of FCNs is limited by the absence of real deconvolution, and full-resolution features are not well preserved. To address this limitation, deconvolution networks [12] have been proposed recently by performing actual deconvolution. The pooling layers are reversed in the decoding stage by unpooling layers which keep track of the maximum activation position selected during the pooling operation. While both of these two approaches are attempts to design novel deep models specifically for dense prediction problems, they do not have explicit mechanisms to address the conflicting goals in dense prediction problems. They still suffer from loss of information due to excessive reduction of resolution as we show in our experiments.

2.1 Residual Deconvolutional Network Model

In the design of our model, we intend to achieve three goals: (1) Generate dense predictions equal in size to any arbitrary-sized input. (2) Increase the receptive fields of output maps to increase the contextual information used to make pixel-level decisions. (3) Achieve pixel-level accuracy by incorporating high resolution feature information.

We build on the deconvolution scheme proposed in [12] to generate dense predictions. We enhance the performance of deconvolution networks by adding residual connections between every several stacks of convolution or deconvolution layers. These shortcut connections perform projection mapping and are added to the output of the stacked layers with minimum additional computation cost. It has been shown in [9] that it is much easier to optimize a residual mapping (with shortcut connections added) rather than the original plain one. Residual networks in [9] also demonstrated a significant performance gain as a result of increased network depth on tasks of image classification and object detection. For our dense prediction network architecture, we propose to introduce projection shortcuts not just on the convolutional stage responsible for extracting the feature representations, but also on the deconvolutional stage responsible for reconstructing the shape and producing the objects segmentation. We believe that with this design, our network is able to acquire more multi-scale contextual information while reducing the effect of the degradation problem [9], [38].

We also propose the use of a novel resolution-preserving path to facilitate the reconstruction of full-resolution output. The resolution-preserving paths are essentially the projection mapping of the pooling layer outputs added to the output of the corresponding deconvolution layer before performing the unpooling operation. These paths are responsible for transferring the missing high resolution information from the encoding stage to the decoding stages. Together, the context-growing and the resolution-preserving paths have significantly boosted the performance of non-residual deconvolutional networks as shown in Section 3. An illustration of the RDN architecture is shown in Figure 1.

2.2 Network Architecture and Training

The RDN architecture is mainly inspired by the ideas in [7], [9], [12] with three main differences:

- The convolution stage of the network has been mirrored in the deconvolution stage to produce dense probability

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMI.2016.2613019, IEEE Transactions on Medical Imaging

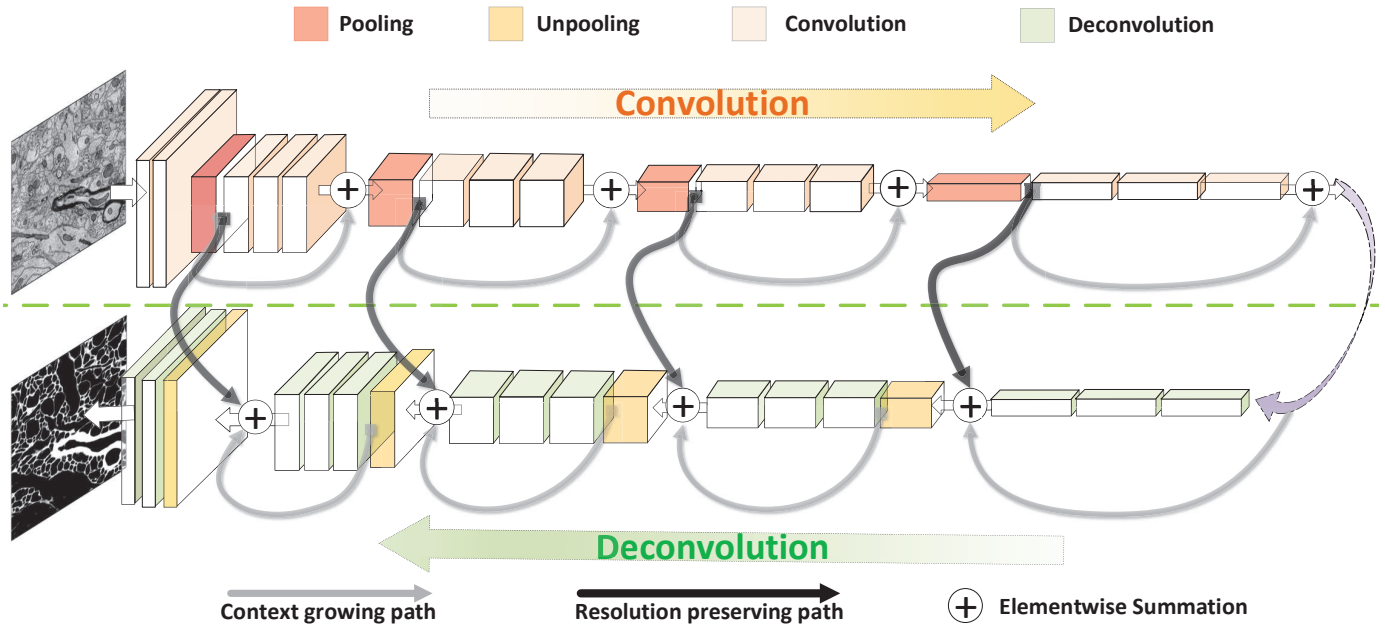


Fig. 1: Architecture of the residual deconvolutional network (RDN). The network consists of two pathways, namely context-growing path and resolution preserving path. All convolution and deconvolution layers in the encoding and decoding stages are of size 3x3. A kernel of size 1x1 is used to implement the projection mappings. Max pooling is used to reduce the feature map sizes in the convolution stage while unpooling is used to restore the original size in the decoding stage.

maps instead of a single value prediction for each training instance. Context-growing paths have been added to the deconvolution as well as the convolution layers. Also, resolution-preserving paths have been added to transfer resolution-specific information from encoding to decoding stages.

