Published in final edited form as:
Med Image Anal. 2010 December ; 14(6): 770–783. doi:10.1016/j.media.2010.06.002.

Detection of Neuron Membranes in Electron Microscopy Images using a Serial Neural Network Architecture

Elizabeth Jurrus[a,b], **Antonio R. C. Paiva**[a], **Shigeki Watanabe**[c], **James R. Anderson**[e], **Bryan W. Jones**[e], **Ross T. Whitaker**[a,b], **Erik M. Jorgensen**[c], **Robert E. Marc**[e], and **Tolga Tasdizen**[a,d]

[a]Scientific Computing and Imaging Institute

[b]School of Computing, University of Utah

[c]Department of Biology, University of Utah

[d]Department of Electrical Engineering, University of Utah

[e]Moran Eye Center, University of Utah School of Medicine

Abstract

Study of nervous systems via the connectome, the map of connectivities of all neurons in that system, is a challenging problem in neuroscience. Towards this goal, neurobiologists are acquiring large electron microscopy datasets. However, the shear volume of these datasets renders manual analysis infeasible. Hence, automated image analysis methods are required for reconstructing the connectome from these very large image collections. Segmentation of neurons in these images, an essential step of the reconstruction pipeline, is challenging because of noise, anisotropic shapes and brightness, and the presence of confounding structures. The method described in this paper uses a series of artificial neural networks (ANNs) in a framework combined with a feature vector that is composed of image intensities sampled over a stencil neighborhood. Several ANNs are applied in series allowing each ANN to use the classification context provided by the previous network to improve detection accuracy. We develop the method of serial ANNs and show that the learned context does improve detection over traditional ANNs. We also demonstrate advantages over previous membrane detection methods. The results are a significant step towards an automated system for the reconstruction of the connectome.

Keywords

Machine Learning; Membrane Detection; Auto-Context; Artificial Neural Networks; Filter Bank; Contour Completion; Neural Circuit Reconstruction

1. Introduction

Neural circuit reconstruction, i.e. the *connectome* [1], is currently one of the grand challenges facing neuroscientists. Similarly, the National Academy of Engineering has listed reverse-engineering the brain as one its grand challenges [1]. While neural circuits are central to the study

of the nervous system, relatively little is known about differences in existing neuronal classes, patterns, and connections. Electron microscopy (EM) is an unique modality for scientists attempting to map the anatomy of individual neurons and their connectivity because it has a resolution that is high enough to identify synaptic contacts and gap junctions. These are important indicators for types of neuron topology and are required for neural circuit reconstruction. Several researchers have undertaken extensive EM imaging projects in order to create detailed maps of neuronal structure and connectivity [2, 3]. Early work in this area, by White *et al*. [4], includes the complete mapping of the nematode *C. elegans* nervous system. This is a simple organism, containing just over 300 neurons and 6000 synapses, yet it took nearly a decade to identify all the relevant structures and reconstruct the connectivity [2]. In comparison, newer imaging techniques are producing much larger volumes of very complex organisms, with thousands of neurons and millions of synapses [5,6]. Thus, automating the reconstruction process is of paramount importance.

The ability to reconstruct neural circuitry at ultrastructural resolution is of substantial clinical importance. Retinal degenerative diseases, including pigmentosa and macular degeneration, result from a loss of photoreceptors. Photoreceptor cell stress and death induces subsequent changes in the neural circuitry of the retina resulting in corruption of the surviving retinal cell class circuitry. Ultrastructural examination of the cell identity and circuitry reveal substantial changes to retinal circuitry with implications for vision rescue strategies [7,8,9,10,11,12,13]. These findings in retinal degenerative disease mirror findings in epilepsy where neural circuits also undergo remodeling in presumed response to abnormal electrical activity clinically manifested as seizures. Scientists are interested in examining normal and pathological synaptic connectivities and how neuronal remodeling contributes to neuronal pathophysiology [14,15, 16]. Examination of synaptic and dendritic spine formation during development provide insight into the adaptivity of neural circuits [17,18]. Ultrastructural evaluation of multiple canonical volumes of neural tissue are critical to evaluate differences in connectivity between wild type and mutants. The complexity and size of the these datasets, often approaching 17 terabytes, makes human segmentation of the complex textural information of electron microscopic imagery a difficult task. Moreover, population or screening studies become unfeasible since fully manual segmentation and analysis would require multiple years of manual effort per specimen. As a result, better image processing techniques are needed to help with automated segmentation of EM data including identification of neurons and the connections.

1.1. Serial-section transmission electron microscopy

The modality we have chosen for reconstructing the connectome at the individual cell level is serial-section transmission electron microscopy (TEM). It provides scientists with images that capture the relevant structures; however, it poses some interesting challenges for image processing. Most importantly, serial-section TEM offers a relatively wide field of view to identify large sets of cells that may wander significantly as they progress through the sections. It also has an in-plane resolution that is high enough for identifying synapses. In collecting images through TEM, sections are cut from a specimen and suspended so that an electron beam can pass through it creating a projection. The projection can be captured on a piece of film and scanned or captured directly as a digital image. An important trade-off occurs with respect to the section thickness. Thinner sections are preferable from an image analysis point of view because structures are more easily identifiable due to less averaging. However, from an acquisition point of view, thinner sections are harder to handle and impose a limit on the area of the section that can be cut. For instance, in the rabbit retina, scientists need to study sections

[1]William Perry, Farouk El-Baz, Wesley Harris, Calestous Juma, Raymond Kurzweil, and Robert Langer, The unveiling of the grand challenges for engineering, in *AAAS Meeting*, Feb 2008.
[2]Emily Singer, A wiring diagram of the brain, *Technology Review*, Nov 2007.

with areas as large as 250μ*m* in diameter to gain a sufficient understanding of neural connectivity patterns. Sections of this size can be reliably cut at 50 – 90nm thickness with the current serial section TEM technology. This leads to an extremely anisotropic resolution, 2 – 5nm in-plane compared to 50 – 90nm out-of-plane, and poses two image processing challenges. First, the cell membranes can range from solid dark curves for neurons that run approximately perpendicular to the cutting-plane, to grazed grey swaths for others which run more obliquely and suffer more from the averaging effect. Consequently, segmentations of neurons in these 2-D images, are difficult given the change in membrane contrast and thickness. Second, due to the large physical separation between sections, shapes and positions of neurons can change significantly between adjacent sections.

There are alternative specimen preparation and EM imaging techniques that can be used for neural circuit reconstruction such as Serial-Block Face Scanning Electron Microscopy. Briggman and Denk proposed a specimen preparation which only highlights extracellular spaces removing almost all contrast from intracellular structures [5]. However, it is not possible to identify synapses with that approach. Identification of synapses is an important part of neural circuit reconstruction because it determines which cells are communicating, and where in the circuitry they connect. To highlight synapses in TEM, scientists must use a stain that also highlights intracellular structures, such as vesicles and mitochondria, as well as neuron membranes. Therefore, image segmentation techniques must account for these data characteristics in order to identify and successfully track neurons across hundreds of sections.

1.2. Neuron segmentation

There are two general approaches for neuron segmentation. One approach focuses first on the detection of neuron membranes in each 2-D section [19,20,21]. These boundaries can be used to identify individual neurons, which are then linked across sections to form a complete neuron. Unfortunately, accurate detection of neuron membranes in EM is a difficult problem given the presence of intracellular structures. This makes simple thresholding, edge detection (i.e., Canny), and region growing methods ineffective for the detection of neuron membranes. Some example images and results with traditional image processing methods are shown in Figure 1. The other approach to neuron segmentation is to directly use the 3-D characteristics of the data [22,23]. However, full 3-D approaches are difficult due to the anisotropic nature of the data. As mentioned earlier, in serial-section EM, there is a trade-off between section thickness and section loss rate. The datasets used in this paper to demonstrate membrane detection are from the *C. elegans* ventral nerve cord and from the rabbit retina. For these datasets, the nerve cord has a resolution of 6nm × 6nm × 33nm and the retina has a resolution of 2nm × 2nm × 80nm. This large section thickness often causes features to shift significantly between sequential images, decreasing the potential advantages of a direct 3-D approach. For these reasons, we follow the first approach which is to first perform a 2-D segmentation followed by a linking of the segmented regions in 3-D. This approach is particularly suitable for datasets in which a majority of the neurons run in a general direction which is roughly orthogonal to the sectioning plane such as the datasets considered in this paper. The main focus of this paper is to improve the 2-D neuron segmentation in each section. This information can then be used to link the segmentation in each section to obtain the full 3-D reconstruction.

Recent related work indicates that machine learning methods are an effective approach for detection of neuron membranes. These methods all use different representations for learning membrane pixels, most of which include training a single instance of a classifier on image derived features, such as Hessian eigenspaces [21,24] and local statistical features [22]. Inspired by Tu's auto-context shape classification approach [25], the method described in this paper uses a series of classifiers to more accurately detect membranes in EM images, which is a necessary step for improved 3-D neuron segmentation as discussed above. However, unlike

Tu's auto-context [25] which uses boosting to select features from a large pool of candidates such as Haar wavelet responses, we use a series of artificial neural networks (ANNs) that operate on a fixed set of features. The first ANN uses as input the intensity values sampled directly from the image. The input to the subsequent ANNs in the series is comprised of the same set of image values, in addition to the output of the previous ANN sampled on a stencil of nearby pixels (as depicted in Figure 4). The ANNs in the series, therefore, have different inputs even though they have a common desired output. The advantages of this method are twofold. First, the classifier uses raw data, that is, the image intensities, rather than a constrained version of the image as given by responses to a large filter bank or statistical features that will not scale well for large datasets. Second, the use of the serial ANNs provides context, which is information from nearby pixels that contributes to the learning, providing increasing amounts of relative information at each stage of the network. As a result, the series of ANNs learns to remove vesicles and mitochondria from the membrane detection and close gaps in places where the membrane is weak. In this paper, we demonstrate the improvement from the combined use of stencils and the series of ANNs for two datasets with distinctly different characteristics.

2. Related Work

There are several methods that attempt to segment EM images of neural tissue. Active contours, in both parametric and level set forms [26,20,27], can provide smooth, accurate segmentations of cells. However, they are very sensitive to initialization, which must be close to the neuron membrane, and often confuse internal structures for neuron membranes. If given an edge term that suppresses internal structures, such as one that is derived from the output of the classifier proposed in this paper, these methods may be more promising. Also, recent work using graph-cut segmentations on EM images produces promising neuron segmentations starting from a manual initialization [28]. All of these methods require an initialization and are more appropriate for segmenting only a few cells. Our goal is the automatic segmentation of thousands of cells which renders manual initialization impractical.

Another set of approaches focus on segmenting the neurons by first performing membrane detection. Simple thresholding methods can be applied after anisotropic directional smoothing to improve membrane continuity [29,19]. This method does not remove internal cellular structures and simultaneously fails to detect a sufficiently high percentage of the true membranes to make accurate segmentations.

Supervised machine learning methods have proved to be useful for detecting membranes in EM images. For example, Jain *et al*. utilizes a multilayer convolutional ANN to classify pixels as membrane or non-membrane in specimens prepared with an extracellular stain [23]. The convolutional ANN has two important characteristics: it learns the filters for classification directly from data, and the multiple convolutions throughout the layers of the network account for an increasing (indirect) filter support region. On the other hand, the proposed ANN contains more than 30,000 parameters and, therefore, is computationally intensive and requires very large training sets. For these reasons, this approach has limited practical usefulness. Andres *et al*. proposes a multi-part segmentation process that uses statistical learning and watersheds to segment neural tissue [22]. Both of these methods produce clear segmentations of the membranes, however, they are aimed at datasets in which the stain used on the specimen suppresses the contrast of intracellular structures leaving only the cell membranes visible [5]. This preparation technique simplifies the segmentation task but, on the other hand, it prevents a full neural circuit reconstruction since this requires the detection of synapses, which are characterized by certain intracellular structures.

In other work based on supervised learning, simple classifiers such as a single perceptron applied to a carefully chosen set of features has been shown to provide promising results in

identifying membranes in EM images [24]. Nevertheless, this method still needs significant post processing to connect membranes and remove internal cellular structures. Similarly, Venkataraju *et al*. proposes using local context features computed from the Hessian matrix to train a boosted classifier to detect membranes, which highlights the importance of context for membrane detection [21]. The results obtained with these methods demonstrate not only the complexity of the problem, but also the potential of supervised machine learning for neuron segmentation.

