Automatic segmentation of nine retinal layer boundaries in OCT images of non-exudative AMD patients using deep learning and graph search

LEYUAN FANG,1,2,* DAVID CUNEFARE,1 CHONG WANG,2 ROBYN H. GUYMER,3 SHUTAO LI,2 AND SINA FARSIU1,4

1Departments of Biomedical Engineering Duke University, Durham, NC 27708, USA
2College of Electrical and Information Engineering, Hunan University, Changsha 410082, China
3Centre for Eye Research Australia University of Melbourne, Department of Surgery, Royal Victorian Eye and Ear Hospital, Victoria 3002, Australia
4Department of Ophthalmology, Duke University Medical Center, Durham, NC 27710, USA

*leyuan.fang@duke.edu

Abstract: We present a novel framework combining convolutional neural networks (CNN) and graph search methods (termed as CNN-GS) for the automatic segmentation of nine layer boundaries on retinal optical coherence tomography (OCT) images. CNN-GS first utilizes a CNN to extract features of specific retinal layer boundaries and train a corresponding classifier to delineate a pilot estimate of the eight layers. Next, a graph search method uses the probability maps created from the CNN to find the final boundaries. We validated our proposed method on 60 volumes (2915 B-scans) from 20 human eyes with non-exudative age-related macular degeneration (AMD), which attested to effectiveness of our proposed technique.

© 2017 Optical Society of America

OCIS codes: (110.4500) Optical coherence tomography; (100.0100) Image processing; (100.2960) Image analysis.

References and links

31. Q. Yang, C. A. Reisman, K. Chan, R. Ramachandran, A. Raza, and D. C. Hood, “Automated segmentation of...


1. Introduction

Optical coherence tomography (OCT) can acquire 3D cross sectional images of human tissue at micron resolutions [1], which has been widely used for a variety of medical and industrial imaging applications [1,2]. The high resolution of OCT enables the visualization of multiple retinal cell layers and biomarkers of retinal and neurodegenerative diseases, including age-related macular degeneration (AMD) [3–5], diabetic retinopathy [6], glaucoma [7], Alzheimer’s disease [8], and amyotrophic lateral sclerosis [9]. For the study of many retinal diseases, accurate quantification of layer thicknesses in the acquired OCT images is crucial to advance our understanding of such factors as disease severity and pathogenic processes, and to identify potential biomarkers of disease progression. Moreover, segmentation of retinal layer boundaries is the first step in creating vascular pattern images from the popular new OCT angiography imaging modalities [10–13]. Since manual segmentation of OCT images is time consuming and subjective, it is necessary to develop automatic layer segmentation algorithms.

Over several decades, a multitude of OCT retinal segmentation algorithms have been developed, which can be generally classified into the following two categories: fixed mathematical model based methods and machine learning based methods [14,15]. Mathematical model based methods construct a fixed or adaptive model based on prior assumptions for the structure of the input images, and include A-scan [16,17], active contour [18–21], sparse high order potentials [22], and 2D/3D graph [23–30] based methods. Machine learning based methods formulate layer segmentation as a classification problem, where features are extracted from each layer or its boundaries and used to train a classifier (e.g. support vector machine, neural networks, or random forest classifiers) for determining layer boundaries [29,31–33].

In recent years, deep learning [34–36] based neural networks have been demonstrated to be a very powerful tool in the field of computer vision [37,38]. One of the most established realizations of deep learning is the convolutional neural network (CNN) [38], which automatically learns a hierarchy of increasingly complex features and a related classifier directly from training data sets. Recent works have extended the CNN framework to complex medical image analysis, such as retinal blood vessels segmentation [39,40], retinal hemorrhage detection [41], brain tumor segmentation [42], and cerebral microbleeds detection [43]. Very recently, a CNN based method was proposed [44] as an alternative to classic machine learning methods [5] for classification of normal and pathologic OCT images. In addition, the CNN model has also been applied to analyze OCT images of skin, aiming to characterize healthy skin and healing wounds [45].

