Deep Supervision with Additional Labels for Retinal Vessel Segmentation Task

Yishuo Zhang and Albert C. S. Chung(✉)

Lo Kwee-Seong Medical Image Analysis Laboratory, Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong, China
ys.zhang@connect.ust.hk, achung@cse.ust.hk

Abstract. Automatic analysis of retinal fundus images is of vital importance in diagnosis tasks of retinopathy. Segmenting vessels accurately is a fundamental step in analysing retinal images. However, it is usually difficult due to various imaging conditions, low image contrast and the appearance of pathologies such as micro-aneurysms. In this paper, we propose a novel method with deep neural networks to solve this problem. We utilize U-net with residual connection to detect vessels. To achieve better accuracy, we introduce an edge-aware mechanism, in which we convert the original task into a multi-class task by adding additional labels on boundary areas. In this way, the network will pay more attention to the boundary areas of vessels and achieve a better performance, especially in tiny vessels detecting. Besides, side output layers are applied in order to give deep supervision and therefore help convergence. We train and evaluate our model on three databases: DRIVE, STARE, and CHASEDB1. Experimental results show that our method has a comparable performance with AUC of 97.99% on DRIVE and an efficient running time compared to the state-of-the-art methods.

1 Introduction

Retinal vessels are commonly analysed in the diagnosis and treatment of various ophthalmological diseases. For example, retinal vascular structures are correlated to the severity of diabetic retinopathy [5], which is a cause of blindness globally. Thus, the precise segmentation of retinal vessels is of vital importance. However, this task is often extremely challenging due to the following factors [1]: 1. The shape and width of vessel vary, which cannot be represented by a simple pattern; 2. The resolution, contrast and local intensity change among different fundus images, increasing the difficulty of segmenting; 3. Other structures, like optical disks and lesions, can be interference factors; 4. Extremely thin vessels are hard to detect due to the low contrast and noise.

In recent years, a variety of methods have been proposed to solve retinal vessel segmentation tasks, including unsupervised methods [9] and supervised methods [11]. Although promising performances have been shown, there is still some room for improvement. As we mentioned before, tiny capillaries are hard

A. F. Frangi et al. (Eds.): MICCAI 2018, LNCS 11071, pp. 83–91, 2018.
https://doi.org/10.1007/978-3-030-00934-2_10

to find and missing these can lead to low sensitivity. Besides, methods that need less running time are preferred in clinical practice. In this paper, we aim to design a more effective and efficient method to tackle these problems.

The emergence of deep learning methods provides a powerful tool for computer vision tasks and these kinds of methods have outperformed other methods in many areas. By stacking convolutional layers and pooling layers, networks can gain the capacity to learn the very complicated representation of features. U-net, proposed in [12], can deal with image patches in an end-to-end manner and therefore is widely used in medical image segmentation.

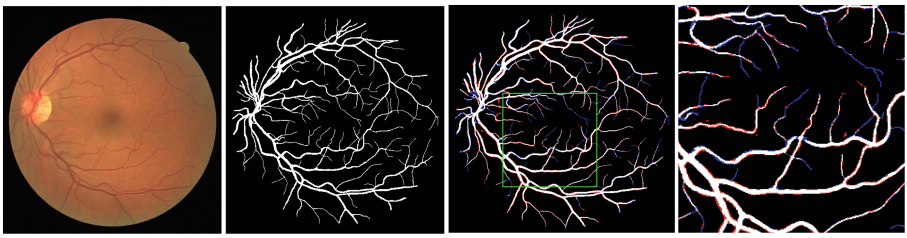

Fig. 1. Images sampled from datasets. From left to right: original fundus image, ground truth, output of a single trained U-net and zoomed segment inside the green rectangle in the third image. In the last image, blue regions denote false negative, while red regions denote false positive.

We analyse the output of a single trained U-net model as shown in Fig. 1. Most mislabelled pixels come from boundaries between foreground and background. Regarding thick vessels, the background areas around vessels are easy to be labelled as positive. However regarding very thin vessels, many of these are ignored by networks and labelled as background. To tackle this problem, we process the ground truth, by labelling the boundary, thick vessels and thin vessels as different classes, which forces the networks to pay different extra attention to error-prone regions. This operation makes the original task become a harder task. If the new task can be solved by our method perfectly, then so could the original task. Besides, we also utilize deep supervision to help networks converge.