Conceptually, of particular relevance to this work is Tu's auto-context framework [25], which uses a series of classifiers with contextual inputs to classify pixels in images. In Tu's method, the "continuous" output of a classifier, considered as a probability map, and the original set of features are used as inputs to the next classifier. The probability map values from the previous classifiers provide context for the current classifier, by using a feature set that consists of samples of the probability map at a large neighborhood around each pixel. This means that a classifier can utilize information relayed by previous classifiers from pixel values beyond the scope of its neighborhood, much like a convolutional network. This works well for the structures being detected in this paper. For example, when detecting smooth and elongated features, context helps identify pixels as belonging to membranes instead of other local structures, such as vesicles, by using information from a broader area. Hence, each subsequent classifier extends the support of the probability map, improving the decision boundary, and thus the system can learn the context, or shapes, associated with a pixel classification problem. Theoretically, the series of classifiers improves an approximation of a posteriori distribution [25]. One of the main contributions of our work is the formulation of a series of ANNs in an architecture similar to auto-context. The particular implementation demonstrated by Tu uses 8,000 nonspecific, spatially dispersed, image features, and a sampling of probability maps in very large neighborhoods. This is appropriate for methods that use a boosting classifier strategy [30] and are being performed on smaller scale machine learning problems. However, in the proposed method, a much smaller set of features allows for flexibility and training of large datasets, such as the full rabbit retina dataset [6], which in total is 16TB, and is more suitable for an ANN classifier.

More generally, the detection of complete membranes, even when portions of the membrane are low in contrast, is closely related to the contour completion and salient contour extraction problems which have been studied extensively in the computer vision literature. A detailed review of the literature on contour completion is beyond the scope of this paper. Various approaches have been proposed including spectral clustering and, graph analysis [31,32,33, 34], tensor voting [35], probabilistic models [36] and conditional random fields [37]. Some related work in this area also uses supervised classification that combines features across different scales to detect edges and close contours [38,39]. This paper applies similar techniques in the use of auto-context, which uses incremental learning to gather information about features at different levels. Each stage of the network learns more information about nearby pixels, closing structures that would otherwise be difficult to identify without an incremental approach.

3. Method

The method developed here for neuron membrane detection combines ANN classifiers and image stencil neighborhood feature vectors. The following sections provide details on each of these components.

3.1. Artificial Neural Network

Given the success of ANNs for membrane detection [24,23] and because auto-context is not specifically tied to any classifier, we implement a multilayer perceptron (MLP) ANN as our

classifier. An MLP is a feed-forward neural network which approximates a classification boundary with the use of nonlinearly weighted inputs. The architecture of the network is depicted schematically in Figure 2. The output of each processing element (PE) (each of node of the ANN) is given as [40,41]

$$y = f(\mathbf{w}^T \mathbf{x} + b), \tag{1}$$

where f is, in our case, the *tanh* nonlinearity, $\mathbf{w}$ is the weight vector, and b is the bias. The input vector $\mathbf{x}$ to PEs in the hidden layer is the input feature vector discussed in more detail in the next section. For the output PEs, $\mathbf{x}$ contains the outputs of the PEs in the hidden layer.

ANNs are a method for learning general functions from examples. They are well suited for problems without prior knowledge of the function to be approximated (a.k.a., ”black box models”). They have been successfully applied to robotics [42,43] and face and speech recognition [44,45], and are robust to noise. Training uses gradient descent to solve for a solution which is guaranteed to find a minimum. However, several trade-offs occur in training ANNs regarding the size of the network and the number of inputs. An ANN with too many hidden nodes can lead to overfitting of the network [40], resulting in a set of weights that fits well to the training data, but may not generalize well to test data. At the other extreme, if the number of hidden nodes is insufficient the ANN does not have enough degrees of freedom to accurately approximate the decision boundary. The number of inputs should also be keep small to mitigate the problem high-dimensional spaces, known as the “curse of dimensionality.” Generally speaking, as the dimensionality of the input space increases, the data becomes increasingly sparse which makes it difficult to accurately learn a decision boundary. Additionally, the training time tends to scale with the amount of training data and size of the network, and therefore training smaller networks is generally preferable. Hence, the number of inputs to each ANN should be large enough to describe the data, while keeping this number to a minimal.

3.2. Image Stencil Neighborhood

Choosing the best set of features to use in training an ANN is crucial for obtaining good segmentations. The field of machine learning has made available several possible strategies. A possible approach uses large sets of statistical features as the input to a learning algorithm. These features can include simple local and non-local properties, including the pixel values, mean, gradient magnitude, standard deviation, and Hessian eigenvalues [22,25,21]. These attempt to present the learning algorithm with a large variety of mathematical descriptors to train on, and are designed to work on a variety of data types. To achieve this generality, however, large numbers of these features are required to train a classifier. Training a classifier, and ANNs in particular, with a large number of features is challenging due to the “curse of dimensionality” which, if not done carefully, can complicate the decision space and make it difficult to find an optimal solution. Another approach is to design a set of match filters and apply them to an image to approximate a pixel’s similarity to a membrane. This works well if the membranes in the image are uniform and respond well using cross-correlation [46,47]. Moreover, the design of the filter bank requires significant a priori knowledge of the problem. Yet, the fixed design may not be optimal for the dataset. Most importantly, the match filters have to be redesigned for datasets with different characteristics. On the other hand, learning these filters from training data, as in the case of convolutional networks [23], has the advantage that no a priori knowledge is required. A similar idea was been used in texture classification where is was shown that direct sampling of the image with a patch is actually a simpler and better approach for training a classifier compared to the use of filter banks [48]. Image patches have also been used successfully for texture segmentation [49] and image filtering [50,51,

52]. Similarly, using image neighborhoods in our case allows the ANNs to learn directly on the input data, giving the classifier more flexibility in finding the correct decision boundary. We define a square image neighborhood as an image patch, shown in Figure 3(a), centered at pixel *k*, *l*,

$$P_{k,l}=\left\{I_{k+i,l+j}:i,\ j=-\frac{R-1}{2},\ldots,\frac{R-1}{2}\right\}. \tag{2}$$

R is the width of the square image patch. Unfortunately, the size of the image patches required to capture sufficient context can be quite large. For this reason, we propose using as input to the ANNs the values from the image and probability map of the previous classifier sampled through a stencil neighborhood, shown in Figure 3(b). A stencil is also centered at pixel *k*, *l* and defined as,

$$S_{k,l}=\cup_{a=1}^{n} B_{k,l,a} \tag{3}$$

where

$$B_{k,l,a}=\left\{I_{k+ai,l+aj}:i,\ j=-1,0,1\right\}, \tag{4}$$

and *n* is the number of rows the stencil spans in the image. The stencil can cover large areas representing the desired feature space, but samples it with a spatially adaptive resolution strategy. In this way, an ANN can be trained using a small number of samples from image data, without having to use the whole image patch. Since the number of weights to be computed in an ANN are dominated by the connection between the input and the hidden layers, reducing the number of inputs reduces the number of weights and helps regularize the learned network. Moreover, using less inputs generally allows for faster training. With this, one aims to provide the classifier with sparse, but sufficient context information and achieve faster training, while obtaining a larger context which can lead to improve membrane detection. This strategy, combined with the serial use of ANNs (described in section 3.3), grows the region of interest for classification within a smaller number of stages and without long training times.

3.3. Serial Artificial Neural Networks

Using principles from auto-context, we implemented a series of classifiers that leverage the output of the previous network to gain knowledge of a large neighborhood. For the first classifier, the input is the image intensities around a pixel sampled using a stencil. For the ANNs in the remaining series, the input vector contains the samples from the original image, used as input to the first ANN, appended with the values from the output of the previous classifier sampled through the stencil neighborhood, yielding a larger feature vector. While the desired output labels remain the same, each ANN is dependent on the information from the previous network and therefore must be trained sequentially, rather than in parallel. Figure 4 demonstrates this flow of data between classifiers. *I* denotes the image, *S* the image values sampled from the image using the stencil, *C* the output from the ANN, and *T* the threshold applied to *C* at zero, yielding the final membrane detection.

The serial structure allows the classifiers to gather with each step context information from a progressively larger image neighborhood to the pixel being classified, as occurs with a convolutional ANN. Indirectly, the classification from the previous ANN contains information about features in surrounding pixels, that is not represented in the original feature set. This

allows the subsequent networks in the series (Figure 4) to make decisions about the membrane classification utilizing nonlocal information. Put differently, each stage in the series accounts for larger structures in the data, taking advantage of results from all the previous networks. This results in membrane detection that improve after each network in the series. Figure 5 visually demonstrates the classification improving between ANNs in the series as gaps in weak membranes are closed and intracellular structures are removed with each iteration in the series. The receiver operating characteristic (ROC) curves in Figures 6 also demonstrate the increase in detection accuracy after each ANN in the series.

Combining the original image features with features sampled from the output of the previous classifier is important because, in this way, the membrane structure relevant for detection is enforced locally and then again at a higher level with each step in the series of classifiers. One of the advantages of this approach is that it provides better control of the training, allowing the network to learn in steps, refining the classification at each step as the context information it needs to correctly segment the image increases. Again, note that the membrane structure is learned directly from the data. Compared to a single large network with many hidden layers and nodes, such as the convolutional ANN of Jain *et al*. [23] which requires 34,000 parameters, the proposed classifier is easier to train. This is mainly because each of the ANNs have a relatively small number of parameters. For example, given a single ANN used to compute the results in Section 4, the number of parameters needed is approximately 500 for the first ANN and 1100 for the remaining ANNs in the series. The number of weights in an ANN with a single-hidden layer is given by $(n + 1)h + (h + 1)$, where n is the number of inputs and h is the number of nodes in the hidden layer. For the first ANN in the series, $n = s$, where s is the number of points in the stencil. For the remaining ANNs in the series, $n = 2s$, since we sample the original image and the output from the previous classifier once. The total number of parameters across the whole series totals to approximately 5000. In contrast, a convolutional ANN needs $(n + 1)h$ for the first layer,and $(nh + 1)h$ for the remaining layers, an h^2 dependence [23]. Hence, much less training data is needed, which is hard to obtain, since the ground truth must be hand labeled[3]. Furthermore, the training is simpler since backpropagation is less likely to get stuck on local minima of the performance surface [40,41], and the network will train much faster.

4. Results

Two TEM datasets are used as test cases for the proposed method. The first dataset is a stack of 50 sections from the ventral nerve cord of the *C. elegans*. The second dataset is a single section from the 16TB rabbit retina dataset. These datasets contain very different types of neural cells. The *C. elegans* data has a resolution of 6nm×6nm×33nm and each 2-D section is 662×697 pixels. Neuron membranes in the *C. elegans* data appear as intensity valleys; however, not all valleys in the data are neuron membranes, i.e. membranes of intracellular structures may also appear locally as valleys. The proposed method successfully learns the appropriate subset of ridges that need to be identified as neuron membranes as will be described in Section 4.2. The rabbit retina data has a resolution of 2nm×2nm×80nm and each 2-D section is 7629×7351 pixels. Unlike the *C. elegans* dataset, neuron membranes in the retina data generally appear as intensity edges. Owing to the flexibility offered by the use of stencils rather than a predefined filter bank, the proposed method is also successful in learning to detect neuron membranes in this dataset as will be discussed in Section 4.3.

[3]According to the "rule-of-thumb" in [41], one needs at least 10× training samples pf the total number of parameters. Thus, compared to Jain *et al*. [23] convolutional ANN, our approach needs about 27× less training samples, for the values given.

4.1. Experimental Setup

Before discussing detailed results of experiments on the two dataset, we will outline the common experimental details. First, our setup for these data sets used 5 ANNs in series. Additional networks could be included; however, for these datasets, the performance converges to a limit (Figure 6) and improvement in membrane detection is minimal. Each ANN used in the experiments contained one hidden layer with 20 nodes. We experimented with more layers and different numbers of nodes but did not find significant advantages. It is important that the number of nodes be large enough to approximate a non linear boundary and small enough that the ANN does not overfit to the training data [53,54]. Results using 10, 20, and 30 nodes turned out to be somewhat similar. Given the time versus performance trade-off, we chose 20 nodes. The networks were trained using backpropagation with a step size of 0.0001 and momentum term of 0.5. We used early stopping as the criterion to determine when to terminate training [40,41]. This means that a small portion of the training data (20% in our case), called the validation set, is used only to test the classifier generalization performance. The training terminates when the lowest error on the validation set is attained. To mitigate problems with local minima, each network is trained for 5 Monte Carlo simulations using randomly initialized weights.

Preprocessing is performed for each image using a contrast limited adaptive histogram equalization (CLAHE) [55] filter. This enhances the contrast of the membranes. Window sizes of 64×64 and 256×256 were used for the *C. elegans* and retina datasets, respectively.

Each image used in the experiments was annotated by an expert who carefully marked neuron membranes with a one-pixel-wide contour. This contour was dilated using a disk shaped kernel with a radius of 2 pixels, ensuring that the positive training examples cover all of the actual membrane pixels. The negative training examples were selected as the remaining pixels in the image, after erosion to remove training pixels that are very close to the membranes. This strategy leaves a thin layer of pixels between the positive and negative training example pixels that are not used for training purposes. This ensures that the network learns on pixels that are either membrane or non-membrane, excluding those that are more prone to labeling errors.

