**BLOOD VESSEL SEGMENTATION**

**A U-Net segmentation model for accurate segmentation of blood vessels in retinal images.**

**Method Proposed:** A novel preprocessing flow which dramatically improves the performance of U-Net in segmenting the blood vessels from the fundus images. The features as learned by the U-Net show accurate representation of the blood vessels.

**Results**: The overall performance of our method and other state-of-the-art deep-learning based methods on DRIVE and CHASE DB1 are tabulated in this paper. Table 2 for DRIVE, 3 for CHASEDB1 respectively represents the performance comparison. The results prove that our method has displayed superior performance compared to recent state-of-the-art methods.

1. **Introduction**

Retinal blood vessels form the root of attention for the clinical diagnosis of few diseases. Realizing automatic blood vessel segmentation from raw fundus images thus is an important and challenging task especially for diseases like Diabetes Retinopathy. In this paper, a deep neural network architecture based on the Dense U-net module is proposed for retinal vessel segmentation with a set of pervasive pre-processing methods and configurations that lead to the most accurate segmentation of the blood vessels.

A significant amount of information regarding the health of a human eye is contained in the blood vessels of the retina. The examination of its vascular or morphological structure is not only a basis in the screening of vascular diseases of the retina but also preliminary in the diagnosis of various other diseases, such as in proliferative diabetes-induced retinopathy and glaucoma, and also stroke and hypertension. Characteristics such as unreliably low contrast between the vessel and retinal background make the manual segmentation task quite challenging. Additionally, high interference that occur in the lesion area plus the complexity of the vascular structure make retinal vessel segmentation a difficult approach for the diagnosis of diseases [5].

In the current times, the most reliable and prevalent method undergone for retinal vessel segmentation is the manual annotation done by doctors with a professional background and experience in handling optical medical images. Due to a advancements in the technologies used for medical imaging, the number of fundus images at hand is exponentially multiplying. Such large quantities of fundus images make it hard to be segmented and annotated one by one by the doctors manually, since, retinal vessels require lot of time and energy for accurate annotation. Moreover, different doctors adopt different segmentation standards while approaching the same fundus image. To overcome this inefficiency faced by manual annotation, lots of automatic retinal vessel segmentation algorithms have been proposed which in turn enable Computer-Aided Diagnosis (CAD). Automatic the tedious work of Retinal Vessel Segmentation can not only directly decrease the doctors’ load of work but also heavily evade the different subjective influence of different doctors’ approach on the segmentation results.

From the summarized algorithms used for vessel segmentation from fundus images employed in [6], [7], they can be broadly divided into two major categories: unsupervised and supervised algorithms. In general, unsupervised learning algorithms mainly use a few pre-set rules to extract vessel features and achieve segmentation without the requirement of manual annotations. These include matching filter-based algorithms [8], and tracking-based algorithms [10]. However, due to the vast diversity in the morphological distribution in the blood vessels, due to its nature, fixed segmentation rules often fail to match. The basis of the idea of leveraging supervised learning algorithms to train the segmentation model using fundus images with segmentation annotations and allow the model to automatically extract the vascular features to achieve vessel segmentation, such as Bayes model-based algorithms [11], support vector machine-based algorithms [12], and deep learning based algorithms [13]–[16]. However, supervised learning algorithms require huge data with manual label, which is hard to get. In recent years, deep learning-based algorithms have continued to develop and performed well in the field of retinal vessel segmentation, which have gradually become the mainstream algorithm.

Deep convolutional neural networks (CNN) have achieved near-radiologist performance in many semantic segmentation tasks in medical image analysis. Fully convolutional neural network (FCN) [4] have succeeded in semantic segmentation on Pascal VOC dataset, and U-Net [5] achieved the top accuracy in the segmentation of neuronal structures in electron microscopic stacks. Other variants of CNN achieve stateof-the-art performance on benchmark semantic segmentation tasks, such as PSPNet [6] and DeepLab [7]. Among all these variants, U-Net is the most widely used structure in medical image analysis, mainly because the encoder-decoder structure with skip connections allows efficient information flow, and performs well when sufficiently large datasets are not available.