Some of the more successful segmentation techniques are hybrids combining two or more approaches, e.g. in [46], active contours is combined with Markov random fields to create a global layer segmentation method. Other methods also combine the CNN model with additional techniques (e.g. conditional random fields, Markov random fields, and graph cut) for specific applications [47–51]. Most recently (after the conference version of our paper was accepted for presentation at the 2017 ARVO conference), a related method based on multi-scale convolutional neural networks combined with graph search was published for segmenting the choroid in OCT retinal images [51]. Our paper is in the same class of segmentation algorithms, which combines the CNN model with graph search methodology (termed as CNN-GS) for the automatic segmentation of nine layer boundaries on human
retinal OCT images. In this method, we first decompose training OCT images into patches. Then, we utilize a CNN to automatically extract representative features from patches centered on specific retinal layer boundaries and train a corresponding classifier to delineate nine layer boundaries. We use the CNN classifier to generate class labels and probability maps for the layer boundaries. Finally, we use a modified variation of our popular graph theory and dynamic programming (GTDP) method [23], in which instead of using gradient-based edge weights, we utilize these CNN-based probability maps to create the final boundaries. An exciting property of our approach is that it requires fewer ad hoc rules as compared to many previous fixed mathematical model-based methods for segmentation of inner retinal layers in non-exudative AMD eyes.

The rest of this paper is organized as follows. Section II briefly reviews the visible retinal layers in human non-exudative AMD OCT images, the 2D graph-based GTDP layer segmentation algorithm, and the CNN model. Section III details the proposed CNN-GS method for the layer segmentation of OCT images. Section IV presents experimental results on clinical human non-exudative AMD OCT data. Conclusions and suggestions for future works are given in Section V.

2. Review

In this Section, we briefly review the retinal layers visible in OCT images of non-exudative AMD subjects, the GTDP layer segmentation algorithm, and the CNN model.

2.1 Human non-exudative AMD OCT image

Figure 1 illustrates a representative retinal OCT image of a patient with non-exudative AMD, with layers labeled. Note that following the terminology of our previous work [52], we refer to the area between the apex of the drusen and RPE layer to Bruch’s membrane as retinal pigment epithelium and drusen complex (RPEDC) [53]. The difficulty in segmenting OCT images of non-exudative AMD as compared to normal eyes is the abnormal deformation (and ultimately atrophy) of the retinal layers, especially in the RPE layer in the form of drusen (highlighted by pink rectangles in Fig. 1). On another front, other normal anatomic (e.g., large vessels) and pathologic features (e.g., hyperreflective foci) affect the accuracy of segmentation algorithms developed for normal and diseased retina (e.g., the very basic implementation of the GTDP algorithm as discussed in the next subsection). Through the years, sophisticated software packages have been developed that apply a myriad of ad hoc rules to enhance the performance of these algorithms in the face of specific pathologies. Machine learning algorithms, such as the CNN-GS method described in the remainder of this paper, can be used as an alternative approach to reduce the reliance of segmentation techniques on ad hoc rules.
2.2. GTDP layer segmentation

The GTDP algorithm [23] represents each OCT B-scan as a graph of nodes where each node corresponds to a pixel. Neighboring pixels are connected in the graph by links called edges, and each edge is assigned a weight. The weights $w_{ab}$ for each edge between two nodes $a$ and $b$ are calculated based on the intensity gradients

$$w_{ab} = 2 - (g_a + g_b) + w_{\text{min}},$$

where $g_a$ and $g_b$ are the vertical intensity gradients at node $a$ and $b$, respectively, and $w_{\text{min}}$ is the minimum possible weight in the graph. The final step is to find a set of connected edges (called a path) that bisects the image. The GTDP method adopts Dijkstra’s algorithm [54] to select the minimum weighted path, which corresponds to a boundary between retinal layers.

2.3. CNN model

A CNN classifier uses a series of transforming layers to extract and classify the features from an input image. Commonly used layers for a CNN classifier include: 1) convolution layers; 2) pooling layers; 3) fully connected layers; and 4) soft max classification layers [38, 39]. Figure 2 illustrates a typical architecture of a CNN model, described in the following.

