Computer Methods and Programs in Biomedicine 183 (2020) 105096

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

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

Retinal vascular junction detection and classification via deep neural networks

He Zhao, Yun Sun, Huiqi Li*

School of Information and Electronics, Beijing Institute of Technology, Beijing 100081, China

ARTICLE INFO

Article history:
Received 15 June 2019
Revised 9 September 2019
Accepted 25 September 2019

Keywords:
Retinal image
Vascular junction detection and classification
Deep learning.

ABSTRACT

Background and Objectives: The retinal fundus contains intricate vascular trees, some of which are mutually intersected and overlapped. The intersection and overlapping of retinal vessels represent vascular junctions (*i.e.* bifurcation and crossover) in 2D retinal images. These junctions are important for analyzing vascular diseases and tracking the morphology of vessels. In this paper, we propose a two-stage pipeline to detect and classify the junction points.

Methods: In the detection stage, a RCNN-based Junction Proposal Network is utilized to search the potential bifurcation and crossover locations directly on color retinal images, which is followed by a Junction Refinement Network to eliminate the false detections. In the classification stage, the detected junction points are identified as crossover or bifurcation using the proposed Junction Classification Network that shares the same model structure with the refinement network.

Results: Our approach achieves 70% and 60% F1-score on DRIVE and IOSTAR dataset respectively which outperform the state-of-the-art methods by 4.5% and 1.7%, with a high and balanced precision and recall values.

Conclusions: This paper proposes a new junction detection and classification method which performs directly on color retinal images without any vessel segmentation nor skeleton preprocessing. The superior performance demonstrates that the effectiveness of our approach.

1. Introduction

Retinal fundus image is a unique type of images to observe circulation system non-invasively. Vascular structure can be observed directly from the image and it can have different morphology due to aging or diseases. Retinal vascular bifurcations and crossovers are important feature points in retinal fundus images as they can provide useful information to clinical diagnosis, such as image analysis [1,2] and biological statistics [3,4]. For example, width changing of vessels at crossover can indicate arteriovenous nicking, which is one of the markers of hypertension retinopathy. Bifurcations are necessary in the calculation of arteriolar-to-venular ratio (AVR) which is an important measurement in clinical study of cardiovascular diseases [5]. Feature points can also be used to extract vessel topology [6] or for image registration [7]. Detection of retinal junctions is an important stage for vessel tracking [8] and vascular reconstruction. It provides essential information to separate every single vessel tree, which is a fundamental step for most disease analysis in retinal images. Human assessment of junction points is a time-consuming and subjective task where the variance of intra-grader and inter-grader also needs to be considered. Thus, automatic detection and classification algorithm is required in clinical study.

However, detecting junction points is challenging due to the fact that the condition of retinal image is very complex. The feature points are close to each other in some cases, or the contrast between the vessels and the background is too low. There has been some research work on detecting junctions of retinal images, and most of them are starting from vessel segmentation map. By observing the process of this kind of methods, it can be noted that they are all heavily dependent on vessel segmentation and vascular geometry. The segmentation-dependent method can produce good detection on the well-segmented images (*e.g.* vessel ground-truth). However, the performance will decrease a lot when segmentation or skeleton generalization is not correct. Typical errors include vessel missing, disconnected or broken vessel segments.

>In this paper, an approach to detect and classify vascular junctions is proposed. Our contribution can be summarized as: first, our detection works on the original image rather than vessel segmentation, which can avoid the mistakes caused by vessel segmentation or skeleton; second, a two-step detection workflow is

* Corresponding author.
E-mail address: huiqili@bit.edu.cn (H. Li).

https://doi.org/10.1016/j.cmpb.2019.105096
0169-2607/

proposed to determine the position of junctions, in which our refinement network is specially designed for retinal vessels so that it can achieve a better performance than the single Junction Proposal Network. Our approach has been tested on two public datasets and the results outperform other state-of-the-art methods.

2. Related work

The existing methods of junction detection can be categorized into two classes: skeleton-based method and model-based method. The skeleton-based methods [9–13] usually count the number of vessel segments within a certain area. They are highly dependent on the segmentation and skeleton results. Inspired by simple cross-point number detector, the work of [9] modify the configuration and combine two scales detectors to make their method more accuracy and robust. A two-step method for feature point detection and classification is proposed in [10], where filters and morphological operations are utilized for detection and the features based on local and topological analysis are employed for identification. A detection filter is designed by Baboiu *et al.* [14] based on scale-space analysis and eigenvalue analysis of bifurcations. Focusing on the misclassification of crossovers, Nguyen *et al.* [15] propose a method that utilizes both local information and vascular geometrical features to distinguish between bifurcation and crossover. Two nearby bifurcations are grouped as a potential crossover and re-examined by the angles between vessels near the junctions. In [16], a branching point detector is proposed similarly as the skeleton-based method, which is applied on the enhanced image instead of vessel segmentation. To avoid the error caused by skeletonization, Morales *et al.* [17] determine the skeleton using stochastic watershed transformation. The junction points are detected by template matching and then classified into crossover and bifurcation by a close loop checking. Researchers are working on novel segmentation methods [18,19] to obtain a better segmentation results, but it's still a challenging task to detect feature points with skeleton-based method.