Many variants of U-Net have been proposed: Alom et al. proposed to use recurrent convolution in U-Net [8]; Ozan et al. proposed to use attention module in U-Net to determine where to look at; Simon et al. proposed Tiramisu [9], where the convolutional layers in a U-Net are replaced with dense blocks. However, all these variants still fall into the encoder-decoder structure, where the number of paths for information flow is limited. We propose LadderNet, a convolutional network for semantic segmentation with more paths for information flow. We demonstrate that LadderNet can be viewed as an ensemble of FCNs, and validate its superior performance on blood vessel segmentation task in retinal images.

1. **METHOD**

The entire algorithm flow is shown in Figure 1. The DRIVE data consists of 40 images out of which 20 are used for training, and the rest half are segregated for testing. The training data is further split in the ratio 90:10 for using the randomly selected images as the training dataset for the model and as the validation dataset respectively used towards for parameter tuning. Then, same data augmentation rules are used for both fundus images and manual vessel labels. Further, these images, both fundus images and their respective masks, are split into patches. These patches are fed into the model and the AUC curve as well as the confusion matrix is plotted. This way, the trained retinal vessel segmentation model along with its performance data is obtained. In the testing stage, testing data is similarly extracted as patches from the original testing data of the same size as the enhanced data following the same order, and then, these patches are fed into the model to arrive at patches with segmented vessels. Finally, the obtained segmented patches are together spliced in an orderly fashion to give the segmentation results of the original image.

1. **DATA PREPROCESSING**

Before being fed into the model, the data requires certain preprocessing in order to improve the performance of the model. This is required since the fundus images are subjected to uneven lighting and noise in the dataset. To reduce such external factors, present in the retinal image in order to improve the visibility of vessels, we adopt image processing techniques such as gray-scale conversion, normalization, histogram equalization, contrast enhancement, etc. Figure 2. Shows minute vessel growth that can escape the naked eye and hence need some amount of pre-processing in order to improve the definition of the vessels in the fundus image.

**Figure 2.** Thin vessels that prove to be difficult to be segmented without pre-processing.

1. Gray- Scale conversion:



**Figure 3**. Flowchart representing the pre-processing techniques.

The input image is initially converted from a three channel RGB image into a singular channel gray image. This image is mathematically derived by splitting the three channels of the BGR image into R channel, B channel and G channel and passing these 2-D matrices into equation 1.

(1)

­The gray-scale image obtained after passing a DRIVE test image through the equation is seen in Figure 5 (b).

1. Z-score normalization:

When the pixels in an image are plotted as a normal distribution curve, Z-score gives an intuition as to where the pixel lies with respect to the mean of all the pixels in the image. A z-score of zero denotes that the value lies exactly at the mean/ average point whereas a score of +3 shows that the value is much farther from the average (probably an outlier). ‘‘Z-score normalization’’ thus is defined as the image signal intensity on a per-pixel basis and is calculated by using equation 2.

(2)

In the above equation, µ represents the mean of the pixels in the Ig and μ denotes the standard

deviation of all the same pixels. The image obtained after Z-score correction is given in Figure 5 (c).

1. CLAHE (Contrast Adaptive Histogram Equalization):

This method is a modified version of Adaptive histogram equalization (AHE). Unlike regular Histogram Equalization methods, CLAHE automatically limits itself from over-amplifying the contrast. It operates on patch-based regions on the image, called tiles, instead of the entire image. Later, bilinear interpolation is used to combine the artificial boundaries of neighboring tiles extracted from the original image. Here, we use this method to improve the contrast between the vessel boundaries and the background of the retinal image. The parameter which is passed as a threshold for contrast limiting, clipLimit, is set as 2.0 in our process. Whereas, the number by which the region along the height or the width of the image is split which is taken into account as a grid or a tile, tileGridSize, is set as 8×8. The image Ic obtained after applying CLAHE is given in Figure 5 (d).

