Deep Convolutional Encoder-Decoder for Myelin and Axon Segmentation

Rassoul Mesbah
Department of Computer Science
University of Otago, New Zealand
Email: rassoul@cs.otago.ac.nz

Brendan McCane
Department of Computer Science
University of Otago, New Zealand
Email: mccane@cs.otago.ac.nz

Steven Mills
Department of Computer Science
University of Otago, New Zealand
Email: steven@cs.otago.ac.nz

***Abstract*—We propose a fully automatic method for segmenting myelin and axon from microscopy images of excised mouse spinal cord based on Convolutional Neural Networks (CNNs) and Deep Convolutional Encoder-Decoder. We compare a two-class CNN, multi-class CNN, and multi-class deep convolutional encoder-decoder with traditional methods. The CNN method gives a pixel-wise accuracy of 79.7% whereas an Active Contour method gives 59.4%. The encoder-decoder shows better performance with 82.3% and noticeably shorter classification time than CNN methods.**

I. Introduction

Segmentation is a common step in medical image processing, and in this work we are interested in identifying myelin in microscope images of the spinal cord. Myelin acts as an insulating material to increase signal transmission speed in mammals [9]. The loss of the myelin sheath is a common symptom of several auto-immune diseases such as multiple sclerosis, leukodystrophy, Charcot-Marie-Tooth, and so on [8]. Moreover, there are other applications for this segmentation task such as studying brain connectivity and maturation [19].

Fig. 1 shows a sample microscopy image of an excised mouse spinal cord and the corresponding labelled ground truth. Myelin is visible as darker material surrounding the brighter axons. However, there are areas with the same intensity as myelin that are axon or background and areas with the same intensity as axon that are usually background. This makes automatic extraction of the myelin challenging.

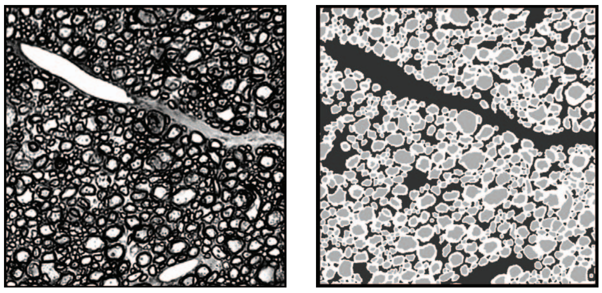
(a) (b)

Fig. 1. (a) Microscopy image of the excised mouse spinal cord ; (b) Ground truth: areas with white colour correspond to myelin regions and the grey regions represent axon.

II. Medical Image Segmentation

There are many techniques which have been proposed for similar applications which can be roughly divided into three categories: those based on geometry or shape such as active contours and Hough transform, intensity based approaches such as histogram, morphology, graph-cut, and those employing learning based approaches such as cascade classifier and support vector machines.

Active contour model [11] is based on fitting *snakes* (connected curves) to nerve boundaries, image energy derived from the features in the frame, and constraint energy that ensures smoothness of the boundary. Active shape model techniques [7] employ variations in statistical information (mostly geometric characteristics) in features to detect shapes in the image. The Hough transform [5] can detect curves and is applicable for nerve boundary detection in our microscopy images. This technique and also similar methods such as fitting ellipses to scattered point data [6] are expensive to compute since Hough transform uses a 5-dimensional parameter space. Moreover, the main problems for using these methods are the very large number of axons per image.

MacDonell [14] employed an active contour method based on two different parameter optimisation techniques using ground truth and raw images for tuning the parameters. He also performed an experiment based on the combination of Canny edge detection and Otsu's methods with active contours. The combination of Canny edge detection with active contours achieved 59.2% accuracy in pixel-wise segmentation task.

Methods based on intensity such as Otsu's analysis [15], [23], multilevel Fukunaga [20] and Renyi entropy [21] segment the image by classifying the pixels into two or more intensity classes. Using intensity as the only feature for segmentation is not very precise as there are some regions with the same intensity as the myelin which are not myelin. Using morphological operations can reduce this error to some degree but cannot eliminate it.

