Benchmarking Automatic Segmentation of Retinal Vascular Structure

Sofia Lima
Computational Biology Department
Carnegie Mellon University
slima2@andrew.cmu.edu

Jen Yi Wong
Computational Biology Department
Carnegie Mellon University
jenyiw@andrew.cmu.edu

Abstract

The purpose of this project is to benchmark a deep-learning U-Net method against a non-deep-learning method for the segmentation of blood vessels in retina fundus images. In this report, we compare our segmentation results on three well-cited datasets: STARE [1], DRIVE[2], and CHASE [3]. We implement our methods while experimenting with different preprocessing and model optimization techniques. The U-Net method significantly out-performed the non-deep learning method on the Jaccard and Dice metric scores on a combined dataset.

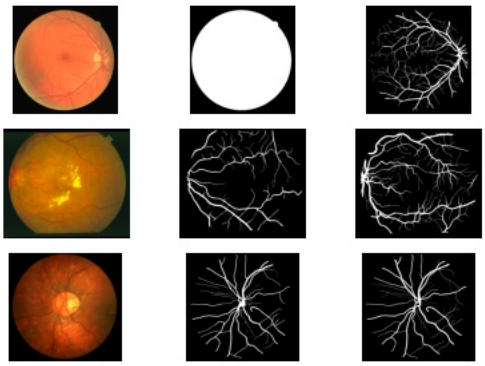

Figure 1: Sample raw images, mask and ground truth(s) from DRIVE (top row), STARE (middle row), and CHASE (bottom row).

1 Project Idea

Assessment of the retinal fundus image can be informative for diagnosing the severity of diseases such as diabetes retinopathy. However, manual annotation and inspection of retinal fundus images can be time-consuming and tedious. Thus, using automated systems to process these images could be helpful to draw conclusions in a high-throughput manner.

A fundamental aspect of the diagnosis are the distribution and characteristics of the blood vessels. As such, the blood vessels must be segmented accurately. The main difficulty in segmenting these images lie in identifying extremely fine blood vessels in images that are unevenly lit and maybe be compounded by blurry areas when disease is evident. Furthermore, a high segmentation accuracy is desired due to the medical nature of the analysis. Several of methods using geometric operators, machine learning or deep learning models already exists, but there is limited literature explicitly benchmarking deep learning and non-deep learning methods for blood vessel images in

retinal fundus images. While deep learning models have often been noted to have a high accuracy in segmenting images, these models also have some drawbacks when compared to geometric methods.

This project aims to design two methods – one using common geometric and machine-learning methods and one using a deep learning model – and to benchmark these two different methods. We aim to explore the advantages and disadvantages of non-deep learning and deep learning methods in the context of analyzing retinal images.

2 Background

Retinopathy refers to retinal disease, primarily due to retinal vascular abnormalities such that there is insufficient blood flow for proper vision [4]. Retinopathy can be a complication seen in different diseases such as diabetes, hypertension, and kidney disease. Detection is important because retinopathy can lead to blindness. Retina fundus images are often used during the diagnosis and monitoring of disease progression for eye diseases [5]. In these images, the distribution and morphology of the blood vessels are frequently indicative of the severity of the disease. However, the inspection of such images can be time-consuming. This process can be made more efficient by using computer-aided diagnostics and automating the inspection of blood vessels in these images in order to properly diagnose patients and study disease progression [6].

Benchmarking is an important process for analyzing segmentation methods. Deep-learning methods for the segmentation of bioimages have become increasingly popular [7], including the convolutional neural networks U-Net structure proposed by Ronneberger *et al*. in 2015 [8] for the purposes of biomedical image segmentation; however, for simpler segmentation tasks there still exist faster, non-deep learning methods which perform fairly accurately which is particularly important to consider when computational resources are limited. The datasets used in this project serve as a particularly useful benchmark because we can compare our methods and results to other impactful studies which use the same data for the same task.