1. Gamma – correction:

 The relationship between a pixel’s value in an image and its actual luminance in the original state is defined by Gamma. Pixels in every image possesses some brightness level, which is known as luminance. This luminance value lies between 0 and 1, where 0 means complete darkness or black, and 1 is complete brightness or white. “Gamma correction function” used is defined by:

Igamma = Ic ^ (1 / γ)

Where Ic is our input image, here, the previously enhanced image using CLAHE, and Igamma is our gamma corrected image. The output image Igamma is then scaled back to the range [0, 255]. This image is represented in the Figure 5 (e).

1. Normalization

By applying Linear Normalization, the range of the pixel values in an image is changed. This Normalization helps bring image to range that is normal to sense. In our case, the image is divided by 255.0 to change the range of the image to 0 to 1.

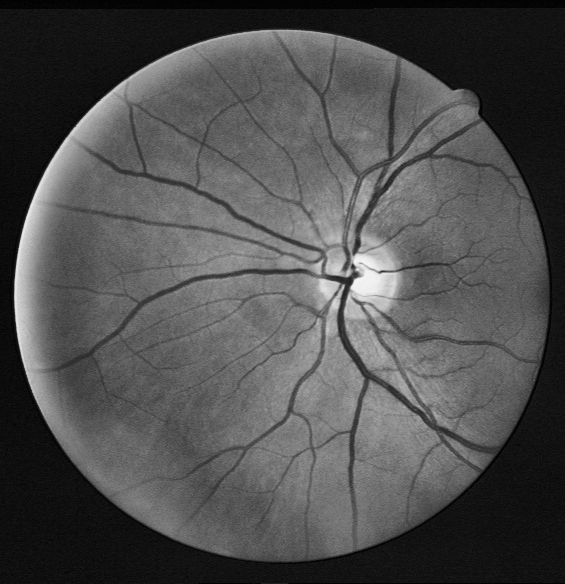
In Figure 4, we can see how the proposed methods of pre-processing has dramatically improved the visibility of the time blood vessels in the fundus image from figure 2.

**Figure 4**. The enhancement of blood vessels after preprocessing.

1. Original Image (b) Gray- scaled image Ig

(c) z-score normalized image – Iz (d) CLAHE Ic

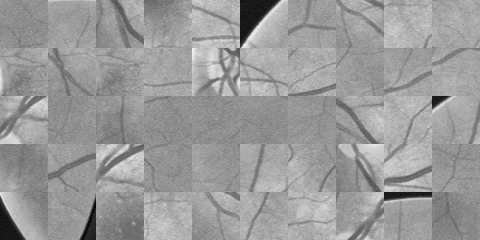


(e) Gamma-corrected image Igamma

**Figure 5.** a) to e) showcases the stages of image pre processing based on the method suggested.

1. **DATASET PREPARATION FOR INPUT:**

The pre-processed full images are not directly fed into the model. Instead, the training of the deep neural network is accomplished on patches, a small sub-part of an image, of the pre-processed full images. The height and width of the patches are assigned as 48x48. These patches are obtained by selecting random centers inside the full image. Apart from patches inside the Field of View (FOV), patches partially outside the Field of View (FOV) are also selected. This enables the neural network to learn to discriminate the blood vessels from the FOV border which seem identical.

1. Sample patches (b) Sample masks

**Figure 6**. Example of 48 \* 48 patches and their corresponding masks

A set of 1,00,000 patches is obtained by extracting 5000 patches in each of the 20 DRIVE images present in the training set randomly. Although the patches overlap, i.e., different patches may contain same part of the original images, no further data augmentation is performed. The first 90% of the dataset is used for training (90000 patches), while the last 10% is used for validation (10000 patches). Figure 6. shows a compilation of sample patches and its corresponding masks extracted from a full image in the training set. This patch extraction process is also done for test images. Here, patches of dimensions 96 \* 96 are extracted. A stride height of 16 and stride width of 16 is also set.