Our main contributions are as follows:

1. Introducing a deep supervision mechanism into U-net, which helps networks learn a better semantically representation;
2. Separating thin vessels from thick vessels during the progress of training;
3. Applying an edge-aware mechanism by labelling the boundary region to extra classes, making the network focus on the boundaries of vessels and therefore get finer edges in the segmentation result.

2 Proposed Method

2.1 U-Net

The architecture of U-net is illustrated in Fig. 2. The left-hand part consists of four blocks, each of which contains stacked convolutional layers (Conv) to learn hierarchical features. The size of the input feature maps is halved after each stage, implemented by a Conv layer with a stride of 2. In contrast, the number of feature channels increases when the depth increases, in order to learn more complicated representations. The right-hand part has a similar structure to the left part. The size of input feature maps is doubled after each stage by a deconvolution layer to reconstruct spatial information.

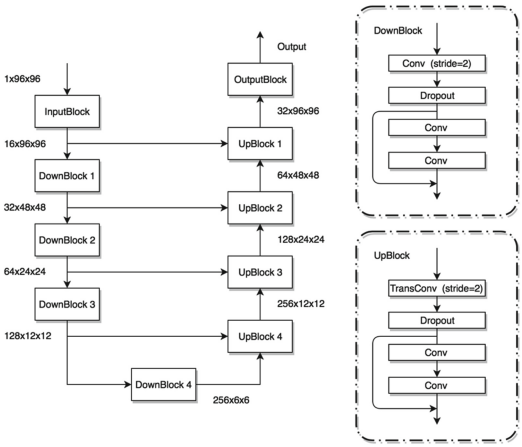

Fig. 2. Architecture of a simple U-net. We annotate shapes about feature maps of each block in the format of 'Channels, Width, Height'. Inner structures of DownBlock and UpBlock are shown on the right, where each Conv layer is followed by two unseen layers: a BatchNorm layer and a ReLU layer.

To utilize feature learned by earlier layers at subsequent layers, feature maps from the left-hand blocks will be fed into the corresponding right-hand blocks. In this way, networks can gain detailed information which may be lost in former downsampling operations but useful for fine boundary prediction. To improve the robustness and help convergence, we apply a residual connection [3] inside each block, which adds feature maps before Conv layers to the output maps pixel-wisely. We also leverage Dropout and BatchNorm inside each block to reduce overfitting and gradient vanishing respectively.

2.2 Additional Label

Additional labels are added to the original ground truth before training, which converts this task into a multi-class segmentation task. Firstly, we distinguish

thick vessels from thin vessels (with a width of 1 or 2 pixels), implemented by an opening operation. Then, we locate the pixels near to the vessel by a dilation operation and label them to the additional class. Therefore, we have 5 classes, which are 0 (other background pixels), 1 (background near thick vessels), 2 (background near thin vessels), 3 (thick vessel) and 4 (thin vessel) (Fig. 3).

The objective of this is to force the networks to treat background pixels differently. As we reported above, the boundary region is easy to be mislabelled. We separate these classes so that we can give more supervision in crucial areas by modifying the class weight in the loss function but not influencing others. Boundary classes have heavier weights in the loss function, which means that these classes will attract a higher penalty if labelled wrongly.

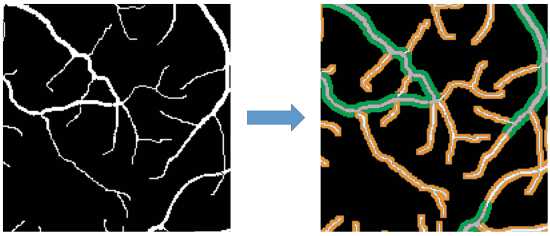

Fig. 3. Generated new multi-class ground truth, where different classes are shown in different colours: 0 (black), 1 (green), 2 (orange), 3 (grey) and 4 (white).