- The input to the network is 3D patches extracted from consecutive slices to exploit the 3D aspect of the data in a way similar to how a human annotator perform segmentation. Square patches were extracted randomly from the entire image stack. We performed mirror-padding for patches extracted from the first and last slices to generate the 3D input to our network.
- We attempt to use minimal post-processing that involves handcrafted features throughout the entire pipeline (see Section 2.4).

The network contains 23 convolutional layers and 20 deconvolutional layers in total as the network is not entirely symmetric. The kernel sizes are either 3x3 or 1x1 when we performed branching before adding the residual paths. Zero padding was used whenever size preserving was needed in the learned layers. We added a batch normalization layer after each learned layer and rectified linear units were used as the non-linearity transformation. We used a patch size of 128x128x3 in training while the entire image was used in testing.

No pre-processing was used on the raw input images. However, we modified the training labels to reduce the segment sizes by increasing the border width in-between them (see Figure 2. The widening of borders was done using a minimum kernel of size 5x5 by assuming that all segments are having a label of 1 and borders are having a label of 0 in the ground truth label stack. Any pixel that was in a neighborhood of size 5x5 of a border pixel was considered to be border as well. Label widening was crucial in allowing the network to differentiate border from non-border pixels.

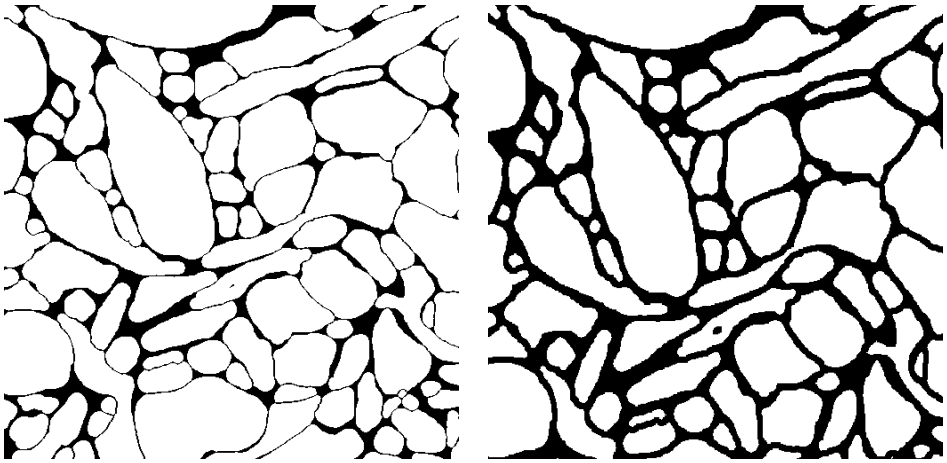
Our model implementation was based on the publicly available C++ Caffe [39]. We trained our RDN using back propagation [4] with stochastic gradient decent. The mini-batch size used was 15 as the dense prediction requires a lot of memory. However, the network requires roughly 15k iterations to achieve its full potential due to the existence of residual paths which speeds up the computations. We used a momentum of 0.9 and weight decay of 0.005. We started with a base learning rate of 10^{-2} with a polynomial decay. Random initialization was used for all learned layers. The experiments were carried out on an NVIDIA K80 GPU machine, taking roughly 2 days of training.

To improve the robustness of the resulting probability maps, we applied 8 variations to the testing images before passing them down through the network. A reverse transformation was then applied to each resulting probability map before taking the average across all variations. The transformations were combinations of horizontal and vertical mirroring, and/or rotations by +90, -90 and 180 degrees.

2.3 EM Image Dense Prediction Problem

In our experiments, we used two separate datasets [40] for training and testing from the ISBI 2013 challenge. Each dataset is a 3D stack of 100 sections from a serial section scanning electron microscopy (ssSEM) of mouse cortex. The pixel resolution is 6x6x30 nm/pixel which covers a microcube of approximately 6x6x3 microns. Both datasets have high x- and y-direction resolution whereas the resolution of z-direction is low. The neurites in the training stack

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMI.2016.2613019, IEEE Transactions on Medical Imaging



Original label image Widened border label image

Fig. 2: An illustration of the effect of border widening on the training labels. We show the original label image (slice 50 of the training stack) on the left and the corresponding altered label on the right. We show that label widening reduces the segments sizes and increases the distance between them.

have been manually delineated, generating a corresponding label stack of 100 sections. The training stack contains 400 neurites that have been labeled consistently across the 100 slices. Some neurites are split into several segments in some slices while still required to preserve their unique label across sections, which increases the complexity of the 3D segmentation task (see Figure 3). The labels of the testing stack are not available to challenge participants.

We formulated the 3D segmentation problem of 400 neurites in the training stack as a single 2D segmentation problem. We built a pixel classifier (Section 2) that accepts patches extracted from the raw input image to generate 2D probability maps. Each resulting probability map indicates the probability of each pixel being either a membrane (border) pixel or non-membrane (neurite). The probability maps have no reference of which neurite a pixel belongs to, had it been identified as a non-membrane pixel. The final 3D segmentation was obtained by a simple post-processing technique described below.

