DeepVessel: Retinal Vessel Segmentation via Deep Learning and Conditional Random Field

Huazhu Fu[1](✉), Yanwu Xu[1], Stephen Lin[2],
Damon Wing Kee Wong[1], and Jiang Liu[1,3]

[1] Institute for Infocomm Research, A*STAR, Singapore, Singapore
fuhz@i2r.a-star.edu.sg
[2] Microsoft Research, Beijing, China
[3] Cixi Institute of Biomedical Engineering,
Ningbo Institute of Materials Technology and Engineering,
Chinese Academy of Sciences, Ningbo, China

Abstract. Retinal vessel segmentation is a fundamental step for various ocular imaging applications. In this paper, we formulate the retinal vessel segmentation problem as a boundary detection task and solve it using a novel deep learning architecture. Our method is based on two key ideas: (1) applying a multi-scale and multi-level Convolutional Neural Network (CNN) with a side-output layer to learn a rich hierarchical representation, and (2) utilizing a Conditional Random Field (CRF) to model the long-range interactions between pixels. We combine the CNN and CRF layers into an integrated deep network called *DeepVessel*. Our experiments show that the DeepVessel system achieves state-of-the-art retinal vessel segmentation performance on the DRIVE, STARE, and CHASE_DB1 datasets with an efficient running time.

1 Introduction

Retinal vessels are of much diagnostic significance, as they are commonly examined to evaluate and monitor various ophthalmological diseases. However, manual segmentation of retinal vessels is both tedious and time-consuming. To assist with this task, many approaches have been introduced in the last two decades to segment retinal vessels automatically. For example, Marin *et al.* employed the gray-level vector and moment invariant features to classify each pixel using a neural network [8]. Nguyen *et al.* utilized a multi-scale line detection scheme to compute vessel segmentation [11]. Orlando *et al.* performed vessel segmentation using a fully-connected Conditional Random Field (CRF) whose configuration is learned using a structured-output support vector machine [12]. Existing methods such as these, however, lack sufficiently discriminative representations and are easily affected by pathological regions, as shown in Fig. 1.

Deep learning (DL) have recently been demonstrated to yield highly discriminative representations that have aided in many computer vision tasks. For

S. Ourselin et al. (Eds.): MICCAI 2016, Part II, LNCS 9901, pp. 132–139, 2016.
DOI: 10.1007/978-3-319-46723-8_16

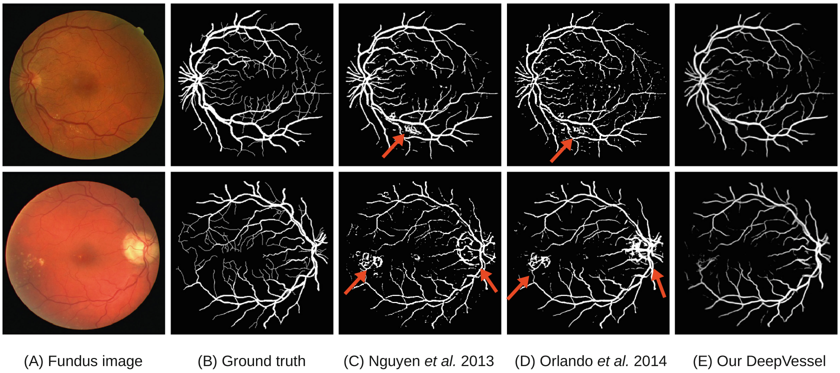

Fig. 1. Retinal vessel segmentation results. Existing vessel segmentation methods (e.g., Nguyen *et al.* [11], and Orlando *et al.* [12]) are affected by the optic disc and pathological regions (highlighted by red arrows), while our DeepVessel deals well with these regions.

example, Convolutional Neural Networks (CNNs) have brought heightened performance in image classification and semantic image segmentation. Xie *et al.* employed a holistically-nested edge detection (HED) system with deep supervision to resolve the challenging ambiguity in object boundary detection [16]. Zheng *et al.* reformulated the Conditional Random Field (CRF) as a Recurrent Neural Network (RNN) to improve semantic image segmentation [18]. These works inspire us to learn rich hierarchical representation based on a DL architecture.