A Haar cascaded classifier [10] is comprised of several simple features placed in a hierarchical topology. This architecture (decision tree) can be used for rapid object detection tasks. Han *et al.* [10] used Haar cascaded classifiers for radicular cysts and odontogenic keratocystic epithelia classification which is roughly similar to our myelin and axon segmentation problem. The accuracy of the classifier is very dependent on the variance of the shape and orientation of cells from the expected norm.

Other work on the myelin segmentation problem includes a semi-automatic method for axon and myelin segmentation based on the combination of thresholding techniques and a series of morphological operations proposed by More *et al.* [16]. The authors claimed that their method achieved 84.3% accuracy in axon identification (not pixel-wise segmentation). The method, however, is dependent on manual supervision for correction of misclassified regions.

These methods mostly have one thing in common: they all work based on generic features (apart from cascade classifiers in which the features to use are chosen from a very large set and therefore the process is similar to learning features). Convolutional neural networks (CNNs) offer the potential to learn specific features for this task, improving segmentation performance. To the best of our knowledge, CNNs have not been used for myelin and axon segmentation before.

The core aspect of CNN is that the features are not defined by the human but learnt automatically. They are characterised by weight sharing (replication), convolution, and subsampling. The convolution of 2-dimensional masks (filters) are shifted step by step across a 2-dimensional array of input values, which may be the image to be processed or the output of an earlier layer of the network. Each layer maps the inputs from the previous layer to the higher-level feature space by a downsampling unit (pooling) that reduces the resolution in order to reduce the network's sensitivity to small variations. The output layer is usually a classifier layer and fully connected. CNNs can find objects in the frame regardless of location and scale [22].

This location and scale invariance is due to the weight sharing and downsampling (pooling) function which makes the output less sensitive to the exact location of the objects including their boundaries. As a result, there is usually a trade-off between the object detection accuracy and localisation. Increasing the depth of the architecture can improve the former while weakening the latter.

Lots of research has been done in order to overcome the trade-off between accurate object recognition and localisation with CNNs. Chen *et al.* [4] propose a method in which they combine CRF and deep CNN (DCNN) architecture for dealing with semantic pixel-wise segmentation problems. The idea behind their algorithm is that adding the conditional random field to the final layer of DCNN helps utilise global information in the classification task and also avoids premature decision-making during the training phase of deep convolutional neural netwroks. Liu *et al.* [12] trained their CRF using CNN feature maps in order to apply local dependencies of the higher level features to the segmentation algorithm. Post-processing based on conditional random fields shows good performance dealing with more dominant classes such as salient objects and background rather than small structures with irregular label frequencies.

Long *et al.* [13] introduced the idea of fully convolutional networks (FCNs) for semantic segmentation in which the output of each fully convolutional module can be mapped onto the bigger frame than the input window(s). The interpolation function is to be learned during training. They devised the unpooling layer as an up-sampling function using the pooling indices of the corresponding MaxPooling layer since it helps to achieve higher precision in the reconstruction of the feature maps. The authors replaced the CNN's fully connected classifier by multiple unpoolings uses the corresponding convolutional layer's feature maps as input and each followed by an up-convolution layer. The fully convolutional layers' outputs are all concatenated in one mask to feed into a softmax classifier.

Fully convolutional module classifies all pixels in the frame simultaneously. Helped by more information about the neighbouring output pixels during the training, FCN shows higher accuracy in semantic pixel-wise segmentation task than similar CNN architectures. Despite the improvement in accuracy and overall classification time, FCN suffers from two main drawbacks. High-resolution structures can be neglected in the feature reconstruction due to the fact that fully convolutional procedure is not capable of estimating very complicated functions. Another shortcoming (which is due to the FCN's fixed-size input frame) is that some pixels of the salient objects or the objects which are relatively smaller than the frame size can be mislabelled [17].