2.4 Post-Processing

Super-pixel level algorithms are commonly used as a building block in most post-processing techniques for 2D and 3D segmentation tasks [11], [41], [42], [43]. They are used mainly to generate an over-segmentation from probability maps or affinity graphs. Later, another classifier is built on top of the results of the super-pixel level algorithms to accurately merge some of the overly segmented regions. The key limitation of these approaches is that they reduce the generality of the overall proposed techniques, since they rely on hand-crafted features to build classifiers on top of super-pixel algorithms. One of the fundamental advantages of the proposed method is the ability to learn features from the data, hence their ability to generalize to many other datasets.

It has been shown before [11] that relying heavily on the learned network while simplifying post-processing could result in a dramatic increase in the speed of computations while maintaining the generalization of the proposed technique. We followed this scheme by applying 3D watershed algorithm directly to the entire probability map stack to generate the final segmentations. The 3D watershed method uses 26-connected neighborhoods to determine the catchment basins in an image. We blurred the probability maps with a Gaussian kernel of size 6x6 and a standard deviation of 1. We also suppressed all minima in the probability maps whose depth were less than a specific threshold. This threshold was mainly used to control the level of over-segmentation and can be tuned using the training data. Our model tends to reduce the predicted segment sizes due to the widening of training label borders described in Section 2.2. As a result, we applied a reverse transformation which used a maximum kernel to increase segments sizes. The overall processing is simple, fast and requires minimum additional computations.

The 3D watershed method does not rely on any hand-crafted features and needs only 1 parameter to be tuned. The quality of the probability maps generated by our RDN is a key for the 3D watershed to be able to generate the final segmentations directly without relying on any additional computations. We demonstrate in Section 3 that its performance on probability maps with lower quality is hampered, making the use of more sophisticated post-processing techniques a necessity.

3 Results

In our experiments, we divided the training stack into 80 slices for training and the rest for validation. We trained our network on the training data for 15K iterations using random sampling of patches. We then evaluated on the validation data. The adapted Rand error metric was used by the ISBI 2013 challenge organizers [35] to assess the segmentation results. The adapted Rand error is defined as: 1 - the maximal F-score of the Rand index (excluding the zero component of the original labels). To evaluate on the testing data, we relied on an automated online system where

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMI.2016.2613019, IEEE Transactions on Medical Imaging

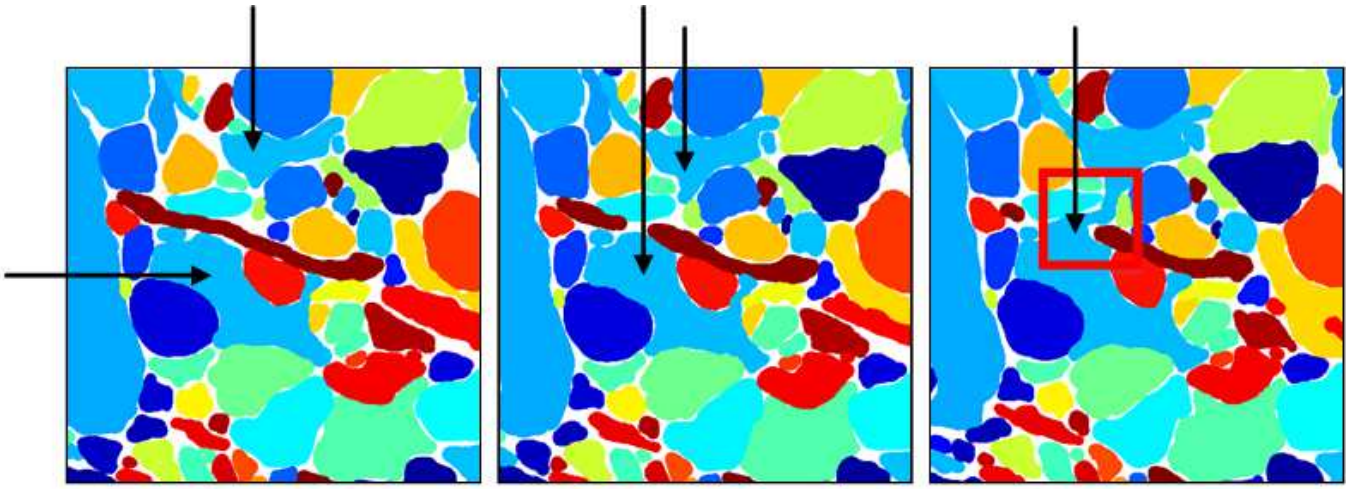


Fig. 3: In this figure, we show the images of the first 3 consecutive slices of the training data cropped at the same position. Segments having the same color across the 3 slices represent the same neurite. The arrows point towards a neurite that has been split in slices 1 and 2 while it appears as a single segment in slice 3. The required segmentation should assign the same label for the splits in slices 1 and 2 even if they are not connected as they belong to the same neurite.

TABLE 1: Comparison between different techniques applied to the validation data. The performance reported is after applying 3D watershed with the best over-segmentation threshold for each set of probability maps independently.

Method	Rand error
RDN	0.0814
IDSIA	0.1184
Deconvolution Network (DN)	0.1514
DIVE CNN	0.1541

our submitted results were compared to the hidden ground truth labels available only to the challenge organizers.