Fundus images are important for the problem of vascular segmentation because they are the only non-invasive visualizations of human blood vessels. Studying the retinal vascular structure is a historic example of an advanced application in biological image analysis. Traditional methods include ridge-based morphological methods which consist of extracting features with special insight into the structures interest [9], and naturally, deep learning based methods have become increasingly popular as well [10]. In a recent 2022 *nature* study, the authors demonstrate excellent performance of this vessel segmentation task using a minimalistic U-Net like model on 20 fundus datasets [11]. Here, we present our project results and discuss this particular segmentation task in the context of model complexity.

3 Methods

3.1 Data

We used three publicly available datasets for this project – Structured Analysis of the Retina (STARE) [1], Digital Retinal Images for Vessel Extraction (DRIVE) [2], and Child Heart and Health Study in England (CHASE) [3]. The three datasets were acquired with different methods, orientations as well as under varying lighting conditions. Examples of raw images are shown in Figure 1 and these differences are summarized in Table 1. Notably, the STARE dataset is more rectangular than DRIVE and CHASE, and the CHASE images are taken such that the optic disc is centered. A combined dataset was created in order to standardize the available data, particularly for the deep-learning method as well as to compare across methods.

In accordance with the literature, we use the green channel from the input images for our methods. We normalized the input images by dividing by the maximum intensity pixel value. STARE and CHASE datasets come with 2 sets of manually labeled ground truth masks, so we randomly selected one label for each sample.

We performed standard image preprocessing including reshaping and resizing to 256 x 256 pixels–the size required for many deep learning model architectures used for transfer learning. We also perform local histogram equalization methods in order to enhance the contrast between the

Table 1: Details of each chosen dataset.

Dataset	Image dimensions	Number of images	Number of observers
DRIVE	584 x 565 x 3	20	1
STARE	605 x 700 x 3	20	2
CHASE	960 x 900 x 3	28	2
Combined	256 x 256 x 1	68	1

Table 2: Features calculated for each identified region and a brief description.

Feature	Description
Area	Area of line segment
Eccentricity	Ratio of the focal distance and the length of the major axis
Orientation	Orientation of region in the image
Length of major axis	Length of the major axis
Perimeter	Perimeter of region
Max intensity	Maximum intensity of region
Mean intensity	Mean intensity of region
Skeleton length	Length of region skeleton
Width	Mean width of region
Circularity	4π Area/Perimeter^2 of region

vessels and the eye tissue background–another standard preprocessing step for this task.

3.2 Models for segmentation

3.2.1 Ridge-based morphological operators and classifier

The workflow of the geometric and machine-learning method is shown below in Fig 1.

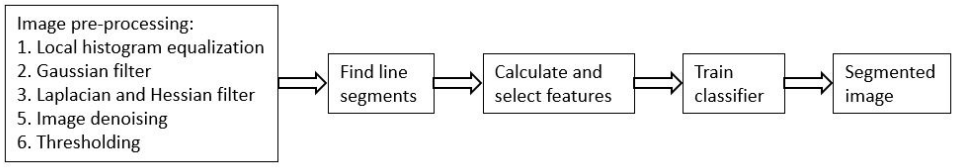

Figure 2: Workflow of the non-deep learning method

Briefly, the image was processed using local histogram equalization as mentioned earlier, smoothed using a Gaussian filter, then processed using a Hessian and a Laplacian filter ($\sigma = (1, 2, 2)$ or $ksize = 3$). Because many of these filters were sensitive to the size and resolution of the image, the size of the filters were scaled proportionately to the dimensions of each dataset. The scaled output of the Hessian was subtracted from the output of the Laplacian. The image was then denoised. The image was then thresholded using local thresholding and connected line segments are found. The line segments are then assessed to be either vessels or not blood vessels by calculating the features (shown in Table 2) for each segment. The features were selected using Sequential Forward Selection (SFS) and classified using either K-nearest Neighbor (KNN), Support Vector Machines (SVM) or Decision Trees (DT). The final prediction mask is then pruned to remove small line segments. This method is similar to that used by Staal *et al* (2004) [2], Jiang *et al* (2017) [12] and Li *et al* (2009) [9].

For the original DRIVE, STARE and CHASE datasets, 5 images were used for each to train the classifiers and the rest of the images were used to test the model. For the combined dataset, 40 images were used to train the model and 14 images were used to test the model.