A DL-based vessel segmentation method is proposed in [9], which addressed the problem as pixel classification using a deep neural network. In [7], Li *et al.* employed cross-modality data transformation from retinal image to vessel map, and outputted the label map of all pixels for a given image patch. These methods has two drawbacks: first, it does not account for non-local correlations in classifying individual pixels/patches, which leads to failures caused by noise and local pathological regions; second, the classification strategy is computationally intensive for both the training and testing phases. In our paper, we address retinal vessel segmentation as a boundary detection task that is solved using a novel DL system called *DeepVessel*, which utilizes a CNN with a side-output layer to learn discriminative representations, and also a CRF layer that accounts for non-local pixel correlations. With this approach, our DeepVessel system achieves state-of-the-art performance on publicly-available datasets (DRIVE, STARE, and CHASE_DB1) with relatively efficient processing.

2 Proposed Method

Our DeepVessel architecture consists of three main layers. The first is a convolutional layer used to learn a multi-scale discriminative representation. The second is a side-output layer that operates with the early layers to generate a companion local output. The last one is a CRF layer, which is employed to

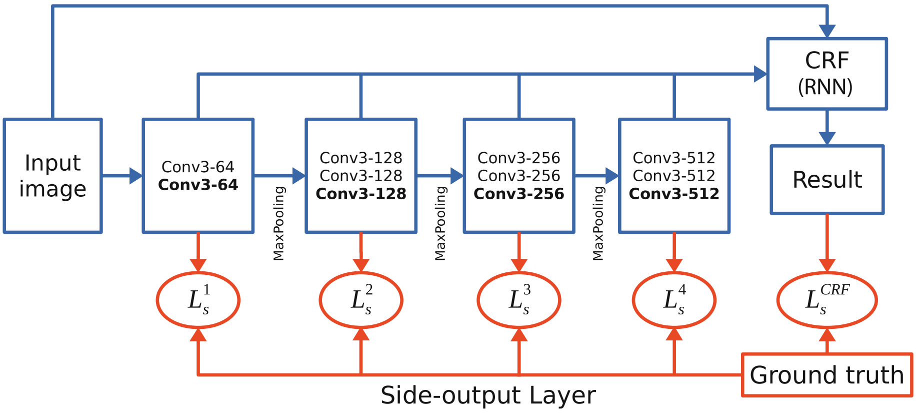

Fig. 2. Architecture of our DeepVessel system, which consists of convolutional, side-output, and CRF layers. The front network is a four-stage HED-like architecture [16], where the side-output layer is inserted after the last convolutional layers in each stage (marked in **Bold**). The convolutional layer parameters are denoted as "Conv<receptive field size>-<number of channels>". The CRF layer is represented as an RNN as done in [18]. The ReLU activation function is not shown for brevity. The red blocks exist only in the training phase.

further take into account the non-local pixel correlations. The overall architecture of our DeepVessel system is illustrated in Fig. 2.

Convolutional Layer is used to learn local feature representations based on patches randomly sampled from the image. Suppose $\mathbf{L}_j^{(n)}$ is the j-th output map of the n-th layer, and $\mathbf{L}_i^{(n-1)}$ is the i-th input map of the n-th layer. The output of the convolutional layer is then defined as:

$$\mathbf{L}_j^{(n)} = f(\sum_i \mathbf{L}_i^{(n-1)} * \mathbf{W}_{ij}^{(n)} + b_j^{(n)}\mathbf{1}), \tag{1}$$

where $\mathbf{W}_{ij}^{(n)}$ is the kernel linking the i-th input map to the j-th output map, $*$ denotes the convolution operator, and $b_j^{(n)}$ is the bias element.