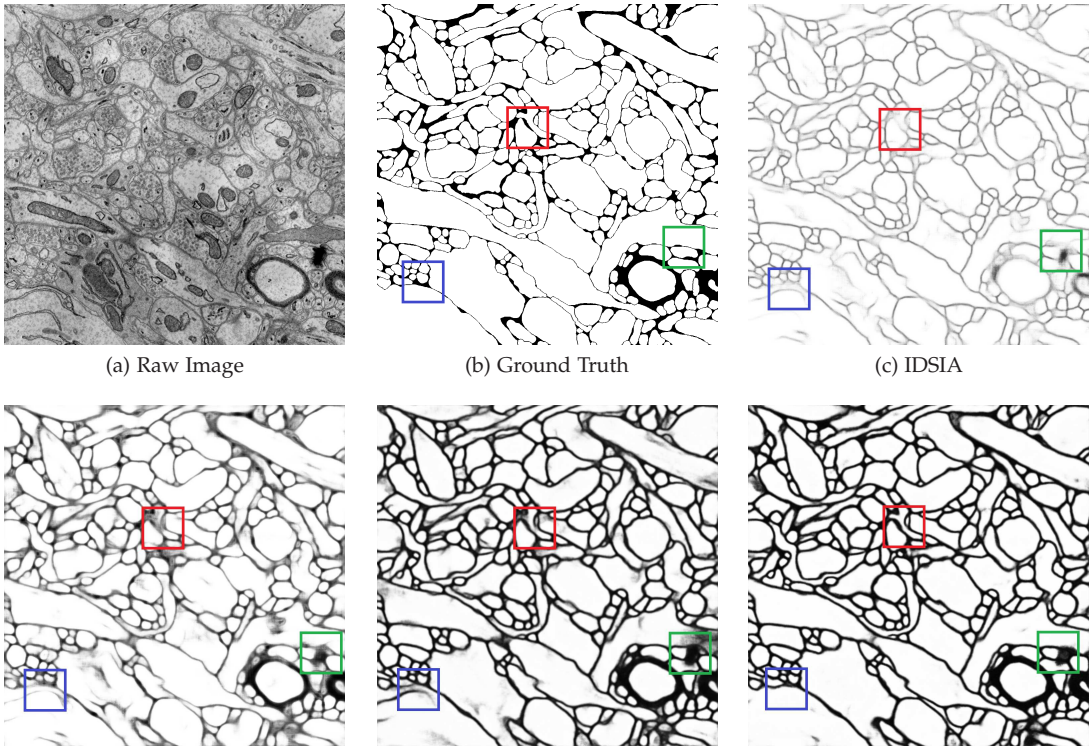
We demonstrated the superiority of our proposed model over other techniques by using the results in Table 1. The results were obtained by applying the 3D watershed method to the validation probability maps as discussed in Section 2.4. We performed grid search to obtain the best parameter to control the over-segmentations for each set of probability maps independently while applying 3D watershed. We compared our proposed RDN to three other techniques:

- IDSIA [16]. Probability maps obtained from training a CNN. These probability maps are provided by the challenge organizers as an optional parameter to use by the participants to evaluate their proposed post-processing techniques.
- DIVE CNN [17]. Probability maps obtained from training a CNN. These probability maps are obtained from one of the leading teams in the ISBI 2012 2D segmentation challenge [36] and we used their proposed model to generate these probability maps for the 3D challenge data.
- Deconvolution Network (DN): Probability maps obtained by training the same exact RDN network without the residual paths.

From Table 1, it is clear that our RDN outperforms all the other CNN-based models by a significant margin. For the IDSIA probability maps, we do not know which slices were used as validation data by the generating team as they provided their probability maps for the entire training stack. However, we assumed that they used the same validation slices as ours (slices 1-20 from the training stack). This assumption is either fair or in favor of the IDSIA probability maps in case the chosen slices were in fact used as training instances by them. Nonetheless, our RDN still achieved a much better segmentation using 3D watershed. A qualitative evaluation of the performance of those models are provided in Figure 4. We showed the probability map generated for the same slice by different networks and we highlighted sample areas of improvement in colored boxes. Our RDN was able to recover most of the missed borders by the CNN trained by IDSIA and DIVE and also improve the certainty of some others. In contrast to DN, our RDN is less sensitive to noise and produces clear probability maps. We compare the final segmentation obtained from our RDN, the IDSIA CNN and the DIVE CNN in Figure 5 after applying 3D watershed. We showed 3 consecutive slices from the validation data where pixels have been consistently given the same color across the 3 slices to denote that they belong to the same neurite. We noticed that the quality of the probability maps generated by RDN has significantly impacted the segmentation results. The CNN-based probability maps result in poor segmentation by either splitting or merging many segments, thus requiring the use of more sophisticated post-processing methods.

We applied our trained model on the testing stack where the labels are hidden and submitted our results to the ISBI 2013 challenge. We achieved the 2nd ranking among many participating teams. Most of the challenge participating teams are working on improving post-processing techniques while relying entirely on the probability maps provided by the IDSIA team. For example, the leading team generated over-segmentations based on the IDSIA probabil-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMI.2016.2613019, IEEE Transactions on Medical Imaging



(a) Raw Image (b) Ground Truth (c) IDSIA

(d) DIVE CNN (e) DN (f) RDN

Fig. 4: Qualitative comparison between the results of several models with respect to training slice number 50. Sample areas with clear differences are marked with colored boxes. A: Raw input image. B: Ground truth 2D label image. C: Probability map generated by team IDSIA with a CNN [16]. D: Probability map generated by team DIVE with a CNN [17]. E: Probability map generated by a deconvolution network (DN). F: Probability map generated by our RDN.

