Medical Image Analysis 58 (2019) 101556

Contents lists available at ScienceDirect

Medical Image Analysis

journal homepage: www.elsevier.com/locate/media

Deep vessel segmentation by learning graphical connectivity

Seung Yeon Shin[a], Soochahn Lee[b,*], Il Dong Yun[c], Kyoung Mu Lee[a]

[a] Department of Electrical and Computer Engineering, Automation and Systems Research Institute, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul, 08826, South Korea
[b] School of Electrical Engineering, Kookmin University, Seoul, 02707, South Korea
[c] Division of Computer and Electronic Systems Engineering, Hankuk University of Foreign Studies, Yongin, 17035, South Korea

ARTICLE INFO

Article history:
Received 18 January 2019
Revised 2 September 2019
Accepted 5 September 2019
Available online 6 September 2019

Keywords:
Vessel segmentation
Convolutional neural network
Graph neural network
Vessel graph network

ABSTRACT

We propose a novel deep learning based system for vessel segmentation. Existing methods using CNNs have mostly relied on local appearances learned on the regular image grid, without consideration of the graphical structure of vessel shape. Effective use of the strong relationship that exists between vessel neighborhoods can help improve the vessel segmentation accuracy. To this end, we incorporate a graph neural network into a unified CNN architecture to jointly exploit both local appearances and global vessel structures. We extensively perform comparative evaluations on four retinal image datasets and a coronary artery X-ray angiography dataset, showing that the proposed method outperforms or is on par with current state-of-the-art methods in terms of the average precision and the area under the receiver operating characteristic curve. Statistical significance on the performance difference between the proposed method and each comparable method is suggested by conducting a paired *t*-test. In addition, ablation studies support the particular choices of algorithmic detail and hyperparameter values of the proposed method. The proposed architecture is widely applicable since it can be applied to expand any type of CNN-based vessel segmentation method to enhance the performance.

1. Introduction

Observation of blood vessels is crucial in the diagnosis and intervention of many diseases. Clinicians have mainly relied on manual inspections, which can be operator-dependent and time-consuming. Over the years, the demand for efficiency has led to the development of numerous methods for automatic vessel segmentation.

Most methods are based on image-processing (Frangi et al., 1998; Soares et al., 2006), optimization (Orlando and Blaschko, 2014; Shin et al., 2016; Zhao et al., 2015; Sun et al., 2014), learning (Becker et al., 2013; Sironi et al., 2015), or their combination. Most conventional image-processing-based methods (Frangi et al., 1998; Soares et al., 2006) either develop or adjust filters for emphasizing blood vessels in a given image. Many optimization methods aim to determine the best global graph structure based on applied prior knowledge. Diverse cost functions, defined on a Markov random field (Orlando and Blaschko, 2014; Shin et al., 2016), an active contour model (Zhao et al., 2015), an arbitrary graph (Sun et al., 2014), etc., are therein minimized. However, the prior knowledge often consists of only simple rules such as local smoothness, limiting the modeling capacity. More complex distributions can be modeled using learning-based methods such as boosting (Becker et al., 2013) or regression (Sironi et al., 2015). However, due to the limited model complexity, only local appearances are mostly learned.

Many recent methods being proposed are based on deep learning (Ganin and Lempitsky, 2015; Khalaf et al., 2016; Li et al., 2016; Liskowski and Krawiec, 2016; Oliveira et al., 2018; Zhang and Chung, 2018; Wu et al., 2018; Ventura et al., 2018; Fu et al., 2016b; 2016a; Maninis et al., 2016; Yan et al., 2018). In earlier methods, the output for the whole image was achieved by aggregating the predictions of local image patches processed independently (Ganin and Lempitsky, 2015; Khalaf et al., 2016; Li et al., 2016; Liskowski and Krawiec, 2016; Oliveira et al., 2018; Zhang and Chung, 2018; Wu et al., 2018). Ganin and Lempitsky (2015) studied a combination of nearest neighbor search with convolutional neural networks (CNNs) to learn complex transforms such as vessel segmentation. Khalaf et al. (2016) modified the vessel segmentation task as a three-class classification problem among large vessels, small vessels, and background areas to reduce the intra-class variation. The task was also re-formulated into another multi-class task by introducing additional labels on boundary areas of vessels in the work of Zhang and Chung (2018). In the work of Oliveira et al. (2018), the stationary wavelet transform by which multi-resolution

[*] Corresponding author.
E-mail address: sclee@kookmin.ac.kr (S. Lee).

https://doi.org/10.1016/j.media.2019.101556
1361-8415/© 2019 Elsevier B.V. All rights reserved.

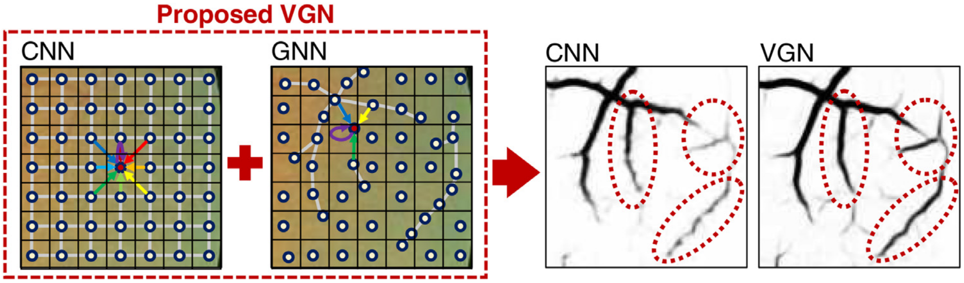

Fig. 1. Motivation of the proposed method. The articulated shape and hierarchical patterns of the vessel structures are not likely to be learned in existing CNN-based vessel segmentation methods. The proposed *vessel graph network* (VGN) utilizes a graph neural network (GNN), to propagate information along vessel structures, together with a CNN to address this issue. In the presented example results, VGN clearly enhances the detection of vessels with weak contrast by considering the vessel graph structure, compared to that of a CNN-only method (Maninis et al., 2016). The resulting vessel probability images are inverted for better visualization. All figures best viewed in color.

information is achieved, was incorporated to enrich the input representation for a network. In the work of Wu et al. (2018), the multiscale network followed network model was proposed to accurately segment capillaries of smaller diameter and lower contrast.

To improve efficiency and effectiveness, more recent methods (Fu et al., 2016b; 2016a; Maninis et al., 2016; Yan et al., 2018) take the whole image as input in the manner of the fully convolutional network method (Shelhamer et al., 2017). In the works of Fu et al. (2016b,a), a fully-connected conditional random field or its neural network implementation (Zheng et al., 2015) was utilized to take into account long-range interactions between pixels. Maninis et al. (2016) made use of the pretrained VGGNet (Simonyan and Zisserman, 2014) as a base network and specialized it for retinal vessel segmentation, which is called as deep retinal image understanding (DRIU). In the work of Yan et al. (2018), a vessel-segment-level loss is jointly utilized with a conventional pixelwise loss for accurate segmentation of thin vessels, where the thickness inconsistency of thin vessels is more penalized than thick vessels to alleviate the severe imbalance in the numbers of comprising pixels.

The CNN-based methods make use of multi-scale features by stacking outputs of intermediate layers for final prediction. However, those features represent only local appearances, with scope that does not exceed the receptive field. Most CNN-based methods for vessel segmentation are shallow with small number of layers. Empirically, just increasing the CNN depth to extend the receptive field does not improve accuracy (Maninis et al., 2016). While research on CNN structures, which increases the field-of-view (FoV) for more contextual information, is being conducted (Chen et al., 2018), it is in regards to the more general problem of semantic image segmentation. Furthermore, due to the vessel shape comprising a hierarchical structure of tubular regions, improvements for CNN structures that still rely on the square kernels and regular image grid will most likely have limited impact. We believe an alternative graphical representation is more likely to be effective, by helping to learn the articulated shape and hierarchical patterns of the vessel structures (Fig. 1).

Neural network models for irregular domains such as graphs are rapidly gaining more interest (Scarselli et al., 2009; Bruna et al., 2014; Henaff et al., 2015; Defferrard et al., 2016; Kipf and Welling, 2017; Niepert et al., 2016; Monti et al., 2017; Hamilton et al., 2017; Veličković et al., 2018). The core application of graph neural networks (GNN) is vertex classification on generic graphs, for example, the classification of articles using citation graphs. Compared to the fixed eight neighborhood system of the regular image grid, general graphs can have arbitrary edge connectivity. Within GNNs, the issue is then how to represent and learn the relational information between connected vertices as linear transforms or convolutions. In previous works, these operations have been defined either in the spectral domain of the connectivity matrix, or in the spatial domain. Our work was motivated with the belief that GNNs are suitable for our problem of learning the structural representation of vessels. Here, we are using the term GNN as an umbrella term for a diverse range of particular methods. We thus provide a brief review below.

The vertex connectivity is often represented as the adjacency matrix or Laplacian matrix. Several methods have been so developed that the connectivity is utilized in the spectral domain of the Laplacian matrix (Bruna et al., 2014; Henaff et al., 2015; Defferrard et al., 2016; Kipf and Welling, 2017). Beginning with the work of Bruna et al. (2014) which proposed a generalized CNN for a graph using the eigenvectors of its Laplacian, works by Henaff et al. (2015) and Defferrard et al. (2016), and Kipf and Welling (2017), have been proposed in order to improve the efficiency by spatially localizing filters using restricted spectral multipliers (Henaff et al., 2015), approximated filters by Chebyshev expansion of order k (Defferrard et al., 2016) and order one (Kipf and Welling, 2017). On the other hand, spatial approaches (Niepert et al., 2016; Monti et al., 2017; Hamilton et al., 2017) directly apply neural network filters on the graph. This is achieved by fixing the number of sampled vertices and neighborhoods in the work of Niepert et al. (2016), by a novel spatial-domain model, the mixture model network (MoNet) in the work of Monti et al. (2017), and by another method, called GraphSAGE (SAmple and aggreGatE), in which features from sampled neighbors are aggregated for each vertex, in the work of Hamilton et al. (2017). In the graph attention network (GAT) method of Veličković et al. (2018), an attention mechanism is incorporated in order to selectively attend over neighbors for each vertex. In the proposed method, we apply the GAT due to the effectiveness of its attention mechanism. For a more detailed summary on GNNs, we refer the reader to Bronstein et al. (2017).

We would like to note that a wide variety of works using GNNs have already been proposed for various computer vision problems ranging from semantic segmentation (Qi et al., 2017; Landrieu and Simonovsky, 2018), image recognition (Li et al., 2017), 3D shape analysis (Verma et al., 2018; Litany et al., 2018), video action recognition (Wang and Gupta, 2018), and person reidentification (Shen et al., 2018). Several methods for medical imaging problems have also been proposed (Selvan et al., 2018; Cucurull et al., 2018; Li et al., 2018). Selvan et al. (2018) cast airway tree extraction as a graph refinement task from an over-complete graph. Cucurull et al. (2018) investigate the usefulness of the GNNs, by which contextual information can be exploited, in the problem of cortical mesh segmentation. Li et al. (2018) model whole slide pathological images as graphs and tried to learn global topological features for survival prediction.

In this paper, we present a novel CNN architecture, the *vessel graph network* (VGN), that jointly exploits the global structure of vessel shape together with local appearances, as shown in Fig. 1.