Finally, to optimize network performance, the total number of training examples from each image includes all of the positive examples and a random selection of negative examples such that there are twice the number of negative examples, than positive. Choosing the optimal number of training examples was difficult given there were many more negative than positive examples in this dataset. If all the negative training examples are used then the ANNS are biased towards classifying pixels as non-membrane. After conducting a series of experiments for considering the results from different ratios of positive and negative examples and the training times, we found the 2:1 ratio resulted in the best segmentation while achieving a reduced training time. Using all the training data (and other increased ratios, such as 4:1) produced a similar ROC curve but results were biased towards false negatives. Clearly, we could have adjusted the threshold in the final stage.Alternatively, one possible solution for this problem would be the use of a weighting factor chosen to obtain unbiased training. However, either approach would have much slower training than the previous described strategy without improving the overall segmentation. For each pixel in our training data, a stencil with a radius of 5 (or $n = 11$ in Equation 3) is used to sample the image data and form the feature vector.

4.2. Results on the *C. elegans* Ventral Nerve Cord

The nematode *C. elegans* is an important dataset for neural circuit reconstruction. Despite being a well studied organism [4], there are still numerous open questions such as how genes regulate wiring [56] or how connectivity is altered to mediate different behaviors, for example between males and females [57]. Reconstructions of the full nervous system reveals topological

characteristics important for researchers studying neuron wiring. The particular dataset used in this paper is from the ventral nerve cord of the *C. elegans* and is important for studying the topological structure resulting from neurons making connections to local targets.

To validate the robustness of the method, five-fold cross-validation was used on a set of 50 annotated images, separated into 5 groups of 10 images in each. The network was trained on each fold according to the procedure described in Section 4.1, and tested on the remaining four. The improvement in the classification after each ANN in the series is visible in the classification of the training data after each stage, shown in Figure 5, and in the receiver operating characteristic (ROC) curves in Figure 6. The output from the network improves quantitatively and qualitatively with each network in the series. Directly sampling the image using a stencil and repeated uses of the network enables the method to accurately estimate the appearance of membrane pixels and pixels in surrounding neighborhoods.

Figure 7(a) shows four sections from the *C. elegans* dataset chosen at random. The final membrane detection with the proposed method is shown in Figure 7(e). Note that these are testing results; that is, these four sections were not used as training data. To demonstrate the advantages of the proposed method, two other methods are presented. The first method, shown in Figure 7(b), performs thresholding after enhancing the membranes with anisotropic directional smoothing [19]. Figure 7(c) shows results from an approach similar to the approach in Mishchenko [24], which learns boundary confidences using Hessian eigenvalues as input to a single layer neural network. It can be seen that the proposed method removes a substantially larger percentage of the intracellular structures from the detection results as well as providing better membrane continuity. It is important to note that in Mishchenko [24] further post-processing is performed to interpolate between broken boundaries and complete contours, resulting in an improved result compared to the one shown here. However, we compare against only the single layer network part of that method since our goal is to demonstrate the improvement achieved by the use of ANNs and auto-context. Of course, the same preprocessing methods could be applied to the results of the proposed method as well. Figure 8 shows enlarged regions demonstrating the removal of large intracellular structures and closing of weak membranes.

To demonstrate the advantages of directly sampling the image with a stencil, we also tested the proposed auto-context ANN strategy but with inputs to the ANNs that are derived from a line detection filter bank rather than sampling the image. We used a filter bank that consists of a set of 32 line detection filters oriented at different angles and 5 circle detection filters with different radii. The circle detection filters were included to help the auto-context ANN to learn to remove vesicles from the membrane detection results. Figure 7(d) is the output obtained with the filter bank/series of ANN approach. While these results are better than the results in Figure 7(b) and (c), they contain more false positives than the results of the full stencil/auto-context ANN approach shown in Figure 7(e). The advantages of using the stencil becomes clearer in a quantitative comparison as discussed in the next paragraph. Furthermore, an important practical advantage of using the stencil is that it does not require any a priori knowledge. Therefore, it can be trained to detect different structures as will be shown for the retina dataset in Section 4.3. In comparison, a filter bank designed to capture the relevant structures for the *C. elegans* dataset is not expected to capture the relevant structures in a different dataset which necessitates the design of a new filter bank.

Figure 9 compares the ROC curves for each method from Figure 7. For this particular data, a single layer ANN using Hessian eigenvalues as inputs (labeled “Hessian”) demonstrates no quantitative differences from thresholding after directional anisotropic diffusion (labeled “Jurrus et al.”). These ROC curves correspond to the qualitative results in Figure 7(b) and (c), respectively. The other three curves all show a large improvement in performance. The use of

membrane detection filters (labeled “Filters”) demonstrates how a carefully chosen set of features can be used for learning to detect membranes. Image patches (labeled “Patches”) are just as successful in training to detect membranes as filters. However, in testing, the patches outperform the filters. We argue that this is due to the fact that patches sample the image directly and give more flexibility to the classifier than a filter bank. Using a stencil (labeled “Stencil”) results in the best performance. The stencil provides the classifier with two important features. First, similar to patches, it trains the classifier on image sample directly, as opposed to a fixed representation as obtained from the filters. Second, it samples a larger area than the patches, while maintaining the same number of features (as seen in Figure 3). The latter feature is very helpful in practice since it ensures improved performance without sacrificing the network training time (actually, in our experiments, using the stencil improved the training reliability and time).

Figure 10 demonstrates the final neuron segmentation, after a very simple region flood fill is applied to the image of detected membranes with the proposed method. This depicts how close the final segmentation is to the true segmentation. Figure 11 is a full 3-D neuron segmentation for four key neurons and nearby muscles from this dataset. Neurons are segmented in each section using a region flood fill and linked across sections using a minimum path finding algorithm similar to Jurrus *et al*. [19]. Hand edits were required to correct some mistakes in the automatic segmentation. This 3-D model shows the motor neurons in the ventral nerve cord and their processes interdigitating along the lateral edge of the nerve bundle (Figure 11(a)) to make contact with the muscles (Figure 11(b)). Multiple muscles, in turn, must send processes to these motor neurons to receive input. Areas where this communication is occurring are marked in red. There are three motor neuron inputs into these muscles: the VA neurons release acetylcholine during backwards movement, the VB neurons release acetylcholine during forward movement, and the VD motor neurons release GABA to relax the muscle to allow sinusoidal movement. These data demonstrate that axons do not precisely interweave. GABA neurons run alongside a group of muscle arms and form multiple synapses to differing subsets of muscles before giving way to acetylcholine motor neurons. By contrast, the two types of acetylcholine neurons usually form contacts to the muscles simultaneously. Again, they form 2 to 3 contacts to the muscles for a segment of axon before giving way to the GABA motor neuron. This demonstrates the importance and diagnostic capabilities of full connectivity diagrams and renderings.

4.3. Results on the Rabbit Retina

The retina is a complex structure containing several layers of neurons. Processing light sets off a series of chemical events and connections among these neurons that scientists would like to model. Most importantly, scientists would like to characterize neural circuitry that is damaged and in a diseased state. However, unraveling the connective patterns in this complex tissue is an enormous task.

To demonstrate the robustness of our method on a very different dataset, an expert hand segmented all of the bipolar, amacrine, and horizontal cells in a single 2-D section through the retina. This section is 7629×7351 pixels and contains approximately 500 neurons. The image was divided into four equal sections and a four-fold cross validation technique is used to assess the performance of the algorithm.

Figure 12 shows the output on the test data. Figure 12(a) shows portions of the TEM image, cropped to show the cellular details. Figure 12(b) is a simple baseline membrane detection obtained by thresholding the intensity gradient after smoothing the input image with a Gaussian kernel (standard deviation 3 pixels). Thresholding the gradient results in some obvious problems. Differences in contrast and the presence of intracellular features make isolation of the neuron edges difficult. Figure 12(c) shows the results of applying the series of ANN method

with a filter bank as input. For this data, 25 Leung-Malik edge filters [46] were used. The Leung-Malik filter banks consists of first derivatives of a Gaussian kernel (standard deviation 3 pixels) at various orientations. The results in Figure 12(d) are from the stencil/serial ANN approach identical in architecture to the one used for the *C. elegans* dataset. From a qualitative perspective, the stencil removes more intracellular structures and is more robust to changes in contrast. When the weights from the final network, which was trained with a filter bank, are applied to the testing images, the edge detection performs poorly. However, sampling the image using a stencil is a more robust way to detect membrane edges and provides more consistent results across images. Figure 13 gives clear examples of how this method removes non membrane structures and closes complex gaps resulting from inconsistent membrane data. Most importantly, the results from this dataset demonstrate the flexibility of our method on different feature types. The feature vectors for both dataset are the same, that is, they are simply ram image values sampled from the input data.

Figure 14 shows the quantitative comparison for the methods demonstrated in Figure 12. The gradient magnitude provides a baseline for how well a simple edge detection method can be expected to perform. While it detects many of the neuron boundaries, it also has a lot of false positive responses for internal structures and fails to close gaps in weak parts of the membranes. The serial networks provide a very large improvement over this simple method when a filter bank is in place. However, the proposed stencil/auto-context ANN method is demonstrated to do still a significantly better job at detecting boundaries than the filters.

5. Conclusion and Future Work

In this paper a new approach for neuron membrane detection is proposed. Inspired by Tu's auto-context framework [25], our approach introduces two major contributions. The first contribution is the introduction of a serial ANN classifier and its application to neuron membrane detection. The use of context allows the classifier to close gaps in weak membranes and suppress intracellular structures by using increasingly non-local information with each ANN in the series. The second contribution is the use of raw image intensities sampled through a stencil as inputs to the series ANN rather than a predetermined filter bank. This provides increased flexibility to the classifier which can then be trained to detect neuron membranes in datasets with significantly different characteristics. Also, it must be noted the choice of sampling the image with a stencil rather than using the more traditional patch neighborhood. As shown in the results, utilizing a stencil yields significantly better results. This is because, for the same number of features, a stencil provides context information for a larger neighborhood. Although larger patches could be utilized, the number of features would grow more rapidly to impractical levels, and would be slower (and more complicated) to train the classifier. These two contributions result in a neuron membrane algorithm that outperforms other methods.

A direct comparison to Tu's auto-context classification using a probabilistic boosting tree is difficult to do. Applying the same filter bank to the data presented here results in storage complications and does not scale to larger datasets like the rabbit retina data, which is 16TB [6]. However, to compare the performances, our method was tested on the Weizmann Horse dataset [58], for which results using Tu's method are available, without significant changes in performance. The method used in this paper had an overall accuracy (or f-value) of 0.834, while Tu's accuracy was 0.84, and the qualitative differences were negligible [59]. Nevertheless, it must be emphasized that a major advantage in our method is that the filters are learning directly from data, tremendously simplifying the user's role. By incorporating a similar architecture into the form of a series of ANNs, we have designed a method that performs well on EM images and aids in the building of 3-D models for neural circuit reconstruction, as depicted in Figure 11.

Given the challenge of full 3D reconstructions, and the extremely anisotropic resolution of serial section TEM, we approach this problem with a two-stage solution that consists of first segmenting neurons in 2-D sections and then linking them up the segments in 3-D. Therefore, the motivation for improving the accuracy of automatic neuron membrane detection methods is to minimize user interaction required to correct the segmentation. Figure 10 demonstrates segmentations obtained by applying a simple flood fill operation to the image of detected membranes without any user corrections. In future work, segmentations obtained using the proposed method can be ex-tended to other sections, taking advantage of segmentations in sequential sections having similar anatomy. Simple gap closing methods can also be applied to close small remaining holes in the membrane for a better 2-D neuron segmentation. Finally, a similar classifier strategy could prove successful also in segmenting long tubular structures such as vasculature in MRI due to the capability of closing gaps in weak areas of elongated structures.

Acknowledgments

This work was supported by NIH R01 EB005832 (TT), NIH EY0015128 (RM), EY002576 (RM), NEI Vision Core EY014800 (RM), HHMI (EMJ), and NIH NINDS 5R37NS34307-15 (EMJ).