Side-output Layer acts as a classifier that produces a companion local output for early layers [6]. Suppose $\mathbf{W}$ denotes the parameters of all the convolutional layers, and there are M side-output layers in the network, where the corresponding weights are denoted as $\mathbf{w} = (\mathbf{w}^{(1)}, ..., \mathbf{w}^{(M)})$. The objective function of the side-output layer is given as:

$$\mathcal{L}_s(\mathbf{W}, \mathbf{w}) = \sum_{m=1}^{M} \alpha_m L_s^{(m)}(\mathbf{W}, \mathbf{w}^{(m)}), \tag{2}$$

where α_m is the loss function fusion-weight or each side-output layer, and $L_s^{(m)}$ denotes the image-level loss function, which is computed over all pixels in the training retinal image X and its vessel ground truth Y. For the retinal image, the pixels of the vessel and background are imbalanced, thus we follow HED [16] to utilize a class-balanced cross-entropy loss function:

$$L_s^{(m)}(\mathbf{W}, \mathbf{w}^{(m)}) = -\frac{|Y^-|}{|Y|} \sum_{j \in Y^+} \log \sigma(a_j^{(m)}) - \frac{|Y^+|}{|Y|} \sum_{j \in Y^-} \log(1 - \sigma(a_j^{(m)})), \quad (3)$$

where $|Y^+|$ and $|Y^-|$ denote the vessel and background pixels in the ground truth Y, and $\sigma(a_j^{(m)})$ is the sigmoid function on pixel j of the activation map $A_s^{(m)} \equiv a_j^{(m)}, j = 1, ..., |Y|$ in side-output layer m. Simultaneously, we can obtain the vessel prediction map of each side-output layer m by $\hat{Y}_s^{(m)} = \sigma(A_s^{(m)})$.

Conditional Random Field (CRF) Layer is used to model non-local pixel correlations. Although the CNN can produce a satisfactory vessel probability map, it still has some problems. First, a traditional CNN has convolutional filters with large receptive fields and hence produces maps too coarse for pixel-level vessel segmentation (e.g., non-sharp boundaries and blob-like shapes). Second, a CNN lacks smoothness constraints, which may result in small spurious regions in the segmentation output. Thus, we utilize a CRF layer to obtain the final vessel segmentation result. Following the fully-connected CRF model of [5], each node is a neighbor of each other, and it takes into account long-range interactions in the whole image. We denote $\mathbf{v} = \{v_i\}$ as a labeling over all pixels of the image, with $v_i = 1$ for vessel and $v_i = 0$ for background. The energy of a label assignment $\mathbf{v}$ is given by:

$$E(\mathbf{v}) = \sum_i \psi_u(v_i) + \sum_{i<j} \psi_p(v_i, v_j), \quad (4)$$

with:

$$\psi_u(v_i) = \frac{1}{M} \sum_{m=1}^{M} a_i^{(m)}, \quad \textbf{and}, \quad \psi_p(v_i, v_j) = \mu(v_i, v_j) \sum_{d=1}^{D} h^{(d)} k^{(d)}(\mathbf{f}_i, \mathbf{f}_j), \quad (5)$$

where $\psi_u(v_i)$ and $\psi_p(v_i, v_j)$ are the unary and pairwise terms, respectively. $a_j^{(m)}$ is the value at pixel i in the activation map $A_s^{(m)}$ of side-output layer m, and $k^{(d)}$ for $d = 1, ..., D$ is the Gaussian kernel applied on feature vectors. The feature vector of pixel i, denoted by $\mathbf{f}_i$, is derived from image features such as spatial location and RGB values. An effective solution to minimize the CRF energy $E(\mathbf{v})$ in Eq. (4) is through mean-field approximation [5]. In our system, we employ the implementation of [18], in which the CRF is reformulated as a Recurrent Neural Network (RNN) layer and can be utilized in an end-to-end DL architecture.