2.3 Deep Supervision

Deep supervision [6] is employed to solve the problem of information loss during forward propagation and improve detailed accuracy. This mechanism is beneficial because it gives semantic representations to the intermediate layers. We implement it by adding four side output layers as shown in Fig. 4. The output of each side layer is compared with the ground truth to calculate auxiliary losses. Final prediction maps are generated by fusing the outputs of all four side layers.

We employ cross-entropy as loss function and calculate it for the final output as well as each side output. Owing to the amounts of different classes being imbalanced, we add a class-balanced weight for each class to correct imbalances. As we discussed before, pixels of boundaries around thick vessels and pixels of thin vessels should be given relatively heavier weights.

$$CE(pred, target) = -\sum_{i} weight_i \times target_i \times log(pred_i). \tag{1}$$

The total loss is defined as below, comprising of loss of fused output, losses of side outputs and L-2 regular term.

$$Loss = CE(fuse, GT) + \sum_{3}^{n} CE(side_i, GT) + \frac{\lambda}{2}\|w\|^2. \tag{2}$$

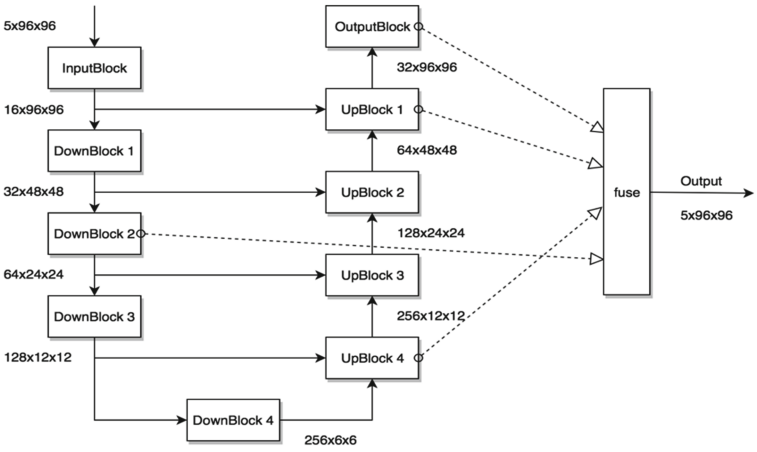

Fig. 4. Diagram of our proposed method.

3 Experiments

We implement our model with PyTorch library. Stochastic gradient descent algorithm (SGD) with momentum is utilized to optimize our model. The learning rate is set to 0.01 initially and halved every 100 epochs. We train the whole model for 200 epochs on a single NVIDIA GPU (GeForce Titan X). The training progress takes nearly 10 h.

3.1 Datasets

We evaluate our method on three public datasets: DRIVE [13], STARE [4] and CHASEDB1 [2], each of which contains images with two labelled masks annotated by different experts. We take the first labelled mask as ground truth for training and testing. The second labelled masks are used for comparison between our model and a human observer. The DRIVE dataset contains 20 training images and 20 testing images; thus, we take them as the training set and testing set respectively. STARE and CHASEDB1 datasets contain 20 and 28 images, respectively. As these two datasets are not divided for training and testing, we perform a four-fold cross-validation, following [10].

Before feeding the original image into networks, some preprocessing operations are performed. We employ contrast-limited adaptive histogram equalization (CLAHE) to enhance the image and increase contrast. Then the whole images are cropped into patches with the size of 96 * 96 pixels. To augment the training data, we perform the flip, affine transformation, and noising operations randomly. In addition, the lightness and contrast of the original images are changed randomly to improve the robustness of the model.

3.2 Results

A vessel segmentation task can be viewed as an unbalanced pixel-wise classification task. For evaluation purpose, measurements including Specificity (Sp), Sensitivity (Se) and Accuracy (Acc) are computed. They are defined as below:

$$Sp = \frac{TN}{TN + FP}, Se = \frac{TP}{TP + FN}, Acc = \frac{TP + TN}{TP + FP + TN + FN}, \quad (3)$$

Here TP, FN, TN, FP denote true positive, false negative, true negative and false positive, respectively. Additionally, a better metric, area under the receiver operating characteristic (ROC) curve (AUC), is used. We believe that AUC is more suitable for measuring an unbalanced situation. A perfect classifier should have an AUC value of 1.