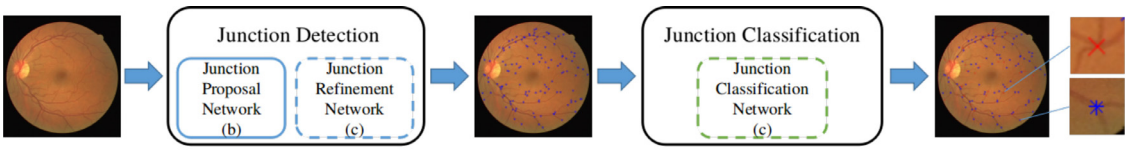
For the model-based methods, a designed model is employed to detect the maximum response. COSFIRE (Combination Of Shifted FIlter REsponses) are proposed for feature point detection in [20] and the response is computed as the combination of shifted Gabor filters. Several prototypes are selective to construct the COSFIRE filters which are effective to detect keypoints similar to prototypes. Its performance really depends on the selected prototypes as well as vessel segmentation, and there is no able to distinguishing between crossover and bifurcation. A probabilistic model has been proposed in [21]. The vascular trees are divided into independent segments and junctions are split into terminals, bridges and bifurcations. Each junction will gain a probability from the Bayesian model for different configurations, and maximum a posteriori is used to assign the most likely configuration. In [22], a new transformation by directional anisotropic wavelets is introduced to convert images into the joint space of positions and orientations in which the candidate junctions are selected based on the geometrical properties and followed by a refinement. Finally, a fusion step of resulting junctions and a skeleton-based method is utilized to get better result. The vessel keypoint detector (VKD) is proposed in [23], which is derived from log-polar map of segmentation patches. The candidate junction points are extracted by VKD and classified using combined features of Random Forest classifier. A hierarchical probabilistic model is proposed to detect bifurcation points in [24] where bifurcation and normal point are classified by the local intensity cross sections modeled by Gaussian function. Exclusion region and position refinement (ERPR) is proposed to improve the accuracy of feature point detection in [25]. This method is based on centerline detection, trying to refine the position of feature point by tracking back from the junction. In [26], junctions are detected by combining Hessian information and correlation matrix, and the number of branches and branch orientations are also provided by the method.

Deep learning methods have also been applied in junction detection in the recent years. Pratt *et al.* [27] propose a deep learning classification baseline for bifurcation and crossover detection. They extract patches for all the vessel centerline points as the junction candidates. For each patch, they use two CNNs to classify crossover, bifurcation or background. A multi-task framework [28] is used in another work to learn the vessel centerline probability followed by eigen-analysis on Hessian decomposition to get potential junctions and a multi-scale intersection search is used for refinement.

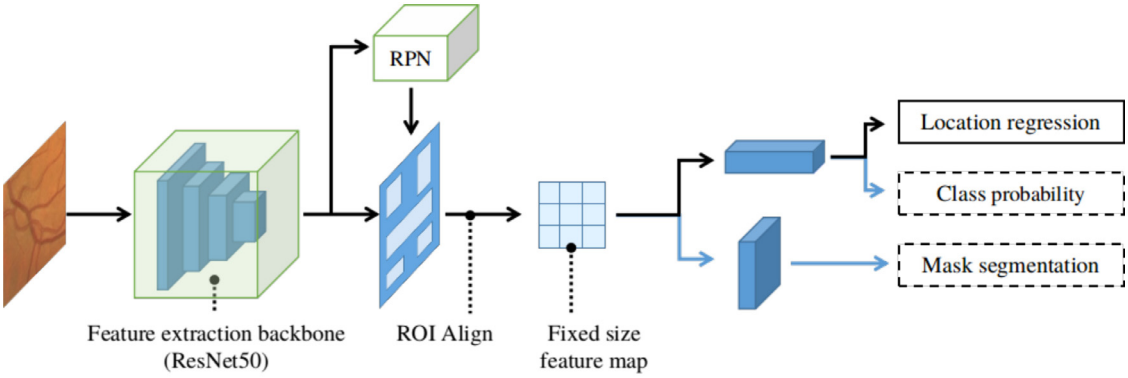
Object detection that locates the object position with a bounding box has attracted interest of many researchers recently. There are two categories of these methods: two-stage detection and one-stage detection. R-CNN [29] is the basis of a series of two-stage detection method. It obtains region proposals by selective search and uses a CNN model to classify and locate the object position. Instead of feeding the region proposals into the CNN every time, Fast-RCNN [30] is proposed to detect object on the feature maps. This improvement makes the feature extraction only performed once and thus the speed is significantly increased. Faster-RCNN [31] utilizes a Region Proposal Network (RPN) to get regions instead of selective search to further improve the speed, while the Mask-RCNN [32] combines the segmentation and detection task together to get better performance. On the other hand, one-stage methods directly give outputs from the input image instead of predicting object position based on region proposals. Classical one-stage methods include YOLO [33] and SSD [34]. YOLO (You only look once) is a neural network which reframes the object detection as a regression problem and predicts the bounding box positions and probabilities directly from images with one evaluation. SSD (Single shot multibox detector) further improves the performance and speed of YOLO with anchors [31] and multi-scale feature maps. The speed of one-stage framework is extremely high, although it is weak at precise locations especially for small objects. Motivated by the success application of these object detection methods, we propose a RCNN-based Junction Proposal Network to give the initial junction locations.

3. Our approach

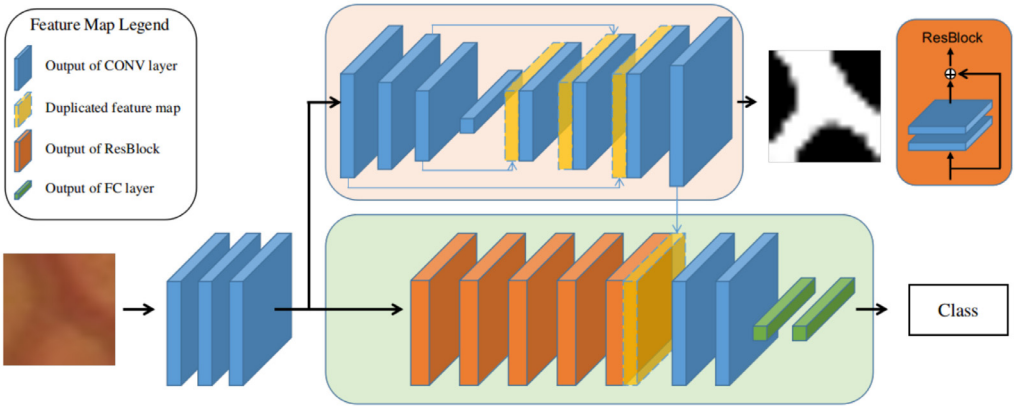
In this work, we aim at detecting and classifying retinal junction points directly from the raw color images. Denote the color retinal image as $\boldsymbol{I} \in \mathbb{R}^{W\times H\times 3}$, the crossover positions as $\boldsymbol{C_{i,j}}$ and bifurcation positions as $\boldsymbol{B_{i,j}}$. Our detection task is to find junction locations $\boldsymbol{J_{i,j}}$ (*i.e.* $\boldsymbol{B_{i,j}}$ and $\boldsymbol{C_{i,j}}$) on the retinal image $\boldsymbol{I}$, while the goal of classification is distinguishing $\boldsymbol{C_{i,j}}$ from $\boldsymbol{B_{i,j}}$. Fig. 1 (a) illustrates our approach pipeline. The proposed approach can be divided into two stages: the detection stage and classification stage. There are three individual networks engaged, which are named Junction Proposal Network (JPN), Junction Refinement Network (JRN) and Junction Classification Network (JCN). A two-step scheme that consists of JPN and JRN is designed for detection stage. It extracts the potential junctions $\boldsymbol{J_{i,j}}$ from the original image $\boldsymbol{I}$ and eliminates the false detection (shown by blue boxes in Fig. 1(a)). As to classification stage, JCN is proposed to identify crossover $\boldsymbol{C_{i,j}}$ and bifurcation $\boldsymbol{B_{i,j}}$ in $\boldsymbol{J_{i,j}}$ (shown by green boxes in Fig. 1(a)). Essentially, JRN and JCN both are classification models but designed for different tasks, so they share the same model structure in this work (shown by dashed boxes in Fig. 1(a)). JPN is a modified RCNN-based object detection model while JRN and JCN share a multi-task classification model. In what follows, we will give the detailed introduction to the detection model and classification model.