Our DeepVessel Architecture is an end-to-end system illustrated in Fig. 2, which contains four CNN stages and one CRF stage. Each CNN stage includes multiple convolutional and ReLU layers, and one side-output layer. The side-output layer is connected to the last convolutional layer in each stage to support deep layer supervision. The objective function of the whole system is:

$$(\mathbf{W}, \mathbf{w}, \mathbf{h}) = \arg\min \left(\mathcal{L}_s(\mathbf{W}, \mathbf{w}) + L_s^{CRF}(\mathbf{W}, \mathbf{w}, \mathbf{h}) \right), \quad (6)$$

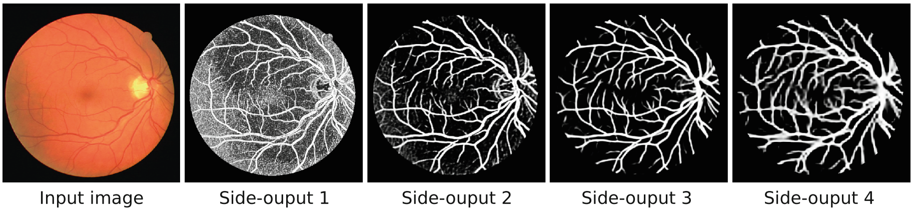

Fig. 3. The vessel prediction map for each side-output layer in our architecture.

where $\mathbf{h}$ is a CRF layer parameter, $\mathcal{L}_s$ is the CNN layer loss function in Eq. (2), and L_s^{CRF} is the CRF layer loss function, specifically the class-balanced cross-entropy loss function in Eq. (3). We minimize the objective function via standard stochastic gradient descent. In our DeepVessel architecture, we only employ four CNN stages with side-output layers. The main reason is that retinal vessels in fundus images are different from general object edges in natural images. An object edge separates two regions of different appearance, which allows the boundary to be detectable even at deeper layers. By contrast, a retinal vessel appears merely as a curved line, which is too thin to respond in the higher stride layers. Thus, we only employ four side-output layers. The vessel prediction map example for each side-output layer is shown in Fig. 3, where earlier side-output layers have a smaller receptive field size and respond to local details, while deeper layers represent appearance at a larger scale.

3 Experiments

We implement our framework using the *Caffe* library and build on top of the implementation of HED [16]. The model parameters follow the configuration used in [16]. We employed a two-step fine-tuning approach that first utilizes the ARIA dataset [2] to fine-tune the initial parameters, and then the DRIVE training set [15] to obtain the final fine-tuning parameters. We rotate all training images to eight different angles, and rescale the ARIA images to the same size as the DRIVE images. The whole fine-tuning phase takes about two days on a single NVIDIA K40 GPU (10,000 iterations). For a 565×584 image, it takes about $1.3\,s$ to generate the final vessel map.

3.1 Experimental Results

We evaluate our method[1] on three publicly datasets: DRIVE [15], STARE [4], and CHASE_DB1 [3]. These datasets provide two manual segmentations generated by two different experts for each image. The first observer is selected as

[1] Our results on all three datasets can be downloaded from http://hzfu.github.io/subpage/deepvessel/deepvessel.html.

ground truth and used for performance evaluation in the literature. We performed the evaluation in terms of Accuracy ($Acc = \frac{TP+TN}{TP+FN+TN+FP}$), and Sensitivity ($Sen = \frac{TP}{TP+FN}$), where TP, TN, FP and FN represent the number of true positives, true negatives, false positives and false negatives, respectively. Note that there is no training set in the STARE and CHASE_DB1 datasets, thus we only utilize the DRIVE training set to fine-tune the final parameters.

We compare our method with several state-of-the-art vessel segmentation methods, and also report the ground truth labeling of the second observer as the performance of a human observer. Our DeepVessel system outputs a probability map, and Otsu's thresholding method [13] is employed to obtain the binary labeling result automatically in the experiments. Table 1 lists the performances on the three datasets, where the reported performance scores from the original papers are used. Our method obtains the best Accuracy scores among the methods, which include the other DL method [9] on the DRIVE dataset. And our method obtains Accuracy performance similar to the human observer on the CHASE_DB1 dataset and a better Accuracy score on the other two datasets.

