

Vessel Extraction from Retinal Images

Final Project Report

Ariane Alix and Baptiste Dehaine

École Normale Supérieure Paris-Saclay
ariane.alix@ens-paris-saclay.fr
baptiste.dehaine@eleves.enpc.fr

Abstract. In this work, we present a description and evaluation of our method for the segmentation of blood vessel of retinas. Retinal vessel segmentation can be of great interest for the diagnosis of retinal vascular diseases such as age-related macular degeneration, glaucoma and diabetic retinopathy. Therefore, in recent years, several automatic segmentation methods have been proposed, ranging from the use of simple filters to Deep Neural Networks.

The methods based on Convolutional Neural Networks mainly rely on the learning of local patterns of vessels, but do not consider the graphical structure of those. We think that taking advantage of the vessels network-like form, inherent to their biological utility, would help improve the vessel segmentation accuracy. Therefore, our method uses both a IterNet [16] architecture, intertwined with a Graph Neural Network with Attention to simultaneously comprehend local, global and structural vessel patterns. The idea of a combination of a Convolutional Neural Network and a Graph Network comes from Shin's "VGN" [14], a method developed in 2018 that was the state-of-the-art on retinal vessels segmentation when it got out.

We evaluated our model on a retinal image called DRIVE, and compared it to the current state-of-the-art methods in terms of the average precision, AUC and F-measure. The scores obtained on our validation set are on-par with the state-of-the-art.

1 Introduction

1.1 The problem

The problem we are interested in is the segmentation of blood vessels in the fundus, which is the interior surface of the eye including the retina and the fovea. The goal is to identify the pixels of the image corresponding to blood vessels. Figure 1 shows an example of retinal fundus and the segmentation obtained with the B-COSFIRE method (see subsection 3.1 for more details).

Our work focuses on developing a new method for the retinal fundus segmentation by combining Shin's "Deep vessel segmentation by learning graphical connectivity" [14], and Li's "IterNet" network [16]. We will evaluate it on the DRIVE dataset [11], composed of forty photographs that were selected randomly from a diabetic retinopathy screening program in The Netherlands.

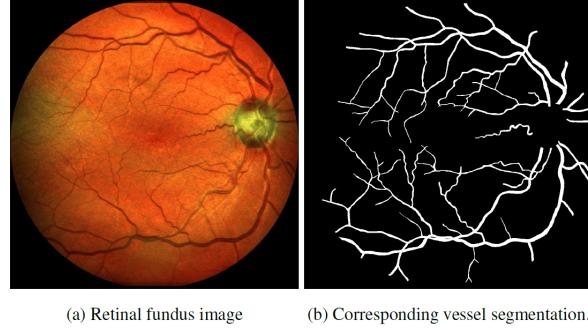


Fig. 1: Example of retinal vessels segmentation [5]

1.2 Why it is important

The inspection of the retinal blood vessels can be important in the determination of various cardiovascular diseases. The study of their length, width, tortuosity, branching patterns and angles are used for the diagnosis of diseases such as diabetes, hypertension, arteriosclerosis... [11]

This can be done manually but it is time-consuming, and it can be a difficult task in a lot of cases, for example with a low contrast or vessel anomalies. Therefore, the automatic detection and analysis of the retinal vessels can assist specialist in the diagnosis of diseases by extracting the vascular structure.

1.3 Other possible applications

Those specific segmentation methods could be used in other fields than for retinal blood vessels segmentation. Using the same principle, we could segment all types of networks or vessel-like structures in pictures, like roads or rivers on satellite images, or even cracks in structures. For example, an article published in 2018 by F. Bastani, S. He et al. [17], presents a model focusing on the automatic generation of road maps from aerial images using an iterative graph construction process based on CNNs.

2 Problem definition

Let us note X the input image and $\{x_i, i = 1 \dots N\}$ its N pixels.

In the manual segmentation that we try to reproduce, each pixel is labeled either "vessel" (1, white) or "non-vessel" (0, black).

Globally, the final goal is to minimize the pixel-wise cross-entropy when predicting if a pixel is part of a vessel:

$$\text{Loss}(X) = -\frac{1}{N} \sum_{i=1}^N (y_i \log(p(x_i)) + (1 - y_i) \log(1 - p(x_i)))$$

where y_i is the true label (1 or 0) of the pixel x_i and $p(x_i)$ is the predicted probability for pixel x_i to be part of a blood vessel.