1. **LadderNET**

All the variants of U-Net and itself in previous literatures are all commonly known to have an encoder-decoder structure. This structure is usually shaped like a U and hence the name. Besides its dominating performance in segmentation tasks, a limited number of paths for information flow makes the learning in U-Net single-pathed. A more versatile learning can be seen in LadderNet, which is a multibranched convolutional neural network allowing for more than one path for information flow. This makes the model more suitable for semantic segmentation with a deep chained network as shown in Figure. 7. Generally, in this network, the features present in the different spatial scales are named using letters from A to E, and the columns are denoted using numbers from 1 to 4. Column 1 and 3 are named as encoder branches due to their nature, and column 2 and 4 are named as decoder branches. Convolution with a stride of 2 is used when the model transitions between small-receptive-field features to large-receptive field features (e.g., A to B). While transitioning from large-receptive field features and use transposed convolution with a stride of 2 to go from large-receptive-field features to features (e.g., B to A). The number of channels is doubled from one level to the next level (e.g., A to B).

As seen in Figure 7. visualization of LadderNet reveals that it is after all a chain of U-Nets. In our model, 2 U-Nets can be seen: one made of Columns 1 and 2 and the other made of Columns 3 and 4. Between two U-Nets, skip connections at levels A-D can be seen. Unlike U-Nets, LadderNet differs at this step where the features from the encoder branches (columns 1 and 3) are directly concatenated with the features from the decoder branches (columns 2 and 4). Thus a summed up vector of all the features from both the branches respectively is obtained. Examples of multiple paths of information flow in the used LadderNet can be seen:

(1) A1 → A2 → A3 → A4,

(2) A1 → A2 → A3 → B3 → B4 → A4,

(3) A1 → B1 → B2 → B3 → B4 → A4,

(4) A1 → B1 → C1 → D1 → E1 → D2 → C2 → B2 → A2 → A3 → A4.

Each of the above paths can be viewed individually as a variant of Fully Connected Network. Obviously, an exponential growth in the total number of paths along with the increase in the number of encoder decoder pairs and the depth of the model, that is, spatial levels (e.g., A to E in Fig. 7). This explains why LadderNet has an advantage over single path U-Nets and its variants in its potential to capture more complicated features and therefore perform at a higher accuracy.

A natural follow-up question is how with an increase in encoder-decoder branch increases the number of parameters and eventually make the training more tedious and slower. To overcome this shortcoming, a shared-weights residual block system is introduced. This block is different from a standard residual block Figure 8. in the fact that here, the same weights are shared by two convolutional layers in the same block. Much like the recurrent convolutional neural network (RCNN) where two convolutional layers present in the same block is viewed as one recurrent layer, except that, in LadderNet the batch normalization layers within a block are different. To avoid the problem of overfitting in this model, a drop-out layer is introduced between two convolutional layers. Even with much fewer parameters compared to a standard residual block, a shared-weights residual block combines the strength of skip connection, recurrent convolution as well as drop-out regularization to produce much higher segmentation results.