3.2.2 U-Net

The U-Net structure uses consecutive convolution and max-pooling steps in the contracting path, followed by consecutive steps of up-sampling and convolution in the expanding path [8]. We implemented variations of the U-Net architecture with Adam optimizer using keras and pytorch.

We constructed two models with varying complexity. Our first model followed the original architecture presented by Ronneberger *et al*. [8] with 4 encoding blocks and a bottleneck layer with 1024 channels in the feature map, and required center-cropping the encoder feature maps before concatenation during decoding. We implemented this model in pytorch. We also implemented another model in keras that followed a small-U-Net architecture with only 2 encoding blocks and a bottleneck layer with 256 channels in the feature map. In each case, our last step in our decoder is a sigmoid activation function. For our final segmentation prediction, we set a threshold at 0.5 in order to classify each pixel as foreground or background.

We used built-in binary cross entropy loss functions and implemented custom dice loss. We also experimented with various preprocessing steps. We varied sizes 128x128, 256x256, 512x512, and 572x572, as well as performing local, global or no histogram equalization. Global histogram equalization was performed using the equalize_hist package from the skimage.exposure library; local histogram equalization was performed with the equalize package from the skimage.filters.rank library with a disk shape 1/20th the size of the original image. In the results presented here, we trained the small-U-net model for 50 epochs on our retina dataset with image size 256x256 with no histogram equalization.

3.3 Evaluation

We evaluated our model using various metrics including loss, accuracy, AUC, intersection over union (IOU; also known as Jaccard Index) and Dice scores. For both approaches, we report the best obtained results according to these metrics and compare performances.

4 Experiments and Results

4.1 Ridge-based morphological operators and classifier

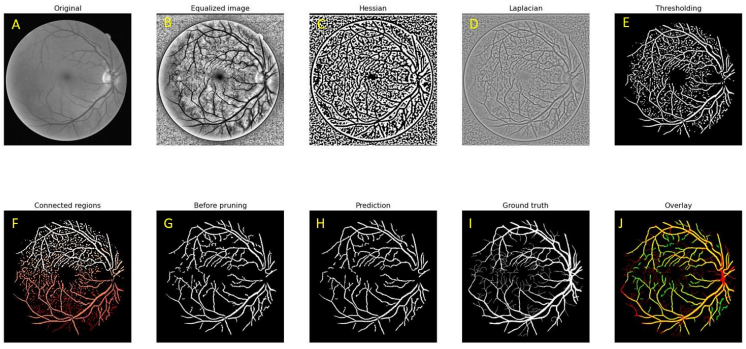

Figure 3: Figure showing the intermediate output following the workflow shown in Fig.2. Fig.3J shows the overlay of the prediction (in green) and the ground truth (in red).

Fig. 3 shows several images from the intermediate image processing steps and the final prediction. The segmentation of blood vessels can be considered quite distinct from many other challenges that aim to segment round, convex shapes. It required extensive use of ridge operators and some commonly-used methods such as watershed did not produce a high accuracy. Both the Hessian (Fig. 3C) and Laplacian (Fig. 3D) operators were used to extract the ridge features, as the Hessian was better at identifying large vessels and overlapping tubes while the Laplacian was better at identifying capillaries. On the flipside, the Hessian produced larger noise particles compared to the Laplacian operator.

Table 3: Number of features selected and evaluation scores for each classifier on the DRIVE dataset.

Classifier	No. of features	Accuracy	AUC	Jaccard	Dice
KNN	5	0.954 ± 0.007	0.819 ± 0.033	0.535 ± 0.046	0.696 ± 0.042
SVM	5	0.953 ± 0.008	0.802 ± 0.036	0.524 ± 0.051	0.686 ± 0.048
DT	5	0.951 ± 0.016	0.792 ± 0.092	0.491 ± 0.149	0.641 ± 0.185

Table 4: Final evaluation scores for each dataset, including the pixel-wise accuracy, area under the curve (AUC), Jaccard score and Dice score.