References

1. Sporns O, Tononi G, Ktter R. The human connectome: A structural description of the human brain. PLoS Comput. Biol 2005;1:e42. [PubMed: 16201007]
2. Fiala JC, Harris KM. Extending unbiased stereology of brain ultra-structure to three-dimensional volumes. J Am Med Inform Assoc 2001;8(1):1–16. [PubMed: 11141509]
3. Briggman KL, Denk W. Towards neural circuit reconstruction with volume electron microscopy techniques. Current Opinion in Neurobiology 2006;16(5):562–570. doi:http://dx.doi.org/10.1016/j.conb.2006.08.010, URL http://dx.doi.org/10.1016/j.conb.2006.08.010. [PubMed: 16962767]
4. White J, Southgate E, Thomson J, Brenner F. The structure of the nervous system of the nematode Caenorhabditis elegans. Phil. Trans. Roy. Soc. London Ser. B Biol. Sci 1986;314:1–340.
5. Briggman KL, Denk W. Towards neural circuit reconstruction with volume electron microscopy techniques. Current Opinion in Neurobiology 2006;16(5):562–570. [PubMed: 16962767]
6. Anderson J, Jones B, Yang J-H, Shaw M, Watt C, Koshevoy P, Spaltenstein J, Jurrus EUVK, Whitaker R, Mastronarde D, Tasdizen T, Marc R. A Computational Framework for Ultrastructural Mapping of Neural Circuitry. PLoS Biology 2009;7(3):e74.
7. Marc RE, Jones BW, Watt CB, Vazquez-Chona F, Vaughan DK, Organisciak DT. Extreme retinal remodeling triggered by light damage: implications for age related macular degeneration. Mol. Vis 2008;14:782–806. [PubMed: 18483561]
8. Marc RE, Jones BW, Anderson JR, Kinard K, Marshak DW, Wilson JH, Wensel T, Lucas RJ. Neural reprogramming in retinal degeneration. Invest. Ophthalmol. Vis. Sci 2007;48:3364–3371. [PubMed: 17591910]
9. Marc RE, Jones BW, Watt CB, Strettoi E. Neural remodeling in retinal degeneration. Prog Retin Eye Res 2003;22:607–655. [PubMed: 12892644]
10. Jones BW, Marc RE. Retinal remodeling during retinal degeneration. Exp. Eye Res 2005;81:123–137. [PubMed: 15916760]
11. Jones BW, Watt CB, Frederick JM, Baehr W, Chen CK, Levine EM, Milam AH, Lavail MM, Marc RE. Retinal remodeling triggered by photoreceptor degenerations. J. Comp. Neurol 2003;464:1–16. [PubMed: 12866125]
12. Jones BW, Watt CB, Marc RE. Retinal remodelling. Clin Exp Optom 2005;88:282–291. [PubMed: 16255687]
13. Peng YW, Hao Y, Petters RM, Wong F. Ectopic synaptogenesis in the mammalian retina caused by rod photoreceptor-specific mutations. Nat. Neurosci 2000;3:1121–1127. [PubMed: 11036269]
14. Sutula T. Seizure-Induced Axonal Sprouting: Assessing Connections Between Injury, Local Circuits, and Epileptogenesis. Epilepsy Curr 2002;2:86–91. [PubMed: 15309153]

15. Pollard H, Khrestchatisky M, Moreau J, Ben-Ari Y, Represa A. Correlation between reactive sprouting and microtubule protein expression in epileptic hippocampus. Neuroscience 1994;61:773–787. [PubMed: 7838377]
16. Koyama R, Yamada MK, Fujisawa S, Katoh-Semba R, Matsuki N, Ikegaya Y. Brain-derived neurotrophic factor induces hyperexcitable reentrant circuits in the dentate gyrus. J. Neurosci 2004;24:7215–7224. [PubMed: 15317847]
17. Sorra KE, Harris KM. Overview on the structure, composition, function, development, and plasticity of hippocampal dendritic spines. Hippocampus 2000;10:501–511. [PubMed: 11075821]
18. DeBello, WM.; Feldman, DE.; Knudsen, EI. Adaptive Axonal Remodeling in the Midbrain Auditory Space Map; J. Neurosci. 2001. p. 3161-3174.URL http://www.jneurosci.org/cgi/content/abstract/21/9/3161
19. Jurrus E, Whitaker R, Jones B, Marc R, Tasdizen T. An Optimal-Path Approach for Neural Circuit Reconstruction. Proceedings of the 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro 2008:1609–1612.
20. Macke J, Maack N, Gupta R, Denk W, Schölkopf B, Borst A. Contour-propagation algorithms for semi-automated reconstruction of neural processes. Journal of Neuroscience Methods 2008;167:349–357. [PubMed: 17870180]
21. V. KU, Paiva A, Jurrus E, Tasdizen T. Automatic markup of neural cell membranes using boosted decision stumps. To Appear, Proceedings of the 6th IEEE International Symposium on Biomedical Imaging 2009:1039–1042.
22. Andres, B.; Köthe, U.; Helmstaedter, M.; Denk, W.; Hamprecht, FA. Segmentation of SBFSEM Volume Data of Neural Tissue by Hierarchical Classification. In: Rigoll, G., editor. Pattern Recognition, vol. 5096 of *LNCS*. Springer; 2008. p. 142-152.ISBN 978-3-540-69320-8, doi: 10.1007/978-3-540-69321-5_15
23. Jain V, Murray J, Roth F, Turaga S, Zhigulin V, Briggman K, Helmstaedter M, Denk W, Seung H. Supervised Learning of Image Restoration with Convolutional Networks. IEEE 11th International Conference on Computer Vision 2007:1–8.
24. Mishchenko Y. Automation of 3D reconstruction of neural tissue from large volume of conventional serial section transmission electron micrographs. J Neurosci Methods. 2009
25. Tu Z. Auto-context and its application to high-level vision tasks. IEEE Conference on Computer Vision and Pattern Recognition 2008:1–8.
26. Jurrus E, Hardy M, Tasdizen T, Fletcher P, Koshevoy P, Chien C-B, Denk W, Whitaker R. Axon Tracking in Serial Block-Face Scanning Electron Microscopy. Medical Image Analysis 2009;13(1): 180–188. [PubMed: 18617436]
27. Vazquez L, Sapiro G, Randall G. Segmenting Neurons in Electronic Microscopy via Geometric Tracing. Proc. of ICIP 1998:814–818.
28. Vu, N.; Manjunath, B. Graph cut segmentation of neuronal structures from transmission electron micrographs. Image Processing, 2008; ICIP 2008. 15th IEEE International Conference on, ISSN 1522-4880; 2008. p. 725-728.doi:10.1109/ICIP.2008.4711857
29. Tasdizen T, Whitaker R, Marc R, Jones B. Enhancement of Cell Boundaries in Transmission Electron Microscopy Images. ICIP 2005:642–645.
30. Freund, Y.; Schapire, RE. A decision-theoretic generalization of on-line learning and an application to boosting; EuroCOLT '95: Proceedings of the Second European Conference on Computational Learning Theory; London, UK: Springer-Verlag; 1995. p. 23-37.ISBN 3-540-59119-2
31. Shashua A, Ullman S. Structural Saliency: The Detection Of Globally Salient Structures using A Locally Connected Network. Computer Vision 1988:321–327. Second International Conference on.
32. Mahamud S, Williams L, Thornber K, Xu K. Segmentation of multiple salient closed contours from real images, Pattern Analysis and Machine Intelligence. IEEE Transactions on 2003;25(4):433–444. ISSN 0162-8828, doi:10.1109/TPAMI.2003.1190570.
33. Fowlkes C, Belongie S, Chung F, Malik J. Spectral Grouping Using the Nyström Method. IEEE Transactions on Pattern Analysis and Machine Intelligence 2004;26:214–225. [PubMed: 15376896]
34. Zhu, Q.; Song, G.; Shi, J. Untangling Cycles for Contour Grouping. Computer Vision, 2007; ICCV 2007. IEEE 11th International Conference on, ISSN 1550-5499; 2007. p. 1-8.doi:10.1109/ICCV.2007.4408929

35. Tang C-K, Medioni G. Curvature-augmented tensor voting for shape inference from noisy 3D data, Pattern Analysis and Machine Intelligence. IEEE Transactions on 2002;24(6):858–864. ISSN 0162-8828, doi:10.1109/TPAMI.2002.1008395.
36. Ren, X.; Malik, J. A Probabilistic Multi-scale Model for Contour Completion Based on Image Statistics; ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part I; London, UK: Springer-Verlag; 2002. p. 312-327.ISBN 3-540-43745-2
37. Ren, X.; Fowlkes, C.; Malik, J. Scale-invariant contour completion using conditional random fields. Computer Vision, 2005; ICCV 2005. Tenth IEEE International Conference on, vol. 2, ISSN 1550-5499; 2005. p. 1214-1221.doi:10.1109/ICCV.2005.213
38. Dollar, P.; Tu, Z.; Belongie, S. Supervised Learning of Edges and Object Boundaries. Computer Vision and Pattern Recognition; IEEE Computer Society Conference on, vol. 2, IEEE Computer Society; Los Alamitos, CA, USA. 2006. p. 1964-1971.ISSN 1063-6919, doi: http://doi.ieeecomputersociety.org/10.1109/CVPR.2006.298
39. Shotton J, Blake A, Cipolla R. Multiscale Categorical Object Recognition Using Contour Fragments. Pattern Analysis and Machine Intelligence, IEEE Transactions on 2008;30(7):1270–1281. ISSN 0162-8828, doi:10.1109/TPAMI.2007.70772.
40. Haykin, S. Neural networks - A comprehensive foundation. 2nd edn.. Prentice-Hall; 1999.
41. Principe, JC.; Euliano, NR.; Lefebvre, WC. Neural and Adaptive Systems: fundamentals through simulations. John Wiley & Sons; 2000.
42. Pomerleau, D. Knowledge-based Training of Artificial Neural Networks for Autonomous Robot Driving. In: Connell, J.; Mahadevan, S., editors. Robot Learning. Kluwer Academic Publishing; 1993. p. 19-43.
43. Wells G, Venaille C, Torras C. Promising Research: Vision-Based Robot Positioning Using Neural Networks. IVC 1996;14(10):715–732.
44. Rabi G, Lu S. Visual Speech Recognition by Recurrent Neural Networks. JEI 1998;7(1):61–69.
45. Cottrell, G. Extracting features from faces using compression networks: face, identity, emotion and gender recognition using holons. Morgan Kaufmann; 1990. p. 328-337.
46. Leung T, Malik J. Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons. Int. J. Comput. Vision 2001;43(1):29–44. ISSN 0920-5691, doi: http://dx.doi.org/10.1023/A:1011126920638.
47. Schmid, C. Constructing models for content-based image retrieval. Computer Vision and Pattern Recognition, 2001; CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, vol. 2, ISSN 1063-6919; 2001. p. II–39-II–45.doi:10.1109/CVPR.2001.990922
48. Varma, M.; Zisserman, A. Texture classification: are filter banks necessary?. Computer Vision and Pattern Recognition, 2003; Proceedings. 2003 IEEE Computer Society Conference on, vol. 2, ISSN 1063-6919; 2003. p. II–691p. II–698doi:10.1109/CVPR.2003.1211534
49. Awate SP, Tasdizen T, Whitaker RT. Unsupervised Texture Segmentation with Nonparametric Neighborhood Statistics. Proceedings of the European Conference on Computer Vision 2006:494–507.
50. Buades A, Coll B, Morel J-M. A non-local algorithm for image denoising. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition 2005:60–65.
51. Awate SP, Whitaker RT. Unsupervised, information-theoretic, adaptive image filtering for image restoration, IEEE Trans. on Pattern Analysis and Machine Intelligence. 2006;28(3):364–376.
52. Tasdizen T. Principal Components for Non-local Means Image Denoising. Proceeding of International Conference on Image Processing. 2008
53. Cybenko G. Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals, and Systems 1989;2(4):303–314. doi:10.1007/BF02551274.
54. Hornik K. Approximation capabilities of multilayer feedforward networks. Neural Netw 1991;4(2): 251–257. doi:10.1016/0893-6080(91)90009-T.
55. Pizer, S.; Johnston, R.; Ericksen, J.; Yankaskas, B.; Muller, K. Contrast-limited adaptive histogram equalization: speed and effectiveness. Visualization in Biomedical Computing; Proceedings of the First Conference on (1990); 1990. p. 337-345.doi:10.1109/VBC.1990.109340
56. Jin Y, Hoskins R, Horvitz HR. Control of type-D GABAergic neuron differentiation by C. elegans UNC-30 homeodomain protein. Nature 1994;372(6508):780–783. [PubMed: 7997265]

57. White JQ, Nicholas T, Gritton J, Truong L, Davidson ER, Jorgensen EM. The sensory circuitry for sexual attraction in C. elegans males. Curr. Biol 2007;17(21):1847–1857. [PubMed: 17964166]

58. Borenstein, E.; Sharon, E.; Ullman, S. Combining Top-Down and Bottom-Up Segmentation; CVPRW '04: Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04) Volume 4, IEEE Computer Society; Washington, DC, USA. 2004. p. 46ISBN 0-7695-2158-4

59. Paiva AR, Jurrus E, Tasdizen T. Using Sequential Context for Image Analysis. Proceedings of the International Conference on Pattern Recognition. 2010