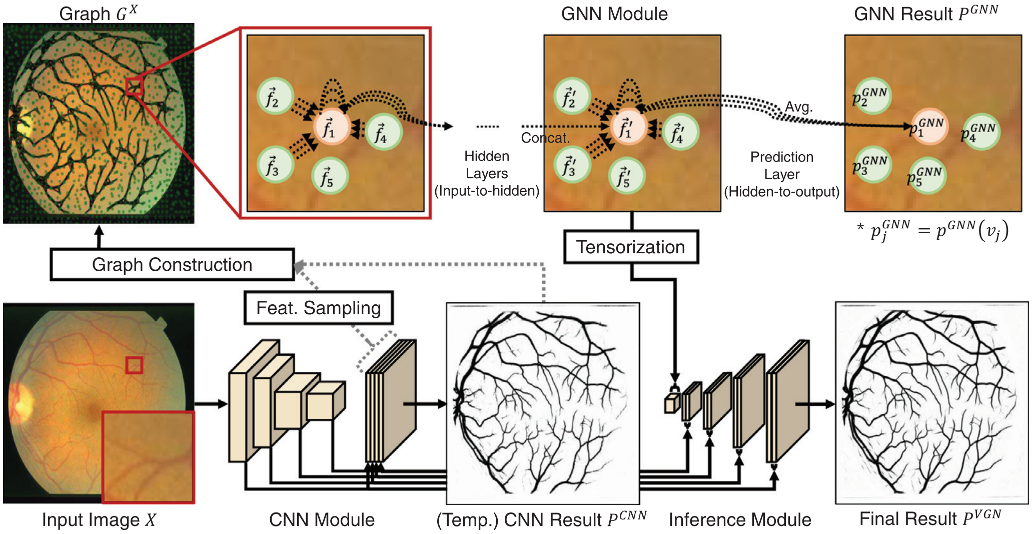

Fig. 2. Overall network architecture of VGN comprising the CNN, GNN, and inference modules. The CNN module generates pixelwise features corresponding to vessel probabilities, whereas the GNN module generates features which reflect the vascular connectivity. $\vec{f}_j$ and $\vec{f}'_j$ are the input and hidden representations of each vertex v_j, respectively, and $p^{GNN}(v_j)$ is the corresponding vessel probability prediction in the GNN module. The CNN and GNN modules are respectively devised to extract local/global evidences for vessels. The CNN and GNN features are then complementarily combined in the inference module to produce the final vessel probability map. The input graph for the GNN is generated in a separate graph construction module. The resulting vessel probability images are inverted for better visualization. Refer to Fig. 3 for the detailed network design.

The VGN comprises three components, i) a CNN module for generating pixelwise features and vessel probabilities, ii) a GNN module to extract features which reflect the vascular connectivity, and iii) an inference module to produce the final segmentation. The input graph for the GNN is generated in an additional graph construction module. The overall network architecture is described in Fig. 2.

The main contributions of our work are as follows. 1) Our work is the first, to the best of our knowledge, method to apply GNN to utilize the graphical structure of blood vessels. 2) We have developed a comprehensive framework in the VGN, including a scheme to define semi-regular graph nodes which enables a seamless combination of GNN features with CNN features in the overall architecture, making possible the joint learning of both local appearance and global structure. We note that the GNN acts as a component within the VGN that is trained and used much like other component modules in complex networks, for instance, the Region Proposal Network (RPN) in the Faster R-CNN method (Ren et al., 2017). Compared to the RPN which infers region proposals, the GNN in the VGN infers likely vessel nodes based on the features of neighboring nodes, that ultimately affects the final pixelwise segmentation. 3) The proposed VGN architecture is completely complementary to CNNs, and can be incorporated with any existing CNN structure to enhance the performance.

We demonstrate the effectiveness of the VGN through extensive comparative evaluations on four retinal image datasets and a coronary artery X-ray angiography dataset, which show the state-of-the-art performance of the VGN.

2. Methods

2.1. Overview of network architecture

The CNN module learns features, on the regular image grid of size $h \times w$, to infer pixelwise vessel probabilities $P^{CNN} = \{p^{CNN}(x_i)\}_{i=1}^{h\times w}$ of input image $X = \{x_i\}_{i=1}^{h\times w}$. The GNN module learns features for vertices on an irregular graph, which is constructed based on an initial inference of P^{CNN}. For each vertex v_j, hidden representations that enable accurate inference of the vessel probability should be learned within the GNN, accounting for the influence of neighboring vertices. For instance, when a vertex is surrounded by confident vessel vertices, it will become more likely to be labeled a vessel based on its GNN features. Due to their complementary characteristics, the CNN and GNN features are combined in the inference module to compute the final improved vessel probability map $P^{VGN} = \{p^{VGN}(x_i)\}_{i=1}^{h\times w}$.

As the CNN module, we adopt the network of DRIU (Maninis et al., 2016), based on the VGG-16 network (Simonyan and Zisserman, 2014). We note that any other CNN-based vessel segmentation method can be used. In DRIU, a vessel probability map is inferred from concatenated multi-scale features from the VGG-16. Before the concatenation, feature maps are resized to have identical spatial resolution. In our VGN, we adopt the pixelwise cross entropy loss $L_{CNN}(X)$ for this CNN module, defined as:

$$L_{CNN}(X) = -\frac{1}{|X|}\sum_i \Big(p^*(x_i)\log\, p^{CNN}(x_i) + (1-p^*(x_i))\log\big(1-p^{CNN}(x_i)\big)\Big), \qquad (1)$$

where $p^*(x_i)$ and $p^{CNN}(x_i)$ are the ground-truth (GT) label and the predicted vessel probability for each pixel x_i by the CNN, respectively. The weights for class-balancing are omitted for brevity. Please refer to Maninis et al. (2016) for more details.

The inference module is designed to combine the respective features from the CNN and GAT, each encoding a more local and more global representation, to infer the final vessel probability map. We adopt the structure of the U-Net (Ronneberger et al., 2015), but augment it with the GAT features. At each layer in the expansive portion, the GAT features are combined with the corresponding intermediate CNN features. The detailed network architecture is as shown in Fig. 3.

2.2. Graph neural network module

A graph must be constructed and given as input for both training and testing of the GNN module. We assume a CNN has been pretrained to generate P^{CNN}, on which vertex sampling and edge construction are performed.

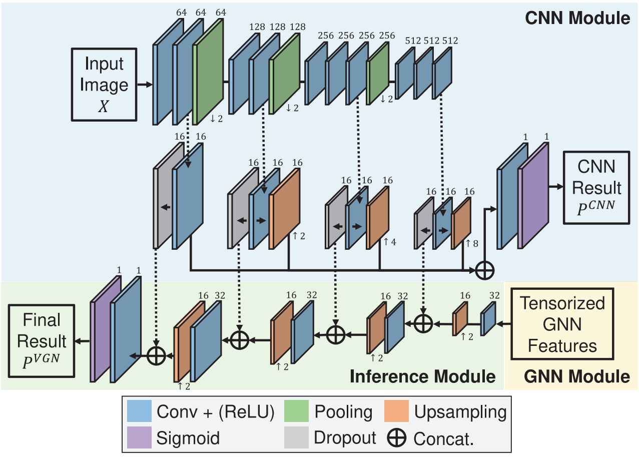

Fig. 3. Detailed network architecture for the proposed VGN. The CNN output layer and the first layer of the inference module, by which the tensorized GNN features are compressed, use 1 × 1 kernels while all the others use 3 × 3 kernels. The number of output channels is denoted on top of each layer. Up/downsampling factor is also provided at the bottom of each upsampling/pooling layer. The arrows denote the flow of the operations. Here, the GNN module, also presented in Fig. 2, is simplified for brevity. We note that this presented architecture is one particular version of the VGN. The numbers of layers in the CNN and inference modules can be varied respectively according to the datasets and the vertex sampling sparsity δ. Refer to Section 3.2 for more details.

2.2.1. Vertex sampling

The input image X of size $h \times w$ is partitioned into non-overlapping regions of size $2^\delta \times 2^\delta$. Within each region, the pixel $\hat{x}$ with maximum vessel probability $\hat{x} = \arg\max_{x_i} p^{CNN}(x_i)$ is sampled as a vertex. In regions where there is no unique maximum vessel probability value, a vertex is randomly sampled among the pixels with the maximal values. For regions where all pixels have all zero probabilities, the center pixel is chosen. This process results in a set of vertices $V = \{v_j\}_{j=1}^N$ for the image X. N is the number of vertices and equal to $\lceil h/2^\delta \rceil \times \lceil w/2^\delta \rceil$. $\lceil\,\rceil$ gives the smallest integer equal or larger than its input. We term this vertex sampling method, the semi-regular vertex sampling (SRVS) process. SRVS enables interaction between vessel-like regions, where the spatial intervals between nodes can be controlled by adjusting δ. In the extreme case where all pixels are nodes, $\delta = 0$. With larger δ values, coarser graphs are constructed with possibly longer-range interactions between vertices. Each vertex v_j is represented by its original image coordinates. An initial feature representation for each v_j is extracted from the intermediate feature map, before inference, computed from the CNN. This results in a set of vertex features, $\mathbf{f} = \{\vec{f}_j\}_{j=1}^N, \vec{f}_j \in \mathbb{R}^C$, where C is initially equal to C^{CNN}, the feature dimension of the CNN.

2.2.2. Edge construction

An undirected edge is assigned between a vertex pair if their geodesic distance is smaller than a threshold d. The geodesic distances are defined based on the CNN probabilities P^{CNN}, and calculated as the smallest accumulated sum of absolute p^{CNN} differences. This edge construction method is likely to connect vertices that are located along a high vessel probability path on P^{CNN}. Thus, the vessel structure is reflected in the constructed graph within which information propagation is enabled. We denote the constructed graph from the image X as $G^X(V, E)$, where E is a set of edges.

2.2.3. Graph attention network

We use the GAT (Veličković et al., 2018) as our GNN module due to its attention mechanism through which the GAT can selectively attend over neighbors for each vertex. We believe that the GAT can not only represent the vessel structure given by the initial graph effectively, but also filter out erroneous vessel neighbors implicitly, by the help of the attention mechanism. The GAT layer is illustrated in Fig. 4. Given the constructed graph $G^X(V, E)$ as input, A GAT layer produces a new set of vertex features, $\mathbf{f}' = \{\vec{f'}_j\}_{j=1}^N, \vec{f'}_j \in \mathbb{R}^{C'}$, where C' is a new feature dimension, from an input vertex feature set $\mathbf{f}$. Here, we describe this process in more detail.

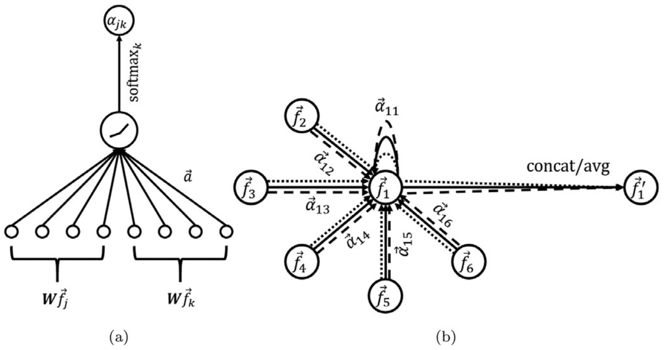

Fig. 4. Illustration of the graph attentional layer of (Veličković et al., 2018). This figure was previously presented in (Veličković et al., 2018) and is reprinted here for the description. (a) The attention mechanism parameterized by a trainable weight vector $\vec{a} \in \mathbb{R}^{2C'}$, and following LeakyReLU and softmax operations. α_{jk} indicates the normalized importance of vertex k for vertex j. (b) Multiple attention heads ($R = 3$ in this figure) applied on vertex 1, where the aggregated features from each head are concatenated or averaged to obtain the output feature $\vec{f}'_1$ of vertex 1.