(a) Pipeline of our approach



(b) Our RCNN-based Junction Proposal Network



(c) Our multi-task classification model

Fig. 1. Our approach flowchart and model structures. (a) our two-stage detection and classification pipeline. (b) RCNN-based Junction Proposal Network structure for potential junction extraction. Only the location regression is used for further processing. (c) multi-task classification model structure that is used both in the detection and classification stage (dashed boxes in (a)).

3.1. RCNN-based junction proposal network

Our Junction Proposal Network is based on the Mask-RCNN model [32], which takes 128×128 image patches as input and outputs the bounding boxes of potential junction locations. Fig. 1 (b) displays the structure of our Junction Proposal Network. It consists of three parts: the backbone for feature extraction; the region proposal network (RPN) for region of interest (ROI) selection; the head module for bounding-box regression, classification and mask generation. In our approach, only the bounding-box regression head branch will be utilized for further processing. All the detected bounding box centers are selected as our initial junction proposals without classification score thresholding. In practice, we use ResNet50 [35] as our feature extraction network backbone structure. Instead of extracting features from ResNet50 directly, a pyramid structure [36] is employed to take multiple scales into consideration. The generated ROIs from RPN are processed to fixed size feature map by ROIAlign. Finally, the head module takes the fixed size feature map as input and outputs the accurate junction locations. Two important components of this network, RPN and ROIAlign, will be introduced in details.

3.1.1. Region proposal network

This component takes large image patch as input and outputs several rectangular junction proposal locations with significant scores. It can be modeled by a fully convolutional network. The network conducts a convolutional operation on the feature map with kernel size of 3 and stride 1. The resulting features are followed by two sibling 1×1 convolutional layers to produce two sets of outputs that are region regression and region classification. Moreover, multiple proposals are predicted simultaneously for each pixel on the feature map with different aspect ratios and scales, which are also called anchors. The final ROIs are generated from these anchors with top-N high confidence.

3.1.2. ROIAlign

ROIAlign is an improved version of ROI pooling and both of them are used for extracting a small feature map from ROI. ROI pooling causes misalignment when quantizing floating-number ROI to a discrete size of feature map. It is addressed by ROIAlign with bilinear interpolation to compute exact value of each sampling point. This operation removes the harsh quantization and properly aligns the ROI with the feature map, which can greatly improve

the accuracy of bounding box locations. Interested readers can refer to [32] for detailed information.

Up to now, we can get the junction proposals within one image patch. By combining all the image patches of one image, we can get the initial positions of junctions. These junction candidates may contain numerous points not belonging to any kind of feature points. So we propose a Junction Refinement Network (*i.e.* classification model) to eliminate the false positive detection.

3.2. Multi-task classification model for junction refinement and junction classification

As aforementioned, our classification model serves for junction refinement in detection stage as well as crossover and bifurcation identification in classification stage. In this section, we give an introduction of our classification model with dual usages. This model plays a role of binary classification that treats junction as foreground in detection stage and crossover as foreground in classification stage.

Our classification model is a multi-task convolutional neural network which combines classification and segmentation tasks together. It takes 16×16 patches as input and produces vessel segmentation and foreground classification simultaneously. Our classification model is shown in Fig. 1 (c). The model contains one backbone for rough feature extraction and two branches for different tasks of classification and segmentation respectively. Green box shows the structure of the classification branch while the pink one is the assistant branch (aka. segmentation branch). Blue cube indicates the feature maps obtained from convolutional layers, and the orange cube refers to the feature maps generated by the ResBlocks. Dashed yellow cube is the feature map duplicated from previous convolutional layer output. Two benefits can be obtained by this design: first, the model can produce more information around the junction area, not only the class of the patch but also the vascular morphology; second, the segmentation information from the assistant branch can help training the main branch to get more accurate result.

The backbone is three stacked convolutional layers with kernel size [5, 5] and stride 1 to get a preliminary feature extraction. After the convolution operation, 32 feature maps are fed into each branch for further calculation. The assistant branch consists of 4 convolutional layers and 5 deconvolutional layers with full skip connections. It takes the feature maps as input and outputs segmentation of corresponding patch. The feature maps are first downsampled into a vector by convolutional layers with kernel size [3, 3] and stride 2, followed by sequential deconvolutional layers with kernel [3, 3] and stride 2 to upsample into original size. Skip connections are utilized between each corresponding feature maps to gain better performance [37]. The filter numbers are set to 32 and 64 for the first half and the second half convolutional layers respectively, while inversely for the deconvolutional layers.

The main branch utilizes a residual module [35] as a basic component to give class prediction of an input patch. Two convolutional layers with kernel [3, 3] and stride 1 together with a skip connection form the residual block, which is displayed in the top-right of Fig. 1 (c). The feature maps from backbone pass 5 residual blocks and are further fused with the features generated from assistant branch. The concatenated feature maps are convoluted by two convolutional layers with kernel size [3, 3, 64] and stride 1, subsequently followed by two fully connected layers with 256 neurons to get the final prediction result.