Assuming an input two dimensional image $x$ (of size $N \times M \times 1$), the first convolution layer convolves the image with $K_1$ different spatial kernels (of size $n_1 \times m_1 \times 1$) to obtain a three dimensional volume of feature maps (of size $N_1 \times M_1 \times K_1$). Later convolutional layers filter input volumes with $K_i$ different spatial kernels (of size $n_i \times m_i \times K_{i-1}$) to obtain a new volume of feature maps (e.g. of $N_i \times M_i \times K_i$). After the convolutions, each unit in a feature map is connected to the previous layer by the corresponding kernel’s weights. Applying multiple kernel convolutions will increase the number of feature maps, creating high computational burdens for future steps. Thus, a pooling layer is often applied after convolution layers, which fuses nearby spatial information (in a kernel window of size $w \times w$) with the max or averaging operations in order to reduce the dimensions of the feature maps [39]. After several convolutional and pooling layers, the high level features are combined in fully connected layers, where the outputs have full connections to all values in the previous layer, with separate weighs for each connection. Finally, a soft max classification layer is applied to the final fully connected layer, which determines the probability of the input image belonging to each class [38]. A CNN that has multiple convolutional, max pooling, and fully connected layers, is termed a deep neural network. In additional to the above four types of basic layers, other commonly used layers include rectified linear units (ReLU) layers and normalization layers [42]. The ReLU layers are usually applied after convolutional layers to non-linearly transform the data using the function $\max(0, x)$ . Here, $x$ is the input to the ReLU.
The simple transformation in ReLU layers can greatly accelerate the CNN training process [38]. The initial CNN layer weights are randomly selected. The training set is split into mini-batches, with \( B \) images per batch. Given a batch of training patches, the CNN uses multiple convolution and pooling layers to extract features and then classify each patch based on the probabilities from the softmax classification layer. After that, the CNN calculates the error between the classification result and the reference label, and then utilizes the backpropagation process [55] to tune all the layer weights to minimize this error. The above process will be repeated several epochs, until the whole CNN model becomes convergent. Here, an epoch is defined as when all batches have been seen, and multiple epochs are used for training [41]. More details about the CNN model can be found in [56].

3. CNN-GS framework for OCT segmentation

We propose the CNN-GS method, which combines CNN and graph search models for the automatic segmentation of OCT images. The CNN-GS method is composed of two main parts: 1) CNN layer boundary classification; and 2) graph search layer segmentation based on the CNN probability maps. The outline of the proposed CNN-GS algorithm is illustrated in Fig. 3.

3.1 CNN layer boundary classification

Since there are variations in intensity ranges between OCT images, we first perform intensity normalization on both the training and testing images. Following the intensity normalization method in [32], we first rescale the intensity values of the B-scan \( X \), \( I_X \), to be between \([0, 1]\)

\[
I \left( \frac{I_{\min}}{\max - \min}, \frac{I_{\max} - I_{\min}}{I_{\max} - I_{\min}} \right)
\]

where \( I_{\max} \) and \( I_{\min} \) stand for the maximum and minimum values in the B-scan \( X \), respectively. We then apply a median filter with a mask of size \( 20 \times 2 \). Next, we find the maximum pixel intensity value \( I_m \) from the whole filtered B-scan. We set the value of all pixels in the unfiltered intensity scaled B-scan that are larger than \( 1.05 \times I_m \) to \( 1.05 \times I_m \). Finally, the intensity values of all pixels are normalized by dividing by the max value in the B-scan. Then, we train a CNN to extract features of specific retinal layer boundaries and to classify nine layer boundaries on OCT images. Specifically, we assign labels “1-9” to layer boundaries from “ILM” to “BrM”. Any pixels that are not on the target layer boundaries, either in or out of the retina, are assigned the label “0”. In the training step, we first extract patches (of size \( 33 \times 33 \) pixels) centered on each pixel of nine manually segmented layer boundaries from OCT B-scans. These extracted patches are regarded as the positive training samples (with labels “1-9”). In addition, for each A-scan of the OCT B-scans, we randomly select one pixel from non-boundary regions (e.g. layer or background regions) and extract its
patch (also of size 33 × 33 pixels) to construct the negative training samples (with label “0”). Both the positive and negative training patches and their classifications are used to train the CNN. In this paper, we use a modified Cifar-CNN [38,57] architecture. After the training process, we obtain a new CNN model with optimized layer weights.