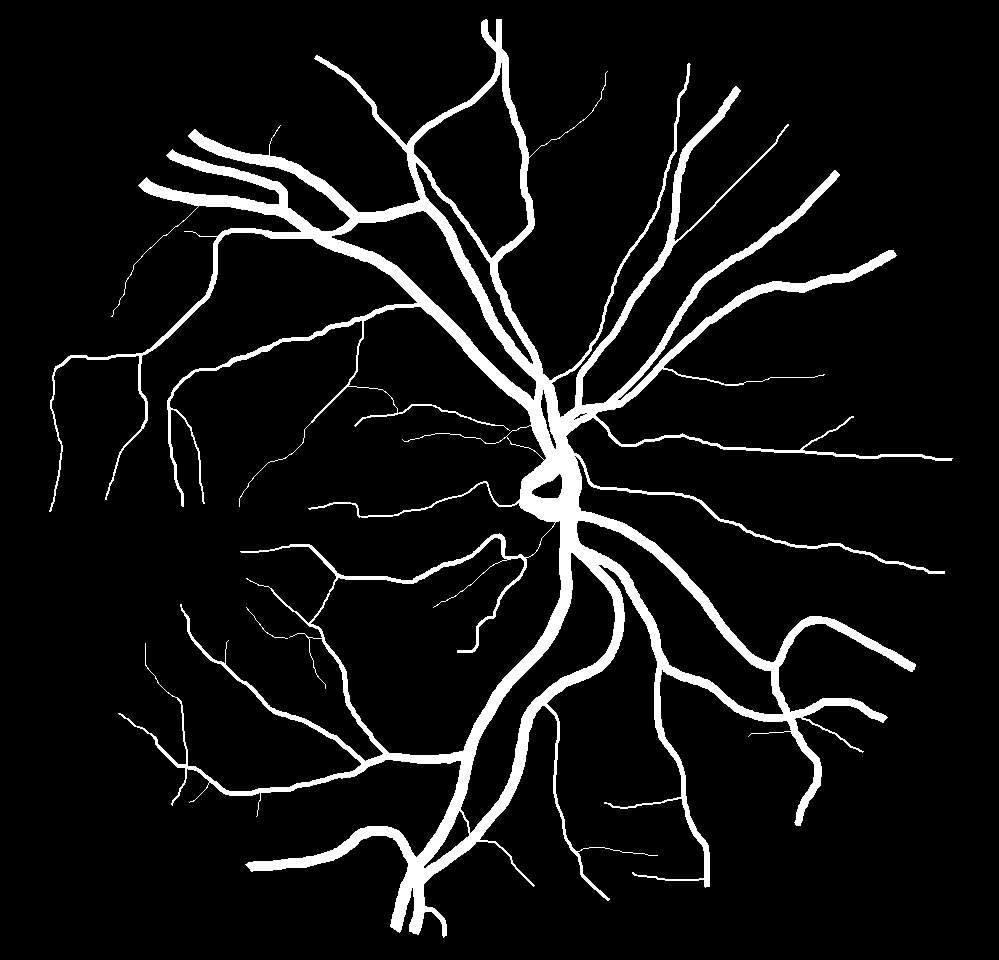


**Figure 8**. Shared weight- residual block system

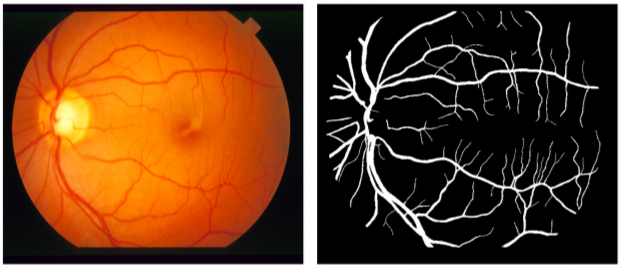
1. **EXPERIMENT**
   1. **Dataset**

The used LadderNet model was evaluated on two datasets which are very well-known for blood vessel segmentation in fundus images: the STARE dataset and the DRIVE dataset. The DRIVE dataset Figure 9 (c) consists a total of 40 color fundus images of retina, 20 of which were used up for training purposes and the remaining 20 images was used for testing. The images are of the dimensions 565×584 pixels. In order to increase the number of training samples for better performance of the model, 1,00,000 patches were randomly sampled from the training images, and 10% of the training samples got used as validation data.