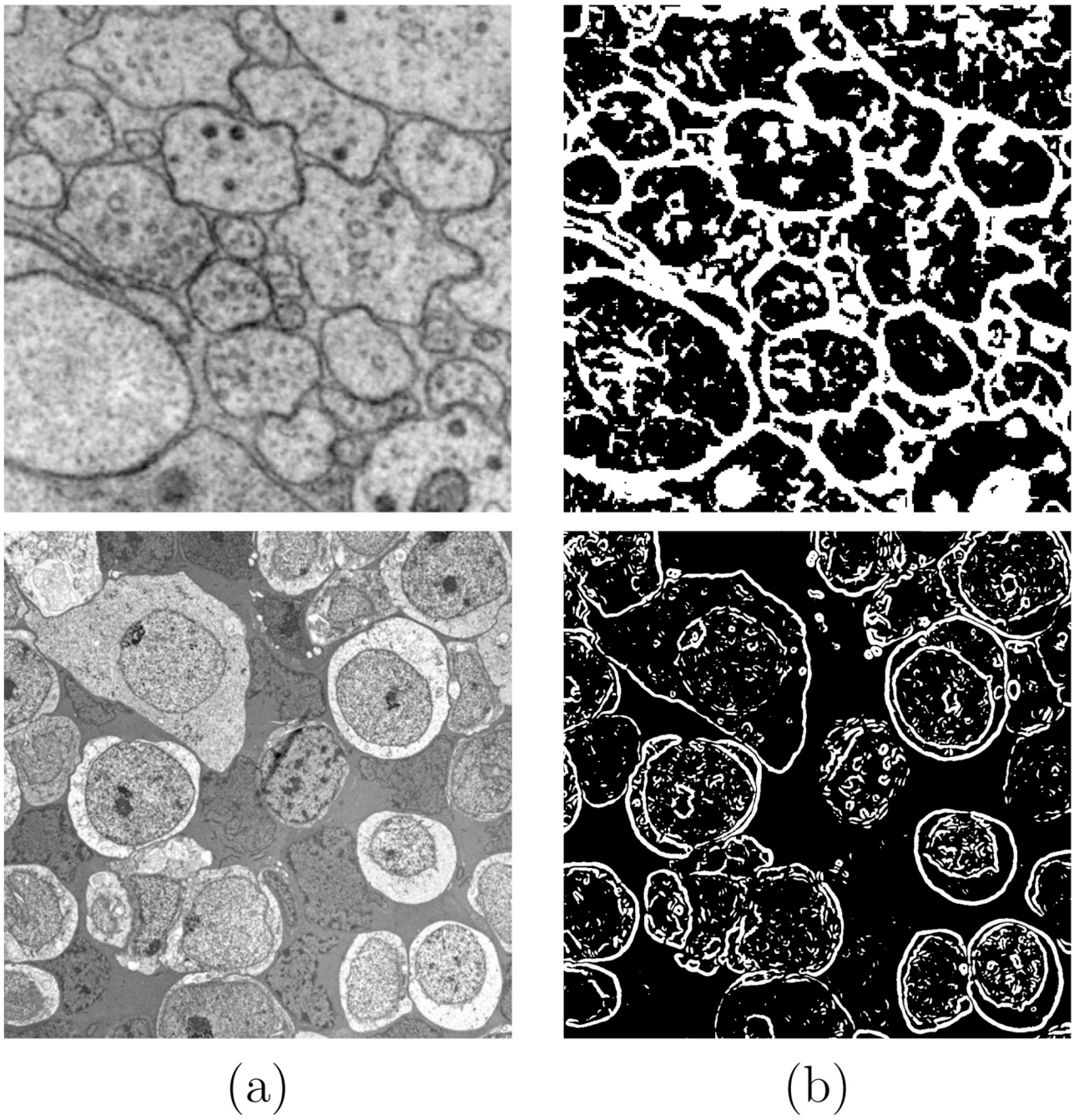

Figure 1.
(a) Example EM images. Top is from the *C. elegans*, bottom is from a rabbit retina. (b) Example membrane detection using thresholding after contrast enhancement and anisotropic directional smoothing to enhance membranes (top), and thresholding on the gradient magnitude (bottom). Both methods highlight the membrane boundaries but fail to remove internal structures.

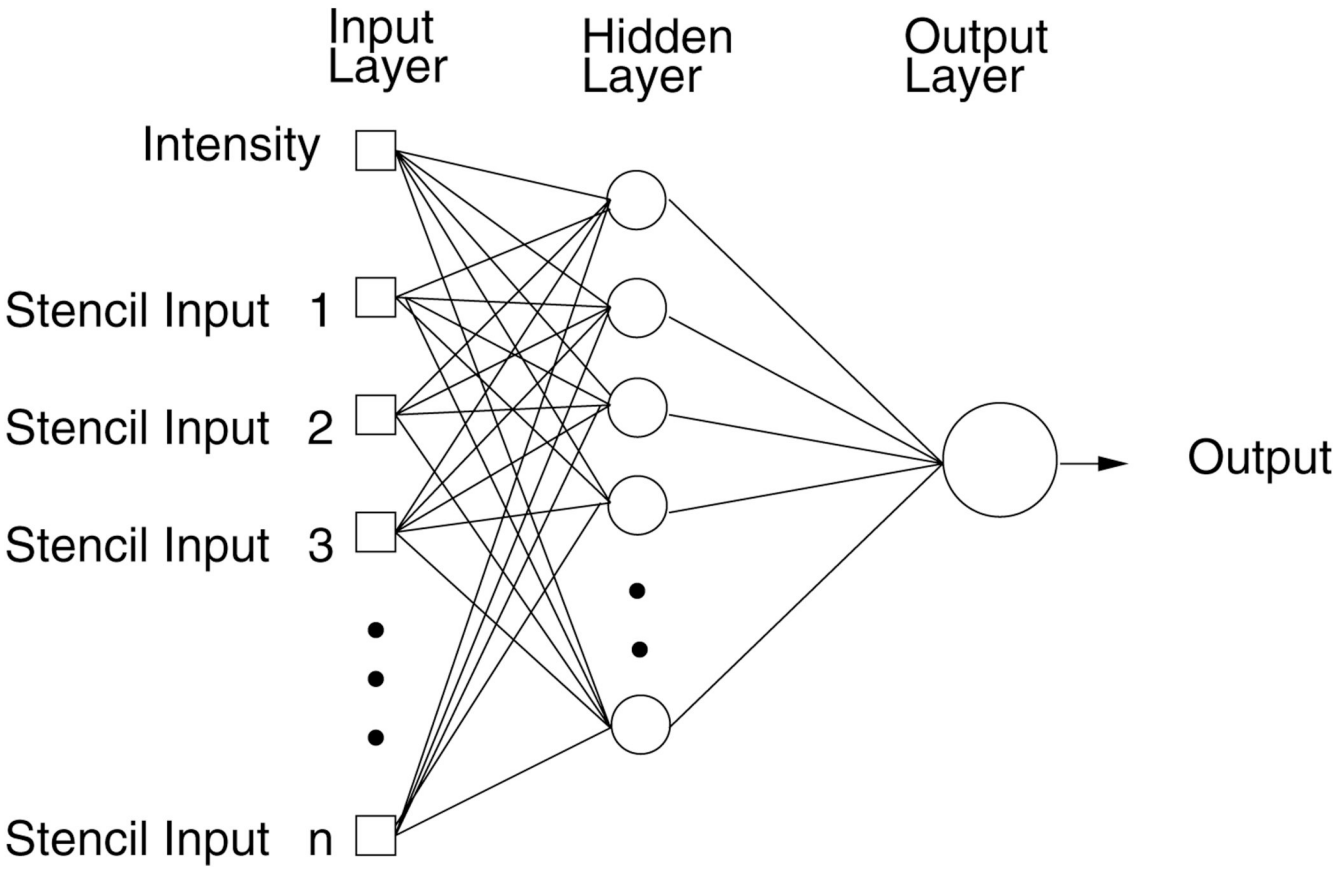

Figure 2.
Neural network diagram with one hidden layer. Inputs to the network include the image intensity and the values of the image at stencil locations.

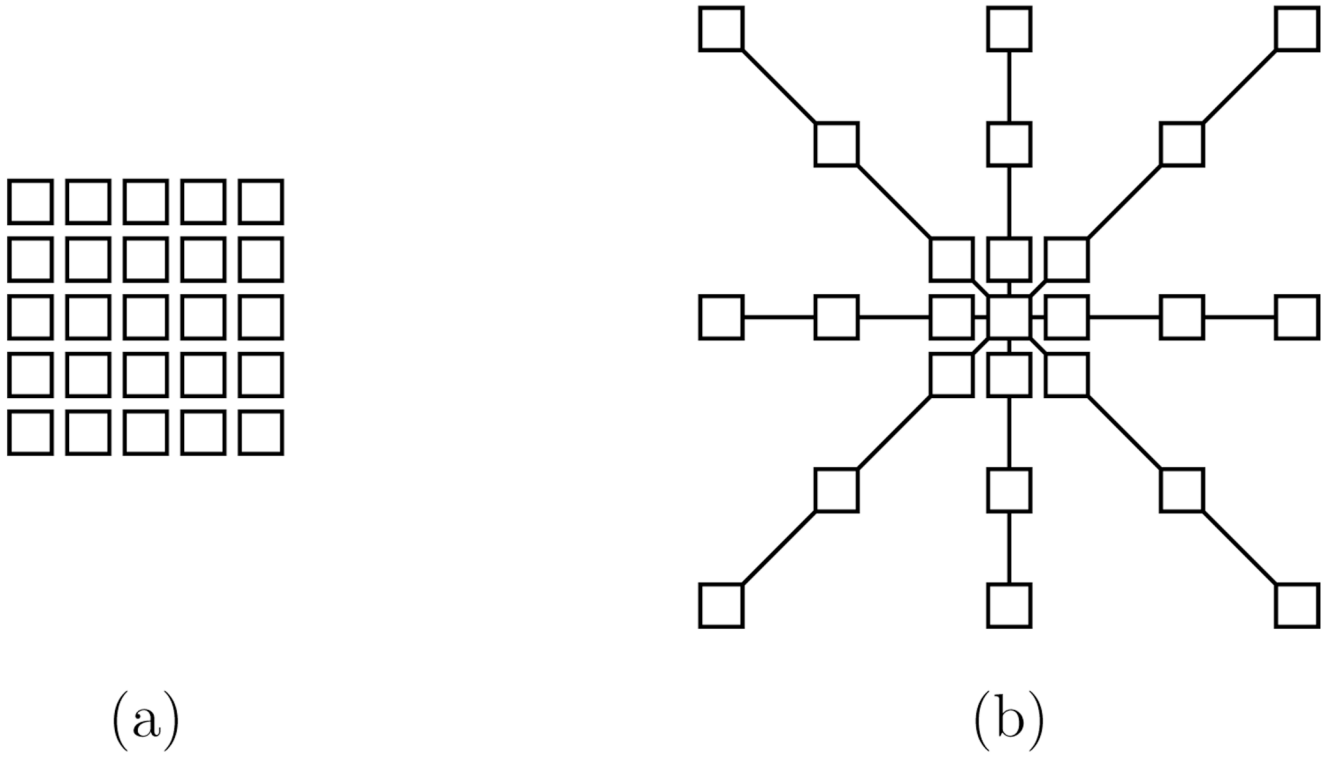

Figure 3.
Two image neighborhood sampling techniques: image pixels sampled using (a) a patch and (b) a stencil. For this example, the stencil contains the same number of samples, yet covers a larger area of the data. This is a more efficient representation for sampling the image space.

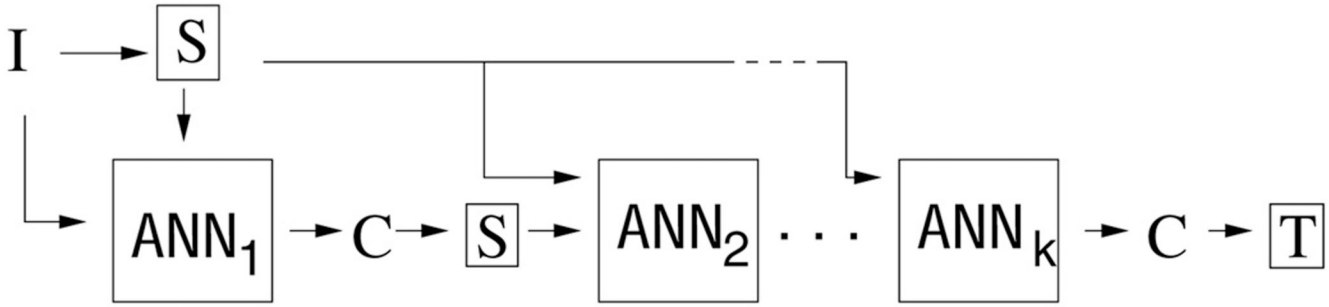

Figure 4.
Serial neural network diagram demonstrating the flow of information between ANNs. *I* is the original image, *C* is the output (probability map) from the classifier before thresholding, *S* is the stencil that samples the image data, and *T* is the final output from the classifier thresholded to produce a binary segmentation.