TABLE 2: Comparison between our method and the other techniques in the ISBI 2013 challenge. This ranking is based on the results published on the challenge leaders board on May 19, 2016. We showed only the top 9 teams.

Team	Rand error
IAL	0.07107
DIVE (our team)	0.09104
Team Gala	0.10041
SCI [42]	0.10829
MIT [43]	0.11361
Anonymous	0.11501
FlyEM [44]	0.12504
rll	0.13111
Rhoana	0.14835

ity maps and then built a random forest classifier based on features computed from the over-segmentations. In contrast to these technique, we do not rely on any hand-crafted features throughout the entire processing pipeline and our method is very fast with minimum additional computations. The corresponding team rankings are shown in Table 2. We note that this challenge is an ongoing one and rankings are subject to change as more teams start joining.

By analyzing the provided dataset, we noticed that the testing stack contains segments with much larger sizes than the ones present in the training stack and with a higher frequency. As a result, a regular deconvolution network is not able to recognize those large segments, resulting in their over-segmentations. We highlight the effectiveness of the proposed RDN model in dealing with this problem and the importance of resolution-preserving paths in Figure 6. We trained a RDN model without resolution-preserving paths and evaluated it on the testing stack where this problem occurs. We noticed that without resolution-preserving paths, the network was not able to reconstruct the full-resolution output effectively, resulting in a poor over-segmentation of very large segments. On the other hand, a regular RDN avoided this over-segmentation, thereby confirming its ability to reconstruct full-resolution output by using the resolution-preserving paths.

Evaluation on the ISBI 2012 challenge dataset

To demonstrate the ability of our proposed model to generalize, we extended our experiments to the ISBI 2012 dataset [45]. The dataset consists of a full stack of EM image slices of *Drosophila* first instar larva ventral nerve cord (VNC). The stack contains 30 grayscale sections of 512 by 512 pixels each. In our experiments, we divided this stack to 20 training slices and 10 for validation. We trained the same RDN classifier explained in Section 2.2 with only a few differences:

- 2D patches were extracted instead of 3D. This is mainly due to the extremely low Z-direction resolution for the provided data.
- Only 2D kernels were used through the entire network.

We used 2D watershed as our post-processing with only 8 neighborhood pixels used to determine the catchment

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMI.2016.2613019, IEEE Transactions on Medical Imaging

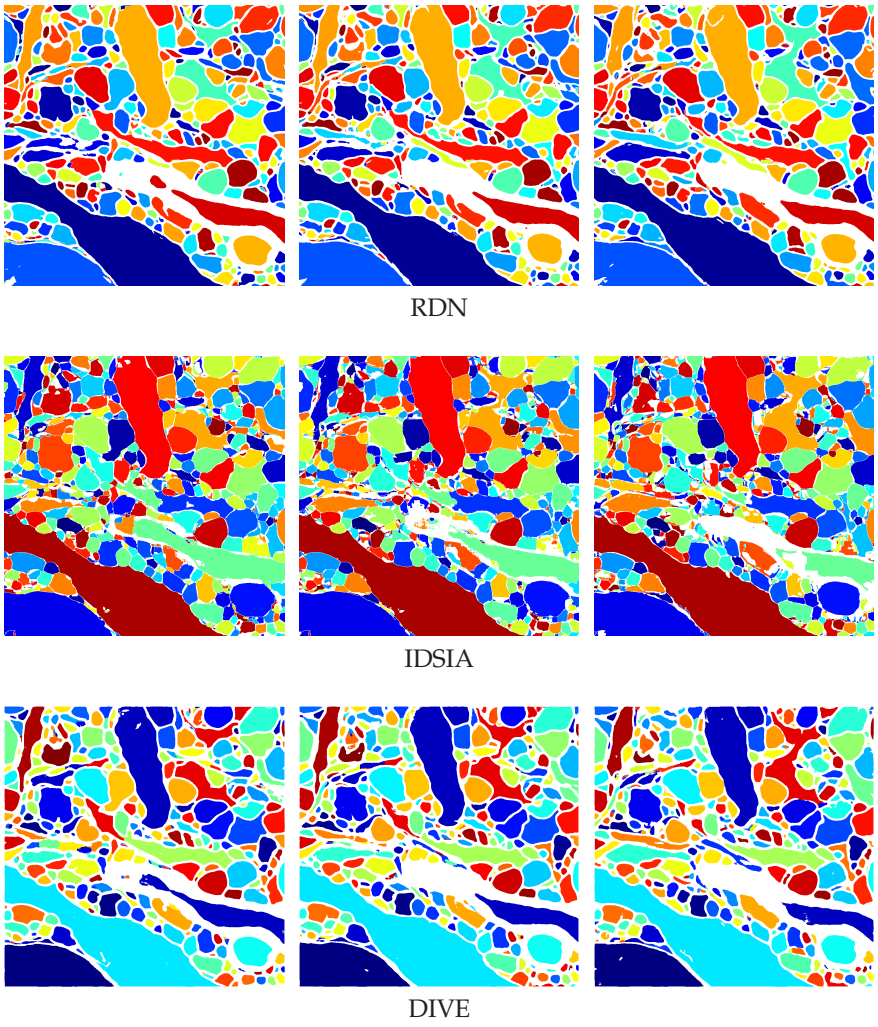


Fig. 5: Comparison between the segmentations obtained by applying the 3D watershed method to our RDN probability maps (top row), the IDSIA probability maps (middle row) and the DIVE probability maps (bottom row). We show 3 consecutive slices from the training stack (slices 2-4) to demonstrate that pixels belonging to the same neurite are segmented consistently (identified by the same color across slices).

basins in the image. Again we did not rely on any problem-specific post-processing technique to ensure the generalization of our technique. We compared our results with a CNN-based classifier trained by the DIVE team [17] participating in the ISBI 2012 EM segmentation challenge [36]. They used an advanced post-processing technique [42], [46], where a random forest classifier is built on top of super pixels output followed by building a Merge Tree (MT). Unlike our technique, the features extracted for the random forest classifier in the MT are generated based on prior knowledge of the data.

