IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 23, 2018, accepted December 10, 2018, date of publication December 24, 2018,
date of current version January 11, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2889321

Semi-Supervised Automatic Segmentation of Layer and Fluid Region in Retinal Optical Coherence Tomography Images Using Adversarial Learning

XIAOMING LIU[1,2], JUN CAO[1,2], TIANYU FU[1,2], ZHIFANG PAN[3], WEI HU[1,2], KAI ZHANG[1,2], AND JUN LIU[1,2]

[1]College of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan 430065, China
[2]Hubei Province Key Laboratory of Intelligent Information Processing and Real-Time Industrial System, Wuhan 430065, China
[3]Information Technology Center, Wenzhou Medical University, Wenzhou 325035, China

Corresponding author: Xiaoming Liu (lxmspace@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61403287, Grant 61472293, and Grant 61572381, in part by the Foundation of Wenzhou Science and Technology Bureau under Grant 2018ZG016, and in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LY16F030010.

ABSTRACT Optical coherence tomography (OCT) is a primary imaging technique for ophthalmic diagnosis due to its advantages in high resolution and non-invasiveness. Diabetes is a chronic disease, which could cause retinal layer deformation and fluid accumulation. It might increase the risk of blindness, and thus, it is important to monitor the morphology change of the retinal layer and fluid accumulation for diabetes patients. Due to the existence of deformation and fluid accumulation, the retinal layer and fluid region segmentation in the OCT image is a challenging task. Machine learning-based segmentation methods have been proposed, but they depend on a significant number of pixel-level annotated data, which is often unavailable. In this paper, we proposed a new semi-supervised fully convolutional deep learning method for segmenting retinal layers and fluid regions in retinal OCT B-scans. The proposed semi-supervised method leverages the unlabeled data through an adversarial learning strategy. The segmentation method includes a segmentation network and a discriminator network, and both the networks are with U-Net alike fully convolutional architecture. The objective function of the segmentation network is a joint loss function, including multi-class cross entropy loss, dice overlap loss, adversarial loss, and semi-supervised loss. We show that the discriminator network and the use of unlabeled data can improve the performance of segmentation. The proposed method is investigated on the duke Diabetic Macular Edema dataset and the POne dataset, and the experiment results demonstrate that our method is more effective than the other state-of-the-art methods for layers and fluid segmentation in the OCT images.

INDEX TERMS Adversarial learning, convolutional neural networks, image processing, layer segmentation, optical coherence tomography.

I. INTRODUCTION

Optical Coherence Tomography (OCT) [1], [2] is an important noninvasive imaging modality for scanning cross-section of biological tissues, it can generate high resolution scans. The widely used commercial OCT technology is spectral domain OCT (SD-OCT) whose axial resolution is $3 \sim 7\mu m$ in biological tissue. In this technology, the reflectance of subsurface structures can be visualized through low coherence infrared light. Because its advantages of great imaging speed and resolution, it has been utilized for imaging the cross-section of the retina and cornea at micrometer resolution level [3]. The cross-sectional information has been widely used in ophthalmology to assist early diagnosis and prognosis of retinal diseases.

Many diseases can cause ophthalmic lesion, for instance, diabetes may suffer from Diabetic Macular Edema (DME) which is caused by hyperglycemia. Hyperglycemia can rupture the epithelium of blood vessels. Moreover, it can also

damage the retinal fluid transport cells which makes the leaked fluid accumulate between retinal layers [4], [5]. Without treatment, DME may result in blindness. Furthermore, with the growth of age, people might suffer from Age-related Macular Degeneration (AMD). For better diagnosis of these diseases, retinal layer and disease affected region segmentation is a crucial step. An example OCT image and its corresponding manual annotations are shown in Figure 1.

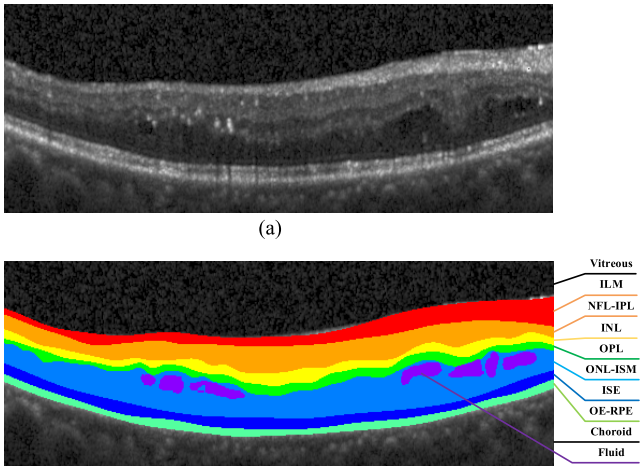


FIGURE 1. Retinal OCT image with nine layers and eight boundaries, (a) is the original image and (b) is the segmented image. NFL is Nerve Fiber Layer; GCL is Ganglion Cell Layer; IPL is Inner Plexiform Layer; INL is Inner Nuclear Layer; OPL is Out Plexiform Layer; ONL is Outer Nuclear Layer; ISM is Inner Segment Myeloid; ISE is Inner Segment Ellipsoid; OS is Outer Segment; RPE is Retinal Pigment Epithelium.

Automatic retinal layer segmentation methods have been widely investigated since manual segmentation is time-consuming and suffers from inter-subject variation. Previous automatic retinal layer segmentation methods include active contours [6], [7], level sets [8], [9], dynamic programming [10]–[12] and graph cut [13], [14]. Duan *et al.* [15] proposed a groupwise curve alignment based method to segmentation retinal layer. All these methods require well-designed prior knowledge of experts. With the rapid development of machine learning [16]–[18], several machine learning based layer segmentation methods have been proposed [19]–[21]. Chiu *et al.* [19] developed a kernel regression based fully automatic method to segment seven retinal layers and fluid-filled regions on retinal SD-OCT images with DME. Karri *et al.* [20] proposed a structured random forest based method with traditional graph theory for eight layer-specific edges detection.

In recent years, deep learning has become the primary approach in computer vision tasks, such as denoising [22], [23], classification [24]–[26], segmentation [27]–[29] and detection [30], [31]. It has also been introduced into medical image processing [32], [33], including a few works on layer segmentation in OCT images [4], [34]–[40].

Although these deep learning-based segmentation approaches have achieved great success on different tasks, they demand a mass of training samples. Simple transformations, such as cropping, flipping, and rotating input images have been widely applied to reduce the overfitting problem, Generative Adversarial Networks (GANs) [41] has been shown as a powerful technique for generation of new images in unsupervised learning.

Inspired by the GANs based segmentation [42], [43] and ReLayNet [4] in OCT image segmentation, in this paper, we proposed a semi-supervised retinal layer and fluid region segmentation method through adversarial learning with encoder-decoder network architectures. Generally, GAN includes a generator and a discriminator. The generator takes a random vector as input and maps it into a synthetic image. The discriminator, which takes a real image or a synthetic image generated by the generator as input, is trained to identify whether the input is real or synthetic. The training of GANs is a max-min game, specifically, the generator is trained through minimizing the GANs loss and the discriminator is trained to maximize the GANs loss. Different from the general GANs, we designed an adversarial learning method for retinal layer and fluid region segmentation.

The proposed method (denoted as SGNet, **S**emi-supervised segmentation with **G**AN **Net**) includes a segmentation network and a discriminator network, both networks are trained alternately. Although the acquisition of medical image is not difficult, the annotation of them is hard to acquire, thus a mass of medical image without annotation are wasted in previous methods. In order to take advantage of these unlabeled data, we introduce semi-supervised learning in this paper, following the work in [43]. The segmentation network is trained to output a segmentation probability maps of all categories through minimizing a joint loss function that combined weighted cross entropy, weighted dice loss, adversarial loss and semi-supervised loss. The discriminator network is trained to discriminate the segmentation prediction maps from the ground truth maps through minimizing discriminate loss. Both the networks are with encoder-decoder architectures. The alternate training strategy encourages the segmentation network to generate predicted label maps closer to the ground truth label maps. Figure 2 shows the training flowchart of the proposed method.

The contributions of the paper include:

(1) We proposed a novel semi-supervised automatic segmentation method for layer and fluid regions segmentation in OCT images, which can take advantage of labeled data more effectively and to make use of unlabeled data, based on the work in [4] and [43]. Both the segmentation and discriminator are with encoder-decoder architectures, and the segmentation network is trained in an adversarial and semi-supervised manner.

(2) Based on the observation that large kernel size which can bring large effective receptive field plays an important role in semantic segmentation, we modify a U-net architecture by global convolutional layer (GCL) [44] to acquire large

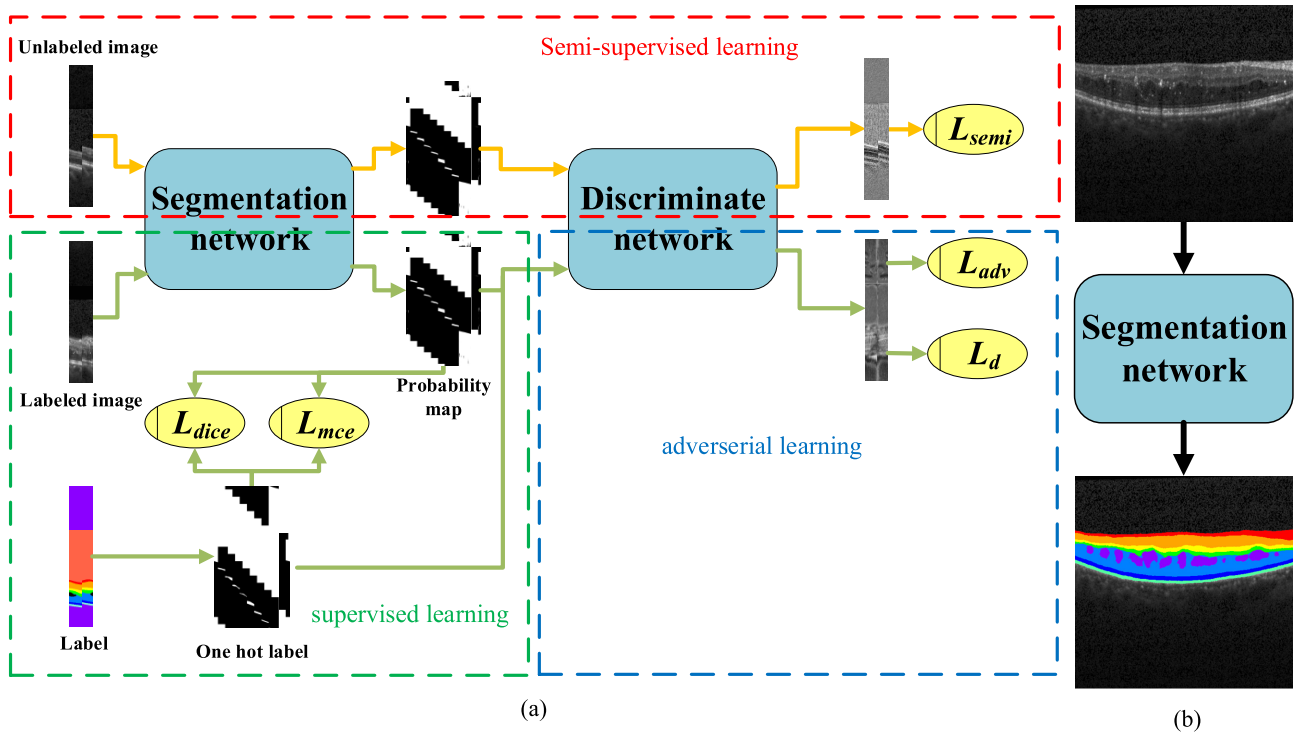


FIGURE 2. The framework of the proposed method for layer and fluid regions segmentation in OCT images. (a) The training stage, the data flow of semi-supervised training (optimized by minimizing L_{semi}), supervised training (optimized by minimizing L_{mce} and L_{dice}) and adversarial training (optimized by minimizing L_{adv} and L_d) are represented in red dashed box, green dashed box and blue dashed box respectively. It is noted that the probability map and one hot label in the figure belong to one input image, since there are 10 classes, they are shown as 10 "images" with each one represents a channel. (b) Testing stage, input a test image and output the segmentation result.

receptive filed with few parameters. Furthermore, boundary refinement layer (BRL) [44] is utilized to further refine the segmentation result. To deal with the class imbalance problem, we weighted the cross-entropy loss with a weighting strategy. Besides, we utilized a weighted dice loss to further improve the segmentation result.