Dataset	Accuracy	AUC	Jaccard	Dice
DRIVE	0.954 ± 0.007	0.819 ± 0.033	0.535 ± 0.046	0.696 ± 0.042
STARE	0.953 ± 0.010	0.875 ± 0.043	0.519 ± 0.052	0.682 ± 0.047
CHASE	0.958 ± 0.003	0.819 ± 0.020	0.490 ± 0.039	0.657 ± 0.036
Combined	0.959 ± 0.008	0.768 ± 0.035	0.395 ± 0.057	0.564 ± 0.061

Line-based and pixel-based classification
Classification by a per-pixel basis was tested against classification by line segments. However, the classification by pixel even for the smallest dataset, DRIVE, was found to be prohibitively slow. Thus, classification by line segment was a more suitable choice given the current constraints.

KNN, SVM and DT
3 different ways of classifying the line segments were tested, namely KNN, SVM and DT. All were tested on the DRIVE dataset coupled with SFS to select the features. It was found that the KNN (n_neighbors = 7) performed slightly better than the SVM and ran much faster. The DT (max_depth=4) had one outlier that performed exceptionally badly; without this outlier, the mean for the Accuracy, AUC, Jaccard and Dice were 0.954, 0.812, 0.525 and 0.686 respectively, which is close to the performace of the other two. KNN was used for all datasets.

Evaluation

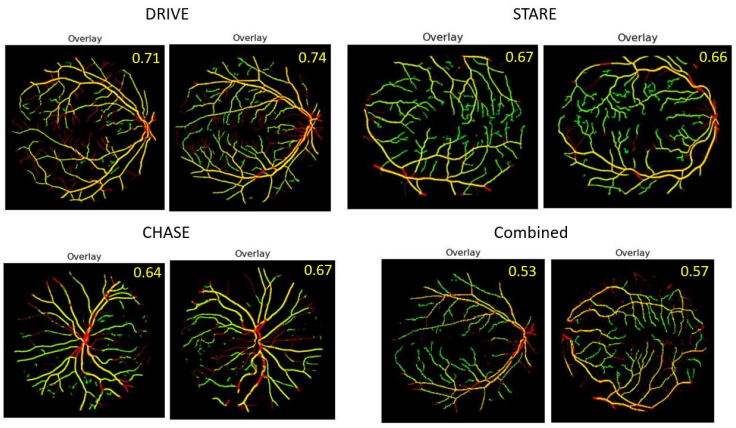

Figure 4: 2 representative segmented images for each dataset. The ground truth is shown in red and the final prediction is shown in green. The Dice score is shown in the top right corner.

The final scores across the four evaluation metrics are shown in Table 4. The non-deep learning method was found to have a high pixel-accuracy and a satisfactory AUC score. However, it performed very poorly according to the Dice and Jaccard scores. This discrepancy may be due to the difference in prevalence of the two classes – the retinal fundus images are sparse and contain predominantly background pixels, skewing the pixel-wise accuracy. In the final overlaid images in Fig. 4, it can be seen that the largest blood vessels were mostly accurately segmented. However, algorithm struggled with the finer capillaries in the middle of the fundus image.

The reliance on image textures meant that this method is not robust to image variability, for instance, resizing and cropping and is specific to a standardized image. This can be most clearly seen in the poor results of the combined dataset in Table 4. This method also performed the best when large, full-resolution images are used as it utilized specific neighbourhood features to obtain the vessel segments. The resized combined image was too small even for the minimum-sized filters, resulting in a drastic loss of sensitivity to the image textures.

4.2 U-Net

For our deep-learning based methods, we implemented two variants of the U-Net architecture and set a threshold of 0.5 for classifying each pixel as foreground or background. We found that the small-U-Net model is advantageous to the original U-Net architecture for this particular segmentation task. The advantages mainly come from computational complexity, as well as IOU scores about 2% higher. Each epoch ran in about 4 seconds using a graphic card with 1024 cuda cores (NVIDIA GeForce GTX 1650). With this small-U-Net, we found that training for 50 epochs resulted in segmentation results with an IOU score of about 52.5% (Table 5).