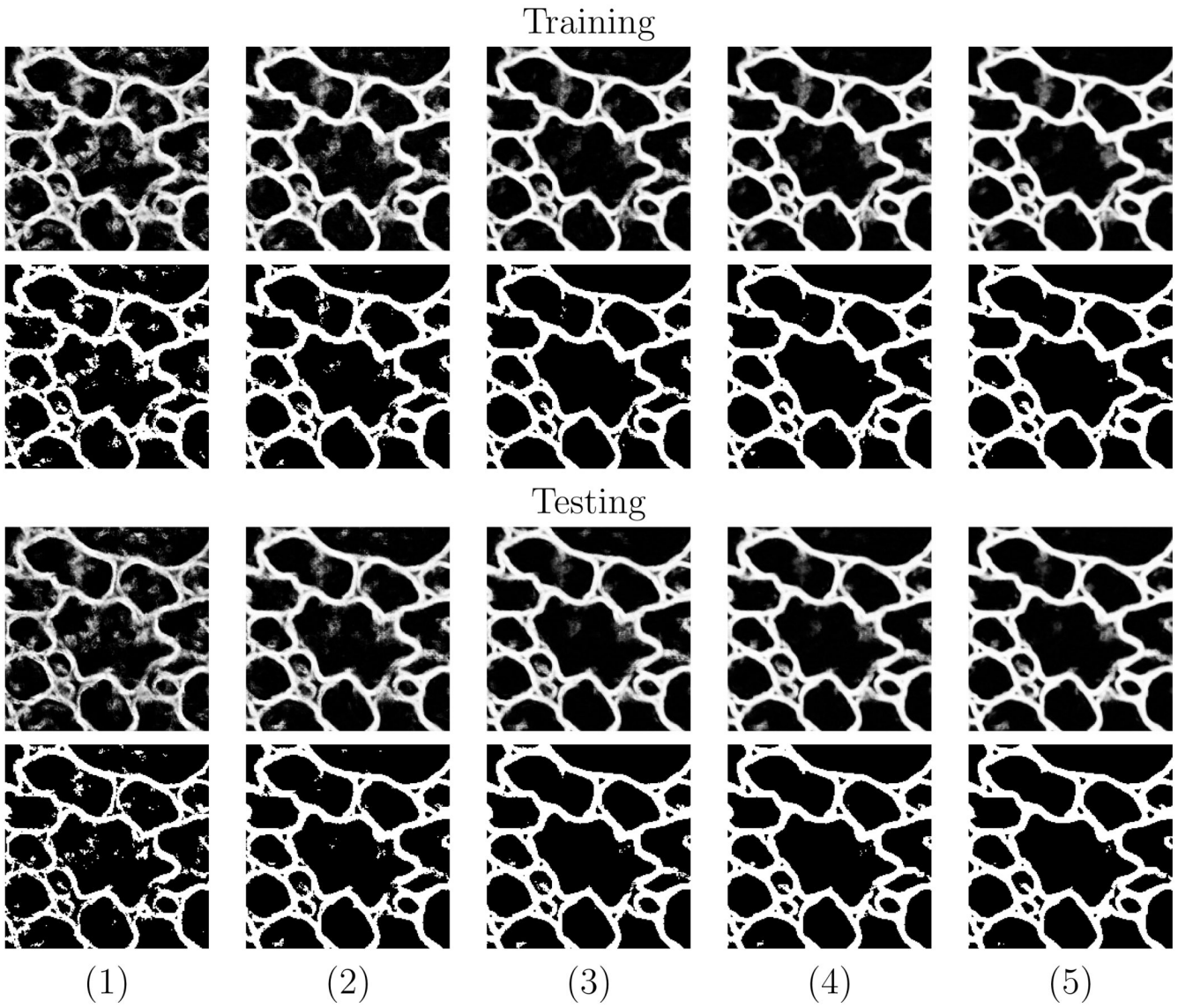

Figure 5.

An example of am image during training (top two rows) and testing (bottom two rows) at each stage (1–5) of the network series. The output from each network is shown in rows 1 and 3. Rows 2 and 4 demonstrate the actual membrane detection when that output is thresholded. The network quickly learns which pixels belong to the membranes within the first 2–3 stages, and then closes gaps in the last couple of stages.

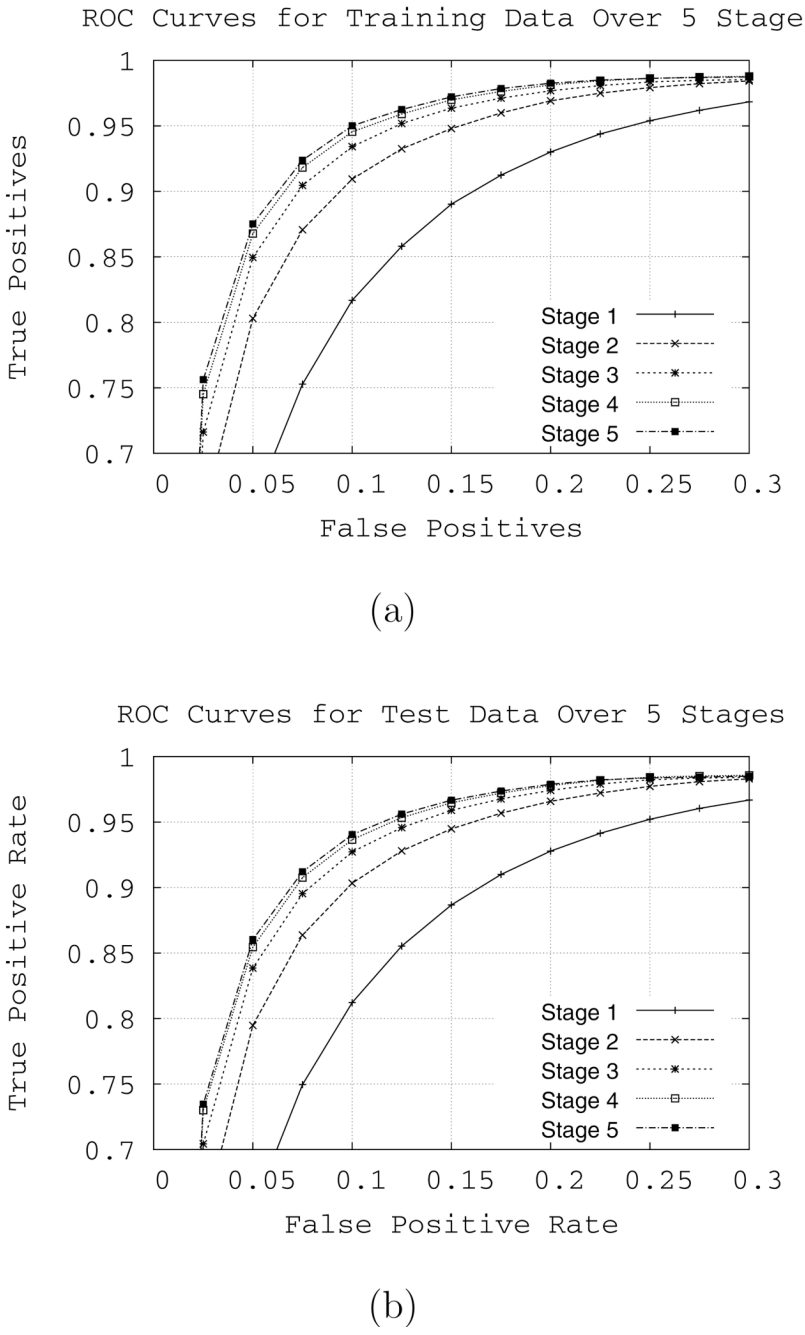

Figure 6.
ROC curves for the (a) training data and (b) testing data for each stage of the network.

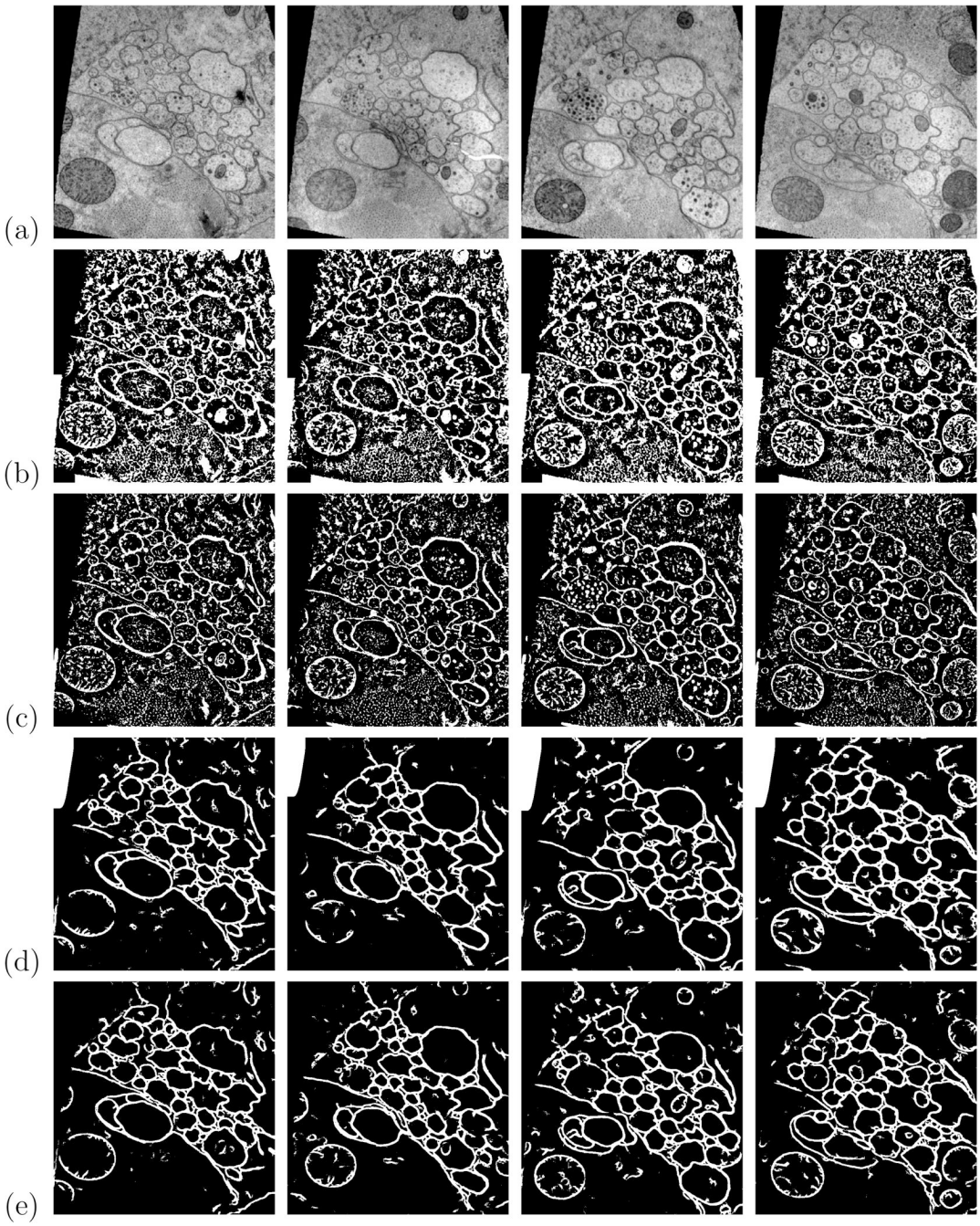

Figure 7.
(a) Cross-sections of the nematode *C. elegans* acquired using EM. Three demonstrated membrane detection techniques: (b) intensity thresholding after directional anisotropic smoothing [19], (c) thresholded boundary confidences from a single ANN trained using Hessian eigenvalues [24], (d) membrane detection from serial ANNs, trained using membrane filter banks and auto-context, and (e) membrane detection from serial ANNs, trained using image data sampled from stencils and auto-context.