In the testing step, for each pixel of the test OCT image, we extract a patch (of size 33 × 33) centered on that pixel. Classification of all patches from each image would create a high computational burden. Therefore, we first utilize the GTDP algorithm [23] to attain a pilot segmentation of the ILM (top) and BrM (bottom) (see Fig. 1) boundaries and only use patches located between these two boundaries. Since there might be slight errors in the segmentation, the pilot estimation of ILM and BrM boundaries are moved up $T_{up}$ pixels and down $T_{down}$ pixels, respectively. Finally, we apply the trained CNN to each patch. The CNN outputs a class label and ten probabilities (corresponding to nine layer boundaries and non-boundary regions) for each patch. The output label and probabilities correspond to the center pixel of that patch taken from the full-sized image.

### 3.2 Graph search layer segmentation based on CNN probability map

The class labels from the CNN are often not precise enough to localize the layer boundaries. Therefore, we use the class probabilities for each pixel with a modified GTDP method to refine the boundaries. Specifically, as described above, we divide each OCT B-scan into 10 classes (consisting of 9 classes of boundaries and 1 class of non-boundary regions). Let $t$ be the class label. For the patch centered at pixel $a$, the CNN outputs 10 classes of probabilities $P_{a,t}, t \in \{0,1,2,...,9\}$. The sum of these probabilities for each patch is 1. The higher the probability value for one specific layer boundary, the more likely the pixel belongs to that boundary. We create probability maps for each class by extracting the corresponding probability for that class from all pixels in the image. The probability maps for 9 layer boundaries are illustrated in Fig. 4. For each pixel in the probability map of the $t$-th layer boundary, larger values correspond to a higher probability that this pixel will be classified as that layer boundary. As shown in Fig. 4(e)-4(m), to better visualize the probability maps, we used a color map where red and blue correspond to larger and smaller probability values, respectively.

To segment the $t$-th layer boundary, we use the probabilities from the $t$-th map to compute graph edge weights $w_{ab}^{Prob}$:

$$w_{ab}^{Prob} = 2 - (P_{a,t} + P_{b,t}) + w_{\text{min}},$$

where $a$ and $b$ are neighboring pixels. Finally, as in [23], we set the minimum graph weight ($w_{\text{min}}$) to $10^{-5}$ and use Dijkstra’s shortest path algorithm [54] to find the optimal path that bisects the image.

### 4. Experimental results

#### 4.1 Data sets descriptions

To validate the effectiveness of the proposed CNN-GS layer segmentation method, our experiments used 117 SD-OCT volumes from 39 participants with non-exudative AMD (50 years of age or older). This data set was originally introduced and described in detail in our previous study [58], which was approved by the human research ethics committee of the Royal Victorian Eye and Ear hospital and conducted in adherence with the declaration of Helsinki. All the participants were imaged at three time points over a 12-month period at 6-month intervals. The SD-OCT volumes were acquired using a Spectralis HRA + OCT device (Heidelberg Engineering, Heidelberg, Germany). Each volume includes 48 to 49 B-scans (of size 496 × 1024 pixels). Each B-scan is the average of up to 25 frames acquired at almost the
same position to reduce noise. Each B-scan was semi-automatically segmented by DOCTRAP software (Duke University, Durham, NC, USA). Specifically, DOCTRAP utilizes the GTDP algorithm [23, 52] as the core segmentation engine combined with a set of ad hoc rules to automatically segment the B-scans into eight layers. Next, the automatically segmented layers were carefully reviewed and corrected by an expert grader to attain the gold standard grading.