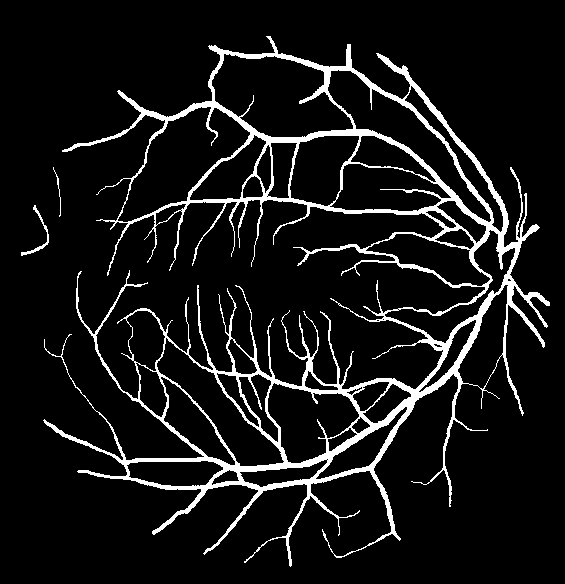
The CHASE DB1 dataset Figure 9 (a) consists of fundus images of both left and right eyes of 14 school children. It consists of a total of 28 color images of the retina, 20 of which were taken separately for training and the other 8 images (from 4 children) for testing. The dimensions of each image is 996 × 960. 760,000 patches of size 48 × 48 were randomly sampled from the training images, and 10% of the training samples was used for validation purpose. All patches underwent pre-processing before training as well as testing. Field of view (FOV) is used for the DRIVE dataset but not the CHASE DB1 dataset due to the unavailability of fundus mask.

1. Example of CHASE DB1



1. Example of STARE

1. Example of DRIVE

**Figure 9.** DATASET used

* 1. **Experimental Setup**

Training and testing of the proposed network was implemented by PyTorch 1.7.1 with CUDA 11.2 as the GPU backend. The experiments are conducted on Google Collaboratory in cloud. As described before, 5000 patches are randomly extracted from each image in the dataset and the total number of training images sums up to 1,00,000 from the 20 images in the DRIVE dataset. For STARE dataset Figure 9 (b), 128000 patches are obtained for training, as the original image set was divided into a 16-image training set and a 4-image test set. In the experiments, 90% of these extracted patches are used for training while the remaining 10% are used for validation. For LadderNet, DRIVE dataset is training on the model for 25 epochs, with a batch size of 64 patches.

* 1. **Evaluation Metrics**

The performance of our model enhanced by pre-processing is checked using the following metrics that are very prevalent. Accuracy (ACCU), Sensitivity (SENS), Specificity (SPEC), and F1-score (F1) are calculated and cross-verified with other models. To compute the above metrics:

Let,

TP denote correct positive predictions

FP denote incorrect positive predictions

TN denote correct negative prediction

FN denote incorrect negative prediction

Therefore,

P = TP + FN denotes total number of positives in the dataset

N = TN + FP denotes total number of positives in the dataset

We further calculated the precision-recall curve and receiver operating characteristics (ROC) curve. The area under the ROC curve was used as our final metric for evaluation.

1. **RESULTS**

Using Adam optimizer with a learning rate of 10-4, the model was trained for 25 epochs.

The training data is tabulated in Table 1.