We used three common metrics to evaluate the segmentations generated:

Minimum Splits and Mergers Warping error is a segmentation metric that penalizes topological disagreements, i.e: the number of splits and mergers required to obtain the desired segmentation.

Foreground-restric Rand error is defined as 1 - the maximal F-score of the foreground-restricted Rand index, a measure of similarity between two segmentations.

Pixel error is defined as 1 - the maximal F-score of pixel similarity, or squared Euclidean distance between the original and the result labels.

We compared the results of our RDN followed by 2D watershed to the DIVE CNN followed by Merge Tree in Table 3. The results are obtained from evaluating on the validation data (slices 21-30). Our RDN with a general post-processing technique clearly outperforms its CNN counterpart across all evaluation metrics. We note that the Rand error is believed to be the most suitable metric to evaluate semantic segmentations as it penalizes over and under segmentations of objects instead of pixel mispredictions. The improvement obtained from using our RDN is mainly demonstrated in the improved Rand error value. We provided a qualitative comparison between both techniques in Figure 7.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMI.2016.2613019, IEEE Transactions on Medical Imaging

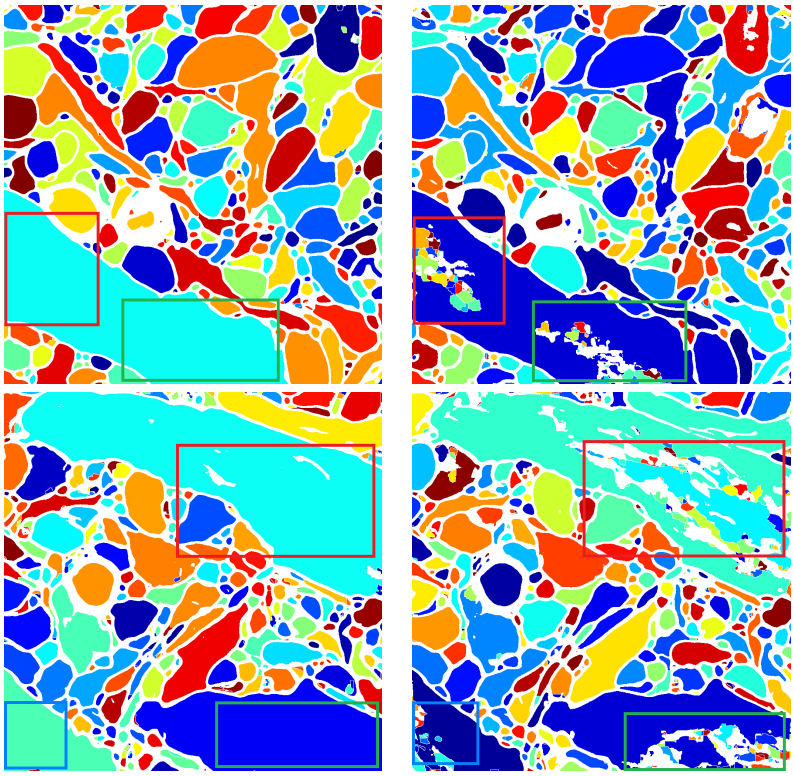


Fig. 6: An illustration of the effect of resolution-preserving paths on the final segmentation. We show the results generated for testing slices number 45 (top row) and 82 (bottom row) by our RDN with and without resolution-preserving paths on the left and right respectively. Colored boxed have been placed on the compared segments.

TABLE 3: Comparison between our RDN and the DIVE CNN segmentations on the ISBI 2012 challenge validation set.

Method	Rand error	Warping error	Pixel error
RDN	0.0282	0.0026	0.0937
DIVE CNN	0.0388	0.0029	0.0939

4 Conclusion

We proposed a computational technique for EM image segmentation by obtaining dense predictions that combined multi-scale contextual reasoning along with full-resolution reconstruction. Our approach achieved promising performance while relying on minimum post-processing. We expect better probability maps be generated with improvement in the z-dimension resolution of the data provided. A limitation in the underlying post-processing techniques is that it requires a specific parameter to control the level of over/under segmentation. Automatic tuning of this parameter is not straightforward and can be data-specific even if it is tuned on the validation dataset. We used semi-automated visualization of the segmentations to overcome this limitation. Nonetheless, our method can be paired with any other post-processing techniques, leading to an overall performance improvement. We did not use hand-crafted features either in the network training or post-processing stages. Consequently, we demonstrated the ability of this model to generalize by applying it to multiple datasets obtained from different species. Our method achieved consistently promising performance. We believe this method can generalize well to other similar dense output prediction tasks.