We provide the results produced by the individual and average fusion results of the side-output layers in Table 1. We also report our results without

Table 1. Performance of different segmentation methods on three datasets.

Methods	DRIVE			STARE		CHASE_DB1	
	Acc	Sen	Time	Acc	Sen	Acc	Sen
Human observer	0.9472	0.7761	-	0.9349	0.8952	0.9538	0.8092
Staal [15]	0.9441	0.7194	15 m	0.9516	0.6970	-	-
Mendonca [10]	0.9452	0.7344	2.5 m	0.9440	0.6996	-	-
Marin [8]	0.9452	0.7067	1.5 m	0.9526	0.6944	-	-
Fraz [3]	0.9480	0.7406	2 m	0.9534	0.7548	0.9469	0.7224
Nguyen [11]	0.9407	0.7429	2.5 s	0.9326	0.8014	0.9213	0.7153
Zhao [17]	0.9477	0.7354	2 m	0.9509	0.7187	-	-
Melinscak [9]	0.9466	0.7276	-	-	-	-	-
Azzopardi [1]	0.9442	0.7655	10 s	0.9497	0.7716	0.9387	0.7585
Roychowdhury [14]	0.9494	0.7395	2.5 s	0.9560	0.7317	0.9467	0.7615
HED [16]	0.9435	0.7364	1.5 s	0.9402	0.7116	0.9380	0.7151
Side-output 1	0.9151	0.5523	-	0.8934	0.6273	0.8102	0.5077
Side-output 2	0.9426	0.6872	-	0.9404	0.7415	0.9290	0.6138
Side-output 3	0.9410	0.7458	-	0.9490	0.7778	0.9468	0.7965
Side-output 4	0.9139	0.5744	-	0.9359	0.6451	0.9465	0.7323
Side-output fusion	0.9472	0.7298	-	0.9531	0.7469	0.9478	0.7485
DeepVessel w/o S	0.9335	0.7164	-	0.9438	0.7216	0.9365	0.7223
DeepVessel	**0.9523**	0.7603	**1.3 s**	**0.9585**	0.7412	**0.9489**	0.7130

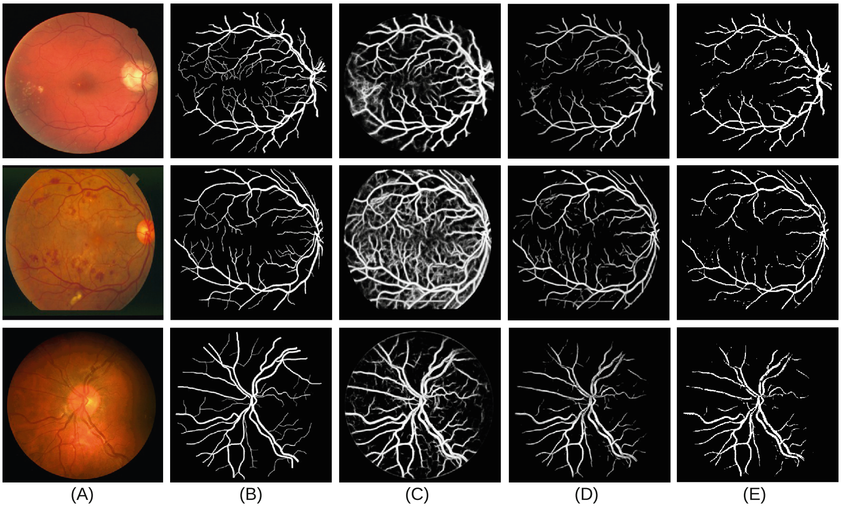

Fig. 4. Examples of results from the dataset. From top to bottom are the fundus images from the DRIVE, STARE, and CHASE_DB1 datasets. From left to right: (A) Fundus images, (B) Ground truth, (C) Fusion results of side-output layers, (D) Our DeepVessel results, (E) Thresholded DeepVessel results

side-output layers (DeepVessel w/o S). We observe that the second and third side-output layers obtain better performance than the other two layers, which is also observed in Fig. 3. The side-output fusion combines all the side-output layer outputs and generally performs better than any of the individual layers and the version without side-output layers. Figure 4 displays some results. It can be observed that our DeepVessel with CRF produces a clearer vessel segmentation result than the fusion result from only the side-output layers, especially for pathological regions as shown in the second row of Fig. 4.