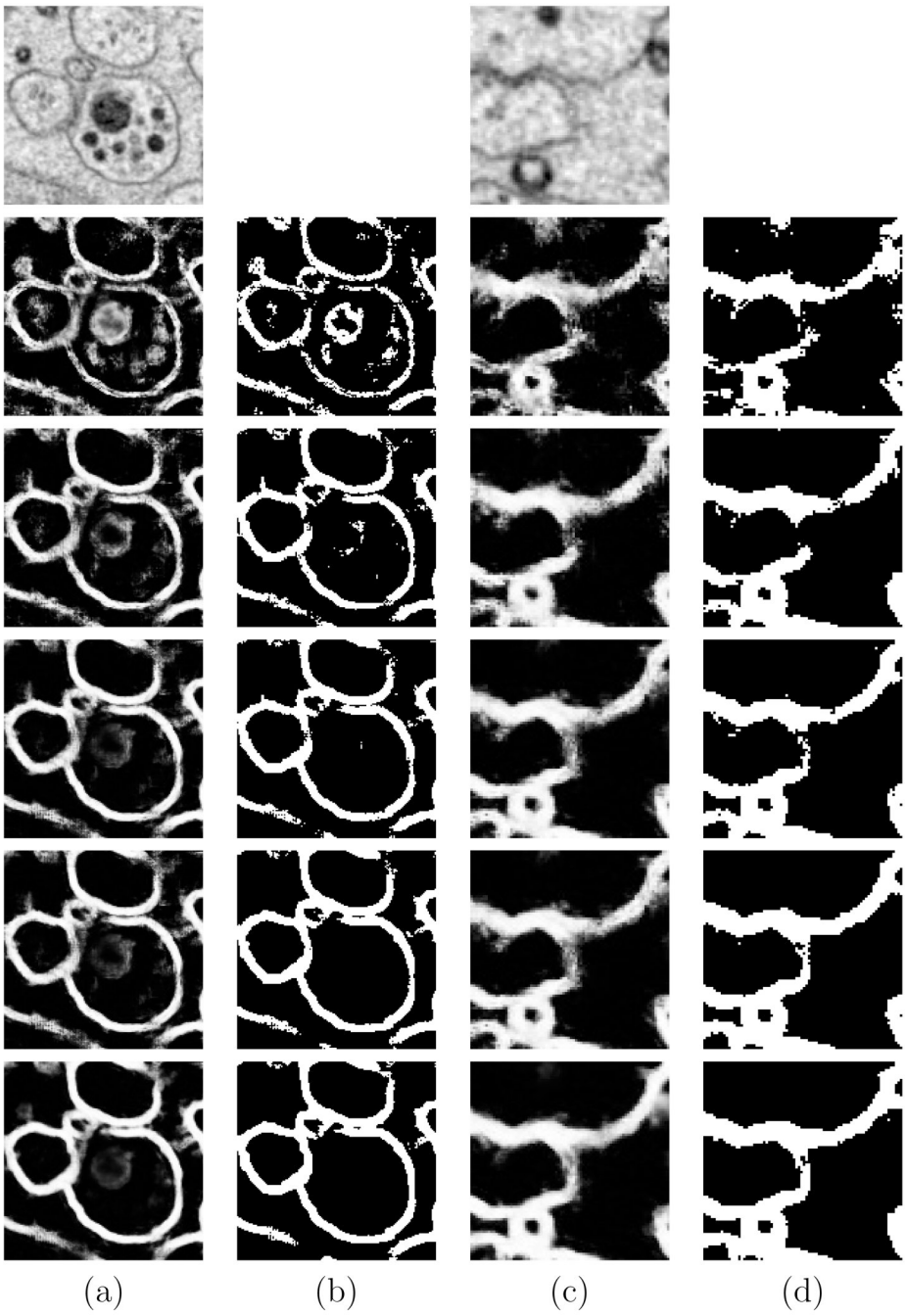

Figure 8.
Examples demonstrating how the proposed method removes intracellular structures (left two columns) and closes gaps in a weak membrane (right two columns). The top row is the original image, columns (a) and (c) show the classifier output, and columns (b) and (d) show the final thresholded segmentation.

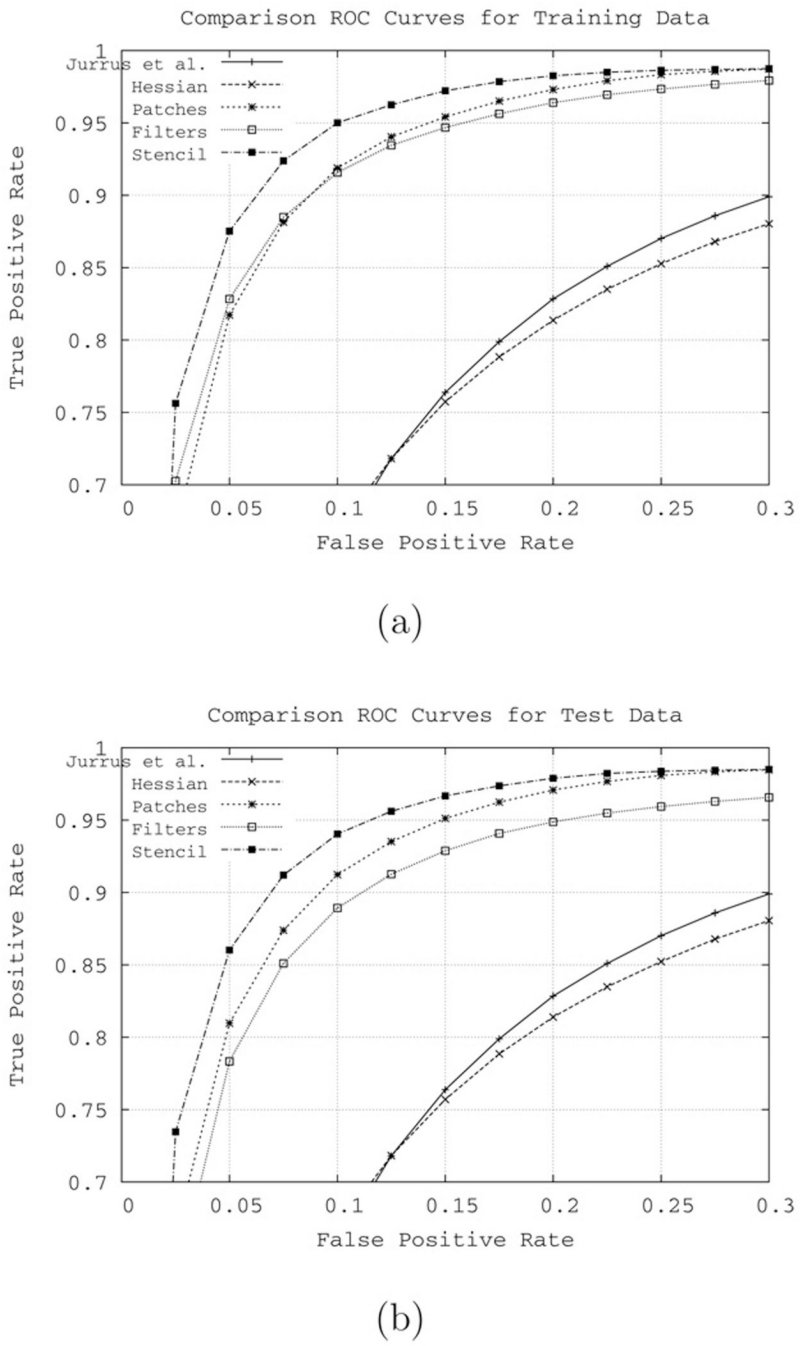

Figure 9.

ROC curves for (a) training and (b) testing on the *C. elegans* data. “Jurrus et al.”: thresholding after directional anisotropic smoothing [19]. “Hessian”: single layer neural network operating on Hessian eigenvalues similar to Mishchenko [24]. The remaining three curves demonstrate the results from different inputs to the proposed auto-context ANN approach.

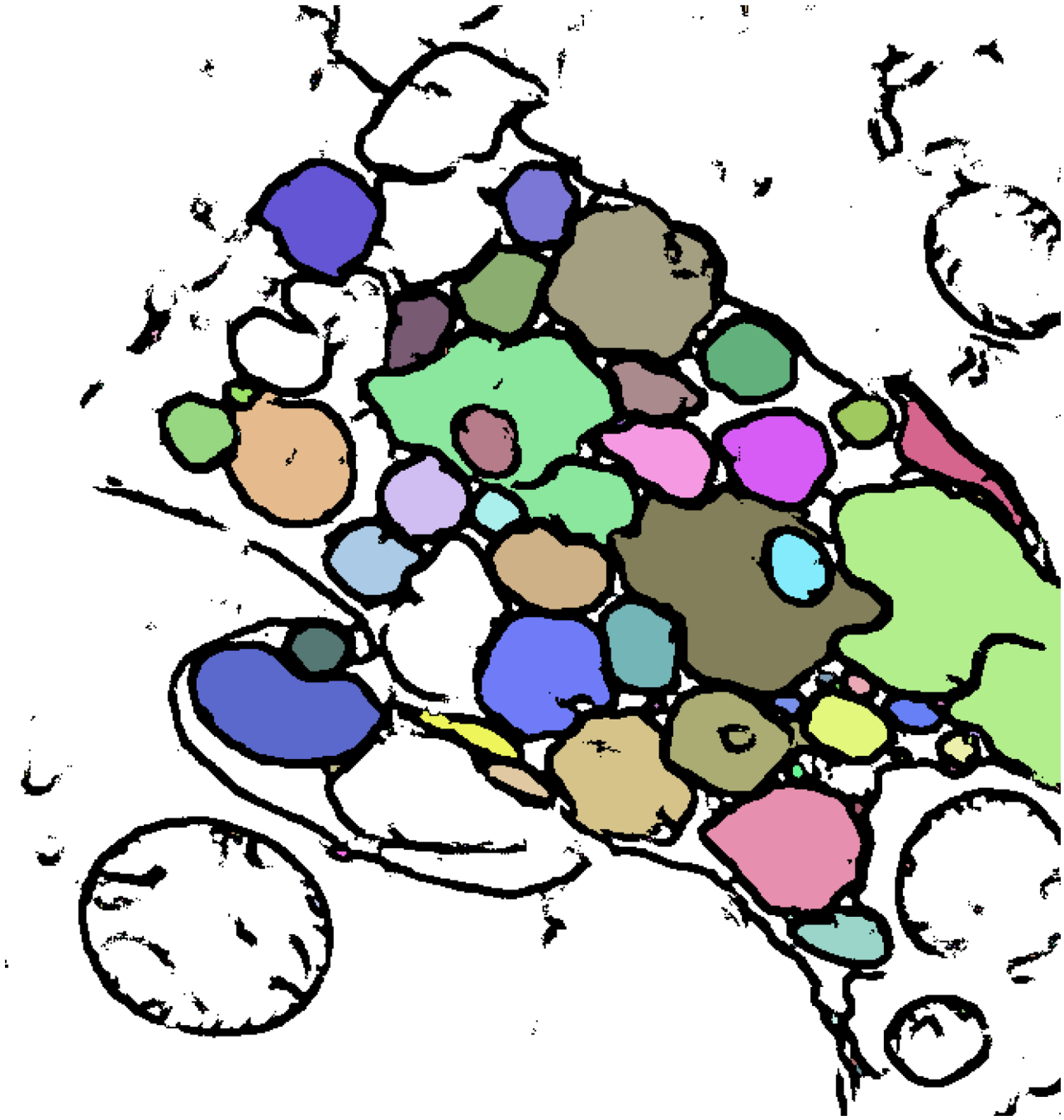

Figure 10.
[color] Segmentation of neurons using a flood-fill on the image of detected membranes. (a) Ground truth and (b) membranes detected with proposed method.

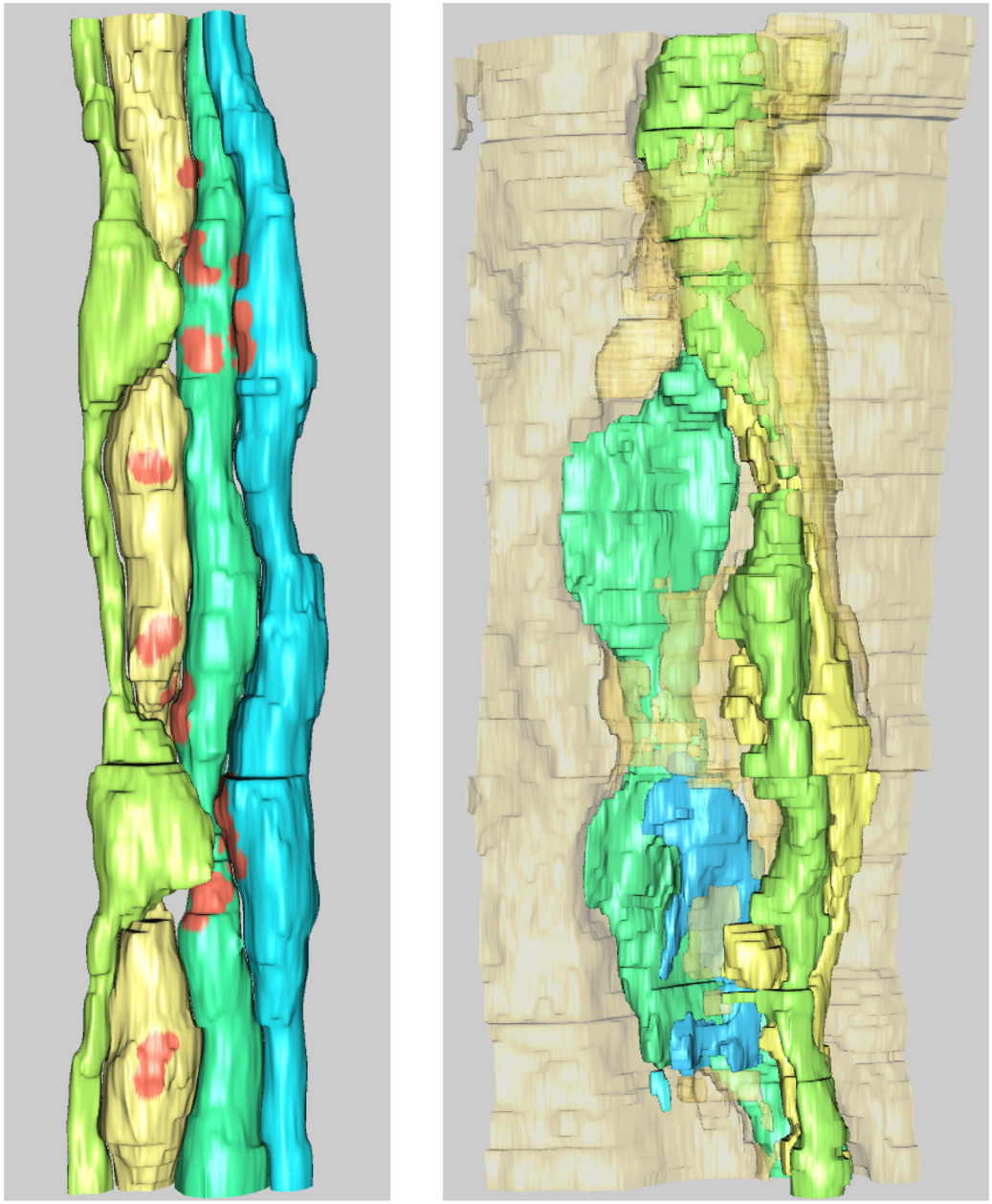

Figure 11.
[color] (a) 3-D renderings of the four neurons competing for information from the muscles. The location of the synapses, which were extracted from user specified locations, are shown in red on the neurons. (b) Similar rendering of the muscles that run alongside the motor neurons.