Firstly, each vertex feature $\vec{f}_j$ is linearly transformed by a shared weight matrix $\mathbf{W} \in \mathbb{R}^{C' \times C}$. Then, attention coefficients e_{jk}, which indicate the importance of vertex k for vertex j, are computed via an attention function for every neighboring vertex pair as follows:

$$e_{jk} = \text{LeakyReLU}\left(\vec{a}^T\left[\mathbf{W}\vec{f}_j \| \mathbf{W}\vec{f}_k\right]\right), \tag{2}$$

where the attention function is a single-layer neural network parameterized by a weight vector $\vec{a} \in \mathbb{R}^{2C'}$. $\|$ is the concatenation operation and the LeakyReLU nonlinearity (negative input slope $\alpha = 0.2$) is applied according to Veličković et al. (2018). The coefficient e_{jk} is further normalized using the softmax function to make them comparable across nodes as follows:

$$\alpha_{jk} = \text{softmax}_k(e_{jk}) = \frac{\exp(e_{jk})}{\sum_{m \in N_j} \exp(e_{jm})}, \tag{3}$$

where N_j is a set of neighboring vertices of vertex j in the graph.

The normalized attention coefficients are then used to compute a weighted sum of the associated features as $\sum_{k \in N_j} \alpha_{jk} \mathbf{W}\vec{f}_k$, which will serve as the output features for each node. In Veličković et al. (2018), multiple independent instances of this output, termed *attention heads*, are employed to stabilize the learning process. Here, we use R number of the attention heads, and concatenate their outputs to obtain the output features of dimension $R \times C'$ as follows:

$$\vec{f}'_j = \big\Vert_{r=1}^{R} \text{ELU}\left(\sum_{k \in N_j} \alpha_{jk}^r \mathbf{W}^r \vec{f}_k\right), \tag{4}$$

where α_{jk}^r and $\mathbf{W}^r$ are normalized attention coefficients and linear transformation matrix for the r-th attention head, respectively. ELU represents the exponential linear unit (ELU) nonlinearity (Clevert et al., 2016). The first-order neighbors are considered in a single GAT layer. Multiple GAT layers can also be used for further interaction between neighbors.

The final prediction layer specifically performs averaging, instead of the concatenation, to produce the vessel probability for each vertex v_j as follows:

$$p^{GNN}(v_j) = \sigma\left(\frac{1}{R}\sum_{r=1}^{R}\sum_{k\in N_j}\alpha^r_{jk}\mathbf{W}^r\vec{f}_k\right), \tag{5}$$

where σ denotes the sigmoid function. $P^{GNN} = \{p^{GNN}(v_j)\}_{j=1}^{N}$ is the vector comprising the vessel probabilities for all vertices. Based on these output probabilities, the GNN module operates on the graph as a classifier for vertices into vessel or non-vessel.

To train the GAT, we use the mean of the vertex-wise cross entropy loss defined as:

$$L_{GNN}(G^X) = -\frac{1}{|V|}\sum_{v_j\in V}\Big(p^*(v_j)\log\, p^{GNN}(v_j) + (1-p^*(v_j))\log\big(1-p^{GNN}(v_j)\big)\Big), \tag{6}$$

$p^*(v_j)$ and $p^{GNN}(v_j)$ are the GT label and the predicted vessel probability for each vertex v_j by the GNN module, respectively. We note that with a slight abuse in notation, v_j denotes its original image coordinates in $p^*(v_j)$, while it denotes the vertex index within $G^X(V, E)$ for $p^{GNN}(v_j)$. The weights for class-balancing are omitted for brevity.

2.3. Inference module

As mentioned, we adopt the structure of the U-Net (Ronneberger et al., 2015) to account for the difference in feature resolution when combining the respective features from the CNN and GAT. More specifically, we first make a tensor of dimension $\lceil h/2^\delta\rceil \times \lceil w/2^\delta\rceil \times C'$ by tensorizing the output features of the penultimate layer of the GNN module, which consist of $\lceil h/2^\delta\rceil \times \lceil w/2^\delta\rceil$ number of C'-dimensional features as described in Section 2.2. The SRVS enables the GNN features to be placed back on a regular grid, even if they were computed assuming an irregular graph. The tensorized GNN features are then upsampled by a factor of two, and concatenated with the corresponding CNN features of the same spatial size. Then, a succeeding convolution layer is applied to locate the upsampled GNN features based on the CNN features. We note that, the graph vertex features can be interpreted as features made sparse by performing max-pooling based on vessel probability. The resulting features computed from the learned GNN module enforce global connectivity for the resulting low resolution vertex grid. Then, the CNN features at each corresponding scale are used to guide the upsampling of the GNN features back to full resolution.

Dropout is applied for the CNN features before concatenation to prevent our inference module from predicting solely from the CNN features. Based on empirical evaluations, we applied the standard per-element dropout. The upsampling, concatenation, and subsequent convolution are repeated until the spatial size reverts to that of the input. Thus, the combined features represent both global and local cues in the inference module. P^{VGN} is produced by applying the sigmoid function on the final one channel output features. We note that the use of different activation functions are based on corresponding reference methods. Particularly, the ReLU for the CNN and inference modules follows the structures of the DRIU and the U-Net, and the LeakyReLU and the ELU for the GAT module follow that of other methods based on the GAT.

We again use the pixelwise cross entropy loss function, similar to that for the CNN module, for this combined VGN inference module, which we denote as L_{INFER}.

We note that, without the GNN module, the VGN just reverts to the CNN, which in our current configuration is the DRIU, since the inference module to combine GNN and CNN features become unnecessary. If the sequential upsampling structure of the inference module is maintained, it basically just becomes a simplified version of U-Net (Ronneberger et al., 2015).

2.4. Network training

We adopt a sequential training scheme composed of an initial pretraining of the CNN followed by joint training, including fine-tuning of the CNN module, of the whole VGN. The P^{CNN} inferred from the pretrained CNN is used to construct the training graphs as described in Section 2.2. Since the computational burden would become prohibitive if a new graph is constructed at each training iteration, the graph construction is performed only at each K_{gc} training iterations. Pre-built graphs are used between the iterations. Compared to when pretraining the CNN, the VGN takes the corresponding graph G^X as well as the image X for jointly training the CNN and the GAT modules. Based on the capability of the GAT to implicitly filter erroneous vessel neighbors, the proposed network learns the graphical structure of the vessels while fine-tuning the CNN module end-to-end. The total loss function used for the VGN combines the respective losses of the CNN, GNN, and combined inference modules as:

$$L_{total}(X) = L_{CNN}(X) + L_{GNN}(G^X) + L_{INFER}(X). \tag{7}$$

When applying the VGN for a test image, CNN module feature generation and inference, graph construction, GNN feature generation, and final VGN inference are each performed sequentially to generate the final segmentation results.

3. Results

3.1. Datasets

The proposed method is evaluated on four retinal image datasets, namely the DRIVE (Staal et al., 2004), STARE (Hoover et al., 2000), CHASE_DB1 (Fraz et al., 2012), and HRF (Budai et al., 2013) datasets, and a coronary artery X-ray angiography (CA-XRA) dataset. We summarize in Table 1 the numbers of images used for training and test, the image resolution, and the availabilities of FoV masks and second manual annotations.

For each dataset, we follow previous works to determine the subsets for training and testing. For DRIVE and STARE sets, we followed the configuration in Maninis et al. (2016). Specifically, the first and last half of images are used for test/training and training/test in each dataset, respectively. The available second manual annotations are used for human performance measurement. For CHASE_DB1 and HRF, we followed Yan et al. (2018) for training/test set split. For CHASE_DB1, we use the first 20 images as the training set and the remaining 8 images as the test set. For HRF, we use the first 5 images each for three subsets of 15 healthy, 15 diabetic retinopathy, and 15 glaucomatous images, 15 images in total, as the training set. The test set comprises the remaining 30 images.

While the whole images are processed for DRIVE, STARE, and CHASE_DB1, the high resolution prohibits this for images of HRF. Thus, we process partly overlapping image patches of size 768 × 768, for both training and testing images. At test time, the vessel probability maps predicted from the patches are aggregated and probability values in overlapped regions are averaged. Since FoV masks are provided, evaluation is only conducted for pixels inside the FoV in all the retinal image datasets.

The CA-XRA set was acquired in a cooperating hospital and comprises 3137 image frames from 85 X-ray angiography (XRA) sequences. All sequences were acquired at 512 × 512 resolution, 8 bit depth, and 15 fps frame rate. We treated each frame as an independent image, without use of temporal information. Frames of the first 80, and the last 5 sequences were assigned as training and test sets comprising 2958 and 179 images, respectively.

Table 1
Retinal and coronary vessel datasets. For each dataset, the numbers of images used for training and test, the image resolution (height × width), and the availabilities of field-of-view (FoV) masks and second manual annotations are presented. We note that: 1) FoV masks are originally not available for STARE and CHASE_DB1, 2) we used ones generated by Oliveira et al. (2018) for STARE, and 3) we have generated them for CHASE_DB1 by a simple thresholding method as in Wu et al. (2018) for comparative evaluation.

Dataset	Retinal				Coronary
	DRIVE	STARE	CHASE_DB1	HRF	CA-XRA
# of images	40	20	28	45	3137
(train/test)	(20/20)	(10/10)	(20/8)	(15/30)	(2958/179)
Resolution	584 × 565	605 × 700	960 × 999	2336 × 3504	512 × 512
FoV mask	Y	Y	Y	Y	N
2nd GT	Y	Y	Y	N	N

Table 2
Summarization of the hyperparameters related to the graph construction and network training for each dataset. The provided value of the vertex sampling sparsity δ is the optimal one for each dataset based on the relevant empirical evaluations in Section 3.4. To construct a sufficient number of edges, the geodesic distance threshold d to be dependent of δ. The learning rate (LR) values for the joint training are presented as a pair of learning rate values for CNN fine-tuning and the remaining module training. Please refer to the text for more details.