(3) To the best of our knowledge, this is the first time that the adversarial learning, which can leverage unlabeled images to improve performance, is used for segmenting retinal layer and fluid region in retinal OCT images.

The rest of the paper is organized as follows. The section II reviews the related work. The following section III describes the proposed method. Experimental results are shown in the section IV. We conclude the paper in the last section V.

II. RELATED WORK

In this section, we review some of the works related to our research, including semantic segmentation, GANs and closely related retinal layer segmentation techniques.

A. SEMANTIC SEGMENTATION

Semantic segmentation is a basic task in computer vision that assigns each pixel of the image a label. Recently, a number of deep learning based methods have been proposed for semantic segmentation. The application of deep learning in semantic segmentation was originally based on image patch classification [45], which has a large computational burden. To reduce computational burden, Shelhamer *et al.* [27] proposed a fully convolutional networks (FCN) that is modified from VGG16 and used pre-trained weights. The FCN combines feature maps of different resolution through upsampling them to the image size, thus FCNs can take arbitrary-size inputs and is faster than patch-based classification method. However, due to the pooling operation, some detailed information in the feature map might be lost. While upsampling is used to compensate the reduction of resolution caused by pooling, it will make the final segmentation result blurred. Till now, a mass of the subsequent state-of-the-art semantic segmentation methods have adopted this paradigm, such as DeepLab [29] series.

Researchers proposed two different forms of structure to improve the segmentation result. One is the encoder-decoder architecture, also a fully convolutional architecture that has be widely utilized. In this structure, the encoder block reduces the spatial dimension of feature maps gradually with subsampling layer (such as pooling layer). In contrast to encoder block, the decoder block gradually recovers spatial dimension and information by upsampling layer (such as interpolation, unpooling layer and deconvolution layer). The feature maps from encoder blocks and those corresponding in decoder blocks are concatenated in the channel dimension to assist in information recovery. In medical image segmentation, one of

the most popular architecture is U-Net [28]. It has been used in many studies and extended to three-dimensional image segmentation [46].

Another structure is to use dilated convolution instead of pooling operation. Pooling operation increases the receptive field which is beneficial for classification. However, the pooling operation reduces the resolution of feature maps which is harmful for semantic segmentation. To mitigate the negative effects of pooling operation, [29] and [47] utilized dilated convolution, which can increase the corresponding receptive field without reducing the space dimension.

To further improve the segmentation performance, DenseCRF [48] can be utilized to further refine the segmentation result of the CNN model.

B. ADVERSARIAL LEARNING

The GANs framework is proposed by Goodfellow [41] and is inspired by two-player game. GAN consists of a generator G and a discriminator D. The generator G, which takes a random noise z that follows a simple distribution (such as Gaussian distribution) as input and outputs a synthesis image to approximate the distribution $p_{x\sim p_{data}(x)}$ of sample data x. The discriminator is a binary classifier which estimates the probability that a sample comes from training data $p_{x\sim p_{data}(x)}$. The optimization of GANs is a minimax two-player game with the following formulation [41]:

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}[\log(D(x))] + E_{z\sim p_z(z)}[\log(1-D(G(z)))] \quad (1)$$

D and G are trained alternately to achieve Nash equilibrium.

For semantic segmentation, Luc *et al.* [49] first introduced adversarial learning into sematic segmentation. They utilized an extra discriminator which just identify the whole input image as real or fake to encourage the results of the segmentation network to be more reliable. Souly *et al.* [42] treated the segmentation network as a discriminator and used a generator to expand the training data, they also used unlabeled data to conduct semi-supervised learning. More recently, different from the above two methods, Hung *et al.* [43] proposed a fully convolutional discriminator to output a predicted probability map, on this basis the whole system can be trained in a semi-supervised and adversarial manner. Different from [42] and [49] while similar to [43], we use two FCN alike networks whose output sizes are the same as the input image. While a general FCN is used in [43], we used two modified U-net [4], [28] as our segmentation network and discriminator network. Besides, the loss function in the proposed method has incorporated Dice loss besides the terms used in [43].

C. LAYER SEGMENTATION IN OCT IMAGES

As presented above, layer segmentation has been intensively investigated in recent decades and several methods have been proposed [6]–[11], [13]–[15], [19]–[21]. Here we only give a brief introduction to some closely related CNN based methods.

Fang *et al.* [34] developed a method that combined convolutional neural networks (CNN) and graph search methods to automatically segment the nine retinal layer boundaries on retinal OCT images. Liu *et al.* [35] integrated hand-designed features and CNN-learned features to train a structured random forest classifier. The trained classifier is used to predict the layer interface, and the final segmentation is obtained with shortest path. Due to the patch-based characteristic, these methods have the disadvantage of high computational burden. Shah *et al.* [37] proposed a method for automatically segmenting 3-D surfaces and evaluated on simultaneous intraretinal layer segmentation of OCT images. Roy *et al.* [4] proposed ReLayNet to simultaneously segment retinal layer and fluid region in retinal OCT images, which greatly reduce computation burden compared to patch-based methods. ReLayNet is based on the U-Net [28], a fully convolutional network that has been widely used in image segmentation. Other convolutional network frameworks have also been used for retinal thickness segmentation [38], [39] and pigment epithelium detachment segmentation [40].

III. THE PROPOSED METHOD

In this section, we present the proposed segmentation method (SGNet) of OCT layers and fluid regions in details. The notations used in the paper are listed in Table 1. We treat this segmentation task as a classification problem with 10 categories. These 10 categories include: Region above the retina (RaR), Inner limiting membrane (ILM), Nerve fiber ending to Inner plexiform layer (NFL-IPL), Inner Nuclear layer (INL), Outer plexiform layer (OPL), Outer Nuclear layer to Inner segment myeloid (ONL-ISM), Inner segment ellipsoid (ISE), Outer segment to Retinal pigment epithelium (OS-RPE), Region below RPE (RbR) and Fluid region.

TABLE 1. Notations used in the paper.

Variable	Note	Variable	Note
X_n	An OCT image with index n from the labeled data	Q_n	Y_n or P_n
Y_n	The ground-truth of an image	i	A pixel in an image
P_n	Probability map	α,β,γ	The weight parameters in loss function
$h\times w\times c$	Height×width×channels	$W_n^{(i)}$	The weight of pixel i in image X_n
λ	Weight parameter in equation (5)	U_n	An OCT image from unlabeled data
N_c	The number of pixels belong to label c	N	The total number of pixels
M	The median number of all N_c/N	T	The threshold value used in semi-supervised loss

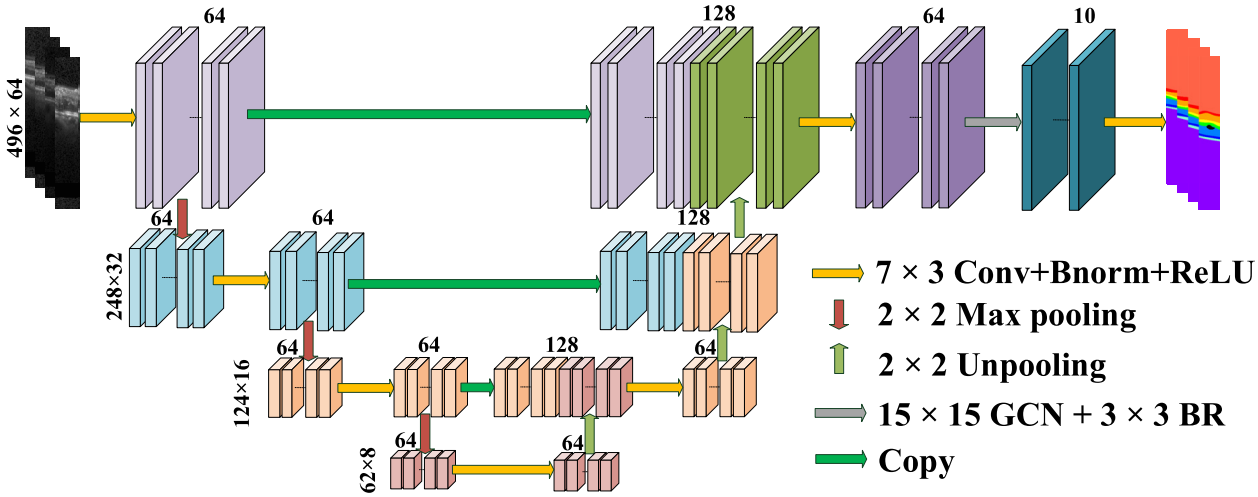


FIGURE 3. The architecture of segmentation network, each colorful overlapped box is a multi-channel feature map, the number on the top of the feature map is the number of channels. The size of feature map is denoted in the left of the feature map row. Note that the feature map with different color means it applied a concatenate operation, and the dark green box represents the feature map of 10 channels, which correspond to 10 classes.

Figure 1 shows an image of the layer structure of retinal, note that RaR is Vitreous and RbR is choroid.

As said above, our SGNet contains two networks, one is segmentation network and the other one is discriminator network. Our method consists of a training stage and a testing stage, an overview can be seen in Figure 2.

In the training stage (Figure 2(a)), the segmentation network takes labeled images as input and outputs the corresponding probability map. The weighted multi-class cross entropy is calculated on the probability map and the corresponding ground truth. Then the probability map and the corresponding ground truth (in one-hot encoding) are inputted into discriminator network respectively, the outputs of discriminator network are utilized to calculate discriminate loss and adversarial loss. In addition, to utilize the information in unlabeled image, the output of segmentation network that take unlabeled data as input is passed to the discriminator network. The semi-supervised loss is calculated based on these outputs of both segmentation network and discriminator network, and optimized after 20 epochs in the training stage. Note that the training image size is 496×64 (*height* $\times$ *width*).