3 Related Work

In this section, we review related work on vessel segmentation.

3.1 Without Deep Learning

Due to the importance of the problem, there is a substantial amount of methods that have been developed over the years; among those a lot are based on "classical" image processing. These methods mainly rely on filters that aim to emphasize the pixels of the vessels in the images (see for instance [2]). Some of those filter-based methods are more complex than a simple change of contrast: B-COSFIRE [1] is an example of a trainable filter that can lean interest point detection; it is trained on vessel-like patterns to be able to select similar structures specifically later on. It reached a 0.9487 AUC and 0.9387 of accuracy on the CHASE dataset (retinal vessels dataset similar to DRIVE).

Other methods consider vessel segmentation as an optimization problem, where we aim to minimize the cost of the graph structure [3,4,9]. The main issue here is the definition of such cost which is arbitrary.

Finally, new methods appeared with Machine Learning learning around 2012. For example Fraz et al.[8] used an Ensemble (of decision trees) method in 2012 and reached a 0.9712 AUC and 0.9469 of accuracy on CHASE. Becker et al. [6] used boosting in 2013, [7] used a regression in 2015 and became the state-of-the art for the F measure. Even if they are more costly computationally-wise, these methods perform better than classical image processing. Nonetheless, those models stay simple compared to the complexity of the structures of vessels, and thus cannot understand accurately all structural levels.

3.2 With Deep Learning

The most recent and performing methods however, are based on Deep Learning. As opposed to "non-deep" methods, the high number of layers and their depth allow the models to comprehend all levels of the vessel structures, by looking simultaneously at local and global features.

In [13] (2015) Olaf Ronneberger, Philipp Fischer, Thomas Brox use a convolutional network for biomedical image segmentation. Their U-net architecture achieved high performance on various segmentation tasks, and became the state-of-the art for the retinal vessel problem, reaching an AUC of 0.9755 on the DRIVE dataset. In 2018, Seung Yeon Shin et al. [14] proposed a method called VGN to improve any CNN vessel segmentation by learning graphical connectivity using graph attention networks, which were presented in [15] as a novel convolution-style neural networks that operate on graph-structured data. They achieved state-of-the-art performances on the DRIVE dataset, with a 0.9802 AUC and 0.8263 F1 score that is still unbeaten.

Shin's VGN method was only beaten (for the AUC) last year by IterNet [16], which consists of multiple iterations of a mini-UNet. It reached the highest AUCs for the three most common retinal dataset: 0.9816, 0.9851, and 0.9881 on the DRIVE, CHASE-DB1, and STARE datasets respectively.

4 Methodology

4.1 Overview of the model's architecture

Our work is mainly structured like Shin's "Deep vessel segmentation by learning graphical connectivity" paper [14]. His version had a classical CNN as the first module out of the 3 main parts; in our approach, we replaced it by our implementation of IterNet. The three main modules of the architecture are therefore:

1. IterNet, a CNN-based model that does a first segmentation by predicting probabilities for each pixel of an image; its loss function is the pixel-wise cross-entropy (see section 2 for the loss computation),
2. a GNN (Graph Neural Network) taking as input a graph G^X extracted from the results of IterNet; its loss is a vertex-wise cross-entropy:

$$L_{GNN}(G^X) = -\frac{1}{|V|} \sum_{v_j \in V} (y_{v_j} \log(p^{GNN}(v_j)) + (1 - y_{v_j}) \log(1 - p^{GNN}(v_j)))$$

where G^X is the graph extracted from the results of IterNet on image X , y_{v_j} the true label of vertex v_j , and $p^{GNN}(v_j)$ the predicted vessel probability for the vertex,

3. an inference module that we will call GVN; it takes as input both the results of the IterNet and of the GNN and predicts the final segmentation; it has a loss criterion L_{GVN} which is the pixel-wise cross-entropy like for the first module.