Dataset	DRIVE	STARE	CHASE_DB1	HRF	CA-XRA
Vertex sampling sparsity δ	2	3	4	5	4
Geodesic distance threshold d	10	20	40	80	40
CNN pretraining LR	10^{-2}	10^{-2}	10^{-3}	10^{-3}	2×10^{-3}
Joint training LR	$10^{-2}/10^{-2}$	$0/10^{-2}$	$5 \times 10^{-3}/5 \times 10^{-3}$	$10^{-3}/10^{-3}$	$2 \times 10^{-3}/2 \times 10^{-3}$
Mini-batch size	1	1	1	1	5
Training iteration	50,000	50,000	50,000	50,000	100,000
Graph update period K_{gc}	10,000	10,000	10,000	10,000	20,000

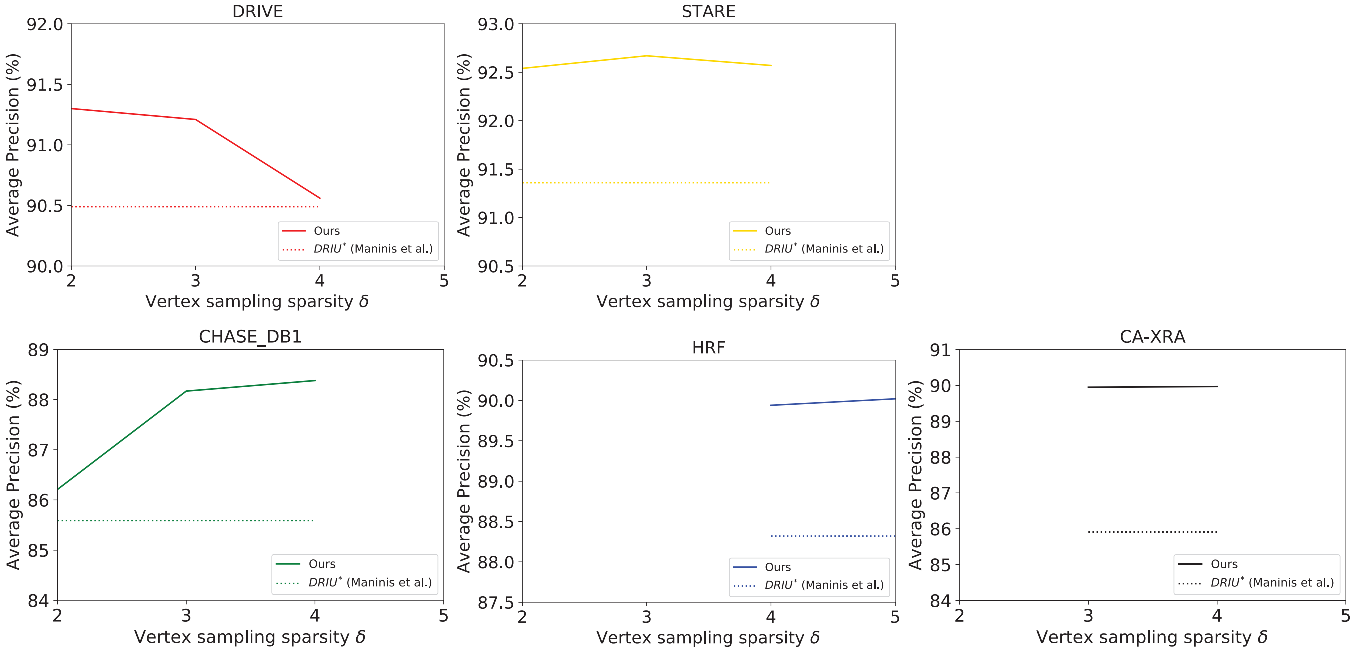

Fig. 5. Average precisions (AP) scored by the proposed VGN according to the vertex sampling sparsity δ for the DRIVE, STARE, CHASE_DB1, HRF and CA-XRA datasets. '*DRIU**' represents our own implementation, which was required as a component of the proposed VGN and is also the baseline for the proposed VGN. We note that setting $\delta = 5$ is allowed only for the HRF set due to the CNN architecture. Also, the tested values of δ for the HRF set were restricted to only larger ones, with an observation from the DRIVE, STARE, and CHASE_DB1 sets that higher resolution images tend to show better performance at larger values, which again shows the same tendency.

3.2. Evaluation details

As noted before, we use the DRIU (Maninis et al., 2016) as the CNN module in the proposed VGN. We used our own implementation of DRIU in the VGN since the authors have not made their training code publicly available. Where available, we provide a comparison between results from our implementation and those reported in Maninis et al. (2016) for reference. CNN architectures are identical to the original DRIU for the DRIVE, STARE, CHASE_DB1, and CA-XRA datasets (C^{CNN}=64=16 × 4), but slightly modified to include all five stages of VGG-16 (C^{CNN}=80=16 × 5) to handle the wider variance of vessel width for the HRF set.

For CNN pretraining, the settings mostly followed that of the original DRIU, but we slightly modify the loss function from the sum to the mean of pixelwise cross entropy, for better combination with other terms of the whole loss function of the VGN.

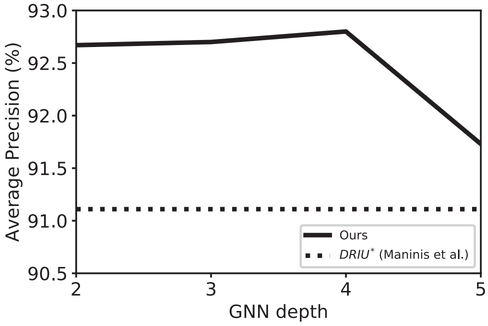

Fig. 6. APs scored by the proposed VGN according to the GNN depth for the STARE dataset. The score of *DRIU**, which is the baseline for the proposed VGN, is also presented as a dotted line for comparison.

For the graph construction of the GNN module, we set the vertex sampling sparsity δ and geodesic distance threshold d as presented in Table 2. The GNN feature dimension C' and the number of the attention heads R are fixed to 16 and 4 for all the datasets, resulting in the output feature of dimension $R \times C' = 4 \times 16 = 64$. The GNN depth was also empirically determined, based on the ablation study.

For training the VGN, we use an Adam optimizer (Kingma and Ba, 2014) and a weight decay of 5×10^{-4}. For each convolution layer, the weights are randomly initialized from a normal distribution and the biases are initialized as zeros while each upsampling layer is initialized as a bilinear filter. The initial learning rates of the pretrained CNN and the remaining modules are set as presented in Table 2. We found that not fine-tuning the pretrained CNN shows better results for STARE due to its small number of training images. We applied horizontal flipping, random brightness and contrast adjustment for data augmentation. Precomputed graphs are flipped accordingly. Mean pixel values are subtracted for each image channel as preprocessing. Dropout with a keep probability of 0.9 is applied for the CNN features before concatenation with upsampled GNN features as described in Section 2.3. The mini-batch size, number of performed training iterations, and graph update period K_{gc} for each dataset are also presented in Table 2.

For comparative evaluation, precision-recall (PR) curves are computed by computing multiple precision-recall pairs using multiple vessel probability thresholds. We also present the average precisions (AP) and the maximum F1 scores as summary measures of the PR curves. To account for methods where resulting vessel probability maps are not available and the PR curves cannot be computed, we also present accuracy, specificity, sensitivity values calculated using a threshold (0.5) for vessel probabilities and the area under the receiver operating characteristic (ROC) curve (AUC). While 'sensitivity' and 'recall' are identical measures, the terms will be used based on convention. We note that: 1) AP and AUC are complementary measures that show the overall performance behavior throughout multiple thresholds, 2) AP focuses more on

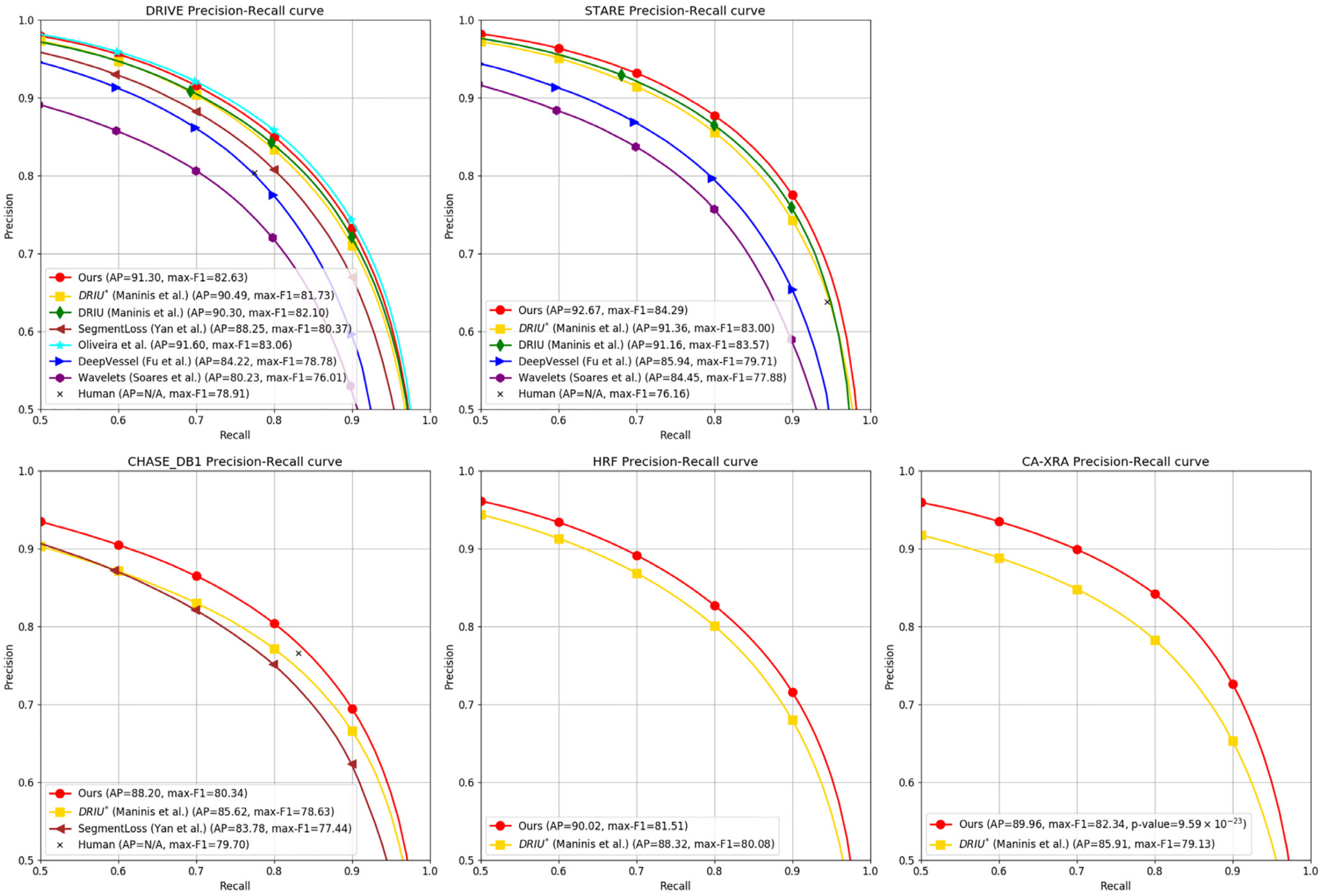

Fig. 7. Precision-recall curves of the proposed VGN and comparable methods on the DRIVE, STARE, CHASE_DB1, HRF and CA-XRA datasets. Average precision (AP) and max F1 scores, in percentages (%) are also given in the legends. 'Human' indicates the performance of the second annotator, which is calculated by comparing the second annotations with the first, if available. '*DRIU**' represents our own implementation, which is the baseline for the proposed VGN. Other results are ones that have been kindly provided by the original authors. We note that provided results that differ in the train/test dataset split have been excluded to ensure fair comparisons. The p-value showing statistical significance that VGN outperforms *DRIU** computed by a paired t-test is also presented for the CA-XRA dataset, which is the largest.