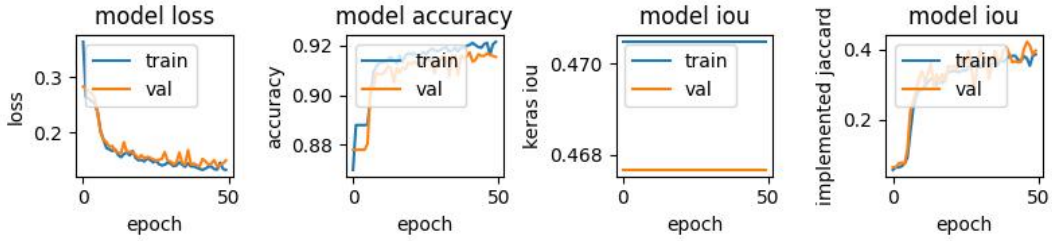

Figure 5: Deep-learning model training performance. As expected, loss decreases during training while accuracy and IOU increase. Note that the meanIOU metric in the keras.metrics library has a bug.

We would like to note that the meanIOU metric in the keras.metrics library has a critical issue: the iou item in the history that is returned from the model.fit operation is constant (Fig 5 middle right). To properly visualize how the IOU was changing during training, we implemented a custom metric. Using this, we were able to capture the appropriate IOU trend (Fig 5 far right).

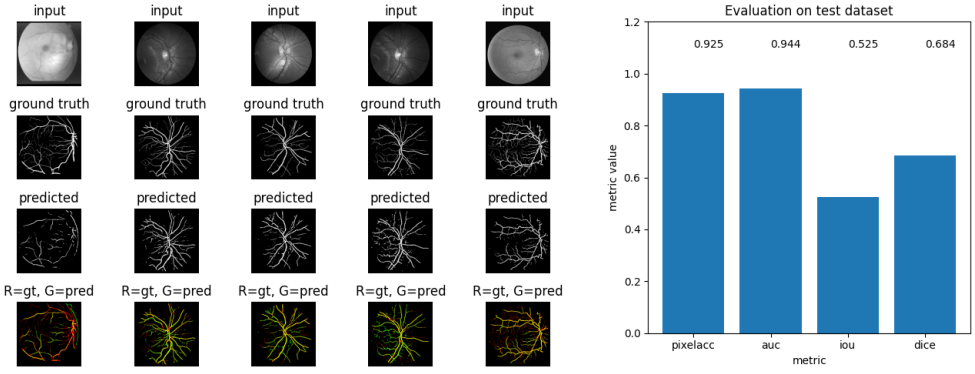

Figure 6: Testing performance of deep-learning with small-U-Net model and preprocessing image size 256x256. **Right**) Segmentation results from 5 sample test images. **Left**) Evaluation metrics using test dataset.

Segmentation results on the testing dataset are shown in Figure 6. The results seem to perform well qualitatively, but it would be quantitatively better if the IOU score was higher. Qualitatively, there seem to be false negatives where there are small vessels near the extremities of the structures. Interestingly, if the model is trained for a shorter period of time, the segmentation results are even more

Table 5: Final evaluation scores for the test dataset under various DL model training conditions.

Training conditions	Accuracy	AUC	Jaccard	Dice
small-U-Net in keras 256x256 input no contrast enhancement BCE loss	0.925	0.944	0.525	0.684
small-U-Net in keras 256x256 input contrast enhancement BCE loss	0.923	0.944	0.507	0.670
original 2015 U-Net in pytorch 256x256 input no contrast enhancement BCE loss	0.935	0.874	0.508	0.662
original 2015 U-Net in pytorch 256x256 input no contrast enhancement Dice loss	0.088	0.500	0.088	0.160
original U-Net in pytorch 512x512 input no contrast enhancement BCE loss	0.944	0.857	0.526	0.685

truncated at the extremities. This suggests that the model learns how to branch out and longer training time may result in a more refined ability. Results from our different models and preprocessing steps are summarized in Table 5 as well as provided in the Appendix.

In the context of computational complexity, smaller-U-Net is preferred for smaller input image sizes and when computational resources are limited. Interestingly, we found that our small-U-Net model implemented in keras had equal running time compared to the larger model which we implemented in pytorch ($\sim$4 seconds per epoch). When comparing image input sizes 256x256 vs 512x512, the runtime of the original 2015 U-Net model increased from $\sim$4 seconds per epoch to $\sim$18-21 seconds per epoch.