The vascular structure is engaged for the assistant branch, but different from other segmentation dependent method, it's only utilized in the training stage for improving the detection accuracy. In the testing phase, the only input of our model is the raw retinal image. The segmentation accuracy doesn't influence our result. So our approach can avoid the mistakes caused by segmentation or skeletonization.

This multi-task convolutional neural network is used in both the refinement of detection stage (JRN) and classification stage (JCN) with different training condition. In detection stage, the JRN eliminates false detections raised by the JPN, so it is trained with junction patches as positive samples and non-junctions as negative samples. In classification stage, positive samples change to crossover patches while negative samples are bifurcation patches to classify two types of feature points using the JCN.

3.3. Loss function

Junction Proposal Network

The loss function of RCNN-based Junction Proposal Network can be summarized as bounding-box regression loss, bounding-box classification loss and segmentation loss. In our case, we consider crossovers and bifurcations together as foreground class of junctions and others as background. There are 3 parallel outputs of Junction Proposal Network. The first layer outputs the bounding-box location $l = (l_x, l_y, l_w, l_h)$, which indicates the bounding-box center pixel coordinates as well as the width and height of the box. The second one gives a probability value to predict each ROI category, that is, junction or not. The last output is the mask of junction points in the bounding-box. The summarized loss can be written as:

$$L^d = L^d_{loc} + L^d_{label} + L^d_{mask} \tag{1}$$

$$smooth_{L1}(x) = \begin{cases} 0.5x^2 & |x| < 1 \\ |x| - 0.5 & otherwise, \end{cases} \tag{2}$$

where L^d_{loc} indicates bounding-box location regression loss that is calculated by smooth L1 loss defined in Eq. (2), L^d_{label} is the cross-entropy loss between bounding-box prediction and ground-truth, and L^d_{mask} refers to the binary cross-entropy loss between segmentation map and ground-truth map. Compared with traditional L2 loss, smooth L1 loss is more robust and has the advantage of eliminating the outlier effect [30].

Multi-task Classification Model

The loss of our multi-task classification model consists of two parts: the pixel-wise difference of segmentation task loss and classification task loss. Similarly as other vessel segmentation methods, we choose pixel-wise intensity difference as the training target. As mentioned above, our classification takes patches as inputs. The training dataset is defined by $\mathcal{C} = \{(\boldsymbol{P}_i, \boldsymbol{Y}_i, c_i)\}_{i=1}^{n_c}$, with $\boldsymbol{P}_i$ being the input patch and $\boldsymbol{Y}_i$ being the corresponding segmentation map while c indicates the patch class. It should be noted that $c = 1$ indicates the patch is junction in detection stage while $c = 1$ changes its meaning in classification stage which refers to crossover patch. Each pixel j value in $\boldsymbol{P}_i$ and $\boldsymbol{Y}_i$ can be written as p^i_j and $y^i_j \in \{0, 1\}$ respectively. Consider the imbalance of vessel pixels and background pixels, we adopt the class-balance entropy instead of the original one [38]. So the loss function of segmentation task can be defined as:

$$L^c_{seg} = -\sum_j \left(\alpha y^i_j log(\hat{y}^i_j) + (1-\alpha)(1-y^i_j) log(1-\hat{y}^i_j)\right) \tag{3}$$

where α is used to handle the imbalance of foreground and background pixels, $\hat{y}^i_j$ is the pixel prediction from assistant branch. For the classification loss, the cross-entropy can be defined as:

$$L^c_{cls} = -\left(c_i log(\hat{c}_i) + (1-c_i) log(1-\hat{c}_i)\right) \tag{4}$$

where $\hat{c}_i$ is the prediction of patch $\boldsymbol{P_i}$. So the total loss for classification leads to:

$$L^c_{total} = \gamma L^c_{seg} + (1-\gamma) L^c_{cls} \tag{5}$$

where γ indicates the weights of these two classes.

The JRN and JCN share the same model structure and training strategy but they are trained separately. The weights of each network are updated with different training data, which will be introduced in the following section.

4. Experiments and results

4.1. Datasets

In our experiments, we evaluate the performance of our proposed approach using two public datasets, DRIVE [39] and IOSTAR [22]. DRIVE dataset includes 40 images with a resolution of 584×565 pixels and Field of View of 45°, where the first 20 images are for training and the other 20 images are for testing. IOSTAR is a scanning laser ophthalmoscope (SLO) image dataset which includes twenty-four 1024×1024 pixel images with a 45° Field of View. The split of training / testing images is 12 / 12. On average, there are 100 bifurcations and 30 crossovers per image in DRIVE and 55 bifurcations and 23 crossovers in IOSTAR. The junction ground-truths of these two datasets[1] are annotated by the authors of [22].

4.2. Data preprocessing

In this section, the details of how to prepare training and testing data for our models are described. Our approach contains two individual models, one is Junction Proposal Network (JPN) and the other is multi-task classification model for JRN and JCN. Although both of them need image patches as input, the details are not the same. In this part, we will introduce the data preprocessing for these two models and the parameter setting.

Junction Proposal Network. Retinal image patches, junction positions, and segmentation around junctions are needed for network training. Firstly, a retinal image and corresponding segmentation map are cropped into patches with a 128×128 sliding window. The segmentation map is further processed with junction positions to remove most of the vessel segments while only keeping the vessel patterns near junction locations. These vessel patterns provide a mask and bounding-box training information (*i.e.* ground-truth) for Junction Proposal Network. In the testing phase, only retinal image patches are needed to give the initial detection results.

Multi-task Classification Model. Detection refinement and junction classification share the same model structure. Though the two networks are for different tasks, the essence of them is the same. In this case, they can share a same training pool with slightly different configuration. The pool is made up of patches extracted from both retinal images and segmentation maps with a size of 16×16, and the patches can be divided into four categories: bifurcation patches, crossover patches, background patches near vessels, and background patches far away from vessels. In bifurcation and crossover patches, the centers are their ground-truth positions. In background patches, the centers are selected on the map except junction regions. A tolerance region is considered for each junction to reduce the offset influence. In practice, centers of junction patches are chosen inside a five-pixel diameter of ground-truth junctions. The refinement network handles crossover and bifurcation patches as positive samples and the background patches are as negatives. While positive and negative samples are crossover and bifurcation patches respectively for junction classification task.