4 Conclusion

In this paper, we have developed a retinal vessel segmentation method, called DeepVessel, based on a novel deep learning architecture. A discriminative representation is learned by a CNN with side-output layers, and a high quality vessel probability map is produced using a CRF layer. We have demonstrated that our system produces state-of-the-art results on three publicly available datasets.

References

1. Azzopardi, G., Strisciuglio, N., Vento, M., Petkov, N.: Trainable COSFIRE filters for vessel delineation with application to retinal images. Med. Image Anal. **19**(1), 46–57 (2015)

2. Farnell, D., Hatfield, F., Knox, P., Reakes, M., Spencer, S., Parry, D., Harding, S.P.: Enhancement of blood vessels in digital fundus photographs via the application of multiscale line operators. J. Franklin Inst. **345**(7), 748–765 (2008)
3. Fraz, M., Remagnino, P., Hoppe, A., Uyyanonvara, B., Rudnicka, A., Owen, C., Barman, S.: An ensemble classification-based approach applied to retinal blood vessel segmentation. IEEE Trans. Biomed. Eng. **59**(9), 2538–2548 (2012)
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. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected CRFs with Gaussian edge potentials. In: Conference on Neural Information Processing Systems, pp. 109–117 (2011)
6. Lee, C., Xie, S., Gallagher, P., Zhang, Z., Tu, Z.: Deeply-supervised nets. In: International Conference on Artificial Intelligence and Statistics (2015)
7. Li, Q., Feng, B., Xie, L., Liang, P., Zhang, H., Wang, T.: A cross-modality learning approach for vessel segmentation in retinal images. IEEE Trans. Med. Imaging **35**(1), 109–118 (2016)
8. Marin, D., Aquino, A., Gegundez-Arias, M., Bravo, J.: 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)
9. Melinscak, M., Prentasic, P., Loncaric, S.: Retinal vessel segmentation using deep neural networks. In: International Conference on Computer Vision Theory and Applications, pp. 557–582 (2015)
10. Mendonca, A., Campilho, A.: Segmentation of retinal blood vessels by combining the detection of centerlines and morphological reconstruction. IEEE Trans. Med. Imaging **25**(9), 1200–1213 (2006)
11. Nguyen, U., Bhuiyan, A., Park, L., Ramamohanarao, K.: An effective retinal blood vessel segmentation method using multi-scale line detection. Pattern Recogn. **46**(3), 703–715 (2013)
12. 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, Heidelberg (2014). doi:10.1007/978-3-319-10404-1_79
13. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. **9**(1), 62–66 (1979)
14. Roychowdhury, S., Koozekanani, D., Parhi, K.: Iterative vessel segmentation of fundus images. IEEE Trans. Biomed. Eng. **62**(7), 1738–1749 (2015)
15. Staal, J., Abràmoff, M., Niemeijer, M., Viergever, M., van Ginneken, B.: Ridge-based vessel segmentation in color images of the retina. IEEE Trans. Med. Imaging **23**(4), 501–509 (2004)
16. Xie, S., Tu, Z.: Holistically-nested edge detection. In: International Conference on Computer Vision, pp. 1395–1403 (2015)
17. Zhao, Y., Wang, X., Wang, X., Shih, F.: Retinal vessels segmentation based on level set and region growing. Pattern Recogn. **47**(7), 2437–2446 (2014)
18. Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., Torr, P.: Conditional random fields as recurrent neural networks. In: International Conference on Computer Vision, pp. 1529–1537 (2015)