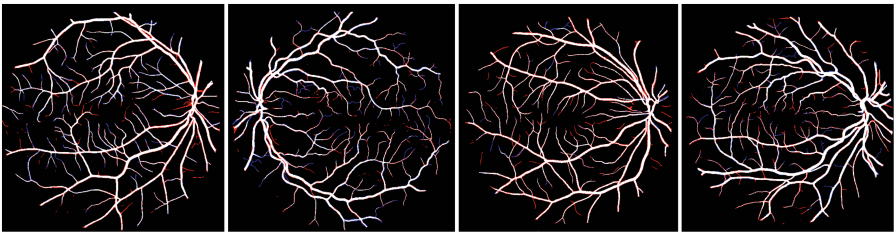

Fig. 5. Examples of our experiment output.

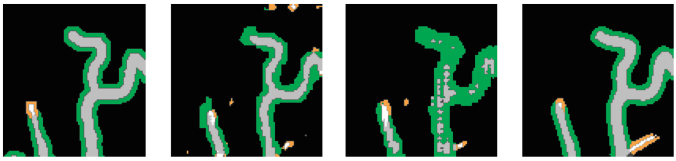

Fig. 6. Comparison between side output and ground truth. From left to right: Ground Truth, 2 side outputs, and final prediction.

We have three observations: 1. Even if the side output cannot locate the vessel, they can locate the boundary region, which can help to find the vessel in final output precisely as a guide. 2. As the resolution of the side output is lower, the tiny vessel may be missed but the boundary region is more distinct and easier to find. This shows mutual promotion between the additional label and deep supervision. 3. The boundary of the boundary region is not refined, but it does not affect the prediction because we will take all of the boundary regions as background (Figs. 5 and 6).

Table 1. Performance comparison with simple U-net on dataset DRIVE

Methods	AUC of all vessels	AUC of thick vessels	AUC of thin vessels
Simple U-net	0.9736	0.9830	0.8678
Our method	0.9799	0.9897	0.9589

To validate the effect of our idea, we perform comparison experiments with a simple U-net. With additional label and well-designed deep supervision, our method has better capabilities of detecting vessels, especially for capillaries. AUC of thin vessels has been increased by 9.11%, as shown in Table 1.

3.3 Comparison

We report performances of our method in respect to the aforementioned metrics, compared with other state-of-the-art methods, as shown in Tables 2 and 3.

Table 2. Performance comparison on the DRIVE dataset

Methods	Acc	Sp	Se	AUC
2nd Observer	0.9472	0.9730	0.7760	N.A.
Fraz et al. [2]	0.9480	**0.9807**	0.7406	0.9747
Liskowski et al. [8]	**0.9535**	**0.9807**	0.7811	0.9790
Mo et al. [10]	0.9521	0.9780	0.7779	0.9782
Leopold et al. [7]	0.9106	0.9573	0.6963	0.8268
Our method	0.9504	0.9618	**0.8723**	**0.9799**

Table 3. Performance comparison on STARE and CHASEDB1 datasets

Methods	STARE				CHASEDB1			
	Acc	Sp	Se	AUC	Acc	Sp	Se	AUC
2nd Observer	0.9353	0.9387	**0.8951**	N.A	0.9560	0.9793	0.7425	N.A.
Fraz et al. [2]	0.9534	0.9763	0.7548	0.9768	0.9468	0.9711	0.7224	0.9712
Liskowski et al. [8]	**0.9729**	0.9862	0.8554	**0.9928**	0.9628	0.9836	0.7816	0.9823
Mo et al. [10]	0.9674	0.9844	0.8147	0.9885	0.9599	0.9816	0.7661	0.9812
Leopold et al. [7]	0.9045	0.9472	0.6433	0.7952	0.8936	0.8961	**0.8618**	0.8790
Our method	0.9712	**0.9901**	0.7673	0.9882	**0.9770**	**0.9909**	0.7670	**0.9900**

We have highlighted the highest scores for each column. Our method achieves the highest Sensitivity on the DRIVE dataset and the highest Specificity on the

other two datasets. Due to the differences of inherent errors among datasets and the class imbalance, we prefer using AUC as an equatable metric for comparison. Our method has the best performance on the DRIVE and CHASEDB1 datasets in terms of AUC.