4.3 Comparison

Several advantages and disadvantages are summarized in Table 6. In the context of retinal fundus images, high image variability is expected. Even though the image has a standardized format, the condition of the eye may obfuscate details due to its diseased state. This gives the U-Net method a distinct advantage over the non-deep learning method in terms of adapting to variable images. In addition, analysis of medical images have a higher accuracy requirement, which is better fulfilled by the U-Net as long as the user has a sufficient resources to train and deploy the learning model. The U-Net produced scores of 0.525 (Jaccard) and 0.684 (Dice) compared to the non-deep learning method which produced significantly lower Jaccard and Dice score of 0.395 and 0.564 respectively. The non-deep learning method also had difficulty in matching the exact width of the blood vessels and the prediction was often too thin for larger vessels but too wide for thin capillaries. This is a significant drawback of this method.

The main advantage of ridge operators is that it requires a very small dataset, and if the dataset has similar images, it can perform very well. In the context of retinal fundus images, it may not always be possible to get datasets large enough to train the U-Net. The U-Net also requires ground truth labels for the entire training dataset, which are time-consuming to create. The ridge operator method also requires the ground truth due to the classifier, but it is possible to remove that and achieve only slightly worse accuracy scores.

Table 6: Advantages and disadvantages of the non-deep learning method versus the U-Net method.

Description	Ridge operators with classifiers	U-Net
Evaluation metrics	Average and acceptable only for large blood vessels	Higher in terms of the Dice and Jaccard scores
Robustness to image variability and noise	Not robust	Robust to image changes if model was trained to generalize
Size of datasets required	Small dataset	Requires large traning dataset
Design, implementation and debugging	Driven by matrix transformations. Debug easily by viewing intermediate steps	Depends on model architecture May not be easy to debug if architecture is complex
Training time and resources	Fast, little resources required	Slow, requires significant CPU/GPU power to train the model
Testing time and resources	Fast, few resources needed	Fast, few resources needed
Requires ground truth	only for classifier	Yes

5 Conclusions

In this study, we compare traditional morphological based methods and deep-learning methods for vessel segmentation. We find that both methods reach relatively high accuracy scores; however, we note that accuracy is not an appropriate metric to evaluate the performance of this task because it will be artificially inflated due to the class imbalance of background to foreground pixels.

The non-deep learning method using ridge operators were able to segment the larger blood vessels acceptably well, but failed at identifying smaller capillaries. It also does not adapt well to image variability and lower resolution images, as shown by the accuracy scores on the combined dataset. However, the non-deep learning method requires fewer training images and computational resources and can perform quite well if the images are highly similar. Thus, this method would only be preferable under certain conditions, namely if the data and computational resources are limited and the images are highly standardized.

We also analyze the computational complexity of the methods used in this project. In terms of input size, larger input image sizes affected the runtime of the deep-learning model significantly, while larger images with higher resolution were preferred for the traditional methods. Smaller input image sizes limited the depth of the deep-learning model, as we were not able to run the original U-Net with 128x128 size images because the dimensions of tensor was smaller than the kernel size for the next convolutional layer.

Another reason to scrutinize input image size: the depth of the neural network also affects the computational complexity. We find that more encoding blocks not only increases the runtime per training epoch, but requires larger images and longer training times to learn all the parameters. When testing the model with the deeper architecture, we need at least 40 epochs to obtain decent results.

With our DL approach, our findings are consistent with published studies demonstrating that relatively simple methods are sufficient for this vessel segmentation task. We find that models with fewer parameters actually outperform models with unnecessary complexity [11]. In our case, our small-U-Net model had higher IOU and Dice scores by about 2% when compared to the larger original U-Net architecture. This is an important finding in our discussion around model complexity in a vessel segmentation task.

References

[1] AD Hoover, Valentina Kouznetsova, and Michael Goldbaum. “Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response”. In: *IEEE Transactions on Medical imaging* 19.3 (2000), pp. 203–210.