All the experiments are conducted on a desktop computer with Intel Core i7 CPU and Titan X GPU. The algorithms are implemented using Python with Tensorflow library. Training time for the detection stage is 187 minutes and for classification stage is 12 minutes, while the testing time are 2.8s and 0.2s per image respectively. When training JPN, we choose 9 anchors for RPN with scales and ratios being $\{4^2, 8^2, 16^2\}$ and {1:2, 2:1, 1:1}. The scales are determined according to the region size around junctions which is characterized by the width of vessels and intersection angles. Thirty-two ROIs are generated by RPN for each image patch and the bounding boxes are fine-tuned by location regression branch. JPN is trained on one GPU for 90 epochs. Only the head module of JPN is trained for the first thirty epochs with learning rate of 0.001, while all the layers are trained for the rest epochs with a learning rate decreased by 10. The parameters are updated with SGD optimizer with momentum 0.9. As to the multi-task classification model, it is optimized by Adam optimizer with a learning rate of 0.001. The coefficients in Eq. (3) and Eq. (5) are set as $\alpha = 0.6$ and $\gamma = 0.5$ respectively.

4.3. Testing phase

Once we have finished training two stage models, we can apply it on new retinal images to detect and classify junctions. Following the pipeline in Fig. 1(a), a retinal image is first cropped into several 128×128 patches which can cover the whole image. Then the patches are fed into JPN to get bounding boxes around junctions. The center of the bounding box is regarded as the junction locations and refined by the JRN to eliminate the false detection. Final junction positions are identified after the above steps. The junctions are further categorized into either crossover or bifurcation by classifying patches centered on junction locations with the size of 16×16.

4.4. Junction detection results

The detection performance is evaluated on the two given public datasets with other state-of-the-art methods. Table 1 displays the quantitative results of our approach as well as other comparison methods on DRIVE and IOSTAR datasets. To evaluate the detection performance, three evaluation metrics are considered here, including precision ($\frac{TP}{TP+FP}$), recall ($\frac{TP}{TP+FN}$) and F1-score ($\frac{2 \cdot Pre \cdot Rec}{Pre+Rec}$). Following [22] and [28], an accepted tolerance with 5 pixels is set to accept the junctions. That is to say, the pixels in the distance of 5 pixels to actual junction points are considered as true positive points. *F1-score* delivers to an overall performance summary, *precision* shows how the method perform on right recognize the junction points, while *recall* indicates the ability to detect the feature points. The left panel shows the performance of DRIVE, while the right one displays the results of IOSTAR. The last two rows show the results of our approach. One is the junction detection directly from JPN without refinement and the other is that junction detection refined by JRN. Performance of other state-of-the-art methods has been shown in the first to fifth rows. Method of Calvo *et al.* [10] is a skeleton-based method with topological analysis. COSFIRE [20] is a model-based method using a bank of filters to detect junctions. BICROS [22] combines orientation-score-based method and skeleton-based method together to identify feature points. The last two [27,28] are deep learning based methods, which engage RBMs and CNN respectively.

Take DRIVE dataset as an example. Compared with other state-of-the-art methods, our final detection achieves the highest F1-score of 0.70, which increases 4.5% than the second high method of Uslu *et al.* and BICROS. Meanwhile, our approach keeps a good

[1] The datasets and junction ground-truths can be downloaded here: http://www.retinacheck.org/datasets.

Table 1
Quantitative evaluation of junction detection on DRIVE and IOSTAR. The comparison methods include Calvo *et al.* [10], COSFIRE [20], BICROS [22], Pratt *et al.* [27], Uslu *et al.* [28]. The performance is evaluated with Precision, Recall and F1-score.

	DRIVE			IOSTAR		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Calvo *et al.* [10]	0.71	0.51	0.59	0.61	0.48	0.54
COSFIRE [20]	0.40	0.74	0.52	0.63	0.33	0.43
BICROS [22]	0.75	0.61	0.67	0.47	0.60	0.52
Pratt *et al.* [27]	0.74	0.57	0.64	0.52	0.54	0.52
Uslu *et al.* [28]	0.65	0.69	0.67	0.52	0.67	0.59
Ours w/o refinement	0.42	0.78	0.55	0.36	0.71	0.48
Ours	0.71	0.70	0.70	0.62	0.57	0.60

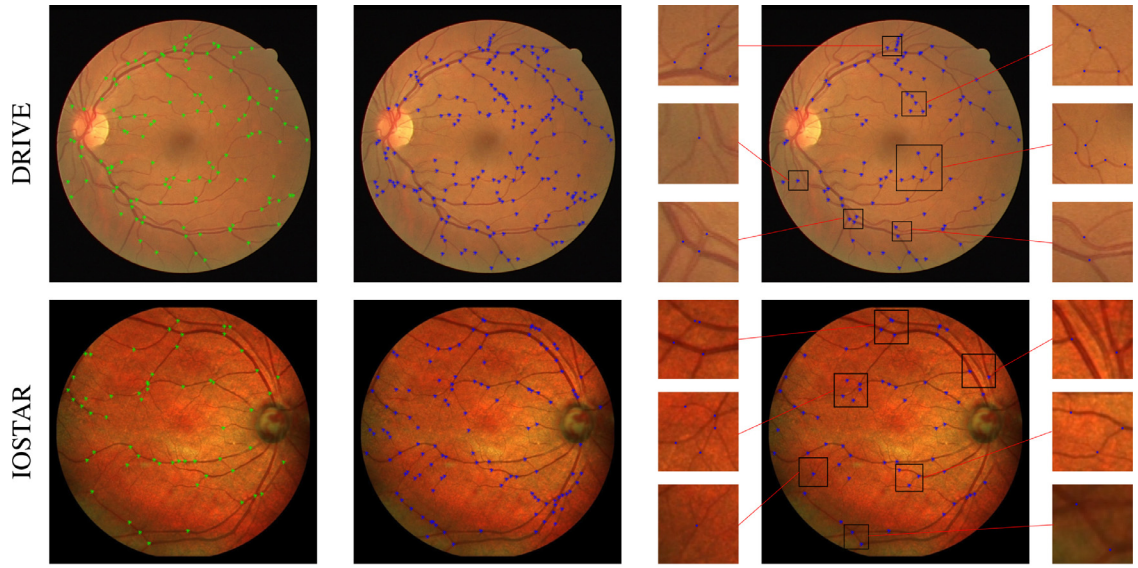


Fig. 2. Visual results of our two-step detection framework and different datasets are viewed in different rows. (a) displays the junction ground-truth, the crossovers and bifurcations are labeled by green stars. (b) is the initial detection results of our Junction Proposal Network. (c) shows the final results by our refinement network where several zoomed-in patches are selected.