In the testing stage, only the trained segmentation network is used (Figure 2(b)). Since the network is fully convolutional, it can receive an image with any size (496×512 in our dataset) as input and outputs a corresponding segmented result with the same size of input.

A. NETWORK ARCHITECTURE

1) SEGMENTATION NETWORK

We use a modified new U-Net like network as our segmentation network architecture, as shown in Figure 3. This architecture is inspired by ReLayNet [4] which has been proposed for retinal layer and fluid region segmentation and global convolutional network (GCN) [44] which contains large convolution kernel. Segmentation task inherently include two challenges which are pixel-wise classification and localization. To deal with the above two challenges simultaneously, global convolutional layer (GCL) [44] is a good choice. To further improve the segmentation performance, we also introduce boundary refinement layer (BRL) [44] which is a residual structure. Specifically, the segmentation network consists of a contracting path with three encoder blocks, a bottom block and an expansive path with three corresponding decoder blocks, as shown in Figure 3.

The encoder block consists of a 7×3 (due to layer structures in the retinal and the change along horizon is much less than vertical direction, we choose a rectangular convolutional kernel) convolutional layer with padding. The convolutional layer is followed by a batch normalization layer to speed up training and a rectified linear unit (ReLU) is introduced to add non-linearity. A 2×2 max pooling with stride 2 is added after ReLU for down-sampling. The bottom block consists of a convolutional layer, a batch normalization layer and ReLU.

The decoder block consists of an upsampling layer, a concatenation operation, a convolutional layer, a batch normalization and a ReLU. The upsampling layer is a unpooling layer that proposed in [50] which upsamples a lower resolution feature map to a higher one. The concatenation operation concatenates the feature maps from the current block and corresponding feature maps from the encoder block.

The final decoder block is followed by a GCL with kernel size 15×15 and a BRL that proposed in [44]. As shown in Figure 4(a), the GCL module combines a $1 \times 15 + 15 \times 1$ convolutional layer and a $15 \times 1 + 1 \times 15$ convolutional layer to enable dense connections within a large 15×15 region in the feature map, thus it is beneficial to classification and localization. The boundary refinement block is modeled as a residual structure to learn the residual between ground truth

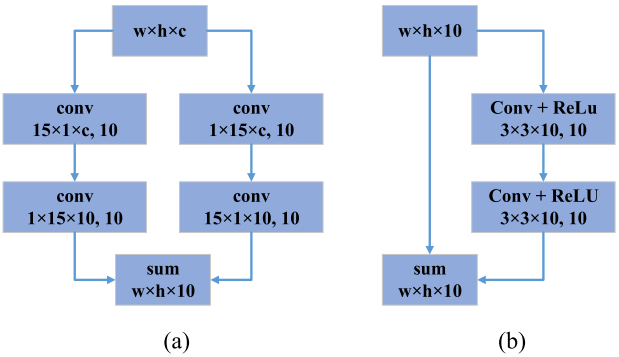


FIGURE 4. The detail of (a) global convolutional network and (b) boundary refinement.

TABLE 2. Architecture of each block (encoder block, bottom block, decoder block, GCN and BR) used in our segmentation network.

	Type	Kernel size	Stride	Padding
Encoder	Conv	7×3	1×1	[3,1]
	Batch Norm			
	Relu			
	Max pool	2×2	2×2	[0,0]
Bottom	Conv	7×3	1×1	[3,1]
	Batch Norm			
	Relu			
Decoder	Unpool	2×2	2×2	[0,0]
	Conv	7×3	1×1	[3,1]
	Batch Norm			
	Relu			
GCL	Conv	15×15	1×1	[7,7]
BRL	Conv	3×3	1×1	[1,1]
	Relu			
	Conv	3×3	1×1	[1,1]
	Relu			

and prediction, which can be seen in Figure 4(b). The detailed architecture of each block is presented in Table 2. Note that the GCL in Table 2 is the combination of a serials of 1d convolutional layers.

2) DISCRIMINATOR NETWORK

Different from the commonly used GANs where discriminator network just identifies the input as real or fake (output a single scalar value), the discriminator network in the proposed SGNet is a fully convolutional architecture that is modified for segmentation (output is a matrix with the same size of input image). It will push the prediction of the segmentation network close to the ground truth. Specifically, the encoder blocks and decoder blocks are the same with segmentation network except for the activation function of them. In discriminator network, following the discriminator network of GAN, all ReLU activation functions are replaced by Leaky ReLU activation function. Compared to ReLu, Leaky ReLu helps to make sure the gradient can flow through the entire architecture which is important for GAN [51]. The GCL and BR are replaced by a classification layer that contains a 1×1 convolutional layer.

B. TRAINING

The segmentation network and discriminator network are trained by optimizing two different loss functions. First of all, we denote the segmentation network as $S(\cdot)$ and the discriminator network as $D(\cdot)$. Given an input OCT image X_n from labeled data of size $h \times w \times 1$ and its corresponding ground truth label Y_n (one-hot encoding), the predicted probability map outputted by segmentation network is $S(X_n)$ of size $h \times w \times c$, which is denoted as P_n, and c is the number of categories. To the discriminator network, we denote the input of it as Q_n(Q_n can be the ground truth label Y_n or the output of segmentation network P_n, the size is $h \times w \times c$). The discriminator network outputs a pixel-wise confidence map which is denoted as $D(Q_n)$ of size $h \times w \times 1$.

1) LOSS OF DISCRIMINATOR NETWORK

The loss function of the discriminator network is the spatial cross-entropy loss which is defined as follow:

$$L_d = -\sum_{i \in X_n} \log(D(Y_n)^{(i)}) - \log(1 - D(P_n)^{(i)}) \tag{2}$$

where i is a pixel belongs to the image X_n. It will push the prediction P_n made by the segmentation network to be close to the ground truth label Y_n.

2) LOSS OF SEGMENTATION NETWORK

The segmentation network is trained via a joint loss function defined as follow:

$$L_s = L_{mce} + \alpha L_{dice} + \beta L_{adv} + \gamma L_{semi} \tag{3}$$

where L_{mce}, L_{dice}, L_{adv}, L_{semi}, denote the multi-class cross entropy, dice loss, the adversarial loss and the semi-supervised loss [43], respectively, and α, β, γ are user defined parameters. Due to the existence of class imbalance, the multi-class cross entropy and dice loss are weighted by a weighting strategy.

The weighting strategy includes median frequency balancing [52] and layer contour weighting. The median frequency balancing assigns a weight to a class in the loss function and the weight is the ratio of the median of class frequencies computed on the entire training set divided by the class frequency. Its formulation is defined as [52]:

$$mfb_c = \frac{M}{N_c/N} \tag{4}$$

where N_c is the number of pixels belong to label c in images of training dataset, N is the total number of pixels in images of training dataset, M is the median number of all N_c/N.

Furthermore, it is difficult to differentiate the pixels adjoining to the tissue boundaries, due to the existence of speckle

noise which may make them diffused. Thus, these pixels adjoining to the boundaries should be more important than others and will be weighted by a constant. The complete weighting strategy is defined as:

$$W_n^{(i)} = mfb^{(i)} + \lambda Indictor(\left|G_n^{(i)}\right| > 0) \tag{5}$$

where i is a pixel in the image X_n, λ is a user defined parameter, G_n is the gradient of label Y_n.

The weighted multi-class cross entropy, which is employed to minimize the empirical risk, encourages the segmentation network to produce predictions which are hard to be distinguished from ground-truth ones by the discriminator network. Given a pixel i in image X_n, its formulation is defined as follow:

$$L_{mce} = -\sum_{i \in X_n} W_n^{(i)} Y_n^{(i)} \log(P_n^{(i)}) \tag{6}$$

where $W_n^{(i)}$ is the weight of pixel i in image X_n calculated as in equation (5).

The dice score coefficient is an important measure to evaluate segmentation performance, and it measures the overlap region between the segmentation result and the ground truth. Thus, training a high-performance segmentation model should maximize the dice score coefficient. To this end, we use a derivable dice loss that proposed in [53]:

$$L_{dice} = 1 - 2\frac{\sum_{c=1}^{10} mbf_c \sum_{i \in X_n^{(c)}} P_n^{(i)(c)} Y_n^{(i)(c)}}{\sum_{c=1}^{10} mbf_c \sum_{i \in X_n^{(c)}} (P_n^{(i)(c)} + Y_n^{(i)(c)})} \tag{7}$$

where $P_n^{(i)(c)}$ is the probability of pixel i belongs to the class c, $Y_n^{(i)(c)}$ is the ground truth of pixel i.

Denote $\sum_{c=1}^{10} mbf_c \sum_{i \in X_n^{(c)}} P_n^{(i)(c)} Y_n^{(i)(c)}$ as Numer_n, and denote $\sum_{c=1}^{10} mbf_c \sum_{i \in X_n^{(c)}} (P_n^{(i)(c)} + Y_n^{(i)(c)})$ as $Deno_n$, the derivative term of this dice loss is as follow:

$$\frac{\partial L_{dice}}{\partial P_n^{(i)(c)}} = -2mfb_c \frac{Y_n^{(i)(c)} Numer_n - Deno_n}{Deno_n^2} \tag{8}$$

For adversarial learning, the adversarial loss L_{adv} is define as [41]:

$$l_{adv} = -\sum_{i \in X_n} \log(D(P_n)) \tag{9}$$

With this adversarial loss, the segmentation network is trained to fool the discriminator network by maximizing the probability of the segmentation prediction belongs to the ground truth distribution. This term in fact will also improve the segmentation result, pushed by the discriminator network $D(\cdot)$.

In order to utilize the large number of unlabeled images to further improve the performance of segmentation network, we incorporate a semi-supervised loss L_{semi} proposed in [43], L_{semi} is optimized via adversarial learning with unlabeled data. Specifically, the trained discriminator network can generate a confidence map which represents whether the prediction results of the regions are close enough to the ground truth distribution. Thus, it can be seen as the supervisory signal to guide the cross-entropy loss in a "self-taught" manner. The confidence map is then binarized with a threshold to highlight the trustworthy region. More specifically, the soft ground truth of semi-supervised is define as [43]:

$$\tilde{Y}_n = onehot(I(D(S(U_n))^{(i)} > T) \cdot \underset{c}{\text{argmax}}(S(U_n))) \tag{10}$$

where $onehot(\cdot)$ denotes one hot encode, $I(\cdot)$ is indictor function, T is a user defined threshold value, U_n is an image from unlabeled data, and then $\tilde{Y}_n$ is the masked segmentation prediction, hence $\tilde{Y}_n$ is of size $h \times w \times c$.