Acknowledgments

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

References

[1] M. Helmstaedter and P. P. Mitra, "Computational methods and challenges for large-scale circuit mapping," *Current Opinion in Neurobiology*, vol. 22, no. 1, pp. 162–169, 2012.

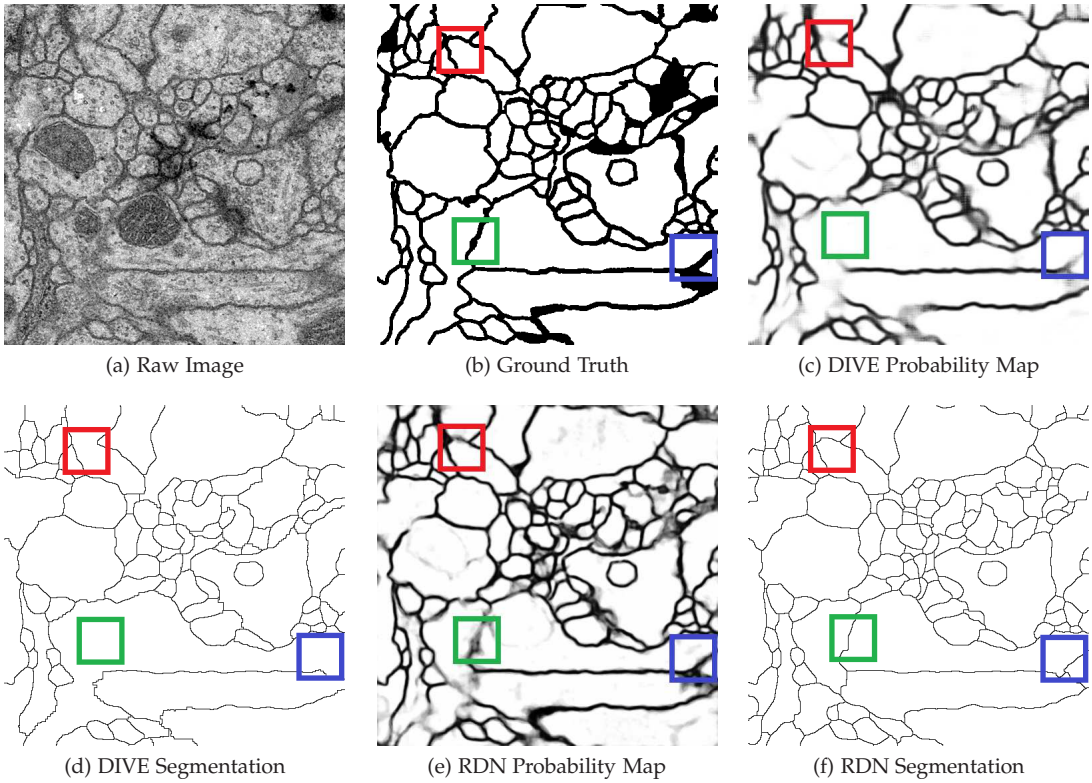
[2] M. Helmstaedter, "Cellular-resolution connectomics: challenges of dense neural circuit reconstruction," *Nature Methods*, vol. 10, no. 6, pp. 501–507, 2013.

[3] 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*, vol. 22, no. 1, pp. 77–88, 2015.

[4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Neural Information Processing Systems*, 2012, pp. 1097–1105.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMI.2016.2613019, IEEE Transactions on Medical Imaging



(a) Raw Image (b) Ground Truth (c) DIVE Probability Map

(d) DIVE Segmentation (e) RDN Probability Map (f) RDN Segmentation

Fig. 7: Qualitative comparison between the probability maps and segmentations obtained from our RDN and the DIVE CNN based on slice 21 from the training stack. Sample areas with clear differences are marked with colored boxes. A: Raw input image. B: Ground truth 2D label image. C: Probability map generated by team DIVE with a CNN. D: Segmentation generated by team DIVE with a CNN and MT. E: Probability map generated by our RDN. F: Segmentation generated by our RDN and watershed.

[6] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision*. Springer, 2014, pp. 818–833.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations*, 2015.

[8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[10] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: Integrated recognition, localization and detection using convolutional networks," in *Proceedings of the International Conference on Learning Representations*, April 2014.

[11] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1915–1929, 2013.

[12] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1520–1528.

[13] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[14] H. Su, F. Xing, X. Kong, Y. Xie, S. Zhang, and L. Yang, "Robust cell detection and segmentation in histopathological images using sparse reconstruction and stacked denoising autoencoders," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 383–390.

[15] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer Assisted Intervention*. Springer, 2015, pp. 234–241.

[16] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," in *Neural Information Processing Systems*, 2012, pp. 2843–2851.

[17] A. Fakhry, H. Peng, and S. Ji, "Deep models for brain EM image segmentation: novel insights and improved performance," *Bioinformatics*, vol. 32, pp. 2352–2358, 2016.

[18] H. Chen, X. J. Qi, J. Z. Cheng, and P. A. Heng, "Deep contextual networks for neuronal structure segmentation," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[19] W. Zhang, R. Li, H. Deng, L. Wang, W. Lin, S. Ji, and D. Shen, "Deep convolutional neural networks for multi-modality isointense infant brain image segmentation," *NeuroImage*, vol. 108, pp. 214–224, 2015.

[20] J. S. Kim, M. J. Greene, A. Zlateski, K. Lee, M. Richardson, S. C. Turaga, M. Purcaro, M. Balkam, A. Robinson, B. F. Behabadi *et al.*, "Space-time wiring specificity supports direction selectivity in the retina," *Nature*, vol. 509, no. 7500, pp. 331–336, 2014.