balance of the precision and recall. The recall score of 0.70 shows that most of the junction points could be detected by our approach, precision of 0.71 indicates that the accuracy of detection points is high. Although some other methods achieve a better precision result, we can find that the recall of their methods is not as high as ours, which finally leads to a lower F1-score. Compared with skeleton-based and model-based methods, the performance of our approach shows significant improvement, which increases 0.11 and 0.18 respectively. Compared with deep learning method, our method also outperforms them especially on the recall evaluation. Following the setting in [22,27,28], we use the model weights trained on DRIVE directly without retraining and the results are shown in the right panel. It seems that the deep learning methods don't work so well on a new modality without training, which causes almost 0.1 F1-score decreasing. We assume that the features extracted by the model trained on DRIVE are not suitable for the new modality dataset and this leads to a performance decrease. Although there is some decreasing on a new modality dataset, the performance of our approach still achieves a satisfactory result compared to others. The F1-score of our approach is 0.6 with precision score of 0.62 and recall score of 0.57. Similar to DRIVE dataset, the detection results have a good balance of precision and recall, which means the model can recognize as more junctions as possible with a satisfied accuracy.

Turn the view into our two-step approach. The first one is our Junction Proposal Network in the second last row. The initial Junction Proposal Network can produce a much higher recall of 0.78, but with a pretty low precision value of 0.42. However, this is reasonable because the role of this network is giving potential junction locations while it may generate large numbers of false positive junction points. This result is still comparable with some state-of-the-art method, such as COSIFRE. Compared with JPN, final results generated by JRN improve precision by 0.3, with recall slightly decreasing to 0.70. This added refinement step improves 27% of F1-score compared with the initial Junction Proposal Network.

The exemplar visual results are shown in Fig. 2. The ground-truth of junction point is displayed in Fig 2 (a), while (b) shows the detection of our Junction Proposal Network and (c) is our final detection results with zoomed-in views of some image part. The DRIVE dataset is in the top row of the figure while IOSTAR is displayed in the bottom. Most of the feature points can be recognized by our Junction Proposal Network. On the other hand, there are many false alarms in (b), such as stacks of junctions detected in such a small region or points on the vessel trunks. Most of the false detections are eliminated by our refined network and the detection map is much cleaner with correct junction points detected. In the zoomed-in views of selected patches, taking DRIVE as an example, the junctions created by large vessels can be well detected even when there are multiple junctions in a very near region (*e.g.* bottom-left zoomed-in view of DRIVE image in (c)). The tiny vessel junctions can also be recognized which are shown in the top-right patch of DRIVE in (c). The bottom-right patch of DRIVE in

Table 2
Quantitative evaluation of bifurcation and crossover classification on DRIVE and IOSTAR. The performance is evaluated with Precision, Recall and F1-score.

	DRIVE			IOSTAR		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Calvo *et al.* [10]	0.62	0.43	0.51	0.39	0.37	0.38
Pratt *et al.* [27]	0.67	0.70	0.68	0.41	0.74	0.53
Ours	0.73	0.66	0.69	0.63	0.51	0.56

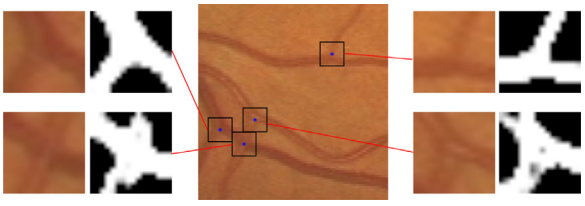
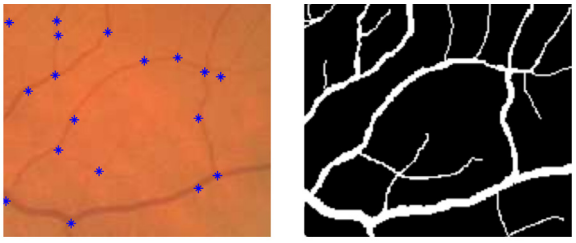


Fig. 3. Zoomed-in views of a region that contains several junctions with extra information of segmentation. The color patches in left and right panels are retinal image patches centered in junctions, while the gray ones are segmentation outputs from our JRN.



(a) Low contrast region around tiny vessels (b) Vessel ground-truths of (a)

Fig. 4. Performance of detection on low contrast image patch containing tiny vessels.

(c) shows that the correct detection can be achieved when two parallel vessels are close. Error usually occurs in this situation by the segmentation-dependent methods, which is avoided by our approach. However, the detection location is not always exactly on the position of the junctions, which can also be found on some thick vessel trunks. In Fig. 3, the detailed information of one exemplar patch is displayed. Junctions in the patch are selected and zoomed-in together with a segmentation map coming from assistant branch of our refinement network. Though the segmentation map is not perfect, it can produce the correct shape of selected patch. The segmentation shows three vessel branches of bifurcation and four branches of crossover. This assistant branch of refinement network can also be used as an indicator to determine whether the detection is correct or not.

To show our approach performance on low contrast regions, exemplar detection results are illustrated in Fig. 4, where (a) is the low contrast image patch containing tiny vessels and (b) is the corresponding vessel ground-truth for better view of bifurcations and crossovers. Our approach is able to detect the junctions, even though the vessels can not be seen clearly. Most of the feature points are detected correctly in Fig. 4, as well as the crossover formed by two thinnest vessels.