![Figure 4](image)

Fig. 4. (a) A sample OCT B-scan from our data set, where a sample A-scan is delineated with red. Examples of zoomed in patches extracted from this A-scan are shown in (b). The vertical light-to-dark and dark-to-light gradient images for (a) used in the GTDP algorithm [23], normalized to values between 0 and 1, are shown in (c) and (d). The probability maps created by the CNN for each of the target nine layer boundaries on this B-scan are shown in (e)-(m). Different colors (from blue to red, as shown in the colorbar on the right) represent the probability values in these probability maps. The CNN classification results for test image (a) are shown in (n), where for each pixel, the assigned class corresponds to the class with the highest probability value.
From the data set, we randomly selected 19 eyes for training the CNN model. 171 OCT B-scans from the 57 volumes were used (for each volume, one B-scan was taken from foveal region and from the peripheral regions above and below the fovea, respectively). We used the remaining 60 volumes from 20 eyes (2915 B-scans) for the testing. Note that, the CNN training data set was completely separate from the data set used for the testing.

4.2 Parameter setting

In our experiment, we adopted the Cifar-CNN architecture from the MatConvNet platform [59] (Downloaded at: http://www.vlfeat.org/matconvnet/) for training and testing the CNN model. The architecture and parameters of the Cifar-CNN used in our experiments are presented in Table 1. The parameters for the Cifar-CNN model in Table 1 were chosen to be the default values of Matconvnet. These parameters were already tuned by the Matconvnet researchers in constructing the Cifar-CNN model [59]. For the sake of completeness, we tested other parameters but did not achieve better results than with the default parameters. The patch size was chosen with respect to the resolution (pixel-pitch) of the images in our data set and the size of anatomical features of interest. The number of patches in each batch, B, for training was 100. The model was trained for 45 epochs, and the weight decay and learning rates were kept at their default values. In our experiments, utilizing more epochs did not significantly decrease the training error, but increased computational cost. The parameters for adjusting initial GTDP segmentation of the ILM and BrM boundaries ($T_{up}$ and $T_{down}$) were set to 15 and 20 pixels, respectively. These can avoid segmentation errors from the GTDP algorithm.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Type</th>
<th>Filter size</th>
<th>Stride</th>
<th>Filter number</th>
<th>Padding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer 1</td>
<td>Convolution</td>
<td>5 × 5 × 1</td>
<td>1 × 1</td>
<td>32</td>
<td>2</td>
</tr>
<tr>
<td>Layer 2</td>
<td>Max pool</td>
<td>3 × 3</td>
<td>2 × 2</td>
<td>—</td>
<td>0</td>
</tr>
<tr>
<td>Layer 3</td>
<td>ReLU</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Layer 4</td>
<td>Convolution</td>
<td>5 × 5 × 32</td>
<td>1 × 1</td>
<td>32</td>
<td>2</td>
</tr>
<tr>
<td>Layer 5</td>
<td>ReLU</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Layer 6</td>
<td>Average Pool</td>
<td>3 × 3</td>
<td>2 × 2</td>
<td>—</td>
<td>0</td>
</tr>
<tr>
<td>Layer 7</td>
<td>Convolution</td>
<td>5 × 5 × 32</td>
<td>1 × 1</td>
<td>64</td>
<td>2</td>
</tr>
<tr>
<td>Layer 8</td>
<td>ReLU</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Layer 9</td>
<td>Average Pool</td>
<td>3 × 3</td>
<td>2 × 2</td>
<td>—</td>
<td>0</td>
</tr>
<tr>
<td>Layer 10</td>
<td>Fully Connected</td>
<td>4 × 4 × 64</td>
<td>—</td>
<td>64</td>
<td>0</td>
</tr>
<tr>
<td>Layer 11</td>
<td>ReLU</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Layer 12</td>
<td>Fully Connected</td>
<td>1 × 1 × 64</td>
<td>—</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>Layer 13</td>
<td>Softmaxloss</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