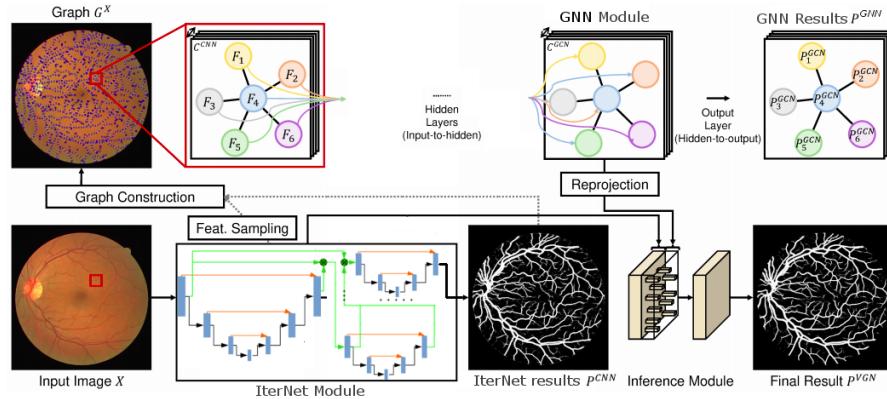


Fig. 2: Overall network architecture comprising the IterNet, Graph Neural Network, and Inference module. Adapted version of a figure extracted from [14]

4.2 IterNet

IterNet is a CNN-based network developed in 2019 by Liangshi Li et al. [16], and is currently the state-of-the-art on the DRIVE dataset in terms of AUC.

The network consists of two slightly different architectures: a UNet, and a simplified version of U-Net, they named 'mini-UNet'. UNet is the base module since it is known for its great performances in segmentation problems; the mini-UNets modules are responsible for the refinement of small parts of the vessels, for example to infer missing pieces.

The first UNet outputs a 32-channel feature map, that has more information than the simple map of probabilities of pixels being on a vessel. Afterward, each mini-UNet uses the output of the last layer of its precedent module, and the concatenated features extracted after the first layers of all the previous modules. Similarly to human annotators who might need to check the specific area in the raw images, the mini-UNets work better when they receive this higher level information as input.

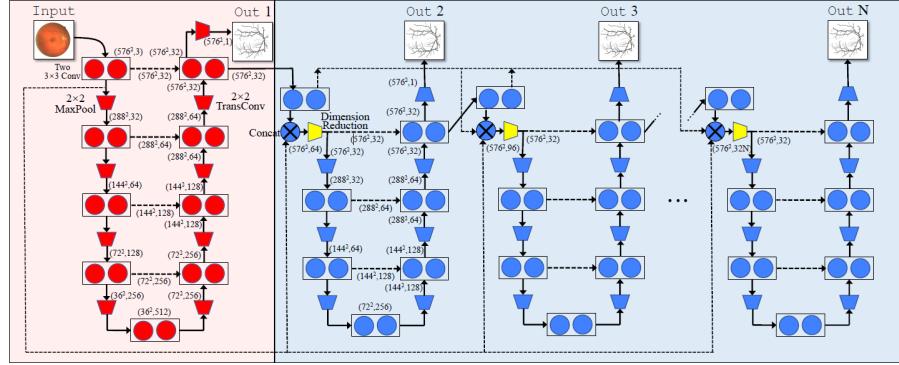


Fig. 3: The structure of IterNet, which consists of one UNet and iteration of N-1 mini-UNets. Figure and caption extracted from [16]

Since the code was only available in Tensorflow, we re-implemented IterNet in Pytorch from scratch using Li's paper. The only difference is that we added some normalization layers that help with the performance of our implementation.

4.3 Graph neural network

Once trained, we use Iternet segmentation results to construct the graphs which represents the vessel structure. The graphs are constructed in the following way:

1. Thresholding: we use a threshold of 0.3.
2. Skeletonization by morphological thinning
3. Vertex generation by equidistant sampling, with distance $\delta = 5$, on the skeleton together with skeletal junctions and endpoints

4. Edge generation between vertices based on the skeletal connectivity.

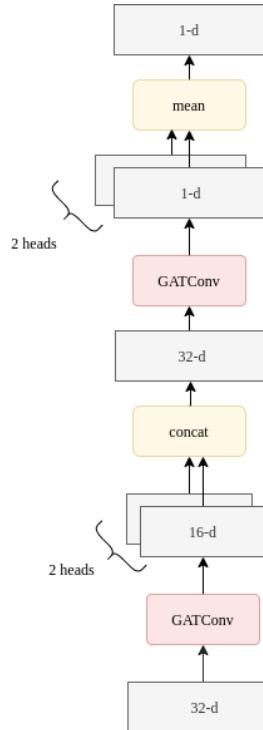


Fig. 4: Diagram of the architecture. Our model takes as input nodes with 32 features. For each node in the graph, a graph attention layer produces features of shape (2,16). The output features are then concatenated and after going through another graph attention layer reducing the dimension to one, we get the vertex probability.