Table 3
Results of the proposed method with different edge construction methods for the STARE dataset. ‘Geodesic’ and ‘Euclidean’ indicate the performances of using the edge construction methods based on geodesic and Euclidean distances, repectively. ‘Geodesic’ is our default method described in Section 2.2. The Euclidean distance is used instead of the geodesic distance in ‘Euclidean’. ‘Full CXT’ represents using fully connected graphs. The score of *DRIU**, which is the baseline for the proposed VGN, is also presented for comparison. We note that the different edge connection methods are only applied at the test stage, using a model trained using graphs constructed with the default geodesic distance method.

Edge const. method	*DRIU** (Maninis et al., 2016)	Geodesic	Euclidean	Full CXT
Average precision (%)	91.36	**92.67**	92.56	90.34

Table 4
Results of different network architectures without the GNN module on the STARE dataset. The networks are: 1) *DRIU**, which is just our baseline CNN model, 2) simplified version of U-Net, which is a structure similar to the proposed network except the GNN module, and 3) DRIU + Deformable convolution, which is a DRIU model with deformable convolutions for variable sampling location of convolution. The numbers of model parameters for each network are also presented.

Network architecture	Average precision (%)	#params (million)
*DRIU** (Maninis et al., 2016)	91.36	7.86
U-Net (Ronneberger et al., 2015)	92.20	7.89
DRIU + Deformable convolution (Dai et al., 2017)	91.82	7.87
Ours	**92.67**	7.91

Table 5
Accuracy (Acc), specificity (Sp), sensitivity (Se), and the area under the receiver operating characteristic (ROC) curve (AUC) of the proposed VGN and comparable methods on the DRIVE, STARE, CHASE_DB1, HRF and CA-XRA datasets. P-values that are obtained by conducting a paired *t*-test between the AUC values of the proposed VGN and each comparable method are also presented for indicating statistical significance of improvements. ‘Human’ indicates the performance of the second annotator, which is calculated by comparing the second annotations with the first if available. ‘*DRIU**’ represents our own implementation, which is the baseline for the proposed VGN. The values of the methods marked with † are taken from respective papers. Meanwhile, the other values are calculated from resulting vessel probability maps provided by the original authors. The bold and underlined values represent the best and second best AUCs, respectively, on each dataset. The results of Oliveira et al. (2018) for the STARE and CHASE_DB1 datasets marked with § were achieved by performing five- and four-fold cross-validations respectively, in which more images were used for training.

Dataset	Method	Acc (%)	Sp (%)	Se (%)	AUC (±STD) (%)	p-value
DRIVE	Wavelets (Soares et al., 2006)	93.08	98.98	52.61	94.36 (±1.35)	4.43×10^{-12}
	DeepVessel (Fu et al., 2016a)	94.52	98.82	65.03	94.04 (±1.26)	2.15×10^{-14}
	Oliveira et al. (2018)†	95.76	98.04	80.39	**98.21** (N/A)	N/A
	SegmentLoss (Yan et al., 2018)	94.81	96.58	82.71	97.13 (±0.68)	2.68×10^{-11}
	MS-NFN (Wu et al., 2018)†	95.67	98.19	78.44	98.07 (N/A)	N/A
	Human	94.73	97.25	77.46	N/A	N/A
	DRIU (Maninis et al., 2016)	91.47	90.98	94.80	97.93 (±0.52)	5.32×10^{-3}
	*DRIU** (Maninis et al., 2016)	91.35	90.88	94.54	97.84 (±0.52)	1.12×10^{-6}
	Ours	92.71	92.55	93.82	98.02 (±0.53)	-
STARE	Wavelets (Soares et al., 2006)	95.34	98.47	68.95	96.99 (±1.39)	2.29×10^{-4}
	DeepVessel (Fu et al., 2016a)	95.58	98.91	67.53	95.83 (±2.26)	7.94×10^{-4}
	Oliveira et al. (2018)†§	96.94	98.58	83.15	**99.05** (N/A)	N/A
	Human	93.72	93.63	94.51	N/A	N/A
	DRIU (Maninis et al., 2016)	93.96	93.87	94.69	98.08 (±1.09)	6.01×10^{-3}
	*DRIU** (Maninis et al., 2016)	94.95	95.25	92.41	98.54 (±0.58)	1.02×10^{-3}
	Ours	93.78	93.52	95.98	98.77 (±0.50)	-
CHASE_DB1	Oliveira et al. (2018)†§	96.53	98.64	77.79	**98.55** (N/A)	N/A
	SegmentLoss (Yan et al., 2018)	95.13	96.03	86.17	97.42 (±0.59)	1.87×10^{-7}
	MS-NFN (Wu et al., 2018)†	96.37	98.47	75.38	98.25 (N/A)	N/A
	Human	96.16	97.46	83.14	N/A	N/A
	*DRIU** (Maninis et al., 2016)	93.39	93.32	94.11	98.10 (±0.54)	2.24×10^{-3}
	Ours	93.73	93.64	94.63	98.30 (±0.53)	-
HRF	SegmentLoss (Yan et al., 2018)†	94.37	95.92	78.81	N/A	N/A
	*DRIU** (Maninis et al., 2016)	94.59	94.88	91.70	98.22 (±0.70)	6.15×10^{-7}
	Ours	93.49	93.29	95.46	**98.38** (±0.61)	-
CA-XRA	*DRIU** (Maninis et al., 2016)	95.77	95.85	94.21	98.92 (±0.80)	1.46×10^{-3}
	Ours	95.17	95.07	97.00	**99.14** (±1.30)	-

the ability to correctly detect the vessels which comprise relatively fewer pixels than the background, and 3) AUC gives equal weights to both the vessel and background regions.

To support the statistical significance of the performance of the proposed method, we also present the p-values obtained using a paired *t*-test with each comparable method.

3.3. Model complexity

For the particular VGN version in Fig. 3, with hyperparameter values of the vertex sampling sparsity $\delta = 4$, GNN feature dimension $C' = 16$, the number of the attention heads $R = 4$, and GNN depth=3, the numbers of parameters of the CNN module and the

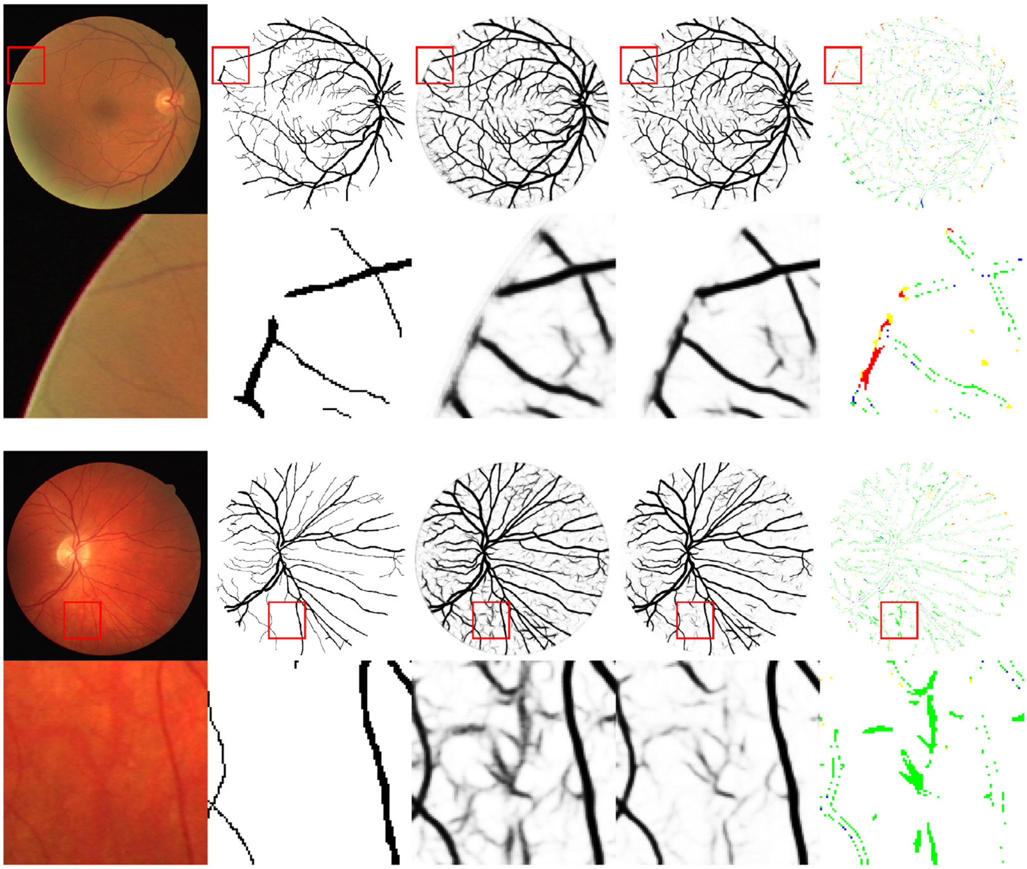

Fig. 8. Each couple of rows represents qualitative results of two representative samples from the DRIVE dataset. The images in the first row from left to right are, the original input image, GT, result of DRIU* representing our own implementation of Maninis et al. (2016), result of VGN, and the highlighted difference between binarized vessel masks of DRIU* and VGN. Refer to Table 7 for color notation of the highlighted difference. The second row in each case is the corresponding zoomed images of the first row. The GT and vessel probability images are inverted for better visualization.

remaining modules are approximately 7.86M and 0.07M, respectively. Thus, any existing CNN model can be extended to the VGN model with only small increase in the number of network parameters.

Training the above specified VGN model on the STARE set takes approximately 7 hours including an initial pretraining of the CNN and intermediate periodic graph updates.

3.4. Ablation study

3.4.1. The effect of vertex sampling density

As described in Section 2.2, the value of δ controls the sampling density of the graph vertices from the image for the GNN module. The larger the value, the coarser the graph becomes relative to the pixels. Details might be lost in coarser graphs but longer range interaction may be enabled.

We empirically determine the optimal δ value for each dataset, as shown in Fig. 5. In retinal datasets, datasets with higher resolution images show their best performance at larger δ. This is reasonable since smaller δ in lower resolution images physically corresponds to larger δ in higher resolution images. Although these results might imply that there should be an optimal δ value corresponding to the image resolution, we found that this cannot be theoretically determined due to the varying appearance and unpredictability of the segmentation output for different images. For applying our method, we believe that empirically determining the best δ value for the given data is the best approach. We note that different characteristics among the retinal datasets will cause discrepancies in terms of the detailed effects. For instance, HRF images are centered at the fovea and CHASE_DB1 images are centered at the optic disc while DRIVE and STARE images are mixed. The VGN outperforms the baseline with significant margins regardless of the values of δ in the CA-XRA set.

3.4.2. The effect of edge construction methods

Table 3 shows evaluations using different edge connection methods. ‘Euclidean’ denotes the method where Euclidean distance is used instead of the aforementioned geodesic distance. ‘Geodesic’ indicates our default method described in Section 2.2. ’Full CXT’ denotes a fully connected graph. We can see that the result of ‘Geodesic’ edge connection outperforms the ‘Euclidean,’ but even the ‘Euclidean’ result is better than that of the DRIU method. We believe that while more false positive edges might be included with the ‘Euclidean’ edge connection, they are likely neglected due to the attention mechanism of our chosen GNN module, the GAT. On the other hand, the result of ’Full CXT’ was significantly worse than all other methods. This is likely to have been caused by the increased difficulty in learning and estimating the appropriate vessel graph structure from a fully connected graph. We believe that these results show the necessity for an edge connection method that represents the actual vessel structure such as our ‘Geodesic’ connection method.