4.5. Junction classification results

As long as we get the positions of junction points, we can divide them into two types: bifurcation and crossover. It outputs the segmentation and category of the input patch. The classification results are displayed in Table 2 together with the comparison with the state-of-the-art methods. Our approach achieves the best performance on both DRIVE and IOSTAR datasets with F1-score of 0.69 and 0.56 respectively. Our classification model obtains the best precision results of 0.73 on DRIVE. For the IOSTAR dataset, it also achieves the highest precision performance of 0.63 while the other two methods are around 0.4. Our recall score on DRIVE and IOSTAR are 0.66 and 0.51 respectively. Compared with our approach, method of Pratt *et al.* achieves a better recall value for both datasets, but the points classified into crossovers have a low accuracy which is 67% on DRIVE and 41% on IOSTAR. Combination of detection rate and the detection accurate rate, our approach is slightly better than Pratt *et al.* as reflected on F1-score.

Table 3
Ablation study where results are produced by variant models trained with different settings on DRIVE dataset. Quantitative evaluation metrics considered here include Precision, Recall and F1-score.

	Precision	Recall	F1-score
JPN	0.42	0.78	0.55
JPN with threshold	0.63	0.69	0.66
Refinement w/o SEG	0.68	0.69	0.68
Refinement with SEG	0.71	0.70	0.70

5. Discussion

5.1. Refinement network

To explain the necessity of our multi-task classification model in the detection stage and how it works, we have carried out some ablation experiments on DRIVE dataset. In our approach, we accept all the junction locations predicted from JPN without considering the class probability of each bounding box. Another idea to use JPN is combining the bounding box locations with thresholding probability, just as what other RCNN-based methods [32] do. We compare our final result with the threshold Junction Proposal Network, and the comparison results are shown in Table 3. The first row displays the performance of JPN which is the same in Table 1. The second row is the results of JPN with suitable threshold. The last row indicates our final detection performance. Choosing a good threshold for the JPN can get satisfied F1-score of 0.66, while the results of our refinement network is 0.70. Compared with our final detection model, the threshold JPN can get a similar recall score, which indicates the threshold network can obtain a good recognition rate. However, its precision is 0.08 lower than that of our final detection model. Comparison of segmentation output from JPN and our refinement network (JRN) is displayed in Fig. 5. The mask branch output of Junction Proposal Network tends to generate vessel segmentation with indistinct edges and corners, which is not like a vessel shape. This phenomenon is more obvious in the crossover segmentation, where the branches of vessels are not extracted very well. While for the similar cases, the assistant branch of our refinement network can produce more accurate segmentation according to the input patches. This advanced segmentation results in turn help the classification task.

Fig. 5. Comparison of segmentation output of JPN mask branch and JRN assistant branch. For each panel, the images from left to right are retinal image, segmentation ground-truth, segmentation from our JRN and the segmentation from JPN.

Our superior performance gives an explanation why our refinement network can improve the junction detection performance. We analyze this phenomenon in two aspects. From the task level, our refinement network is designed only for classification, while the classification and bounding box detection are both performed for Junction Proposal Network. From the model level, our refinement network is more complicated in the aspect of model structure, which means it is more powerful to deal with the complex situation of retinal image. On the contrary, the classification and mask branches in JPN are simple and originally designed for the natural images that have higher contrast and easy content.

5.2. Assistant branch

To see how the assistant branch of our classification model influences the performance, we build our classification model in two ways: with or without assistant branch (aka. segmentation branch). Taking the detection stage as an example, the classification model is used for eliminating the false positive detections and it takes the result from JPN as input. The quantitative results are shown in Table 3. We observe that the model trained only with classification task has a F1-score of 0.68, recall score of 0.69 and precision of 0.68. This performance is still better than threshold Junction Proposal Network with a higher precision score. However, the model trained with multi-task can gain much higher precision score of 0.71 with a bit recall improvement which further leads to the highest F1-score of 0.70. The experiments suggest that the multi-branch network structure not only can provide additional analysis information but also can improve the classification performance.

6. Conclusion

In this paper, we propose a two-stage retinal junction detection and classification framework. It works directly on the original color retinal image without any preprocessing, such as segmentation. This avoids the mistakes caused by segmentation or skeleton and it is flexible to apply on any new dataset. We utilize a RCNN-based Junction Proposal Network to locate the initial junction positions followed by a multi-task classification model as Junction Refinement Network to refine the detection. Our classification model is specially designed for retinal images, thus it can improve the performance of initial junction detection obtained by the Junction Proposal Network. The same classification model is also used for classifying the crossover and bifurcation. The experiments on DRIVE and IOSTAR shows our advantage performance among state-of-the-art methods. Our approach can be easily applied to other medical field with long and thin tubular structures, such as pulmonary vessels, neuronal trees. Furthermore, it can also be utilized for detecting lesion area, such as tumor and aneurysm. In the future work, we will investigate our approach on other kinds of images and extent it to 3D neuron images for bifurcation detection and structure reconstruction.

Acknowlgedgments

The authors have no conflict of interest.