4.3 Layer segmentation results

After training the CNN model, we then evaluated its performance using a separate test data set. We applied the proposed CNN-GS method, DOCTRAP software, and the publicly available OCTExplorer software (downloaded at: https://www.iibi.uiowa.edu/content/iowa-reference-algorithms-human-and-murine-oct-retinal-layer-analysis-and-display) [26, 60, 61] to the 60 OCT volumes from the test set and compared their results with the manually corrected segmentations. OCTExplorer is a 3D OCT layer segmentation software, which utilizes correlations among nearby B-scans for segmentation [60]. The latest version OCTExplorer 4.0.0 (beta) was used in our experiments. To have a fair comparison with OCTExplorer, we strived to match layer boundary delineations of the OCTExplorer and the gold standard grading. Note that there might exist a bias between the manual and OCTExplorer segmentations for each boundary. Such biases arise when the convention in marking the location of a certain layer boundary by one method is consistently different to that of another method. To correct for any bias, we applied pixel shifts to each segmented boundary from OCTExplorer and found the shift that minimized the absolute pixel difference.
with respect to the corresponding manually segmented boundary across the test data set. We found that the best results are attained when we shift each boundary in the automatic segmentation of the OCTExplorer down by bias values of 1, 1, 1, 1, 1, 1, 2, and 1 pixels, respectively.

To attain quantitative performance metrics, first, for each B-scan in the test data set, we calculated the mean thickness difference (in pixels) between the automated and manual segmentations for all layers (eight in the cases of DOCTRAP and CNN-GS, and seven in the case of OCTExplorer). Next, after taking the absolute value of these differences, the mean and standard deviation across all 2915 B-scans from the 60 volumes were calculated. These values are shown in Table 2. Note that, OCTExplorer does not segment the ONL-IS boundary, so a combination of the two layers (ONL + IS) is reported in Table 2 to allow for comparison between the methods. The total retinal thickness (in Table 2) stands for the thickness between the ILM and BrM boundaries. Figure 5 also illustrates the visual comparison results of the CNN-GS, DOCTRAP, OCTExplorer, and manual segmentation. As can be seen, the proposed CNN-GS method performed better than the OCTExplorer software on all segmented layers except the OS in terms of mean difference. However, it is important to note that the manual grader and CNN-GS aimed to segment the Brm boundary, while the OCTExplorer software targeted the outer boundary of RPE [26, 60, 61]. This mismatch in boundary definition in large has contributed to the differences of OCTExplorer with manual grading for the RPEDC layer and total retina thicknesses. On another front, for the sake of completeness, we have also reported the results of the automated DOCTRAP software before manual correction. It should be noted that since the manual grading is based on correcting the DOCTRAP results, there is a positive bias towards the reported accuracy of the DOCTRAP software.

Our CNN-GS algorithm was implemented in MATLAB R2016b and run on a desktop PC with a GPU NVIDIA GeForce GTX 980 equipped on an Intel Core i7 3.5 GHz machine. The average run time of our CNN-GS algorithm per B-Scan is 43.1 seconds.

Table 2. Differences (in pixels) in segmentation between manual grading and automated grading using our CNN-GS method, OCTExplorer software, and DOCTRAP software. The best results for each layer are labeled in bold. (*) emphasizes that OCTExplorer does not delineate the challenging Bruch’s membrane boundary and instead is targeted at segmenting the outer boundary of RPE, which in part explains the large differences with manual grading. (**) emphasizes that since manual grading was based on correcting the DOCTRAP results, there is a positive bias towards the reported DOCTRAP accuracy.

<table>
<thead>
<tr>
<th>Retinal Layer</th>
<th>CNN-GS vs manual grader</th>
<th>OCTExplorer vs manual grader</th>
<th>DOCTRAP ** vs manual grader</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean Difference</td>
<td>Standard Deviation</td>
<td>Mean Difference</td>
</tr>
<tr>
<td>NFL</td>
<td>0.57</td>
<td>0.58</td>
<td>0.68</td>
</tr>
<tr>
<td>GCL + IPL</td>
<td>0.46</td>
<td>0.67</td>
<td>0.65</td>
</tr>
<tr>
<td>INL</td>
<td>0.22</td>
<td>0.30</td>
<td>0.65</td>
</tr>
<tr>
<td>OPL</td>
<td>0.66</td>
<td>0.89</td>
<td>0.84</td>
</tr>
<tr>
<td>ONL</td>
<td>0.77</td>
<td>1.00</td>
<td>—</td>
</tr>
<tr>
<td>IS</td>
<td>0.16</td>
<td>0.29</td>
<td>—</td>
</tr>
<tr>
<td>ONL + IS</td>
<td>0.71</td>
<td>0.93</td>
<td>1.40</td>
</tr>
<tr>
<td>OS</td>
<td>0.98</td>
<td>0.96</td>
<td>0.76</td>
</tr>
<tr>
<td>RPEDC*</td>
<td>1.15</td>
<td>1.61</td>
<td>2.25</td>
</tr>
<tr>
<td>Total Retina*</td>
<td>1.26</td>
<td>1.24</td>
<td>3.40</td>
</tr>
</tbody>
</table>