3.4.3. The effect of GNN depth

Fig. 6 shows the results for GNN modules with differing depths. The measured AP value increases only slightly with depth increased from 2 to 4, but sharply decreases at depth=5. We believe that these results show the adverse effects of excessively

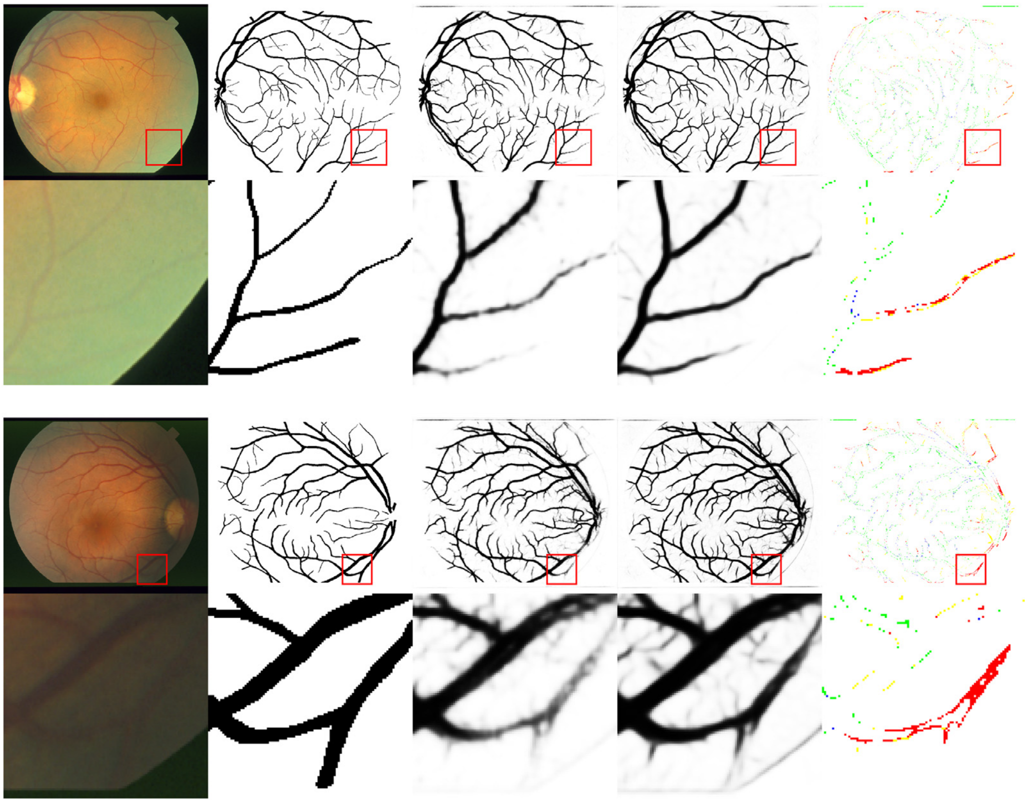

Fig. 9. Each couple of rows represents qualitative results of two representative samples from the STARE dataset. The images in the first row from left to right are, the original input image, GT, result of *DRIU** representing our own implementation of (Maninis et al., 2016), result of VGN, and the highlighted difference between binarized vessel masks of *DRIU** and VGN. Refer to Table 7 for color notation of the highlighted difference. The second row in each case is the corresponding zoomed images of the first row. The GT and vessel probability images are inverted for better visualization.

long-range interactions between vertices, since the *n*-th order neighbors might actually be irrelevant.

3.4.4. The effect of network architecture

To show the effectiveness of using the GNN, the results of different network architectures, which are without the GNN module, are compared in Table 4. A DRIU model with deformable convolutions (Dai et al., 2017), where the spatial sampling location of the convolution is not fixed and learned together with the kernel values, is also compared. We can see from the improvement compared to the U-Net that the GNN helps enhance the performance by incorporating the vessel structure information into the VGN architecture. Also, applying the deformable convolution to the DRIU model only shows a modest improvement in our experiment. We hypothesize that it is due to an ambiguity by the aperture problem, which hinders the offset learning. Instead of this more degree of freedom, we first construct a graph which reflects the vascular connectivity using the geodesic distance, and the attention mechanism of the GAT is utilized to focus on adequate vassel neighbors in the proposed method. We believe this process might lower the difficulty.

3.5. Comparison with previous methods

3.5.1. Quantitative evaluation

We provide comparative evaluations in terms of the PR curves of the proposed VGN and other current state-of-the-art methods, namely Fu et al. (2016a); Maninis et al. (2016); Yan et al. (2018); Oliveira et al. (2018) as well as the more conventional approach of Soares et al. (2006) in Fig. 7. Based on the PR curves, the AP and maximum F1 scores are also presented. We note that except for the *DRIU**, only the results of methods that we could obtain from the original authors and also match the evaluation settings, such as training/test dataset split, are presented.

The proposed method clearly shows better performance than the baseline '*DRIU**', which denotes our implementation of the DRIU method. The relative improvements compared to the baseline in terms of AP are 0.90%, 1.43%, 3.01%, 1.92%, and 4.71% for DRIVE, STARE, CHASE_DB1, HRF, and CA-XRA, respectively.

In Table 5, the segmentation accuracy, specificity, sensitivity, and AUC values are presented for a direct comparison to the reported quantitative results of the previous methods. We can see that: 1) the proposed VGN seems to be slightly more focused on sensitivity in regards to the tradeoff between specificity and sensitivity at the specific threshold of 0.5, and 2) nevertheless, the proposed VGN outperforms the compared methods of Fu et al. (2016a); Maninis et al. (2016); Yan et al. (2018) in terms of AUC. We note that given the increase in sensitivity (recall) compared to other methods, specificity and accuracy values are relatively low. For specificity, this is because it has a tradeoff relationship with sensitivity. For accuracy, this is because it over emphasizes increases in false positives relative to increases in true positives for data where foreground and background are unbalanced, since the direct sum of true positive and true negative pixels are measured.

When considering the results for the DRIVE, STARE, and CHASE_DB1 datasets in Fig. 7 together with those of Table 5, we can see that the proposed VGN outperforms all methods, except the methods of Wu et al. (2018); Oliveira et al. (2018). The VGN is on par with that of Wu et al. (2018), where histogram equalization and gamma adjusting algorithms are applied to improve input images. We note that currently no preprocessing except the mean subtraction is applied for the results of the proposed VGN, but may be easily added to the framework. We note that the results

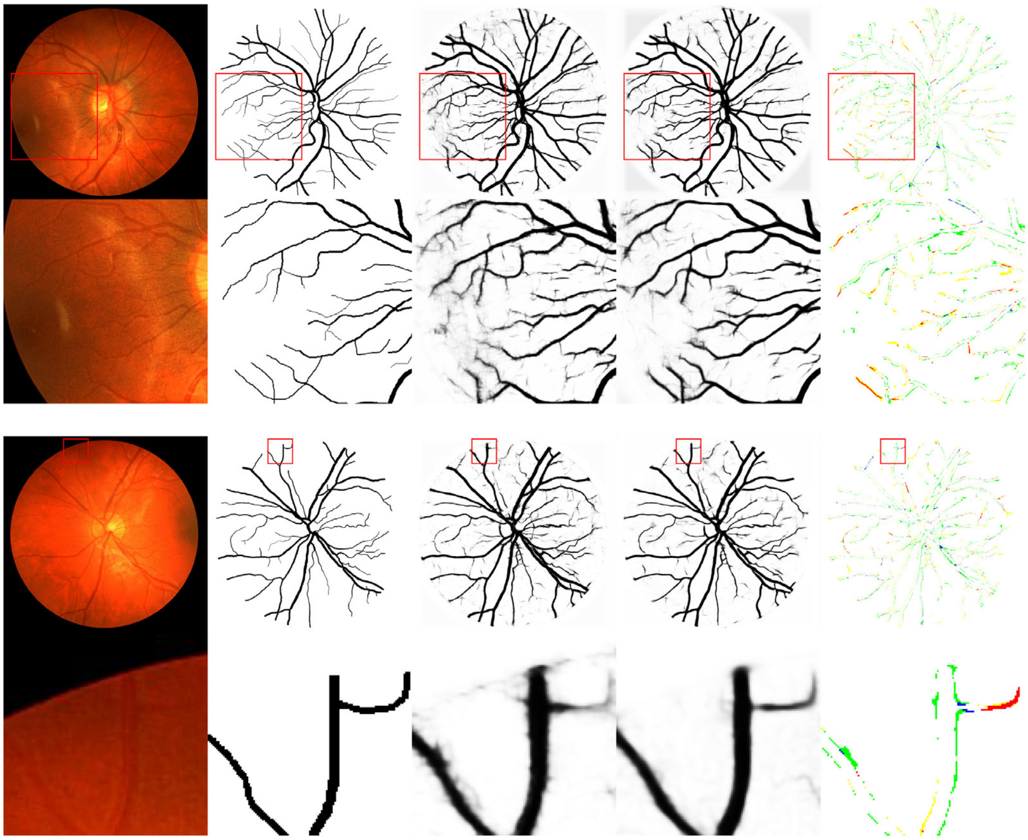

Fig. 10. Each couple of rows represents qualitative results of two representative samples from the CHASE_DB1 dataset. The images in the first row from left to right are, the original input image, GT, result of DRIU* representing our own implementation of Maninis et al. (2016), result of VGN, and the highlighted difference between binarized vessel masks of DRIU* and VGN. Refer to Table 7 for color notation of the highlighted difference. The second row in each case is the corresponding zoomed images of the first row. The GT and vessel probability images are inverted for better visualization.

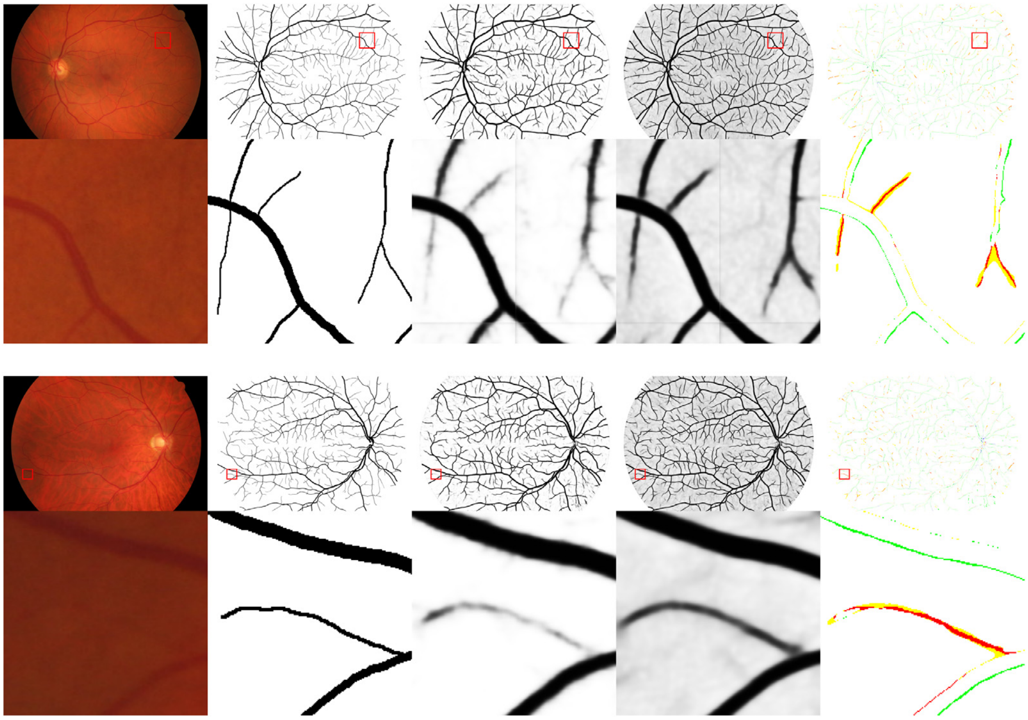

Fig. 11. Each couple of rows represents qualitative results of two representative samples from the HRF dataset. The images in the first row from left to right are, the original input image, GT, result of DRIU* representing our own implementation of Maninis et al. (2016), result of VGN, and the highlighted difference between binarized vessel masks of DRIU* and VGN. Refer to Table 7 for color notation of the highlighted difference. The second row in each case is the corresponding zoomed images of the first row. The GT and vessel probability images are inverted for better visualization.