With the soft ground truth $\tilde{Y}_n$, the semi-supervised loss L_{semi} can be defined as:

$$L_{semi} = -\sum_{i \in X_n} \sum_{c \in C} \tilde{Y}_n^{(i,c)} \log(S(U_n)^{(i,c)}) \tag{11}$$

Equation (11) can be regarded as a masked spatial cross entropy loss due to $\tilde{Y}_n$ and indictor function $I(\cdot)$ in equation (10) are treated as constant during training.

The weighted cross entropy loss and the adversarial loss of segmentation network are optimized with labeled data, as well as the loss of discriminator network. Different from these losses, the semi-supervised loss is only optimized with unlabeled data. Moreover, it is only used to update the parameters of segmentation network. The segmentation network and the discriminator network are trained alternately, and the procedure is presented in algorithm 1.

IV. EXPERIMENTATION AND RESULTS

A. DATASET

The proposed SGNet is evaluated on the Duke DME dataset [19] and the POne dataset [54]. The Duke dataset is collected from 10 DME subjects, each contains 61 SD-OCT B-scan images with size of 512 × 740. Only 11 B-scan images are labeled in each subject, thus we totally have 110 labeled B-scan images annotated by two experts, and we use the annotation from the expert one as the ground truth for training the network. The labeled data set is divided into train set (randomly chosen 5 subjects) and test set (the remaining 5 subjects). The experiments are performed with five different divisions, and the reported results are the average of the five runs. In addition, we randomly sample 110 unlabeled B-scan images from the unlabeled data for semi-supervised learning. Following the ReLayNet [4], we also used a data slicing method that is slightly different from the ReLayNet. For a given OCT B-scan I of size $m \times n$ (*height* × *width*), it is divided width-wise into g non-overlapping slices. Each slice is sampled from I by a stride of half of the slice width. Furthermore, we employed data augmentation (random horizontal flips and random crop) to these sliced data.

Algorithm 1 The Proposed Algorithm

Require: τ, ξ are the learning rate of segmentation network and discriminator network respectively. m, the batch size. n, the number of iterations. n_{semi_start}, the start iteration of training segmentation network with semi loss.
Require: θ, the parameters of segmentation network. ω, the parameters of discriminator network.

While θ has not converged do
  for $t = 0, ..., n$ do
    $g_{semi} = 0$
    If $t \geq n_{semi\_start}$ do
      Sample $\{U_i\}_i^m \sim P_{data}(U)$ a batch from the unlabeled data
      $g_{semi} \leftarrow \nabla_\theta L_{semi}(U_i)$
    Sample $\{X_i\}_i^m, \{Y_i\}_i^m \sim P_{data}(X, Y)$ a batch from the labeled data
    end if
    $g_\theta^{mce} \leftarrow -\nabla_\theta \frac{1}{m} \sum_{i=1}^m L_{mce}(S(X_i), Y_i)$, see equation (6)
    $g_\theta^{dice} \leftarrow -\nabla_\theta \frac{1}{m} \sum_{i=1}^m L_{dice}(S(X_i), Y_i)$, see equation (7)
    $g_\theta^{adv} \leftarrow -\nabla_\theta \frac{1}{m} \sum_{i=1}^m L_{adv}(S(X_i))$, see equation (9)
    $\theta \leftarrow \theta + \tau \cdot SGD(\omega, g_{mce} + \alpha g_{dice} + \beta g_{adv} + \gamma g_{semi})$
    $g_w^d \leftarrow -\nabla_\omega \frac{1}{m} \sum_{i=1}^m L_d(S(X_i), Y_i)$, see equation (2)
    $\omega = \omega - \xi Adam(\omega, g_w^d)$
  end for
end While

Eventually, the slice size that used for training is 496×64 ($height \times width$).

The POne dataset [54] is collected from 10 healthy adult subjects with Spectralis device. The enrolled subjects were selected based on a visual acuity test and none of them suffers from ocular or systematic disease. The scanned volume of each subject contains 61 images with 496×768 pixels ($height \times width$). The axial resolution was 3.9 microns, the transversal resolution varied from 10 to 12 microns, and the inter B-scan spacing was from 120 microns to 140 microns. The B-scans were denoised with averaging of five aligned images. Among the 61 B-scans of each volume, 10 of them were annotated by two experts, and the results of their proposed OCTRIMA 3D method [54] were also provided. The annotations are eight layer-contours, and since all subjects are healthy, the dataset does not have annotation of fluid region. It is noted that the dataset they provided contains only labeled B-scans, that is, there are only 10 B-scans for each subject. The experiment settings on POne dataset is similar to those on Duke dataset.

B. IMPLEMENTATION DETAILS

The proposed SGNet is implemented in Python with Tensorflow library [55], the model is trained on a GTX970 GPU. The segmentation network is trained with Stochastic Gradient Descent (SGD) in which momentum is set as 0.9. The initial learning rate which is decreased with exponent decay with power of 0.9 is set into 0.1. We have investigated the effects of the weight parameters α, β and γ in equation (3), and results will be listed below. Empirically we found they work well with values 0.5, 0.1 and 0.1 respectively. If not specified, they are set as these values. Follow [43], we tried different T in range of 0.1 to 0.3, and we found that the best value of threshold T in equation (11) is 0.2. For the discriminator network, it is trained with Adam optimizer with learning rate of 1×10^{-4} and beta2 with 0.99.

The segmentation and discriminator network are trained alternately, the parameters in segmentation network are optimized with labeled (through cross entropy loss, dice loss and adversarial loss) and unlabeled data (through semi-supervised loss). The testing time is about 0.1 seconds per B-scan with the GTX970 GPU, and more powerful GPU can reduce the testing time further.

C. COMPARATIVE METHODS AND METRIC

We compared the proposed SGNet with several state-of-the-art methods which include (1) kernel regression based classification method (KR-GTDP) [19], (2) Learning layer-specific edges method (LSE-GTDP) [20], (3) convolutional neural networks (CNN) and graph search combined method (CNN-GS) [34] and (4) ReLayNet [4]. Among these methods, KR-GTDP and LSE-GTDP are traditional machine learning based methods, CNN-GS and ReLayNet are deep learning based methods. KR-GTDP, CNN-GS and LSE-GTDP all used graph search for layer segmentation. The basic graph search is not suitable for fluid region segmentation, since it is hard to define start point and end point and the number of fluid regions is not fixed. CNN-GS and LSE-GTDP cannot segment the fluid regions. In KR-GTDP, a separate classifier is specially trained with several manual designed features for fluid regions, and the possible fluid regions are further filtered with shape analysis. KR-GTDP, ReLayNet and the proposed SGNet can segment both layer and fluid regions.

We have also investigated the effect of $GCN + BR$, adversarial loss and semi-supervised loss, which will be presented in the subsection of ablation experiment later. We implemented CNN-GS and ReLayNet with Tensorflow and trained it with the same hyper parameters as mentioned in the original papers.

We utilize the *dice* coefficient and error of layer contour, which have been reported on recent retinal layer segmentation papers as the comparative metric [4], [20], [34]. The *dice* coefficient depends on overlap region between the segmentation result and the ground truth, and its formulation is defined as follow:

$$dice = \frac{2\,|P \cap Y|}{|P| + |Y|} \tag{12}$$

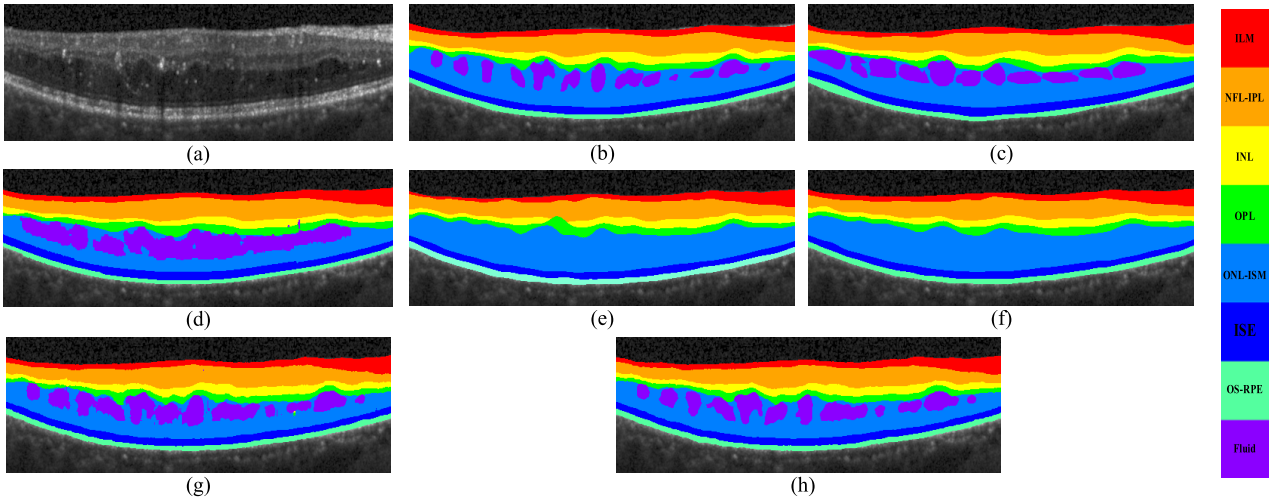


FIGURE 5. Layer and fluid segmentation results on an abnormal OCT B-scan sample with DME. (a) an abnormal OCT B-scan sample, (b) expert 1 annotation, (c) expert 2 annotation, (d) KR, (e) LSE-GTDP, (f) CNN-GS, (g) ReLayNet, (h) the proposed SGNet.

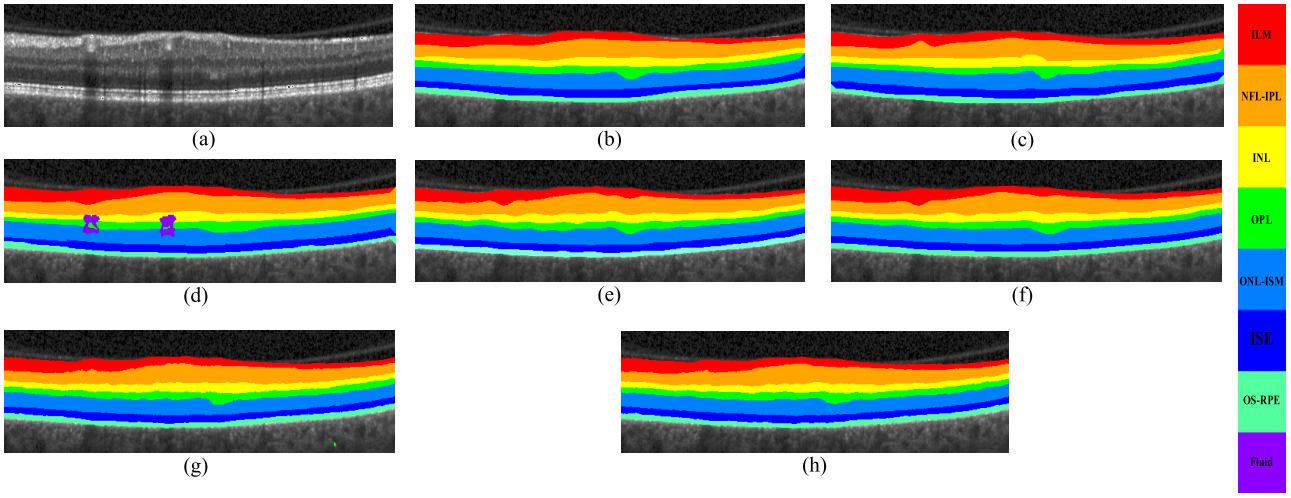


FIGURE 6. Layer and fluid segmentation results on a normal OCT B-scan sample. (a)the normal OCT B-scan sample, (b) expert 1 annotation, (c) expert 2 annotation, (d) KR, (e) LSE, (f) CNN-GS, (g) ReLayNet, (h) the proposed SGNet.

where P and Y are prediction and ground truth of each category, respectively.