Figure 6 shows one graph obtained after segmentation. The input feature vector for each vertex is sampled from the intermediate feature map generated from the CNN at the pixel coordinate of each vertex.

Once the graph obtained, we use a graph attention network to classify the vertex. We use a two-layer feed-forward model with two heads in each layer. The diagram of the architecture used is given in Figure 4.

4.4 Inference module

To conduct inference, we combine features obtained with the CNN and GNN modules. The features from the CNN modules are the features feeding the last convolutional layer. And the features from the GNN module are the features

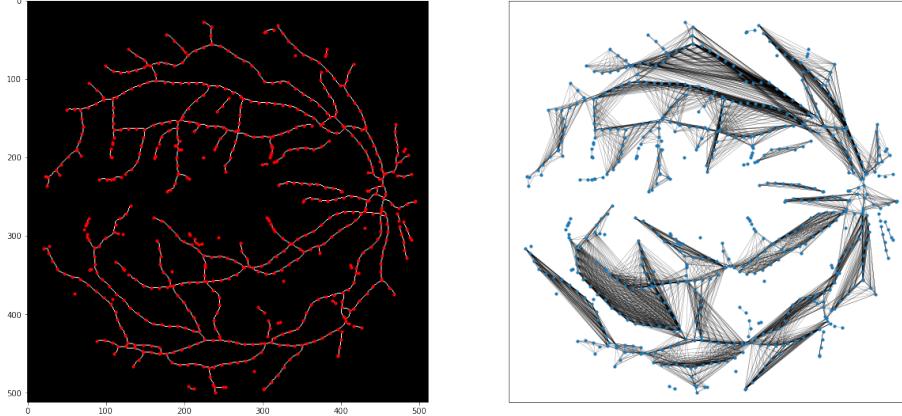


Fig. 5: Graph construction above a segmentation. In the left image, the red dots are the nodes obtained by equidistant sampling. In the right image, the nodes are in blue and edges have been drawn.

obtained before the last attention layer. Both modules provide features of 32 dimensions. They are concatenated together and fed through a neural network with five convolutional layers of kernel size 3×3 . Each layer except the last one is followed by a relu activation. We use a weighted pixelwise cross entropy loss for training to give more importance to pixel which belongs to the graph (weight of δ^2 instead of 1).

4.5 Network training

For training, we initially pretrained Iternet. It is followed by jointly training the GNN and the inference module while fine tuning the CNN.

The probabilities inferred from the pretrained CNN are used to construct the training graphs. Every $K_{gc} = 20$ epochs, we construct new graphs according to the probabilities given by the fine tuned CNN. Compared to the pretrained CNN module, our GVN takes as input the CNN features, as well as the graphs features generated by the GNN module. The total loss function use to train the whole network is

$$L_{TOTAL}(X) = L_{CNN}(X) + L_{GNN}(X) + L_{INFER}(X) \quad (1)$$

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

5 Evaluation

Due to computational limitations, we could only test a version with 1 iteration of mini-UNet in the IterNet part of the model.

We evaluated the segmentation on the DRIVE challenge dataset [11], a widely-used retinal fundus dataset composed of 40 .JPEG-compressed pictures, along with masks and manual segmentation of the vessels (ground truth) for 20 of them. When developing the model, we used 16 of the annotated images as training, the 4 others as validation. The 20 unlabeled ones are used for the challenge: we predicted the segmentation and uploaded the result on the challenge's page.

The metrics we look at are the Area Under the Curve and the F1 score. The F1 score is defined as the weighted harmonic mean of the precision and recall of the test: $F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$ and is a classical scoring for segmentation problems (it is used for the ranking in the DRIVE challenge <https://drive.grand-challenge.org/evaluation/results/>).