Brosch *et al.* [3] propose encoder-decoder architecture in which they use ground truth image as output reference. Encoder builds a hierarchy of features in which the highest level features can be extracted in the last layer. These low-resolution features are appropriate for object detection task while they include not much information about location. In order to reconstruct the input image (in the absence of noise), the decoder architecture is used to up-sample the feature maps and use a convolutional filter to learn the lower level features with higher resolution. Brosch *et al.* [3] also propose an optimised loss function in order to deal with unbalanced class distribution problem. Although this feed-forward architecture with single-step training phase benefits from denoising and feature reconstruction properties, it still suffers from lack of accuracy in detecting high resolution structures.

In order to address this problem, Badrinarayanan *et al.* [1] and Noh *et al.* [18] employed an encoder-decoder architecture in which they placed an up-convolution layer between encoder and decoder units. They used unpooling function in the decoder architecture, each unpooling followed by an up-convolution layer. Then, Badrinarayanan *et al.* [2] proposed a more optimised architecture regarding memory and computations. In this encoder-decoder architecture, the up-convolution layers in the decoder unit are replaced by convolutional layers while the up-convolution layer between the encoder and decoder units is preserved as an optional module. The authors claimed that they propose a architecture which can be used for a variety of semantic pixel-wise segmentation tasks.

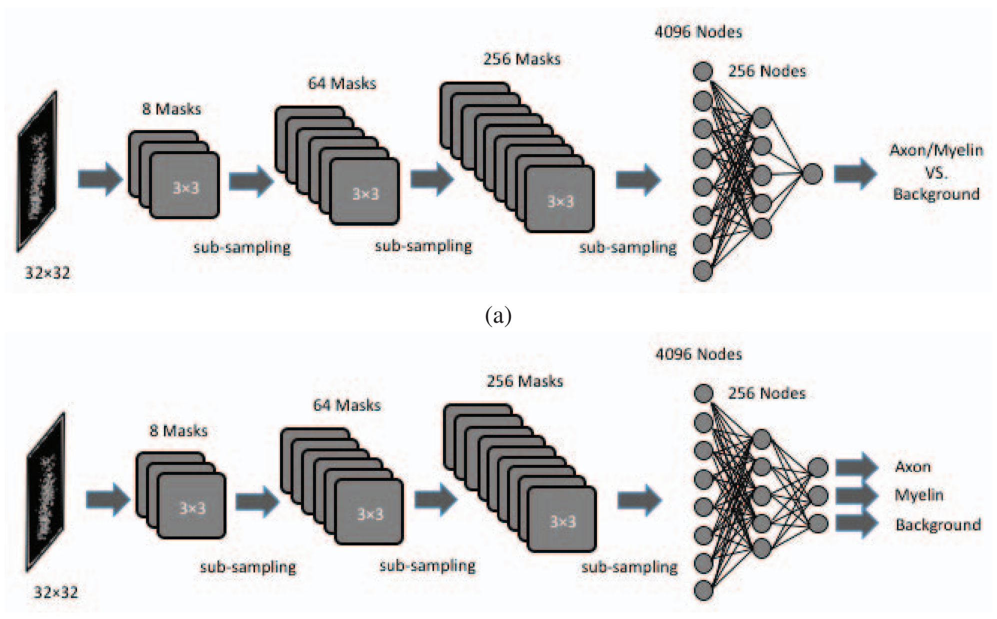

Fig. 2. Our convolutional neural networks' architecture: (a) Two-class CNN; (b) Three-class CNN

III. Method

We employ two different feed-forward convolutional network architectures: CNN and convolutional encoder-decoder in which the former designed to classify one pixel of 32×32 input frame while the latter aims the whole frame simultaneously. The deep convolutional encoder-decoder implementation is inspired by the architecture proposed by Badrinarayanan *et al.* [2] and based on our three-class CNN architecture.