[2] Joes Staal et al. “Ridge-based vessel segmentation in color images of the retina”. In: *IEEE transactions on medical imaging* 23.4 (2004), pp. 501–509.

[3] Christopher G Owen et al. “Retinal arteriolar tortuosity and cardiovascular risk factors in a multi-ethnic population study of 10-year-old children; the Child Heart and Health Study in England (CHASE)”. In: *Arteriosclerosis, thrombosis, and vascular biology* 31.8 (2011), pp. 1933–1938.

[4] William L Stone et al. “Retinopathy”. In: *StatPearls* (2022). URL: https://www.ncbi.nlm.nih.gov/books/NBK541131/.

[5] Muhammad Arsalan et al. “Detecting retinal vasculature as a key biomarker for deep Learning-based intelligent screening and analysis of diabetic and hypertensive retinopathy”. In: *Expert Systems with Applications* 200 (2022), p. 117009.

[6] Stephanie J Chiu et al. “Validated automatic segmentation of AMD pathology including drusen and geographic atrophy in SD-OCT images”. In: *Investigative ophthalmology & visual science* 53.1 (2012), pp. 53–61.

[7] Erik Meijering. “A bird’s-eye view of deep learning in bioimage analysis”. In: *Computational and structural biotechnology journal* 18 (2020), pp. 2312–2325.

[8] Olaf Ronneberger, Philipp Fischer, and Thomas 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.

[9] Ying Li et al. “Ridge-branch-based blood vessel detection algorithm for multimodal retinal images”. In: *Medical Imaging 2009: Image Processing*. Vol. 7259. SPIE. 2009, pp. 1475–1486.

[10] Zhenwei Li et al. “Blood Vessel Segmentation of Retinal Image Based on Dense-U-Net Network”. In: *Micromachines* 12.12 (2021). ISSN: 2072-666X. DOI: 10.3390/mi12121478. URL: https://www.mdpi.com/2072-666X/12/12/1478.

[11] Adrian Galdran et al. “State-of-the-art retinal vessel segmentation with minimalistic models”. In: *Scientific Reports* 12.1 (2022), pp. 1–13.

[12] Zhexin Jiang et al. “Fast, accurate and robust retinal vessel segmentation system”. In: *Biocybernetics and Biomedical Engineering* 37.3 (2017), pp. 412–421.

A Appendix

Below are complete results for various deep learning conditions corresponding to Table .

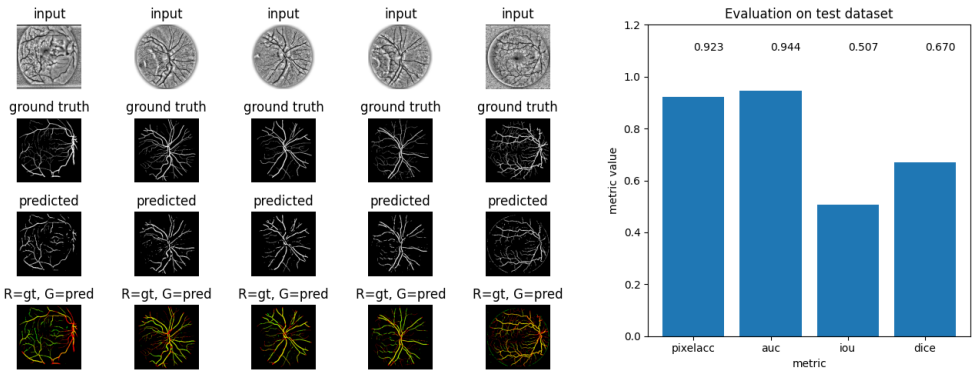

Figure 7: Testing performance of deep-learning with small-U-Net model and preprocessing image size 256x256 with local histogram equalization. **Right**) Segmentation results from 5 sample test images. **Left**) Evaluation metrics using test dataset.