5.1 On training and validation

Here below are the Receiver Operating Characteristic curves obtained for the IterNet only, and for the full VGN model on the training and validation sets:

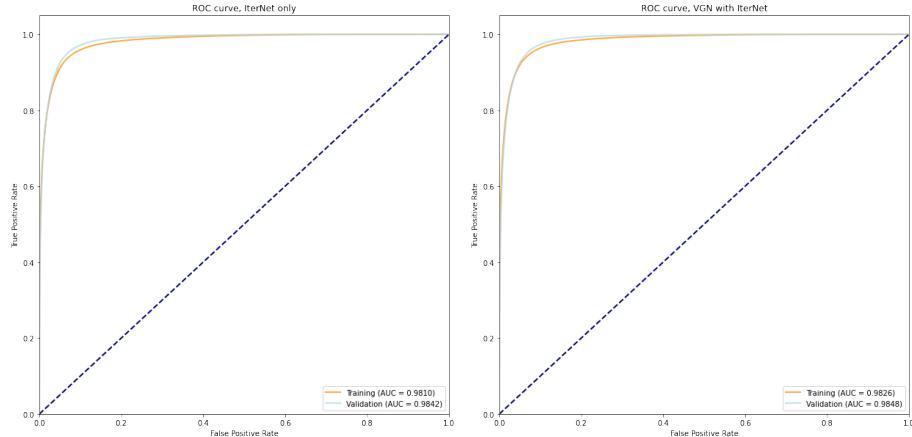


Fig. 6: ROC curves for the IterNet alone (left), and full VGN (right)

The following table sums up the main metrics obtained:

	IterNet only		VGN with IterNet	
	Training	Validation	Training	Validation
AUC	0.9810	0.9842	0.9826	0.9848
F1-score	0.7914	0.7965	0.8062	0.7962

Table 1: Scores obtained on training/validation sets

Figure 7 shows our segmentation results on validation set, using only Iternet first and then with VGN. We can see that our VGN helps extract thinner vessel.

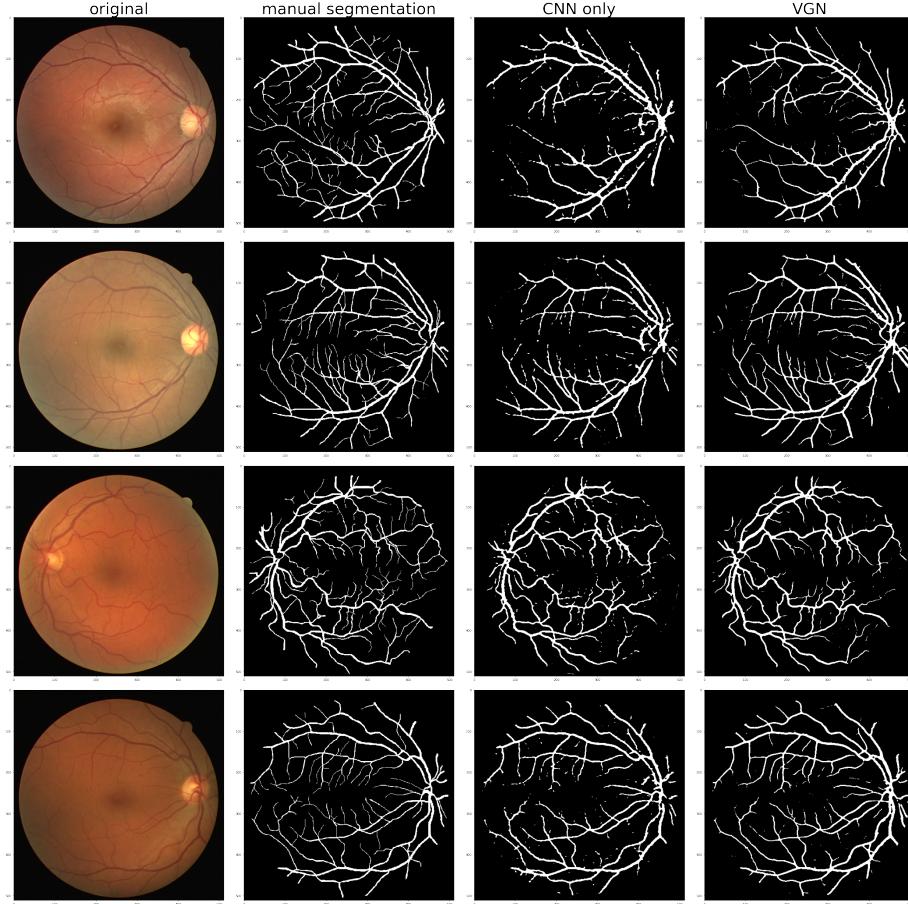


Fig. 7: Segmentation obtained on validation set.