A. Convolutional Neural Network Design

Based on the classification accuracies during experiments, we have chosen our CNN to be comprised of 3 convolutional layers (CL). Each CL is followed by a sub-sampling layer. We used a fully connected architecture with one hidden layer as our classifier. The non-linear activation function is chosen to be hyperbolic tangent.

We performed experiments based on different architectures for our CNN. We tested our network with two and three convolutional layers, 4, 8, 9, and 12 features in our first layer, 16, 64, 81, and 144 features in the second convolutional layer, and 256, 512, and 1024 in the third layer of convolution. We also used variety of training iterations up to 150.

The most accurate architecture among the CNNs been tested during our experiments contains eight 3×3 convolutional filters at the first layer. Using MaxPooling function as a sub-sampling layer, the dimensions of the feature maps reduce to 16×16 as the input is a 32×32 pixel greyscale image. The next CL consists of sixty four 3×3 convolutional filters, each is connected to the previous feature maps. The same downsampling function is used to reduce the size of the feature maps to 8×8. We employed 256 feature masks (3×3) for our third CL, each is connected to all second layer's masks. The output is 256 feature maps with a size of 4×4 pixels which represent the higher level features.

We have experimented with two variants of the CNN architecture. The first uses two two-class CNNs (myelin or not myelin, and axon or not axon); and the second uses a single CNN for all three classes (myelin, axon, background). The fully connected layer has $256 \times 4 \times 4$ input nodes fully connected to the 256 nodes in the hidden layer. The number of output node(s) is one for two-class CNN, and three in multi-class CNN (Fig. 2)

B. Deep Convolutional Encoder-Decoder Design

The encoder unit is similar to our three-class CNN architecture excluding the fully connected layers. Decoder contains three units each consisting of an unpooling followed by a convolutional layer in which the unpooling layers up-sample the feature maps using the corresponding encoder's MaxPooling indices.

All the convolutional masks' sizes are chosen to be 3×3. We employed 256 convolutional masks in the first decoder unit, 64 and 8 masks in the second and third layers, respectively. Three 32×32 feature maps in the last layer of our decoder unit are fed into a SoftMax classifier to be able to have one correspondingly-sized output image.

The convolutional layer that connects encoder and decoder units has 256 masks each of size 1×1. All the activation functions are chosen to be rectified linear units (ReLU) in

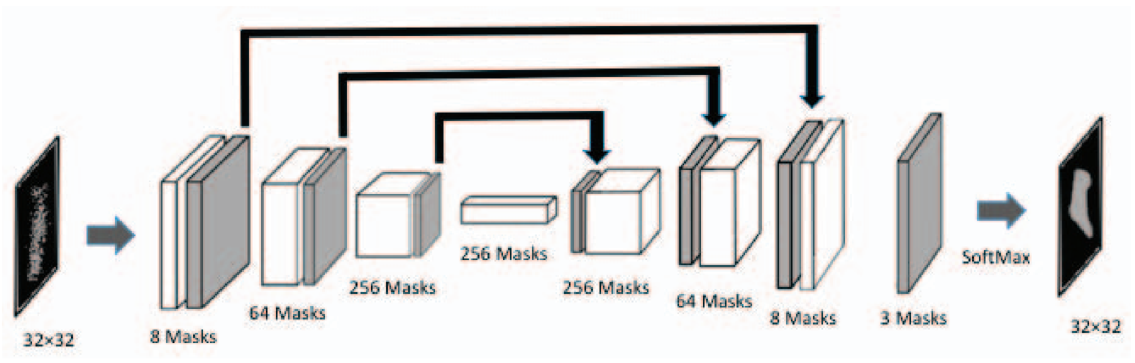

Fig. 3. Our Deep Convolutional Encoder-Decoder's architecture

TABLE I
THE DISTRIBUTIONS OF OUR DATASETS

Dataset	Axon	Myelin	Background
Encoder-Decoder	24.8%	43.5%	31.7%
CNN: Three-class	24.8%	43.5%	31.7%
CNN: Two-class 1	24.8%	-	75.2%
CNN: Two-class 2	-	43.5%	56.5%

order to make the reconstruction of the feature maps theoretically possible [13]. Fig. 3 represents our deep convolutional encoder-decoder's architecture.