The error of layer contour (denote it as CE) is the mean absolute difference between the predicted layer boundary and the ground truth layer boundary along each column. It is formulated as:

$$CE = \frac{1}{n}\sum_{i}^{n}(p_i - g_i) \tag{13}$$

where p_i and g_i are the predicted row location of each boundary and the expert annotated one, n is the number of pixels in the layer contour.

D. RESULTS

In order to compare the proposed SGNet with these comparative methods, we have conducted qualitative analysis and quantitative analysis.

1) QUALITATIVE ANALYSIS

The qualitative comparison between the proposed method and these comparative methods are presented in this section. The results on typical OCT B-scan with fluid region and without fluid region are shown in Figure 5 and Figure 6 respectively.

For the OCT B-scan with fluid region (Figure 5), due to the existence of several small accumulated fluid regions, it is one of the most difficult B-scans to be segmented. It should be noted that not all these comparative methods can jointly segment retinal layers and fluid region. Among these methods, only KR-GTDP (Figure 5(d)) and ReLayNet (Figure 5(g)) can jointly segment retinal layers and fluid regions, the LSE-GTDP (Figure 5(e)) and CNN-GS (Figure 5(f)) can only segment retinal layers. As can be seen from the Figure 5(d), Figure 5(g), Figure 5(h), almost all the fluid regions segmented by KR are fused together (Figure 5(d)).

ReLayNet is better than KR, but it still has fused fluid regions (Figure 5(g)). The proposed SGNet distinguished more different fluid regions (Figure 5(h)), this may be benefited from the integration of GCN and the pixel-wise adversarial learning strategy. For the retinal layer, it can be seen that layer boundary of the proposed method are smoother than ReLayNet, this might be due to the advantage of GCN and BR. The layer boundary segmented by KR (it does not consider the presence of fluid region) is also smoother than ReLayNet due to Graph-based Dynamic Programming.

The results on a normal OCT B-scan is shown in Figure 6. From the figure, we can see that false fluid regions appeared in the segmentation result of KR (Figure 6(d)). It should be noted that the LSE-GTDP and CNN-GS have no ability to segment fluid mass, thus the fluid region will not appear in their segmentation results (Figure 6(e)). As for ReLayNet, from the Figure 6(g), we can see that some pixels below the OS-RPE are obviously misclassified. The proposed SGNet, which has no obvious misclassification, has better performance than ReLayNet.

2) QUANTITATIVE ANALYSIS

The quantitative comparison on *dice* is shown in Table 3, it lists the *dice* value (in mean and standard deviation) of each category and *mean Dice* of all regions. Statistical significance of the differences was determined using 2-tailed paired t-test for which p value of 0.05 were considered significant.

TABLE 3. Quantitative comparison between the proposed SGNET and the comparative method by *Dice* on Duke dataset. The mean (standard deviation) values are reported. the best results are bold and the second-best results are italic.

	ILM	NFL-IPL	INL	OPL	ONL-ISM	ISE	OS-RPE	Fluid	mean
Expert 2	0.849 (0.067)	0.884 (0.044)	0.792 (0.060)	0.725 (0.077)	0.930 (0.029)	0.853 (0.055)	0.822 (0.060)	0.582 (0.217)	0.805
KR-GTDP	0.839 (0.053)	0.879 (0.059)	0.774 (0.078)	0.748 (0.104)	0.918 (0.039)	0.876 (0.047)	0.812 (0.050)	0.612 (0.272)	0.807
LSE-GTDP	0.872 (0.049)	0.910 (0.057)	0.801 (0.071)	0.768 (0.069)	*0.931 (0.061)*	0.874 (0.041)	0.864 (0.053)	NA	0.860
CNN-GS	0.825 (0.026)	0.891 (0.053)	0.741 (0.075)	0.736 (0.071)	0.904 (0.062)	0.867 (0.047)	0.869 (0.039)	NA	0.833
ReLayNet	*0.901 (0.042)*	*0.938 (0.039)*	*0.869 (0.046)*	*0.835 (0.058)*	0.921 (0.042)	*0.913 (0.041)*	*0.900* (0.034)	*0.771 (0.208)*	*0.881*
SGNet (our proposed)	**0.921 (0.033)**	**0.955 (0.030)**	**0.892 (0.034)**	**0.850 (0.054)**	**0.941 (0.039)**	**0.929 (0.039)**	**0.911 (0.027)**	**0.800 (0.185)**	**0.900**

As shown in Table 3, for all the methods, the performance on fluid regions is lower compared to other layer regions, including the result of expert 2. This is due to the low contrast between fluid region and background, also due to the irregular shape and varying number of occurrences. The performances of KR-GTDP and CNN-GS are poor compared to other methods. It is worth to mention that the dice value of KR-GTDP without taking fluid region into account is 0.835, which is similar to the value of CNN-GS (0.833). The proposed method has the best performance for all the 8 categories which include ILM, NFL-IPL, INL, OPL, ONL-ISM, ISE, OSE-RPE and Fluid. ReLayNet has achieved the second-best results for 7 times among the 8 categories, and LSE-GTDP achieved the second-best result for ONL-ISM. For the segmentation of fluid region, the proposed SGNet achieve a *dice* of 0.800 which is with a relative improvement of 3.5%, compare to the performance achieved by ReLayNet. The mean *dice* of all categories is also improved from 0.881 to 0.900, a relative improvement of 2.1%. Since ReLayNet performed best in the four compared methods, the statistical analysis is performed by comparing our SGNet and ReLayNet. The improvements are significant for 5 layers, including ILM ($p < 0.01$), NFL-IPL ($p < 0.05$), INL ($p < 0.01$), ONL-ISM ($p < 0.05$) and ISE ($p < 0.05$).

The quantitative comparison between the proposed method and the comparative methods with contour error (CE) are presented in Table 4. All best results are bold, and it can be apparently seen that the proposed SGNet achieved best result in 6 retinal layer contours out of 7 retinal layer contours compare to the existing best result. These 6 retinal layer contours are Vitreous- ILM, ILM-NFL, IPL-INL, INL-OPL, ISM-ISE and ISE-OS, each of them are reduced by 15.8%, 20.7%, 11.9%, 8.1%, 0.5% and 7.3% respectively on the basis of the existing best result. The mean error of all contours is 1.068 pixels which is also reduced by 10.9%.

TABLE 4. Quantitative comparison between the proposed SGNET and the comparative method by contour error (CE) on Duke dataset. The mean (standard deviation) of CE (in pixels) are reported. the best results are bold and the second-best results are italic.

	Vitreous-ILM	ILM-NFL	IPL-INL	INL-OPL	OPL-ONL	ISM-ISE	ISE-OS	mean
Expert 2	1.298 (1.243)	1.797 (2.029)	1.834 (1.762)	1.812 (1.831)	1.843 (2.126)	1.125 (1.162)	1.246 (1.394)	1.565
KR-GTDP	1.529 (1.267)	1.948 (2.385)	2.132 (2.768)	2.470 (3.832)	2.546 (3.312)	1.126 (1.254)	1.192 (1.344)	1.849
LSE-GTDP	0.973 (0.914)	1.635 (2.542)	1.676 (2.492)	1.684 (2.165)	2.203 (2.457)	0.896 (1.264)	1.256 (1.240)	1.475
CNN-GS	1.884 (1.312)	1.727 (1.893)	1.996 (1.472)	1.516 (1.582)	**1.752 (1.632)**	1.182 (1.231)	1.874 (1.236)	1.704
ReLayNet	*0.852 (1.453)*	*1.137 (1.316)*	*1.258 (1.164)*	*1.343 (1.659)*	2.047 (1.714)	*0.841 (1.162)*	*0.905 (1.214)*	*1.198*
SGNet (our proposed)	**0.717 (0.901)**	**0.902 (1.153)**	**1.108 (1.314)**	**1.234 (1.130)**	*1.840 (1.598)*	**0.837 (1.058)**	**0.839 (1.107)**	**1.068**

Only the error on OPL-ONL of the proposed SGNet is not the best result, the best result of this contour is achieved by CNN-GS, and our SGNet obtained the second-best results. Compared to the result of ReLayNet, it still reduces the error by 10.1%. From the low *dice* of layer OPL between the two experts, it is apparently that OPL layer is the most difficult layer contour to be segmented, this might be the reason that all automatic layer segmentation methods could not achieve better performance on OPL-INL contour. We have also performed statistical analysis on the contour error, due to the relatively large standard deviation, the difference between the proposed SGNet and ReLaynet is not statistically significant ($p > 0.05$).

E. ABLATION EXPERIMENTS

In the proposed SGNet, we incorporated GCN and BR into the U-Net architecture, and for the loss function of segmentation network, we added adversarial loss and semi-supervised loss besides the usually used multi-class cross entropy and dice loss in segmentation methods. In this section, we conduct ablation experiment to evaluate the effect of GCN, BR, adversarial loss and semi-supervised loss. We will also investigate the effect of weight parameters in the loss function of segmentation network. Note the experiments in this subsection are performed on a single division of the dataset.

Since the previous best result is achieved by ReLayNet, thus we denote RelayNet as baseline 1, and the baseline 2 is ReLayNet with GCN. The baseline 3 is the baseline 2 integrated with BR, which is also the segmentation network used in our method, and the baseline 3 with adversarial learning is denoted as baseline 4. The proposed SGNet includes *GCN*, *BR*, L_{adv}, L_{semi}. A summary can be seen in Table 5.

TABLE 5. The characteristics of baselines.

	GCN	BR	L_{adv}	L_{semi}
Baseline 1	No	No	No	No
Baseline 2	Yes	No	No	No
Baseline 3	Yes	Yes	No	No
Baseline 4	Yes	Yes	Yes	No
SGNet	Yes	Yes	Yes	Yes

1) GCN AND BR

In section III.A, we modified the ReLayNet by the GCN (introduce dense connections between classifiers and features) and BR (refine the layer boundary). To verify the effect of GCN and BR in Duke DME dataset, we conduct a comparison between the baseline 1, baseline 2 and baseline 3.

TABLE 6. Quantitative comparison between the baseline 1, baseline with GCN and the baseline 2 by *Dice*, the best results are bold and the second-best results are italic.