6 Discussion

As opposed to us, other works in the literature had access to the labels of the 20 images we used for the challenge, therefore, their Receiver Operating Characteristic curves and thus AUC were not computed on the same dataset as us. As a consequence, we cannot compare exactly the AUC of our method to theirs; but we still can see that our validation AUC (0.9848) is in the same range or higher than the state-of-the-art (IterNet with 0.9816, VGN with 0.9802).

6.1 Potential improvements

The performances of our method could be improved with more computational power, which would allow us to have more than 1 mini-UNet iteration in the IterNet part.

We could also do some data augmentation: as we have only 20 images, it could help with overfitting. For example, we could add images in the training set that are variations of the ones given, by changing the balance of colors, the shape, the brightness or the orientation. To add data, we could have used another public dataset for vessel extraction: the STARE dataset contains 40 images with manual segmentation.

Another point of improvement would be to use multiple values of threshold for our graphs construction. We used an arbitrary value of 0.3 but we should use a wider range of values. If we used all threshold and computed the mean of all probabilities obtained to get our final probabilities , our results would probably be more robust. Concerning the graphs, two vertex are considered neighbors if they are connected on the skeleton. As a consequence, we could be paying attention to irrelevant vertex if they are very far away in space. We should limit the distance separating two neighbors in order to construct a more relevant graph. Indeed the immediate neighbors are more likely to help predict the label of a vertex. Finally we could find more optimal hyper-parameters by conducting a grid search.

References

1. G. Azzopardi, N. Strisciuglio, M. Vento, and N. Petkov, "Trainable cosfire filters for vessel delineation with application to retinal images," *Medical image analysis*, vol. 19, pp. 46–57, 1. (2015)
2. Soares, J.V.B., Leandro et al., "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. (2006)
3. Y. Zhao, Y. Liu et al., "Retinal vessel segmentation: An efficient graph cut approach with retinex and local phase." *PloS One* 10 (4), e0122332. doi: 10.1371/journal.pone.0127486. (2015)
4. S.Y. Shin , S. Lee et al. , "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 . (2016)
5. Straat, Michiel and Jorrit Oosterhof. "Segmentation of blood vessels in retinal fundus images." *ArXiv* abs/1905.12596 (2019)
6. Becker et al., "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* (2013).
7. A. Sironi, V. Lepetit, P. Fua, "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. (2015)
8. M. M. Fraz, P. Remagnino et al., "An ensemble classification-based approach applied to retinal blood vessel segmentation." *IEEE Transactions on Biomedical Engineering*, vol. 59, pp. 2538–2548. (2012)
9. J. I. Orlando, E. Prokofyeva and M. B. Blaschko, "A Discriminatively Trained Fully Connected Conditional Random Field Model for Blood Vessel Segmentation in Fundus Images," in *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 1, pp. 16-27, (Jan. 2017)
10. P. Liskowski and K. Krawiec, "Segmenting Retinal Blood Vessels With Deep Neural Networks," in *IEEE Transactions on Medical Imaging*, vol. 35, no. 11, pp. 2369-2380, (Nov. 2016)
11. "DRIVE Dataset." <http://www.isi.uu.nl/Research/Databases/DRIVE/>, (2004)
12. R. Sharma , A. Mugeesh and K. Nama, "DRIVE - Digital Retinal Images for Vessel Extraction", <https://github.com/rohit9934/DRIVE-Digital-Retinal-Images-for-Vessel-Extraction>
13. Olaf Ronneberger, Philipp Fischer, Thomas Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", Computer Science Department and BIOSS Centre for Biological Signalling Studies, University of Freiburg, Germany. (2015)
14. S. Y. Shin et al., "Deep vessel segmentation by learning graphical connectivity", *Medical Image Analysis* 58 (2019)

15. Petar Veličković and Guillem Cucurull and Arantxa Casanova and Adriana Romero and Pietro Liò and Yoshua Bengio, "Graphs Attention Networks". (2017)
16. Liangzhi Li, Manisha Verma, Yuta Nakashima et al., "IterNet: Retinal Image Segmentation Utilizing Structural Redundancy in Vessel Networks". (2019)
17. Favyen Bastani, Songtao , Sofiane Abbar et al., "RoadTracer: Automatic Extraction of Road Networks from Aerial Images". (2018)