C. Training and Experimental Design

We used 35 microscopy images of the excised mouse spinal cord as our dataset. The images in this data set are 700×700 pixels, and have been previously used by MacDonell [14]. We performed 35 trials, one with each image excluded and the others used for training. To create the training set, we randomly sampled the remaining 34 images using a 32×32 window size for 3,000 samples from each image.

At runtime, for the CNN architectures, a 32×32 window is shifted across the image with one pixel shifts, thus creating 447,561 classification tasks for each image (only the pixel at 16,16 is classified). For the encoder-decoder architecture, each pixel within a 32×32 sub-window is classified simultaneously, and therefore only 441 classification tasks are needed to cover the entire image. The label to be learned by CNN for each 32×32 image is the label of the pixel at coordinate (16, 16).

The output of the encoder-decoder is a three dimensional matrix of size $3 \times 32 \times 32$. The first dimension is a one-hot encoding of the class label (either 1 or 0 in each element, with only one element equal to one). Table 1 shows the distribution of our dataset.

We have used automatic stochastic gradient descent, to avoid overfitting to the training data [22]. We observed the error during the training process of our CNNs, and found that there is a minimum error at the fourth or fifth iteration, for this specific problem. As the majority of our CNNs have the minimum training error at the fifth iteration, we have chosen the maximum number of iterations to be five. In the same way, 25 iterations are chosen for the encoder-decoder training phase.

The Torch7 library was used for all implementation (`www.torch.ch`). All experiments were performed on an iMac with a 3.2 GHz Core i5 quad-core processor for our implementations. Each CNN training iteration took approximately 7 minutes to run; and, consequently, the training time (including 5 iterations) for each module was about 35 minutes. The total encoder-decoder training time (including 25 iterations) was nearly 375 minutes since each iteration over the dataset took roughly 15 minutes.

IV. Results & Discussion

MacDonell [14] includes the most recent and accurate results for automatic segmentation of this data set. In order to compare our methods with his we adopt the same evaluation metric as he used. He employed Active Contour Models (ACM) based on two different parameter optimisation functions: using ground truth and raw image for optimisation of the error function (abbreviated as GTEF & RIEF). In addition, the combination of Canny and Otsu's methods with active contour models were also examined [14].

While these methods reached an accuracy of 53.5% to 59.2%, our two-class CNN can classify the pixels with 72.8% correct-classification rate. When two independent CNNs are used (one to detect Myelin and one to detect Axon), it is possible that a pixel can be classified as both Myelin and Axon. In this case we treat the pixel as background, since it is ambiguous. The multi-class CNN can classify 79.7% of the pixels, accurately. Helped by more information about the spatial correlation of the output pixels during the training phase, the multi-class deep convolutional encoder-decoder has even better performance on pixel-wise classification with accuracy of 82.3% (see Fig. 4). The standard deviation of the classification rate across the 35 trials for two-class CNN, multi-class CNN, and encoder-decoder methods are 4.9%, 5.7%, and 4.7%, respectively. We applied paired t-test to our three-class CNN and deep convolutional encoder-decoder to examine the significance of the accuracy improvement. The corresponding p-value is 3.98×10^{-8}.

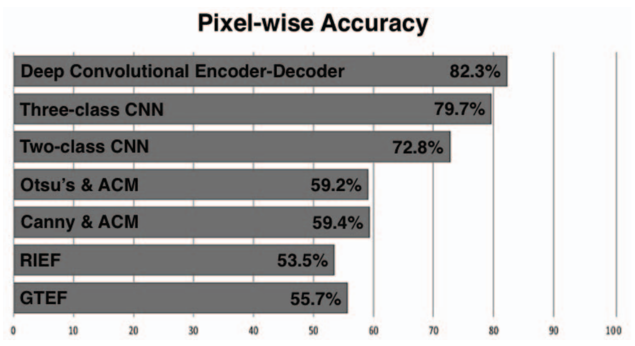

Fig. 4. Comparing the results for Pixels-wise Accuracy Measure. The horizontal axis represents the pixel-wise accuracy measure in percentage.