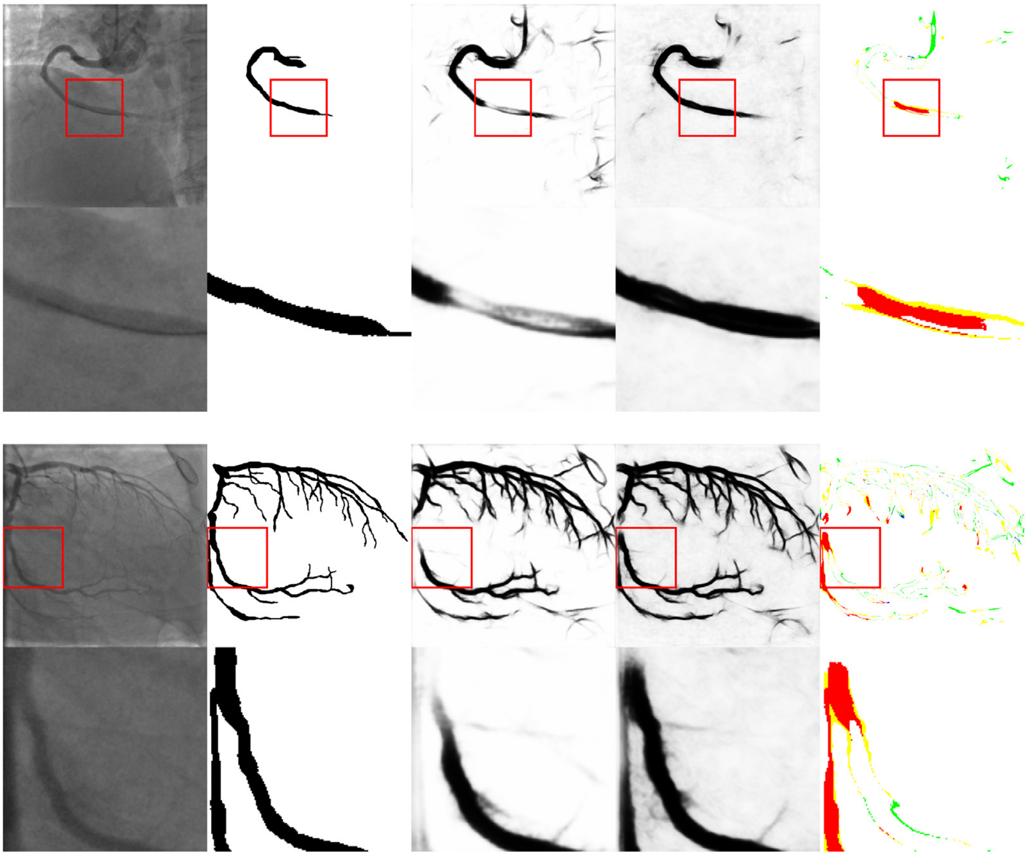

Fig. 12. Each couple of rows represents qualitative results of two representative samples from the CA-XRA dataset. The images in the first row from left to right are, the original input image, GT, result of *DRIU** representing our own implementation of Maninis et al. (2016), result of VGN, and the highlighted difference between binarized vessel masks of *DRIU** and VGN. Refer to Table 7 for color notation of the highlighted difference. The second row in each case is the corresponding zoomed images of the first row. The GT and vessel probability images are inverted for better visualization.

Table 6
APs (% points) scored by the proposed VGN for the three subsets of the HRF test set, each of which contains 10 images from patients that are healthy (H), have diabetic retinopathy (DR), or are glaucomatous (G), as described in Section 3.1. The score of *DRIU**, which is the baseline for the proposed VGN, is also presented for comparison. The improvement in percentage points of the results by VGN compared to that of DRIU are also presented. We can see that the improvement by the VGN is higher for images with pathologies.

Method	H	DR	G
*DRIU** (Maninis et al., 2016)	93.68	83.01	87.44
Ours	94.17	86.25	89.41
Difference	+0.49	+3.24	+1.97

Table 7
Possible types of difference for vessel segmentation mask comparison. T, F, P, N in TP, FP, TN, FN denote true, false, positive, and negative, respectively.

Type	Ground truth	*DRIU**	Ours	Color in Figs. 8, 9, 10, 11, 12
FN → TP	Vessel	Background	Vessel	Red
FP → TN	Background	Vessel	Background	Green
TP → FN	Vessel	Vessel	Background	Blue
TN → FP	Background	Background	Vessel	Yellow

of Oliveira et al. (2018) for the STARE and CHASE_DB1 datasets in Table 5 do not provide a fair comparison since they were achieved by five- and four-fold cross-validations, respectively, which are different from the experimental settings of our method as well as all the other methods. With identical evaluation settings for the DRIVE dataset, it still outperforms the proposed VGN in terms of both the PR curve as well as the AUC. The differences will most likely be due to the following key components of their method, both of which require increased computation. Namely, 1) the difference in input data configuration where the output responses of wavelet transforms are fed into the CNN, and 2) the application of patch-wise inference on multiple rotations, which makes the method effectively an ensemble inference method. We note that these components can easily be incorporated into the proposed VGN.

In Table 6, the AP values on the three subsets of the HRF test set, each of which comprises only healthy, diabetic retinopathy, and glaucomatous images, are presented. The improvements compared to *DRIU**, which is the baseline for the VGN, in terms of AP are 0.49, 3.24, and 1.97 percentage points, respectively. The VGN shows larger improvements in diabetic retinopathy and glaucomatous images than in healthy images. This implies that the VGN is especially adapt to variances in appearances that may be caused by various diseases.

3.5.2. Qualitative evaluation

Figs. 8, 9, 10, 11, and 12 show qualitative results for the DRIVE, STARE, CHASE_DB1, HRF, and CA-XRA datasets, respectively. In all Figures, the difference maps between binarized segmentation maps of the *DRIU** and the VGN is given in the rightmost column, where changes in the labels based on the ground truth labels are color encoded as summarized in Table 7. Compared to *DRIU**, VGN enhances the detection of weak vessels connected to stronger vessels by considering the vessel graph structure, and thus reduces false negatives, which can be observed as red pixels in the difference

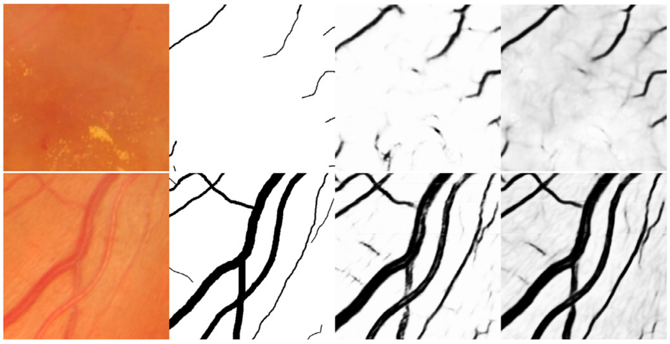

Fig. 13. Example results on challenging cases in the HRF dataset. The top and bottom rows are with lesions and central vessel reflex, respectively. Each row is consisted of an image patch, GT, result of *DRIU** representing our own implementation of Maninis et al. (2016), and result of VGN. The vessel probability images are inverted for better visualization.

maps. Furthermore, the VGN can significantly reduce false positives as well, as represented as the green pixels in the difference maps. Despite these successes, we also found that the VGN method can produce quite *fattened* vessel branches, similar to results of morphological dilation of the segmentation masks of *DRIU**. While these results can qualitatively seem to be clearer in terms of separating vessels from background, quantitatively they contribute to false positives.

Fig. 13 shows example results on challenging cases in the HRF dataset. The top row shows an example where the VGN is able to suppress the false positives that occur for *DRIU**, which can be caused by bright lesions. On the other hand, the bottom row shows an example where the false negative regions for *DRIU** in the middle of the vessels caused by the central vessel reflex are corrected in the results of the VGN.

4. Conclusion

We have proposed a novel deep-learning-based system for vessel segmentation, which we term the VGN, which elaborately incorporates a GNN into a unified CNN architecture to jointly learn both local appearances and global vessel structures. The proposed method, which is a novel network architecture augmenting a CNN with a GNN, is widely applicable since it can be applied to any existing CNN-based vessel segmentation method to further model the graphical vessel structure.

Extensive experiments on four retinal image datasets and a coronary artery X-ray angiography dataset prove the effectiveness of the proposed VGN. VGN not only enhances the detection of weak vessels connected to stronger vessels, but also interestingly suppresses false positives such as catheter tips and ribs around coronary vessels, by considering the graphical vessel structure.

Compared to previous methods, the proposed VGN suffers from increased false positives, mainly due to the thickened, or dilated, vessel branch segmentations. We believe that this may in turn be due to the diffusion of GNN vertex features in the U-Net inspired inference module. In future works, we hope to improve upon this problem. Moreover, we plan to extend the proposed method to 3D imaging modalities such as the computed tomography angiography or to use the temporal information of video data, e.g., fluoroscopic x-ray sequences.

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

We report grants from National Research Foundation of Korea (NRF), grants from National Research Foundation of Korea (NRF), grants from National Research Foundation of Korea (NRF), during the conduct of the study.

As well as grants from Korea Sanhak Foundation, grants from Samsung Electronics Co., grants from Microsoft Korea, Inc., grants from LG Electronics, grants from Koh Young Technology, Inc., grants from Samsung Fire & Marine Insurance Co., Ltd., grants from NAVER Corp., grants from SK Telecom Co., Ltd, grants from Samsung Medison Co., Ltd., grants from AnaPass, Inc., outside the submitted work.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) funded by the Korean Government (MSIT and MOE) under Grants NRF-2017R1A2B2011862, NRF-2019R1F1A1063656, and NRF-2019R1A2C1085113.

References

Becker, C., Rigamonti, R., Lepetit, V., Fua, P., 2013. Supervised feature learning for curvilinear structure segmentation. In: Mori, K., Sakuma, I., Sato, Y., Barillot, C., Navab, N. (Eds.), Medical Image Computing and Computer-Assisted Intervention (MICCAI). Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 526–533.

Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P., 2017. Geometric deep learning: Going beyond Euclidean data. IEEE Signal Process. Mag. 34 (4), 18–42. doi:10.1109/MSP.2017.2693418.