References

[1] A.V. Stanton, B. Wasan, A. Cerutti, S. Ford, R. Marsh, P.P. Sever, S.A. Thom, A.D. Hughes, Vascular network changes in the retina with age and hypertension, J. Hypertension 13 (12) (1995) 1724–1728.
[2] T. Teng, M. Lefley, D. Claremont, Progress towards automated diabetic ocular screening: a review of image analysis and intelligent systems for diabetic retinopathy, Med. Biol. Eng. Comput. 40 (1) (2002) 2–13.
[3] J.J. Kanski, B. Bowling, Clinical ophthalmology: a systematic approach, Elsevier Health Sciences, 2011.
[4] T.Y. Wong, R. Klein, F.J. Nieto, B.E.K. Klein, A.R. Sharrett, S.M. Meuer, L.D. Hubbard, J.M. Tielsch, Retinal microvascular abnormalities and 10-year cardiovascular mortality: a population-based case-control study, Ophthalmology 110 (5) (2003) 933–940.
[5] A.S. Neubauer, M. Luedtke, C. Haritoglou, S. Priglinger, A. Kampik, Retinal vessel analysis reproducibility in assessing cardiovascular disease, Optometry Vision Sci. 85 (4) (2008) E247–E254.
[6] J. De, L. Cheng, X. Zhang, F. Lin, H. Li, K.H. Ong, W. Yu, Y. Yu, S. Ahmed, A graph-theoretical approach for tracing filamentary structures in neuronal and retinal images, IEEE Trans. Med. imaging 35 (1) (2016) 257–272.
[7] F. Zana, J.-C. Klein, A multimodal registration algorithm of eye fundus images using vessels detection and hough transform, IEEE Trans. Med. Imaging 18 (5) (1999) 419–428.
[8] J. Zhang, H. Li, Q. Nie, L. Cheng, A retinal vessel boundary tracking method based on bayesian theory and multi-scale line detection, Comput. Med. Imaging Graph. 38 (6) (2014) 517–525.
[9] A.M. Aibinu, M.I. Iqbal, A.A. Shafie, M.J.E. Salami, M. Nilsson, Vascular intersection detection in retina fundus images using a new hybrid approach, Comput. Biol. Med. 40 (1) (2010) 81–89.
[10] D. Calvo, M. Ortega, M.G. Penedo, J. Rouco, Automatic detection and characterisation of retinal vessel tree bifurcations and crossovers in eye fundus images, Comput. Methods Programs Biomed. 103 (1) (2011) 28–38.
[11] M.E. Martinez-Perez, A.D. Highes, A.V. Stanton, S.A. Thorn, N. Chapman, A.A. Bharath, K.H. Parker, Retinal vascular tree morphology: a semi-automatic quantification, IEEE Trans. Biomed. Eng. 49 (8) (2002) 912–917.
[12] A. Fathi, A.R. Naghsh-Nilchi, F.A. Mohammadi, Automatic vessel network features quantification using local vessel pattern operator, Comput. Biol. Med. 43 (5) (2013) 587–593.
[13] A. Bhuiyan, B. Nath, J. Chua, K. Ramamohanarao, Automatic detection of vascular bifurcations and crossovers from color retinal fundus images, in: Third International IEEE Conference on Signal-Image Technologies and Internet-Based System, IEEE, 2007, pp. 711–718.
[14] D.-M. Baboiu, G. Hamarneh, Vascular bifurcation detection in scale-space, in: 2012 IEEE Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA), IEEE, 2012, pp. 41–46.
[15] U.T.V. Nguyen, A. Bhuiyan, L.A.F. Park, R. Kawasaki, T.Y. Wong, K. Ramamohanarao, Automatic detection of retinal vascular landmark features for colour fundus image matching and patient longitudinal study, in: IEEE International Conference on Image Processing (ICIP), 2013, pp. 616–620.
[16] B. Lin, Y. Sun, J.E. Sanchez, X. Qian, Efficient vessel feature detection for endoscopic image analysis, IEEE Trans. Biomed. Eng. 62 (4) (2015) 1141–1150.
[17] S. Morales, V. Naranjo, J. Angulo, A.G. Legaz-Aparicio, R. Verdú-Monedero, Retinal network characterization through fundus image processing: significant point identification on vessel centerline, Signal Process. 59 (2017) 50–64.
[18] Y. Zhao, J. Xie, P. Su, Y. Zheng, Y. Liu, J. Cheng, J. Liu, Retinal artery and vein classification via dominant sets clustering-based vascular topology estimation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2018, pp. 56–64.
[19] Y. Zhao, L. Rada, K. Chen, S.P. Harding, Y. Zheng, Automated vessel segmentation using infinite perimeter active contour model with hybrid region information with application to retinal images, IEEE Trans. Med. imaging 34 (9) (2015) 1797–1807.
[20] G. Azzopardi, N. Petkov, Automatic detection of vascular bifurcations in segmented retinal images using trainable COSFIRE filters, Pattern Recognit. Lett. 34 (8) (2013) 922–933.
[21] T. Ahmad Qureshi, A. Hunter, B. Al-Diri, A bayesian framework for the local configuration of retinal junctions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3105–3110.
[22] S. Abbasi-Sureshjani, I. Smit-Ockeloen, E. Bekkers, B. Dashtbozorg, B. ter Haar Romeny, Automatic detection of vascular bifurcations and crossings in retinal images using orientation scores, in: International Symposium on Biomedical Imaging (ISBI), IEEE, 2016, pp. 189–192.
[23] C.L. Srinidhi, P. Rath, J. Sivaswamy, A vessel keypoint detector for junction classification, in: International Symposium on Biomedical Imaging (ISBI 2017), IEEE, 2017, pp. 882–885.
[24] S. Kalaie, A. Gooya, Vascular tree tracking and bifurcation points detection in retinal images using a hierarchical probabilistic model, Comput. Methods Programs Biomed. 151 (2017) 139–149.
[25] C.-L. Tsai, C.V. Stewart, H.L. Tanenbaum, B. Roysam, Model-based method for improving the accuracy and repeatability of estimating vascular bifurcations

and crossovers from retinal fundus images, IEEE Trans. Inf. Technol. Biomed. 8 (2) (2004) 122–130.
[26] R. Su, C. Sun, T.D. Pham, Junction detection for linear structures based on hessian, correlation and shape information, Pattern Recognit. 45 (10) (2012) 3695–3706.
[27] H. Pratt, B.M. Williams, J.Y. Ku, C. Vas, E. McCann, B. Al-Bander, Y. Zhao, F. Coenen, Y. Zheng, Automatic detection and distinction of retinal vessel bifurcations and crossings in colour fundus photography, J. Imaging 4 (1) (2017) 4.
[28] F. Uslu, A.A. Bharath, A multi-task network to detect junctions in retinal vasculature, arXiv preprint arXiv:1806.03175 (2018).
[29] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.
[30] R. Girshick, Fast r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.
[31] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems, 2015, pp. 91–99.
[32] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2961–2969.
[33] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788.
[34] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, Ssd: Single shot multibox detector, in: European Conference on Computer Vision, Springer, 2016, pp. 21–37.
[35] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[36] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2117–2125.
[37] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical image Computing and Computer-assisted Intervention, Springer, 2015, pp. 234–241.
[38] S. Xie, Z. Tu, Holistically-nested edge detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1395–1403.
[39] J. Staal, M.D. Abràmoff, M. Niemeijer, M.A. Viergever, B. Van Ginneken, Ridge-based vessel segmentation in color images of the retina, IEEE Trans. Med. Imaging 23 (4) (2004) 501–509.