Not only the three-class CNN and deep encoder-decoder perform about 10% better than the two-class method but also they give a segmentation that is qualitatively more similar to human labelling than the two-class CNN, as shown in Fig. 5.c to 5.e. The encoder-decoder's output is less noisy than the two other methods; however, rectangular patterns correspond to the borders of non-overlapping shifting windows are slightly distinguishable in some regions.

The encoder-decoder method can classify a 700×700 image in less than 6 seconds. In contrast, the CNN takes more than 40 minutes to segment the same image. This is because the encoder-decoder classifies 1024 pixels simultaneously, whereas the CNN architecture has to classify one pixel at a time.

For the encoder-decoder architecture, in order to examine the relation between accuracy and position of the pixel in the frame, we calculated the average accuracy for each pixel of the frame using 357000 randomly chosen 32×32 windows across the dataset (Fig. 6). The distribution of accuracy across the 32×32 sampling frame explains how the position of the pixel (which is not the only circumstance) can effect it's classification rate.

V. Conclusion

In this paper, we employed three different deep learning architectures. The first architecture was designed based on two-class classification for segmenting axon versus non-axon, and also myelin versus non-myelin pixels. The second CNN employed multi-class method to classify each pixel as axon, myelin, or background. The third method is based on the convolutional encoder-decoder architecture in which the pixels in the frame can be segmented either as axon, myelin, or background, simultaneously. The encoder-decoder architecture has performed statistically significantly better than the others, achieving a classification accuracy of 82.3%. All architectures performed much better than previous methods used on this data set. Since our images have no specific characterisations, the deep learning architecture can be applied to similar applications.

Acknowledgment

We would like to express our gratitude to Dr Felix Werner Wehri from University of Pennsylvania Medical Center for providing us microscopy images of the excised mouse spinal cord and also the human labelled ground truth.

References

[1] V. Badrinarayanan, A. Handa and R. Cipolla. "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Robust Semantic Pixel-Wise Labelling", *CVPR*, 2014.

[2] V. Badrinarayanan, A. Kandell and R. Cipolla. "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation", *CVPR*, 2015.

[3] T. Brosch, Y. Yoo, L. Tang, D. Li, A. Traboulsee and R. Tam. "Deep Convolutional Encoder Networks for Multiple Sclerosis Lesion Segmentation", *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015: 18th International Conference Munich, Germany*, vol. 9351, pp. 3–11, 2015.

[4] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy and A. L. Yuille. "Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs", *ICIR*, pp. 1–14, 2014.

[5] A. Fernandez, J. L. Florez, J. R. Alonso, and J. A. Ferrari. "Image segmentation by nonlinear filtering of optical Hough transform", *Applied Optics*, vol. 55, pp. 3632–3638, 2016.

[6] A. Fitzgibbon, M. Pilu, and R. B. Fisher. "Direct least square fitting of ellipses", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, pp. 476–480, 1999.

[7] B. Ginneken, A. Frangi, J. Staal, B. M. T. H. Romeny, and M. A. Viergever. "Active shape model segmentation with optimal features", *IIEEE Transaction on Medical Imaging*, vol. 21, pp. 924–933, 2002.

[8] C. E. Hafer-Macko, K. A. S. MBBS, C. Y. Li, T. W. Ho, D. R. Cornblath, G. M. McKhann, A. K. Asbury, and J. W. Griffin. "Immune attack on the schwann cell surface in acute inflammatory demyelinating polyneuropathy", *Annals of Neurology*, vol. 39, pp. 625–635, 1996.