Bruna, J., Zaremba, W., Szlam, A., Lecun, Y., 2014. Spectral networks and locally connected networks on graphs. In: Proceedings of International Conference on Learning Representations (ICLR).

Budai, A., Bock, R., Maier, A., Hornegger, J., Michelson, G., 2013. Robust vessel segmentation in fundus images. Int. J. Biomed. Imaging 2013. doi:10.1155/2013/154860.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H., 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of European Conference on Computer Vision (ECCV).

Clevert, D.-A., Unterthiner, T., Hochreiter, S., 2016. Fast and accurate deep network learning by exponential linear units (elus). In: Proceedings of International Conference on Learning Representations (ICLR).

Cucurull, G., Wagstyl, K., Casanova, A., Veličković, P., Jakobsen, E., Drozdzal, M., Romero, A., Evans, A., Bengio, Y., 2018. Convolutional neural networks for mesh-based parcellation of the cerebral cortex. In: Proceeding of International Conference on Medical Imaging with Deep Learning (MIDL).

Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y., 2017. Deformable convolutional networks. In: Proceedings of IEEE International Conference on Computer Vision (ICCV), pp. 764–773. doi:10.1109/ICCV.2017.89.

Defferrard, M., Bresson, X., Vandergheynst, P., 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 29. Curran Associates, Inc., pp. 3844–3852. http://papers.nips.cc/paper/6081-convolutional-neural-networks-on-graphs-with-fast-localized-spectral-filtering.pdf.

Frangi, A.F., Niessen, W.J., Vincken, K.L., Viergever, M.A., 1998. Multiscale vessel enhancement filtering. In: Proceedings of Medical Image Computing and Computer Assisted Intervention (MICCAI), pp. 130–137.

Fraz, M.M., Remagnino, P., Hoppe, A., Uyyanonvara, B., Rudnicka, A.R., Owen, C.G., Barman, S.A., 2012. An ensemble classification-based approach applied to retinal blood vessel segmentation. IEEE Trans. Biomed. Eng. 59 (9), 2538–2548. doi:10.1109/TBME.2012.2205687.

Fu, H., Xu, Y., Lin, S., Kee Wong, D.W., Liu, J., 2016. DeepVessel: Retinal vessel segmentation via deep learning and conditional random field. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (Eds.), Medical Image Computing and Computer-Assisted Intervention (MICCAI). Springer International Publishing, Cham, pp. 132–139.

Fu, H., Xu, Y., Wong, D.W.K., Liu, J., 2016. Retinal vessel segmentation via deep learning network and fully-connected conditional random fields. In: Proceedings of IEEE International Symposium on Biomedical Imaging (ISBI), pp. 698–701. doi:10.1109/ISBI.2016.7493362.

Ganin, Y., Lempitsky, V., 2014. N^4-Fields: Neural network nearest neighbor fields for image transforms. In: Cremers, D., Reid, I., Saito, H., Yang, M.-H. (Eds.), Proceedings of Asian Conference on Computer Vision (ACCV). Springer International Publishing, Cham, pp. 536–551.

Hamilton, W., Ying, Z., Leskovec, J., 2017. Inductive representation learning on large graphs. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 30. Curran Associates, Inc., pp. 1024–1034. http://papers.nips.cc/paper/6703-inductive-representation-615learning-on-large-graphs.pdf.

Henaff, M., Bruna, J., LeCun, Y., 2015. Deep convolutional networks on graph-structured data. CoRR abs/1506.05163.

Hoover, A.D., Kouznetsova, V., Goldbaum, M., 2000. Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. IEEE Trans. Med. Imaging 19 (3), 203–210. doi:10.1109/42.845178.

Khalaf, A.F., Yassine, I.A., Fahmy, A.S., 2016. Convolutional neural networks for deep feature learning in retinal vessel segmentation. In: Proceedings of IEEE International Conference on Image Processing (ICIP), pp. 385–388. doi:10.1109/ICIP.2016.7532384.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. CoRR abs/1412.6980.

Kipf, T.N., Welling, M., 2017. Semi-supervised classification with graph convolutional networks. In: Proceedings of International Conference on Learning Representations (ICLR).

Landrieu, L., Simonovsky, M., 2018. Large-scale point cloud semantic segmentation with superpoint graphs. In: The Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Li, Q., Feng, B., Xie, L., Liang, P., Zhang, H., Wang, T., 2016. A cross-modality learning approach for vessel segmentation in retinal images. IEEE Trans. Med. Imaging 35 (1), 109–118. doi:10.1109/TMI.2015.2457891.

Li, R., Tapaswi, M., Liao, R., Jia, J., Urtasun, R., Fidler, S., 2017. Situation recognition with graph neural networks. In: Proceedings of IEEE International Conference on Computer Vision (ICCV).

Li, R., Yao, J., Zhu, X., Li, Y., Huang, J., 2018. Graph CNN for survival analysis on whole slide pathological images. In: Proceedings of Medical Image Computing and Computer-Assisted Intervention (MICCAI). Springer, pp. 174–182.

Liskowski, P., Krawiec, K., 2016. Segmenting retinal blood vessels with deep neural networks. IEEE Trans Med Imaging 35 (11), 2369–2380. doi:10.1109/TMI.2016.2546227.

Litany, O., Bronstein, A., Bronstein, M., Makadia, A., 2018. Deformable shape completion with graph convolutional autoencoders. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Maninis, K.-K., Pont-Tuset, J., Arbeláez, P., Van Gool, L., 2016. Deep retinal image understanding. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (Eds.), Medical Image Computing and Computer-Assisted Intervention (MICCAI). Springer International Publishing, Cham, pp. 140–148.

Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., Bronstein, M.M., 2017. Geometric deep learning on graphs and manifolds using mixture model CNNs. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Niepert, M., Ahmed, M., Kutzkov, K., 2016. Learning convolutional neural networks for graphs. In: Balcan, M.F., Weinberger, K.Q. (Eds.), Proceedings of The 33rd International Conference on Machine Learning (ICML). PMLR, New York, New York, USA, pp. 2014–2023. http://proceedings.mlr.press/v48/niepert16.html.

Oliveira, A., Pereira, S., Silva, C.A., 2018. Retinal vessel segmentation based on fully convolutional neural networks. Expert Syst. Appl. 112, 229–242. doi:10.1016/j.eswa.2018.06.034.

Orlando, J.I., Blaschko, M., 2014. Learning fully-connected CRFs for blood vessel segmentation in retinal images. In: Golland, P., Hata, N., Barillot, C., Hornegger, J., Howe, R. (Eds.), Medical Image Computing and Computer-Assisted Intervention (MICCAI). Springer International Publishing, Cham, pp. 634–641.

Qi, X., Liao, R., Jia, J., Fidler, S., Urtasun, R., 2017. 3D graph neural networks for RGBD semantic segmentation. In: Proceedings of IEEE International Conference on Computer Vision (ICCV).

Ren, S., He, K., Girshick, R., Sun, J., 2017. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. 39 (6), 1137–1149. doi:10.1109/TPAMI.2016.2577031.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional networks for biomedical image segmentation. In: Proceedings of Medical image computing and computer-assisted intervention (MICCAI). Springer, pp. 234–241.

Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G., 2009. The graph neural network model. IEEE Trans. Neural Networks 20 (1), 61–80. doi:10.1109/TNN.2008.2005605.

Selvan, R., Kipf, T.N., Welling, M., Pedersen, J.H., Petersen, J., de Bruijne, M., 2018. Extraction of airways using graph neural networks. CoRR abs/1804.04436.

Shelhamer, E., Long, J., Darrell, T., 2017. Fully convolutional networks for semantic segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 39 (4), 640–651. doi:10.1109/TPAMI.2016.2572683.

Shen, Y., Li, H., Yi, S., Chen, D., Wang, X., 2018. Person re-identification with deep similarity-guided graph neural network. In: Proceedings of European Conference on Computer Vision (ECCV).

Shin, S.Y., Lee, S., Noh, K.J., Yun, I.D., Lee, K.M., 2016. Extraction of coronary vessels in fluoroscopic X-ray sequences using vessel correspondence optimization. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (Eds.), Medical Image Computing and Computer-Assisted Intervention – (MICCAI). Springer International Publishing, Cham, pp. 308–316.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. CoRR abs/1409.1556.

Sironi, A., Lepetit, V., Fua, P., 2015. Projection onto the manifold of elongated structures for accurate extraction. In: Proceedings of IEEE International Conference on Computer Vision (ICCV), pp. 316–324. doi:10.1109/ICCV.2015.44.

Soares, J.V.B., Leandro, J.J.G., Cesar, R.M., Jelinek, H.F., Cree, M.J., 2006. Retinal vessel segmentation using the 2-D gabor wavelet and supervised classification. IEEE Trans. Med. Imaging 25 (9), 1214–1222. doi:10.1109/TMI.2006.879967.

Staal, J., Abramoff, M.D., Niemeijer, M., Viergever, M.A., van Ginneken, B., 2004. Ridge-based vessel segmentation in color images of the retina. IEEE Trans. Med. Imaging 23 (4), 501–509. doi:10.1109/TMI.2004.825627.

Sun, S.-Y., Wang, P., Sun, S., Chen, T., 2014. Model-guided extraction of coronary vessel structures in 2D X-ray angiograms. In: Golland, P., Hata, N., Barillot, C., Hornegger, J., Howe, R. (Eds.), Medical Image Computing and Computer-Assisted Intervention (MICCAI). Springer International Publishing, Cham, pp. 594–602.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Li, P., Bengio, Y., 2018. Graph attention networks. In: International Conference on Learning Representations (ICLR). https://openreview.net/forum?id=rJXMpikCZ.

Ventura, C., Pont-Tuset, J., Caelles, S., Maninis, K.-K., Van Gool, L., 2018. Iterative deep learning for road topology extraction. In: Proceedings of British Machine Vision Conference (BMVC).

Verma, N., Boyer, E., Verbeek, J., 2018. FeaStNet: Feature-steered graph convolutions for 3D shape analysis. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Wang, X., Gupta, A., 2018. Videos as space-time region graphs. In: Proceedings of European Conference on Computer Vision (ECCV).

Wu, Y., Xia, Y., Song, Y., Zhang, Y., Cai, W., 2018. Multiscale network followed network model for retinal vessel segmentation. In: Proceedings of Medical Image Computing and Computer-Assisted Intervention (MICCAI). Springer, pp. 119–126.

Yan, Z., Yang, X., Cheng, K., 2018. Joint segment-level and pixel-wise losses for deep learning based retinal vessel segmentation. IEEE Trans. Biomed. Eng. 65 (9), 1912–1923. doi:10.1109/TBME.2018.2828137.

Zhang, Y., Chung, A., 2018. Deep supervision with additional labels for retinal vessel segmentation task. In: Proceedings of Medical Image Computing and Computer-Assisted Intervention (MICCAI). Springer, pp. 83–91.

Zhao, Y., Liu, Y., Wu, X., Harding, S.P., Zheng, Y., 2015. Retinal vessel segmentation: An efficient graph cut approach with retinex and local phase. PloS One 10 (4), e0122332. doi:10.1371/journal.pone.0127486.

Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., Torr, P.H.S., 2015. Conditional random fields as recurrent neural networks. In: Proceedings of International Conference on Computer Vision (ICCV).