[21] V. Jain, J. F. Murray, F. Roth, S. Turaga, V. Zhigulin, K. L. Briggman, M. N. Helmstaedter, W. Denk, and H. S. Seung, "Supervised learning of image restoration with convolutional networks," in *Proceedings of the IEEE 11th International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.

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

[23] M. Helmstaedter, K. L. Briggman, S. C. Turaga, V. Jain, H. S. Seung, and W. Denk, "Connectomic reconstruction of the inner plexiform layer in the mouse retina," *Nature*, vol. 500, no. 7461, pp. 168–174, 2013.

[24] K. Lee, A. Zlateski, V. Ashwin, and H. S. Seung, "Recursive training of 2D-3D convolutional networks for neuronal boundary prediction," in *Neural Information Processing Systems*, 2015, pp. 3559–3567.

[25] M. F. Stollenga, W. Byeon, M. Liwicki, and J. Schmidhuber, "Parallel multi-dimensional LSTM, with application to fast biomedical volumetric image segmentation," in *Neural Information Processing Systems*, 2015, pp. 2980–2988.

[26] S. C. Turaga, J. F. Murray, V. Jain, F. Roth, M. Helmstaedter, K. Briggman, W. Denk, and H. S. Seung, "Convolutional networks can learn to generate affinity graphs for image segmentation," *Neural Computation*, vol. 22, no. 2, pp. 511–538, 2010.

[27] M. Berning, K. M. Boergens, and M. Helmstaedter, "SegEM: Efficient image analysis for high-resolution connectomics," *Neuron*, vol. 87, no. 6, pp. 1193–1206, 2015.

[28] T. Hu, J. Nunez-Iglesias, S. Vitaladevuni, L. Scheffer, S. Xu, M. Bolorizadeh, H. Hess, R. Fetter, and D. B. Chklovskii, "Electron microscopy reconstruction of brain structure using sparse representations over learned dictionaries," *IEEE Transactions on Medical Imaging*, vol. 32, no. 12, pp. 2179–2188, 2013.

[29] M. Seyedhosseini, M. Sajjadi, and T. Tasdizen, "Image segmentation with cascaded hierarchical models and logistic disjunctive normal networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2168–2175.

[30] E. Jurrus, A. R. Paiva, S. Watanabe, J. R. Anderson, B. W. Jones, R. T. Whitaker, E. M. Jorgensen, R. E. Marc, and T. Tasdizen, "Detection of neuron membranes in electron microscopy images using a serial neural network architecture," *Medical image analysis*, vol. 14, no. 6, pp. 770–783, 2010.

[31] 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 *International Conference on Medical Image Computing and Computer-assisted Intervention*. Springer, 2011, pp. 670–677.

[32] C. Sommer, C. Straehle, U. Köthe, and F. A. Hamprecht, "Ilastik: Interactive learning and segmentation toolkit," in *2011 IEEE international symposium on biomedical imaging: From nano to macro*. IEEE, 2011, pp. 230–233.

[33] N. Krasowski, T. Beier, G. Knott, U. Koethe, F. Hamprecht, and A. Kreshuk, "Improving 3d em data segmentation by joint optimization over boundary evidence and biological priors," in *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*. IEEE, 2015, pp. 536–539.

[34] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proceedings of the International Conference on Learning Representations*, 2016.

[35] ISBI, "3d segmentation of neurites in EM images challenge - ISBI 2013." 2013.

[36] ——, "Segmentation of neuronal structures in EM stacks challenge - ISBI 2012." 2012.

[37] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," in *Proceedings of the International Conference on Learning Representations*, 2015.

[38] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," *arXiv preprint arXiv:1505.00387*, 2015.

[39] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[40] N. Kasthuri, K. J. Hayworth, D. R. Berger, R. L. Schalek, J. A. Conchello, S. Knowles-Barley, D. Lee, A. Vázquez-Reina, V. Kaynig, T. R. Jones *et al.*, "Saturated reconstruction of a volume of neocortex," *Cell*, vol. 162, no. 3, pp. 648–661, 2015.

[41] V. Jain, S. C. Turaga, K. Briggman, M. N. Helmstaedter, W. Denk, and H. S. Seung, "Learning to agglomerate superpixel hierarchies," in *Neural Information Processing Systems*, 2011, pp. 648–656.

[42] T. Liu, M. Seyedhosseini, M. Ellisman, and T. Tasdizen, "Watershed merge forest classification for electron microscopy image stack segmentation," in *International Conference on Computer Vision*, vol. 2013, 2013, p. 4069.

[43] A. Zlateski and H. S. Seung, "Image segmentation by size-dependent single linkage clustering of a watershed basin graph," *arXiv preprint arXiv:1505.00249*, 2015.

[44] J. Nunez-Iglesias, R. Kennedy, T. Parag, J. Shi, and D. B. Chklovskii, "Machine learning of hierarchical clustering to segment 2D and 3D images," *PloS ONE*, vol. 8, no. 8, p. e71715, 2013.

[45] 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*, vol. 8, no. 10, p. e1000502, 2010.

[46] T. Liu, E. Jurrus, M. Seyedhosseini, M. Ellisman, and T. Tasdizen, "Watershed merge tree classification for electron microscopy image segmentation," in *International Conference on Pattern Recognition*. IEEE, 2012, pp. 133–137.