[9] J. E. Hall and A. C. Guyton, *Guyton and Hall Textbook of Medical Physiology*, Saunders, Philadelphia: pa, 2011.

[10] G. Han, J. Wan, T. Breckon, and D. Randell. "Radicular cysts and odontogenic keratocysts epithelia classification using cascaded haar classifiers", *Proc. 12th Annual Conference on Medical Image Understanding and Analysis*, 2008.

[11] M. Kass, A. Witkin, and D. Terzopoulos. "Snakes: Active contour models", *International Journal of Computer Vision*, vol. 1, pp. 321–331, 1998.

[12] F. Liu, G. Lin, and C. Shen. "CRF learning with CNN features for image segmentation", *Pattern Recognition*, vol. 48, pp. 2983–2992, 2015.

[13] J. Long, E. Shehamer, and T. Darrell. "Fully Convolutional Networks for Semantic Segmentation", *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, 2014.

[14] B. MacDonell. "Spinal cord axon segmentation", Master's thesis, University of Otago, Dunedin, New Zealand, 2014.

[15] R. Miani and H. Aggrawal. "A comprehensive review of image enhancement techniques", *Journal of Computing*, vol. 2, pp. 28–13, 2010.

[16] H. L. More, J. Chen, E. Gibson, J. M. Donelan, and M. F. Beg. "A semi-automated method for identifying and measuring myelinated nerve fibers in scanning electron microscope images", *Journal of Neuroscience Methods*, vol. 201, pp. 149–158, 2011.

[17] D. Nie, W. Li, Y. Gao, and D. Shen. "Fully Convolutional Networks for Multi-Modality Isointense Infant Brain Image Segmentation", *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, pp. 1342–1345, 2016.

[18] H. Noh, S. Hong and B. Han. "Learning Deconvolution Network for Semantic Segmentation", *ICCV*, vol. 1, pp. 1520–1528, 2015.

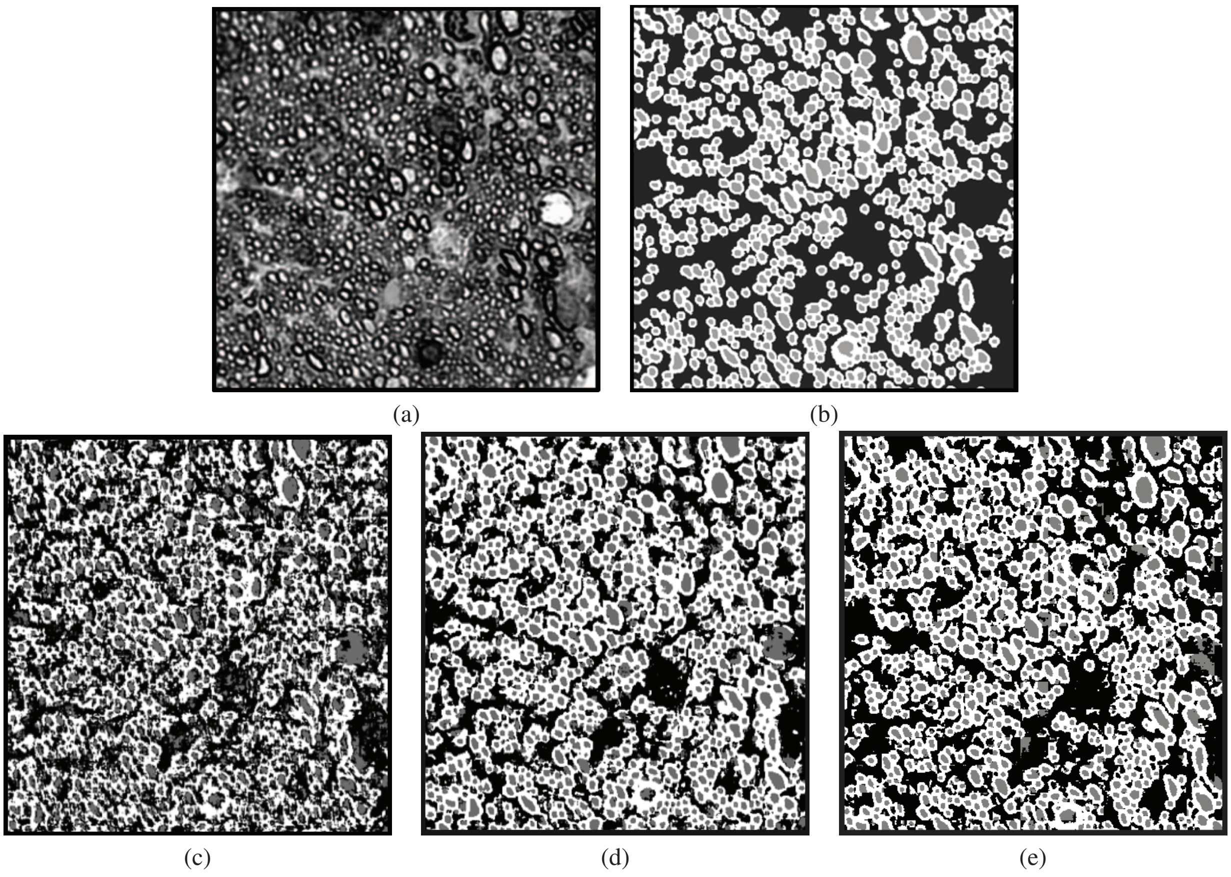
(a) (b) (c) (d) (e)