	ILM	NFL-IPL	INL	OPL	ONL-ISM	ISE	OS-RPE	Fluid	mean
Baseline 1	0.903 (0.033)	0.935 (0.021)	0.871 (0.053)	0.833 (0.046)	0.934 (0.034)	0.918 (0.015)	0.900 (0.038)	0.773 (0.215)	0.883
Baseline 2	0.917 (0.042)	0.943 (0.037)	0.886 (0.071)	0.842 (0.072)	0.938 (0.046)	0.908 (0.037)	0.885 (0.046)	0.778 (0.185)	0.887
Baseline 3	0.915 (0.043)	0.944 (0.047)	0.886 (0.071)	0.841 (0.075)	0.936 (0.048)	0.919 (0.036)	0.893 (0.048)	0.783 (0.182)	0.889
Baseline 4	*0.921 (0.032)*	*0.948 (0.021)*	*0.891 (0.042)*	**0.847 (0.040)**	*0.942 (0.032)*	*0.925 (0.031)*	*0.908 (0.036)*	*0.795 (0.140)*	*0.897*
SGNet	**0.923 (0.031)**	**0.952 (0.021)**	**0.896 (0.049)**	*0.846 (0.040)*	**0.945 (0.033)**	**0.928 (0.014)**	**0.914 (0.030)**	**0.803 (0.193)**	**0.900**

The ablation experimentation results of baseline 1 and baseline 2 and baseline 3 on *dice* are present in Table 6. From the Table 6, it can be seen that the *dice* of baseline 2 (baseline 1 with BR) on ILM, NFL-IPL, INL, OPL, ONL-ISM and fluid are better than the baseline 1, and it is increased by 0.4% on *dice* of all retinal layers (from 0.883 to 0.887). However, the *dice* of baseline 2 on ISE and OS-RPE are worse than the baseline 1. When added *BR* (baseline 3), the *dice* of ISE is increased by 1.2% (from 0.908 to 0.919) which slightly outperforms the baseline 2. The *mean dice* of baseline 3 is also slightly improved from 0.887 to 0.889, this is benefited from the BR layers. The experiment presented on this subsection demonstrate the impact of GCN and BR layers. Paired t-test analysis shows that the differences between baseline 2, 3 and the ReLayNet are not statistically significant.

2) ADVERSARIAL LOSS AND SEMI-SUPERVISED LOSS

In this sub section, we present the ablation experiments on the baseline 3, baseline 4 and proposed SGNet to investigate the effect on adversarial loss and semi loss. Experiment results on a typical example image is shown in Figure 7, from the figure we can see that the result of the proposed SGNet is the best. The *dice* of baseline 3, baseline 4 and the proposed SGNet are shown in Table 6. As can be seen from Table 6, when adversarial loss is added, the *mean dice* of baseline 4 on all retinal layers is improved from 0.889 to 0.897 (0.9% improvement).

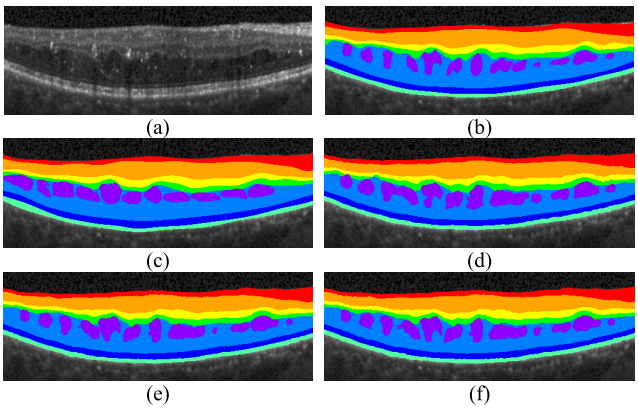


FIGURE 7. Ablation experiment results on an image, (a) the original image, (b) the annotation 1, (c) the annotation 2 (d) baseline 3, (e) baseline 4 and (f) the proposed SGNet.

As for the individual retinal layers, the *dice* of them are all improved. For instance, the *dice* of baseline 4 on Fluid has an 1.5% improvement compared to the baseline 3. This comparison demonstrates that the adversarial loss can improve the segment performance to some extent. Furthermore, after adding the semi-supervised loss, the *mean dice* is further improved from 0.897 to 0.900. Except for OPL layer which is 0.001 lower on *dice*, the *dice* of other retinal layers are all slightly improved. This shows that semi-supervised loss also contributes to the improvement of the segmentation performance, although the improvement is slight. Paired t-test analysis shows that the differences between baseline 4 and the ReLayNet are statistically significant for ILM ($p < 0.01$), NFL-IPL and INL($p < 0.05$). The differences between proposed SGNet and baseline 4 are not statistically significant.

To further investigate the effect of semi-supervised loss (the use of unlabeled data), we have performed experiments with different number of labeled subjects in the training

TABLE 7. Results of the proposed SGNET with different number of labeled B-Scans in the training dataset. The mean (standard deviation) values are reported.

No. labeled B-scans in training dataset	Without semi-supervised loss		With semi-supervised loss	
	Dice	Contour error	Dice	Contour error
11	0.749 (0.089)	3.748 (2.116)	0.782 (0.076)	3.107 (1.830)
22	0.788 (0.068)	2.918 (1.856)	0.819 (0.067)	2.574 (1.804)
33	0.831 (0.051)	2.342 (1.652)	0.852 (0.050)	2.146 (1.643)
44	0.867 (0.044)	1.791 (1.260)	0.878 (0.048)	1.606 (1.432)
55	0.897 (0.029)	1.124 (0.783)	0.900 (0.027)	1.056 (0.645)
66	0.902 (0.025)	1.118 (0.710)	0.908 (0.023)	1.024 (0.349)

dataset. The ratio of the labeled images to the unlabeled images is also kept 1:2, and the number of training subjects varies from 1 (11 labeled B-scans and 22 unlabeled B-scans) to 5 (55 labeled B-scans and 110 unlabeled B-scans). The results are listed in Table 7. From the table, we can see that the performance improvement obtained with the utilization of unlabeled data is large when the number of labeled training images is small. The differences of Dice measurement are statistically significant ($p < 0.5$) when the number of labeled B-scans in the training dataset is less than or equal to 33. With the increases of labeled training B-scans, the effect of utilizing unlabeled data diminishes.

3) EFFECTS OF WEIGHT PARAMETERS ON LOSS FUNCTION

The loss function of the segmentation network in Equation (3) consists of four components, and they are integrated with weight parameters for the final loss function. Although in theory, they have different value ranges (L_{mce}, L_{adv} and L_{semi} are in $[0, +\infty)$ and L_{dice} is in [0, 1]), we find their values are in similar scales in practice. The investigated value of weights α, β, γ for L_{mce}, L_{adv} and L_{semi} are [0.1, 0.5, 1, 10], and the results are listed in Table 8. From the table, we can see that the results are not greatly affected by the weights when they are in a reasonable range. The training may not converge when the weight of L_{adv} is too large ($\beta = 10$), and the best result is obtained when α, β, γ are set as 0.5, 0.1, 0.1 respectively.

F. FIVE-FOLD CROSS-VALIDATION

As noted above, we performed k-fold cross-validation by splitting the dataset differently. The dataset is split 5 times, and in each splitting 5 subjects are used for training and the other 5 subjects are used for testing. The network structure and hyperparameters are set as presented above. The results are listed in Table 9 and 10. We can see that the results are similar, and paired t-test analysis also shows the differences of results from different splitting are not statistically significant.

TABLE 8. Averaged *Dice* results of the proposed SGNET with different weights for segmentation network loss function.

α	$\beta = 0.1$				$\beta = 0.5$			
	$\gamma = 0.1$	0.5	1	10	$\gamma = 0.1$	0.5	1	10
0.1	0.869	0.882	0.883	0.868	0.874	0.879	0.865	0.870
0.5	0.900	0.889	0.894	0.873	0.882	0.872	0.841	0.836
1	0.887	0.886	0.875	0.864	0.867	0.843	0.868	0.853
10	0.892	0.885	0.878	0.862	0.854	0.872	0.861	0.834
α	$\beta = 1$				$\beta = 10$			
	$\gamma = 0.1$	0.5	1	10	$\gamma = 0.1$	0.5	1	10
0.1	0.852	0.877	0.866	0.873	0.712	NAN	NAN	NAN
0.5	0.838	0.853	0.867	0.831	NAN	NAN	NAN	NAN
1	0.844	0.789	0.846	0.822	NAN	NAN	NAN	NAN
10	0.865	0.848	0.865	0.844	0.716	NAN	0.724	NAN

TABLE 9. Dice Results of 5-fold Cross-Validation on Duke dataset with 5 subjects for training and the other 5 subjects for testing. The mean (standard deviation) values are reported.

Fold	ILM	NFL-IPL	INL	OPL	ONL-ISM	ISE	OS-RPE	Fluid
1	0.923 (0.031)	0.952 (0.021)	0.896 (0.049)	0.846 (0.040)	0.945 (0.033)	0.928 (0.014)	0.914 (0.030)	0.803 (0.193)
2	0.919 (0.038)	0.954 (0.034)	0.887 (0.061)	0.849 (0.064)	0.938 (0.044)	0.930 (0.037)	0.908 (0.049)	0.784 (0.182)
3	0.921 (0.053)	0.948 (0.076)	0.879 (0.096)	0.851 (0.075)	0.940 (0.092)	0.931 (0.045)	0.916 (0.037)	0.798 (0.127)
4	0.925 (0.045)	0.951 (0.051)	0.893 (0.059)	0.854 (0.069)	0.939 (0.047)	0.924 (0.045)	0.915 (0.054)	0.806 (0.219)
5	0.917 (0.047)	0.954 (0.036)	0.903 (0.076)	0.852 (0.071)	0.941 (0.060)	0.930 (0.052)	0.916 (0.053)	0.809 (0.202)
Average	0.921	0.952	0.892	0.850	0.941	0.929	0.914	0.800

TABLE 10. Contour error results (in pixels) of 5-fold Cross-Validation with 5 subjects for training and the other 5 subjects for testing. The mean (standard deviation) values (in pixels) are reported.