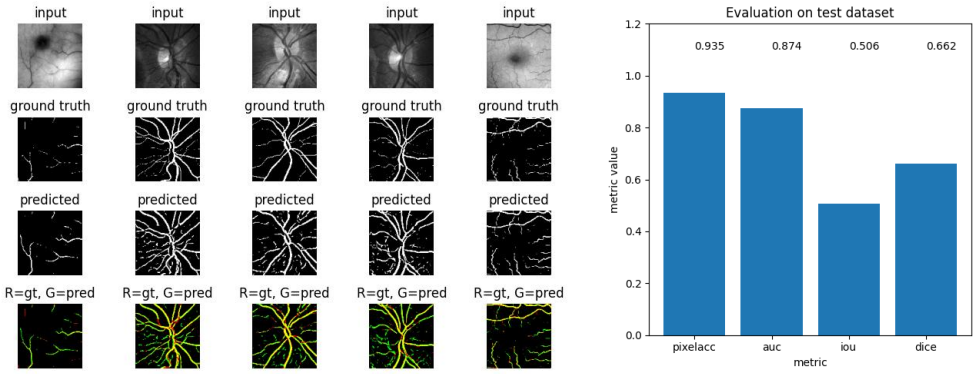

Figure 8: Testing performance of deep-learning with original 2015 U-Net model with BCE and preprocessing image size 256x256. **Right**) Segmentation results from 5 sample test images. **Left**) Evaluation metrics using test dataset.

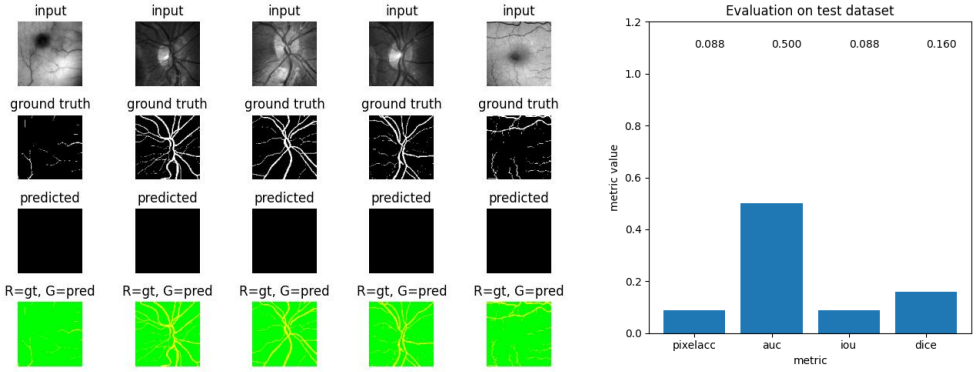

Figure 9: Testing performance of deep-learning with original 2015 U-Net model with Dice Loss and preprocessing image size 256x256. **Right**) Segmentation results from 5 sample test images. **Left**) Evaluation metrics using test dataset.

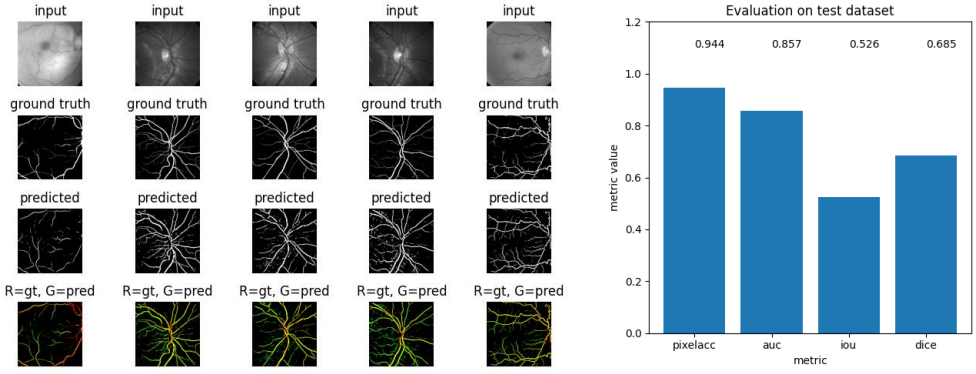

Figure 10: Testing performance of deep-learning with original 2015 U-Net model with Dice Loss and preprocessing image size 512x512. **Right**) Segmentation results from 5 sample test images. **Left**) Evaluation metrics using test dataset.