Table 4. Time comparison with other methods

Method	Training time (h)	Running time (s)
Liskowski et al. [8]	8	92
Mo et al. [10]	10	0.4
Our method	10	0.9

In terms of running time, our method is also computationally efficient when compared to other methods (Table 4). Our proposed method can deal with an image size of 584*565 in 1.2 s, much faster than the method proposed in [8]. This benefit is obtained from our method by using the U-net architecture which works from patch to patch, instead of using a patch to predict the central pixel alone. The method proposed in [10] is a little faster than ours, as their network has less up-sampling layers. However, removing up-sampling leads to a decrease in fine prediction and especially sensitivity. By overall consideration, we choose proper numbers of layers as used in our presented method, which can achieve the best performance with highly acceptable running time.

4 Conclusion

In this paper, we propose a novel deep neural network to segment retinal vessel. To give more importance to boundary pixels, we label thick vessels, thin vessels and boundaries into different classes, which makes a multi-class segmentation task. We use a U-net with residual connections to perform the segmentation task. Deep supervision is introduced to help the network learn better features and semantic information. Our method offers a good performance and efficient running time compared to other state-of-the-art methods, which can give high efficacy in clinical applications.

References

1. Fraz, M.M., et al.: An approach to localize the retinal blood vessels using bit planes and centerline detection. Comput. Methods Programs Biomed. **108**(2), 600–616 (2012)
2. Fraz, M.M., et al.: An ensemble classification-based approach applied to retinal blood vessel segmentation. IEEE Trans. Biomed. Eng. **59**(9), 2538–2548 (2012)
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)

4. Hoover, A., Kouznetsova, V., Goldbaum, M.: Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. IEEE Trans. Med. Imaging **19**(3), 203–210 (2000)
5. Jelinek, H., Cree, M.J.: Automated Image Detection of Retinal Pathology. CRC Press, Boca Raton (2009)
6. Lee, C.Y., Xie, S., Gallagher, P., Zhang, Z., Tu, Z.: Deeply-supervised nets. In: Artificial Intelligence and Statistics, pp. 562–570 (2015)
7. Leopold, H.A., Orchard, J., Zelek, J.S., Lakshminarayanan, V.: Pixelbnn: augmenting the pixelcnn with batch normalization and the presentation of a fast architecture for retinal vessel segmentation. arXiv preprint arXiv:1712.06742 (2017)
8. Liskowski, P., Krawiec, K.: Segmenting retinal blood vessels with deep neural networks. IEEE Trans. Med. Imaging **35**(11), 2369–2380 (2016)
9. Marín, D., Aquino, A., Gegúndez-Arias, M.E., Bravo, J.M.: A new supervised method for blood vessel segmentation in retinal images by using gray-level and moment invariants-based features. IEEE Trans. Med. Imaging **30**(1), 146–158 (2011)
10. Mo, J., Zhang, L.: Multi-level deep supervised networks for retinal vessel segmentation. Int. J. Comput. Assist. Radiol. Surg. **12**(12), 2181–2193 (2017)
11. Orlando, J.I., Blaschko, M.: Learning fully-connected CRFs for blood vessel segmentation in retinal images. In: Golland, P., Hata, N., Barillot, C., Hornegger, J., Howe, R. (eds.) MICCAI 2014. LNCS, vol. 8673, pp. 634–641. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10404-1_79
12. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
13. Staal, J., Abràmoff, M.D., Niemeijer, M., Viergever, M.A., Van Ginneken, B.: Ridge-based vessel segmentation in color images of the retina. IEEE Trans. Med.Imaging **23**(4), 501–509 (2004)