Fold	Vitreous-ILM	ILM-NFL	IPL-INL	INL-OPL	OPL-ONL	ISM-ISE	ISE-OS
1	0.705 (0.790)	0.910 (0.931)	1.076 (1.210)	1.275 (1.247)	1.936 (1.775)	0.762 (0.777)	0.729 (0.759)
2	0.814 (0.972)	0.835 (1.124)	1.014 (1.104)	1.150 (1.329)	1.853 (1.889)	0.922 (0.889)	0.841 (0.982)
3	0.711 (0.863)	0.922 (1.258)	1.126 (1.512)	1.344 (1.453)	1.763 (1.624)	0.827 (0.996)	0.718 (1.156)
4	0.642 (1.076)	1.021 (1.357)	1.172 (1.606)	1.218 (1.867)	1.756 (1.454)	0.706 (1.124)	0.787 (0.925)
5	0.712 (0.805)	0.821 (1.096)	1.151 (1.138)	1.184 (1.258)	1.894 (1.246)	0.968 (1.502)	1.121 (1.713)
Average	0.717	0.902	1.108	1.234	1.840	0.837	0.839

G. RESULTS ON PONE DATASET

We have also investigated the performance of the proposed method on the POne dataset [54]. As introduced above,

the dataset contains 100 B-scans from 10 subjects, with 10 B-scans from each subject. Similar to the experiments on Duke dataset, the dataset is also split 5 times, and in each splitting 5 subjects are used for training and the other 5 subjects are used for testing. Since the dataset does not contain unlabeled images, we used annotations of 5 slices in each subject and treated the other 5 slices as unannotated. In other words, the training dataset contains 50 slices, among them 25 are annotated and the other 25 slices are unannotated. CNN-GS, ReLayNet and our proposed SGNet were trained on the training dataset, and the performances were evaluated on the other 50 slices. It should be noted that OCTRIMA does not involve a training procedure, and we used their provided results directly.

TABLE 11. Quantitative comparison between the proposed SGNET and the comparative method by *Dice* on POne dataset, the best results are bold and the second-best results are italic. The mean (standard deviation) values are reported.

	ILM	NFL-IPL	INL	OPL	ONL-ISM	ISE	OS-RPE	mean
Expert 2	0.854 (0.073)	0.913 (0.032)	0.797 (0.051)	0.811 (0.036)	0.945 (0.015)	0.792 (0.073)	0.829 (0.052)	0.849
OCTRIMA	0.891 (0.050)	0.939 (0.017)	0.862 (0.036)	0.832 (0.052)	0.950 (0.018)	0.840 (0.076)	0.882 (0.066)	0.885
CNN-GS	0.872 (0.047)	0.941 (0.016)	0.826 (0.029)	0.821 (0.032)	0.955 (0.014)	0.886 (0.024)	0.933 (0.018)	0.891
ReLayNet	*0.924 (0.022)*	*0.942 (0.018)*	*0.890 (0.017)*	*0.870 (0.015)*	0.965 (0.007)	*0.923 (0.022)*	*0.928 (0.029)*	*0.920*
SGNet (our proposed)	**0.931 (0.024)**	**0.946 (0.015)**	**0.901 (0.012)**	**0.876 (0.016)**	**0.972 (0.005)**	**0.938 (0.020)**	**0.941 (0.021)**	**0.929**

TABLE 12. Quantitative comparison between the proposed SGNET and the comparative method by contour error (CE) on POne dataset, the best results are bold and the second-best results are italic. The mean (standard deviation) values (in pixels) are reported.

	Vitreous-ILM	ILM-NFL	IPL-INL	INL-OPL	OPL-ONL	ISM-ISE	ISE-OS	mean
Expert 2	0.958 (0.742)	1.395 (1.331)	1.450 (1.086)	1.308 (1.018)	1.445 (1.185)	0.754 (0.586)	2.248 (1.398)	1.365
OCTRIMA	***0.705 (0.616)***	1.069 (1.226)	0.972 (0.822)	1.046 (0.992)	1.397 (1.961)	0.560 (0.444)	1.670 (1.394)	1.060
CNN-GS	1.207 (0.918)	0.943 (0.714)	1.084 (0.742)	1.377 (0.842)	0.997 (0.926)	0.785 (0.473)	0.786 (0.674)	1.026
ReLayNet	1.254 (0.972)	*0.896 (1.152)*	*0.840 (0.804)*	*0.935 (1.274)*	*0.902 (0.846)*	*0.384 (0.484)*	*0.538 (0.637)*	*0.821*
SGNet (our proposed)	*0.723 (0.714)*	**0.894 (1.216)**	**0.792 (0.778)**	**0.832 (0.789)**	**0.856 (0.798)**	**0.318 (0.442)**	**0.448 (0.534)**	**0.695**

The results of different methods on the POne dataset are listed in Table 11 and 12. Figure 8 shows results of different methods on a B-scan sample. From the tables, we can observe that all methods work well on the POne dataset, since B-scans in the dataset all come from healthy subjects. While the differences are small, the proposed method also performed best on the POne dataset. Compared to ReLayNet, the improvements of SGNet are statistically significant for 4 layers, including INL ($p < 0.01$), ONL-ISM ($p < 0.01$), ISE ($p < 0.01$) and OS-RPE ($p < 0.05$).

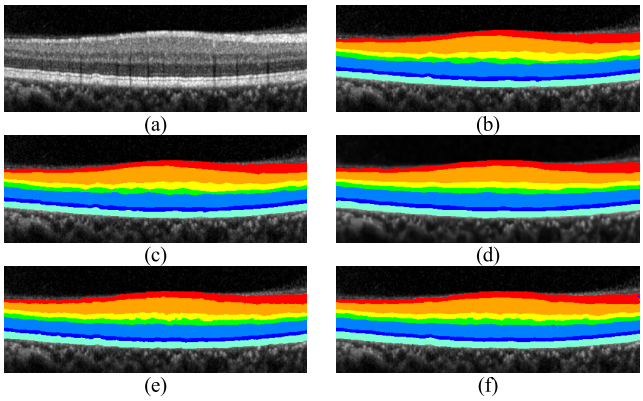


FIGURE 8. Layer segmentation results on an OCT B-scan sample in PONE dataset. (a) an OCT B-scan sample, (b) expert 1 annotation, (c) expert 2 annotation, (d) CNN-GS, (e) ReLayNet, (f) the proposed SGNet.

H. FAILURE CASES

Although our proposed SGNet has achieved promising segmentation results on most of the B-scans in the two datasets, the performance needs further improvement for some B-scans. Since ReLayNet and SGNet have shown better performances than CNN-GS and can segment fluid regions, we only shown results of these methods. Figure 9 shows some challenging cases and the results. The OPL layer (green

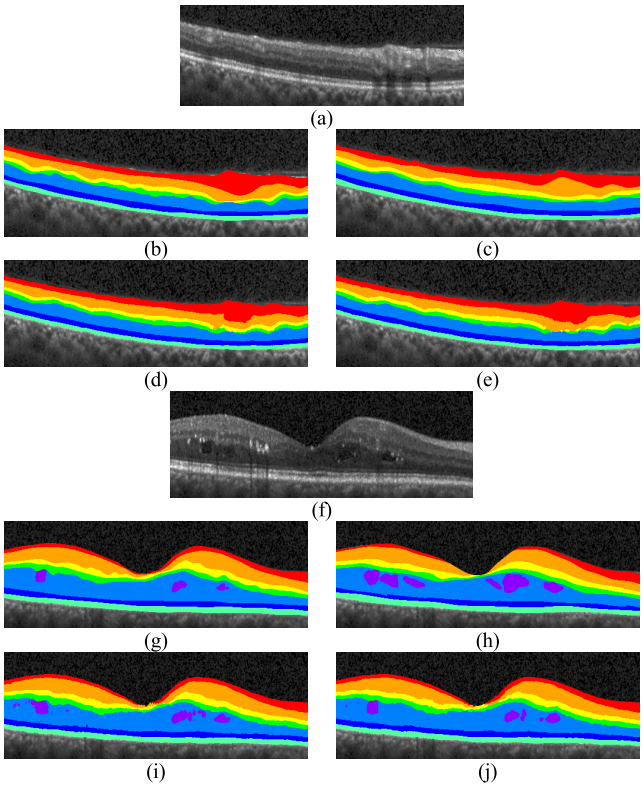


FIGURE 9. Cases where our results were suboptimal. (a) and (f) are the original images, (b) and (g) are annotations of expert 1, (c) and (h) are annotations of expert 2, (d) and (i) are the results of ReLayNet, (e) and (j) are the results of the proposed SGNet.

color) is very thin in the B-scan shown in Figure 9(a), and the shadow (caused by vessel) makes the contrast between OPL and neighbor layers lower. Both ReLayNet (Figure 9(d)) and SGNet (Figure 9(e)) cannot accurately segment the layers, and the continuity of OPL are broken. Post-processing such as graph search may improve the results. Another approach is to incorporate the layer order and continuity information into the network architecture or loss function, while it is not intuitive to incorporate these prior information into deep learning methods [56].

Figure 9(f)-(j) show the results on another B-scan with DME. In this image, the contrast between fluid regions and background is low, and the difference between annotations of two experts is large. Compared to the ground truth (annotation of expert 1), both ReLayNet and the proposed SGNet segmented more fluid regions, although these regions are labeled as fluid regions by the expert 2. Besides, the continuity of ILM and NFL-IPL at the fovea are also broken. Similarly, prior information may also be utilized to deal with the discontinuity problem.

V. CONCLUSIONS

In this paper, we proposed a semi-supervised method (named SGNet) to segment 9 retinal layer and fluid regions automatically on OCT B-scans through an adversarial learning strategy. The proposed method is validated on Duke DME dataset and POne dataset and compared with four state-of-the-art retinal layer segmentation methods. To the best of our knowledge, this is the first adversarial learning based method to segment retinal layer and fluid on OCT B-scans. The segmentation method, which is trained end-end, including a new segmentation network modified from ReLayNet and a discriminator network, both are fully convolutional network. The segmentation network is trained by optimizing a joint loss function including weighted cross entropy loss, weighted dice loss, adversarial loss and semi-supervised loss. Semi-supervised loss is only trained with unlabeled data, all the other losses are trained with labeled data. It should be noted that only the parameters of segmentation network are updated when trained with unlabeled. The testing time is about 0.1 seconds per OCT B-scan on GTX970 GPU. The experiment results on duke and PONE datasets show that the proposed method outperforms the state-of-the-art retinal layer and fluid segmentation methods.

In the future work, we will explore the integration of prior knowledge and deep learning for retinal layer segmentation [56], investigate some advanced segmentation network architectures [25], [29], [57], we will also explore the segmentation of retinal layer with other diseases.

REFERENCES

[1] W. Drexler and J. G. Fujimoto, "State-of-the-art retinal optical coherence tomography," *Prog. Retinal Eye Res.*, vol. 27, no. 1, pp. 45–88, Jan. 2008.

[2] D. Huang *et al.*, "Optical coherence tomography," *Science*, vol. 254, no. 5035, pp. 1178–1181, 1991.

[3] F. A. Medeiros, L. M. Zangwill, C. Bowd, R. M. Vessani, R. Susanna, Jr., and R. N. Weinreb, "Evaluation of retinal nerve fiber layer, optic nerve head, and macular thickness measurements for glaucoma detection using optical coherence tomography," *Amer. J. Ophthalmol.*, vol. 139, no. 1, pp. 44–55, 2005.

[4] A. G. Roy *et al.*, "ReLayNet: Retinal layer and fluid segmentation of macular optical coherence tomography using fully convolutional networks," *Biomed. Opt. Express*, vol. 8, no. 8, pp. 3627–3642, 2017.