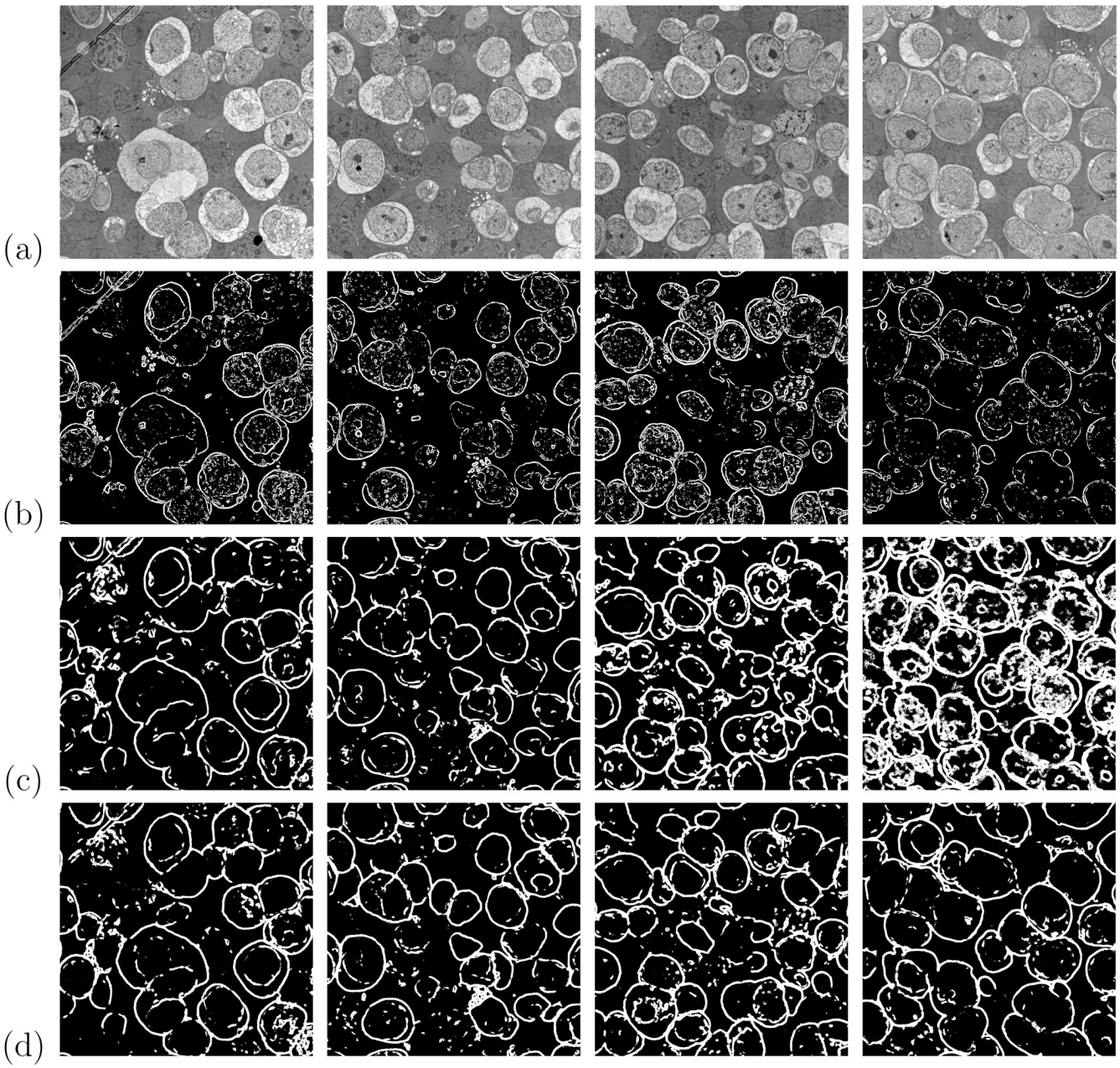

Figure 12.
(a) TEM images from a rabbit retina. Membrane detection with: (b) thresholding on the gradient magnitude, (c) serial ANNs using the output of an edge detection filter bank, and (d) serial ANNs using image intensities sampled from a stencil.

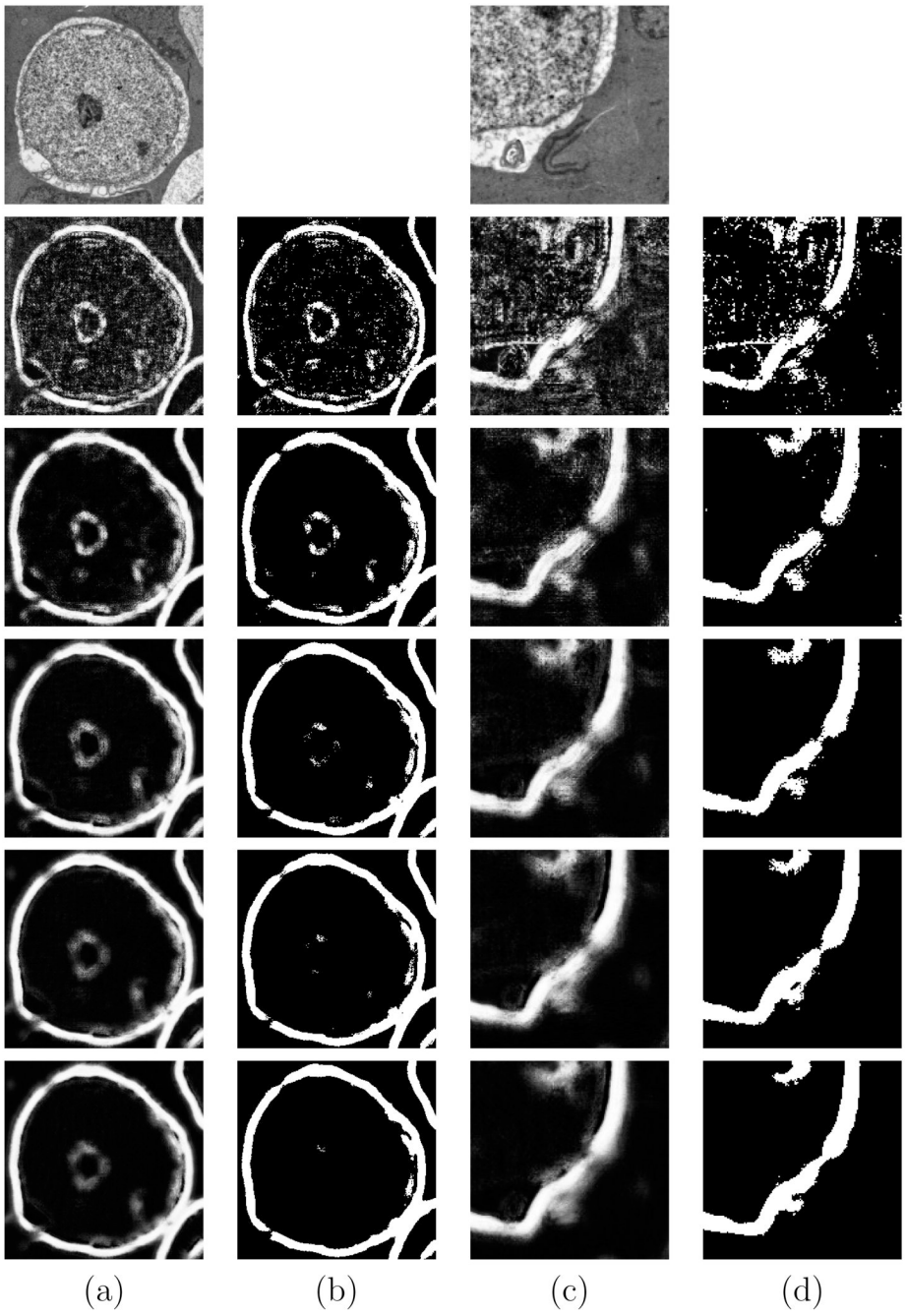

Figure 13.
Examples of locations in the data where intracellular structures are removed (left two columns) and gaps in membranes are closed (right two columns). The top row shows the raw images, columns (a) and (c) show the classifier output, and columns (b) and (d) are the final thresholded segmentations.

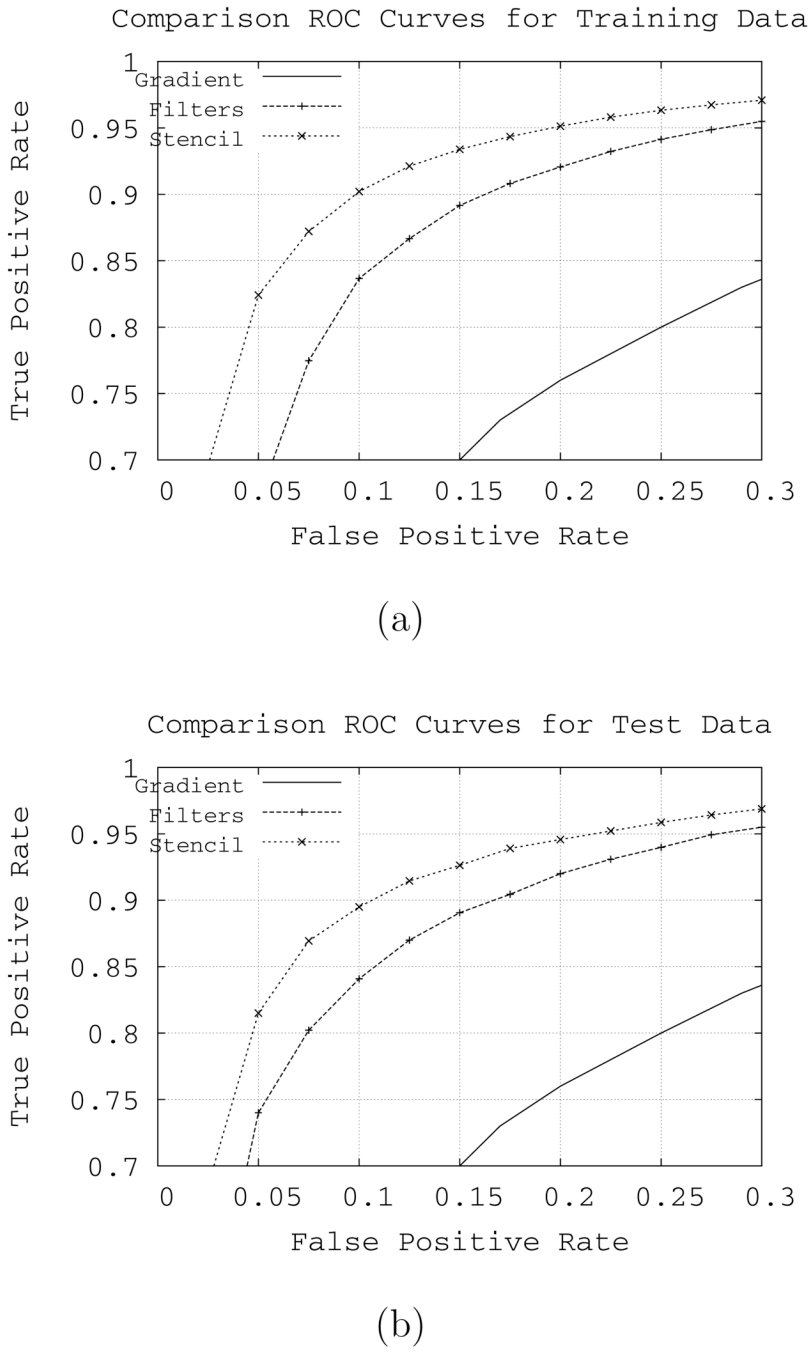

Figure 14.
ROC curves computed on the retina data for the (a) training data and (b) testing data. For comparison, an ROC curve is included that shows the best membrane detection when the gradient magnitude is thresholded.