5. Conclusions

In this paper, we presented a novel convolutional neural networks and graph search based method named CNN-GS for automatic segmentation of nine layer boundaries on non-
exudative AMD OCT images. The CNN-GS method utilizes a CNN to extract effective features of specific retinal layer boundaries and train a corresponding classifier to delineate eight layers. In addition, we further applied graph search methods on the probability maps generated by the CNN to obtain the final boundaries. Our experimental results on a relatively large data set of 60 OCT volumes (2915 B-scans) from non-exudative AMD eyes demonstrate the effectiveness of the proposed CNN-GS method. Note that we could have used a deeper CNN, which is expected to further improve our segmentation performance, but at the cost of a higher computational burden.

Fig. 5. Visual comparisons among the manual segmentations, OCTExplorer, DOCTRAP, and CNN-GS results on three non-exudative AMD images from the testing set.

A major disadvantage of the proposed method is its reliance on the availability of a large annotated data set. The black-boxed design of the CNN makes customization and
performance analysis of each step less tractable and reduces the options for customization. Also, there are still some parameters such as patch and filter size that are empirically selected. Moreover, in its current implementation, CNN-GS is more computationally intensive as compared to competing methods. We note that the majority of the computational cost is incurred by the CNN classification of each patch of the B-scan. Our CNN model is implemented in the Matconvnet platform and is expected to be accelerated by using the Caffe or Python platforms. In addition, classification of the large number of patches per B-scan creates a very high computational burden, and in our future work we plan to design a deep convolutional network model to process each B-scan as a whole to increase efficiency.

The results of this study are encouraging because of the simplicity and versatility of the proposed method. We emphasize that the core framework of the CNN-GS method is not specifically tailored for non-exudative AMD structures. This is in contrast to many previous automatic algorithms which required a multitude of ad hoc rules to make them suitable for segmenting images from AMD eyes (e.g. [52]). We expect that this framework will be applicable for many other types of disease by simply replacing or extending the training data set with manually segmented images of the disease of interest. It is expected that in some more challenging cases, modifications and customizations to this learning based technique will be needed. Such modifications are expected to be much less extensive than what is required for repurposing fixed mathematical model based methods. Thus, the proposed method is an important step toward the ultimate goal of attaining a universally applicable OCT segmentation software.

Note that, each 33 × 33 patch (e.g. Fig. 4(b)) is used to calculate only the probability values of the central pixel in that patch for the layer probability maps (e.g. Figs. 4(e)-4(m)). The fixed patch sizes and the (5 × 5) convolutional filters adopted in the CNN model are not considered to be optimal as they were chosen empirically. Therefore, one of our ongoing works is to design shape adaptive filters which can adjust their sizes according to the resolution of the OCT system, and the size of the anatomic and pathologic structures of interest.

Our segmentations were demonstrated to be close to semi-automatic grading, which we deem as the gold standard. However, we should emphasize that there is no guarantee that the gold standard human marking, even if we do not consider inter or intra grader variabilities, perfectly represent the true anatomic and pathological features of interest. OCT images, as in other ophthalmic imaging technologies [62], contain imaging artifacts. A good example is the variability in visualizing Henle's fiber layer, which severely affects the delineation of the OPL-ONL boundary [63].

In this paper, the proposed CNN-GS method was only trained and tested on non-exudative AMD SD-OCT images. In further publications, we will extend the proposed CNN-GS model to handle other kinds of pathologies seen in other diseases of the eye. In addition, there are strong incentives to apply the CNN methodology to other OCT image applications (e.g. image denoising, interpolation, and lesion detection).

**Funding**

Duke/Duke-NUS pilot collaborative grant and National Natural Science Foundation of China (NSFC) (61325007, 61501180).