Fig. 5. (a) Original Microscopy 700 × 700 Image; (b) The corresponding ground truth image; (c) The output of applying two-class segmentation method to the image; (d) The output of applying multi-class segmentation method to the image; (e) The output of applying deep convolutional encoder-decoder segmentation method to the image. In images (b) to (e), white pixels and grey pixels are associated with the myelin and axon labels, respectively.

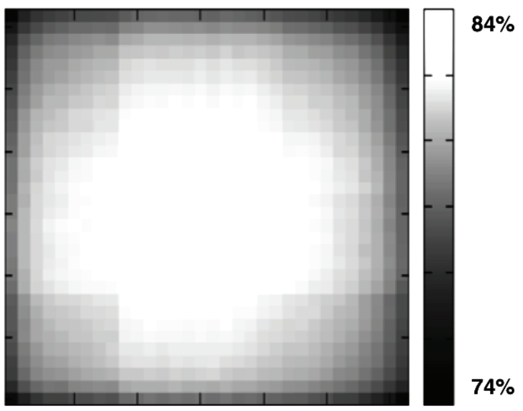

Fig. 6. Positional accuracy of deep convolutional encoder-decoder for 32×32 output window. Maximum and minimum accuracies can be seen in the central region with bright colour and the areas close to the frame borders which are darker, respectively.

[19] H. H. Ong, A. C. Write, S. L. Wehrli, A. Souza, E. D. Schwartz, S. N. Hwang, and F. W. Wehri. “Indirect measurement of regional axon diameter in excised mouse spinal cord with q-space imaging: Simulation and experimental studies”, *NeuroImage*, vol. 40, pp. 1619–1632, 2008.
[20] P. Y. Yin and L. H. Chen. “A fast iterative scheme for multilevel thresholding methods”, *Signal Processing*, vol. 60, pp. 305–313, 1997.
[21] HP. Sahoo, C. Wilkins, and J. Yeager. “Threshold selection using renyi’s entropy”, *Pattern Recognition*, vol. 30, pp. 71–84, 1997.
[22] J. Schmidhuber. “Deep learning in neural networks: An overview”, *Elsevier*, vol. 61, pp. 85–117, 2015.
[23] J. Zhang and J. Hu. “Image segmentation based on 2d otsu method with histogram analysis”, *2008 International Conference on Computer Science and Software Engineering*, vol. 6, pp. 105–108, 2008.