[5] X. Liu, Z. Yang, W. Hu, J. Liu, and K. Zhang, "Detection of macular diseases in optical coherence tomography image," *Int. J. Parallel, Emergent Distrib. Syst.*, pp. 1–13, Jun. 2018, doi: 10.1080/17445760.2018.1472261.

[6] M. Mujat *et al.*, "Retinal nerve fiber layer thickness map determined from optical coherence tomography images," *Opt. Express*, vol. 13, no. 23, pp. 9480–9491, 2005.

[7] A. Mishra, A. Wong, K. Bizheva, and D. A. Clausi, "Intra-retinal layer segmentation in optical coherence tomography images," *Opt. Express*, vol. 17, no. 26, pp. 23719–23728, 2009.

[8] J. Novosel, G. Thepass, H. G. Lemij, J. F. de Boer, K. A. Vermeer, and L. J. van Vliet, "Loosely coupled level sets for simultaneous 3D retinal layer segmentation in optical coherence tomography," *Med. Image Anal.*, vol. 26, no. 1, pp. 146–158, 2015.

[9] C. Wang, Y. X. Wang, and Y. Li, "Automatic choroidal layer segmentation using Markov random field and level set method," *IEEE J. Biomed. Health Inform.*, vol. 21, no. 6, pp. 1694–1702, Nov. 2017.

[10] S. J. Chiu, X. T. Li, P. Nicholas, C. A. Toth, J. A. Izatt, and S. Farsiu, "Automatic segmentation of seven retinal layers in SDOCT images congruent with expert manual segmentation," *Opt. Express*, vol. 18, no. 18, pp. 19413–19428, 2010.

[11] J. Y. Lee *et al.*, "Fully automatic software for retinal thickness in eyes with diabetic macular edema from images acquired by cirrus and spectralis systems," *Invest. Ophthalmol. Vis. Sci.*, vol. 54, no. 12, pp. 7595–7602, 2013.

[12] X. Liu, D. Liu, T. Fu, K. Zhang, J. Liu, and L. Chen, "Shortest path with backtracking based automatic layer segmentation in pathological retinal optical coherence tomography," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 2770–2774.

[13] M. K. Garvin, M. D. Abramoff, X. Wu, S. R. Russell, T. L. Burns, and M. Sonka, "Automated 3-D intraretinal layer segmentation of macular spectral-domain optical coherence tomography images," *IEEE Trans. Med. Imag.*, vol. 28, no. 9, pp. 1436–1447, Sep. 2009.

[14] P. A. Dufour *et al.*, "Graph-based multi-surface segmentation of OCT data using trained hard and soft constraints," *IEEE Trans. Med. Imag.*, vol. 32, no. 3, pp. 531–543, Mar. 2013.

[15] W. Duan *et al.*, "A generative model for OCT retinal layer segmentation by groupwise curve alignment," *IEEE Access*, vol. 6, pp. 25130–25141, 2018.

[16] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2006.

[17] Y. Deng, Y. Li, Y. Qian, X. Ji, and Q. Dai, "Visual words assignment via information-theoretic manifold embedding," *IEEE Trans. Cybern.*, vol. 44, no. 10, pp. 1924–1937, Oct. 2014.

[18] Y. Deng, Q. Dai, R. Liu, Z. Zhang, and S. Hu, "Low-rank structure learning via nonconvex heuristic recovery," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 3, pp. 383–396, Mar. 2013.

[19] S. J. Chiu, M. J. Allingham, P. S. Mettu, S. W. Cousins, J. A. Izatt, and S. Farsiu, "Kernel regression based segmentation of optical coherence tomography images with diabetic macular edema," *Biomed. Opt. Express*, vol. 6, no. 4, pp. 1172–1194, 2015.

[20] S. P. K. Karri, D. Chakraborthi, and J. Chatterjee, "Learning layer-specific edges for segmenting retinal layers with large deformations," *Biomed. Opt. Express*, vol. 7, no. 7, pp. 2888–2901, 2016.

[21] D. Xiang *et al.*, "Automatic retinal layer segmentation of OCT images with central serous retinopathy," *IEEE J. Biomed. Health Inform.*, to be published, doi: 10.1109/JBHI.2018.2803063.

[22] K. G. Lore, A. Akintayo, and S. Sarkar, "LLNet: A deep autoencoder approach to natural low-light image enhancement," *Pattern Recognit.*, vol. 61, pp. 650–662, Jan. 2017.

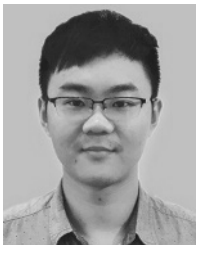
[23] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[25] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.
[26] Y. Deng, Z. Ren, Y. Kong, F. Bao, and Q. Dai, "A hierarchical fused fuzzy deep neural network for data classification," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 4, pp. 1006–1012, Aug. 2017.
[27] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.
[28] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2015, pp. 234–241.
[29] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2017.
[30] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
[31] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.
[32] G. Litjens *et al.*, "A survey on deep learning in medical image analysis," *Med. Image Anal.*, vol. 42, pp. 60–88, Dec. 2017.
[33] D. Shen, G. Wu, and H. Suk, "Deep learning in medical image analysis," *Annu. Rev. Biomed. Eng.*, vol. 19, pp. 221–248, Jun. 2017.
[34] L. Fang, D. Cunefare, C. Wang, R. H. Guymer, S. Li, and S. Farsiu, "Automatic segmentation of nine retinal layer boundaries in OCT images of non-exudative AMD patients using deep learning and graph search," *Biomed. Opt. Express*, vol. 8, no. 5, pp. 2732–2744, 2017.
[35] X. Liu *et al.*, "Automated layer segmentation of retinal optical coherence tomography images using a deep feature enhanced structured random forests classifier," *IEEE J. Biomed. Health Inform.*, 2018, doi: 10.1109/JBHI.2018.2856276.
[36] X. Liu, T. Fu, Z. Pan, D. Liu, W. Hu, and B. Li, "Semi-supervised automatic layer and fluid region segmentation of retinal optical coherence tomography images using adversarial learning," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 2780–2784.
[37] A. Shah, L. Zhou, M. D. Abrámoff, and X. Wu, "Multiple surface segmentation using convolution neural nets: Application to retinal layer segmentation in OCT images," *Biomed. Opt. Express*, vol. 9, no. 9, pp. 4509–4526, 2018.
[38] J. Hamwood, D. Alonso-Caneiro, S. A. Read, S. J. Vincent, and M. J. Collins, "Effect of patch size and network architecture on a convolutional neural network approach for automatic segmentation of OCT retinal layers," *Biomed. Opt. Express*, vol. 9, no. 7, pp. 3049–3066, 2018.
[39] F. G. Venhuizen *et al.*, "Robust total retina thickness segmentation in optical coherence tomography images using convolutional neural networks," *Biomed. Opt. Express*, vol. 8, no. 7, pp. 3292–3316, 2017.
[40] Y. Xu *et al.*, "Dual-stage deep learning framework for pigment epithelium detachment segmentation in polypoidal choroidal vasculopathy," *Biomed. Opt. Express*, vol. 8, no. 9, pp. 4061–4076, 2017.
[41] I. J. Goodfellow *et al.*, "Generative adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 3, 2014, pp. 2672–2680.
[42] N. Souly, C. Spampinato, and M. Shah, "Semi supervised semantic segmentation using generative adversarial network," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5689–5697.
[43] W.-C. Hung, Y.-H. Tsai, Y.-T. Liou, Y.-Y. Lin, and M.-H. Yang. (2018). "Adversarial learning for semi-supervised semantic segmentation." [Online]. Available: https://arxiv.org/abs/1802.07934
[44] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters—Improve semantic segmentation by global convolutional network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1743–1751.
[45] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, Aug. 2013.
[46] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: Learning dense volumetric segmentation from sparse annotation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2016, pp. 424–432.
[47] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. Int. Conf. Learn. Represent.*, 2016, pp. 1–13.
[48] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected CRFs with Gaussian edge potentials," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2011, pp. 109–117.
[49] P. Luc, C. Couprie, S. Chintala, and J. Verbeek, "Semantic segmentation using adversarial networks," in *Proc. NIPS Workshop Adversarial Training*, 2016, pp. 1–12.
[50] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2016, pp. 1520–1528.
[51] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, vol. 30, no. 1, 2013, pp. 1–3.
[52] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 2650–2658.
[53] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. 4th Int. Conf. 3D Vis.*, Oct. 2016, pp. 565–571.
[54] J. Tian, B. Varga, G. M. Somfai, W.-H. Lee, W. E. Smiddy, and D. C. DeBuc, "Real-time automatic segmentation of optical coherence tomography volume data of the macular region," *PLoS ONE*, vol. 10, no. 8, p. e0133908, 2015.
[55] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *Proc. OSDI*, vol. 16, 2016, pp. 265–283.
[56] Y. He *et al.* (2018). "Topology guaranteed segmentation of the human retina from OCT using convolutional neural networks." [Online]. Available: https://arxiv.org/abs/1803.05120
[57] R. Zhang *et al.*, "Automatic segmentation of acute ischemic stroke from DWI using 3-D fully convolutional DenseNets," *IEEE Trans. Med. Imag.*, vol. 37, no. 9, pp. 2149–2160, Sep. 2018.

XIAOMING LIU received the Ph.D. degree from Zhejiang University, China, in 2007. From 2014 to 2015, he was a Visiting Scholar with The University of North Carolina at Chapel Hill, NC, USA. He is currently a Full Professor with the College of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, China. His research interests include medical image processing, pattern recognition, and machine learning.

JUN CAO is currently a Graduate Student with the College of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, China. His research interests include medical image processing and machine learning.

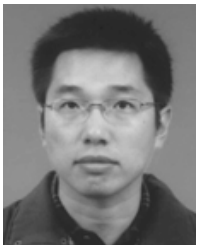


TIANYU FU is currently a Graduate Student with the College of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, China. His research interests include medical image processing and machine learning.

ZHIFANG PAN received the Ph.D. degree from the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China, in 2013. In 1999, he joined Wenzhou Medical University, where he is currently an Associate Professor and the Deputy Director of the Information Technology Center. His research interests include the analysis of medical image and big data.

WEI HU received the Ph.D. degree from Zhejiang University, China, in 2008. He is currently a Full Professor with the College of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, China. His research interests include system on chip and computer architecture.

KAI ZHANG received the Ph.D. degree from the Huazhong University of Science and Technology, China, in 2008. From 2008 to 2010, he held a Postdoctoral position at Peking University. In 2017, he was a Visiting Scholar with Oklahoma State University. He is currently a Full Professor with the College of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, China. His research interests include computational intelligence and deep learning.



JUN LIU received the Ph.D. degree from the Wuhan University of Science and Technology, China, in 2012. From 2015 to 2016, he was a Visiting Scholar with the University of Michigan, Ann Arbor, MI, USA. He is currently a Full Professor with the College of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, China. His research interests include medical image processing, ultrasound image analysis, and machine learning.