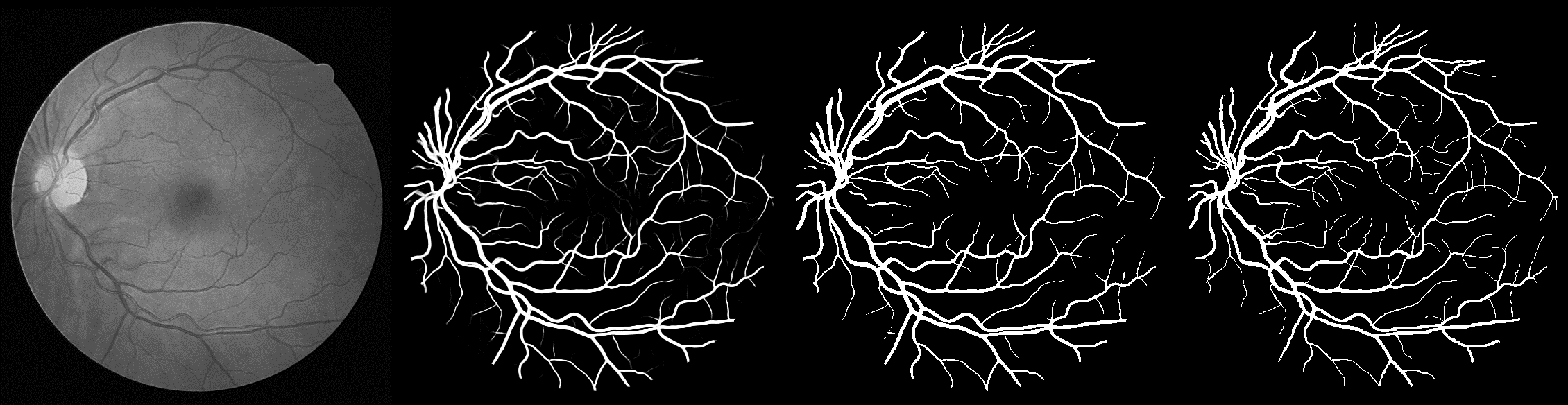
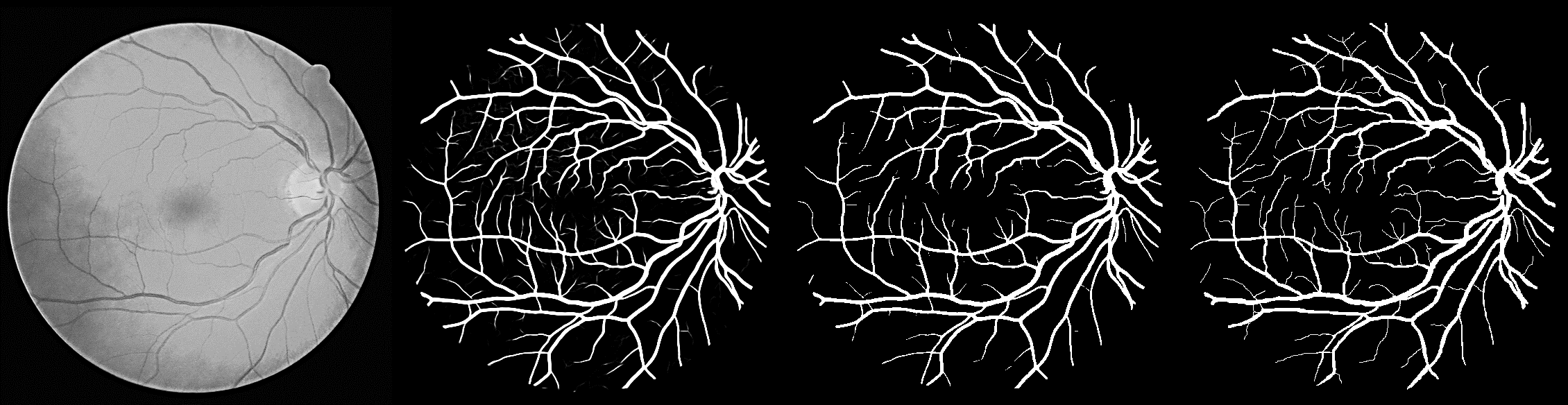
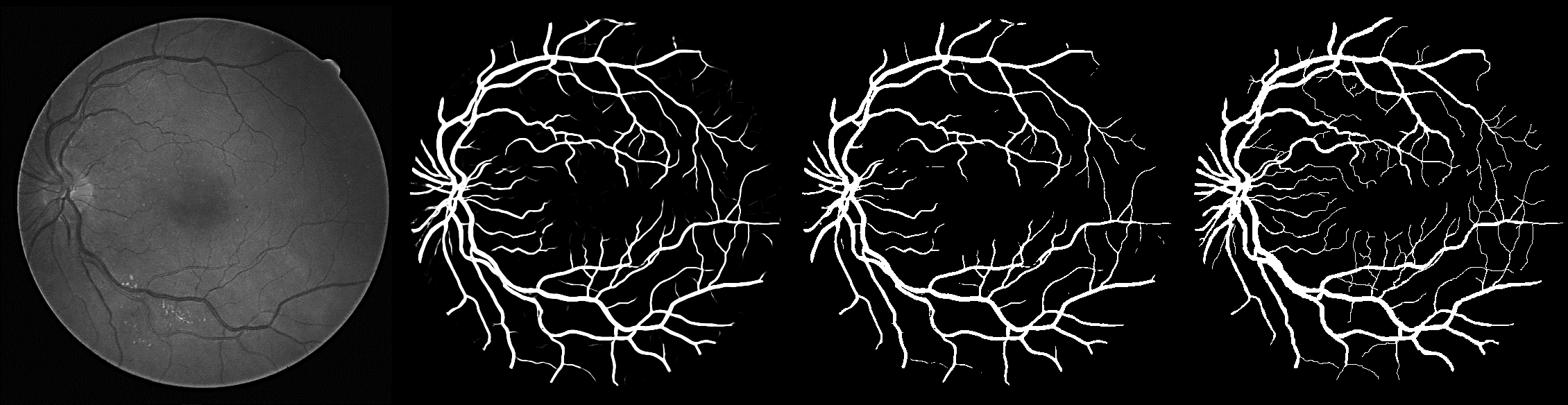
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Epoch** | **Train\_Loss** | **Val\_AUC\_ROC** | **Val\_F1** | **Val\_ACCU** | **SENS** | **SPEC** |
| 0 | 0.125312 | 0.977679 | 0.816315 | 0.954579 | 0.792917 | 0.978158 |
| 1 | 0.122715 | 0.978394 | 0.816967 | 0.955324 | 0.783317 | 0.980411 |
| 2 | 0.12065 | 0.97864 | 0.817655 | 0.955505 | 0.783737 | 0.980558 |
| 3 | 0.119086 | 0.979241 | 0.820542 | 0.955672 | 0.796167 | 0.978936 |
| 4 | 0.117645 | 0.979713 | 0.823767 | 0.955488 | 0.817301 | 0.975643 |
| 5 | 0.116572 | 0.979962 | 0.821736 | 0.956303 | 0.791223 | 0.980381 |
| 6 | 0.115547 | 0.980054 | 0.823024 | 0.956213 | 0.799877 | 0.979015 |
| 7 | 0.114614 | 0.980278 | 0.824173 | 0.956401 | 0.802773 | 0.978808 |
| 8 | 0.113849 | 0.980281 | 0.82374 | 0.956441 | 0.79965 | 0.979309 |
| 9 | 0.113033 | 0.980562 | 0.823676 | 0.956587 | 0.796624 | 0.979918 |
| 10 | 0.112425 | 0.980613 | 0.824707 | 0.95657 | 0.80261 | 0.979026 |
| 11 | 0.111841 | 0.980564 | 0.824606 | 0.956524 | 0.802906 | 0.97893 |
| 12 | 0.111292 | 0.980812 | 0.824608 | 0.956778 | 0.798236 | 0.979901 |
| 13 | 0.110787 | 0.980995 | 0.825776 | 0.95687 | 0.803 | 0.979313 |
| 14 | 0.110266 | 0.980958 | 0.826575 | 0.956755 | 0.80963 | 0.978214 |
| 15 | 0.109935 | 0.981093 | 0.825956 | 0.956965 | 0.802248 | 0.97953 |
| 16 | 0.10947 | 0.981 | 0.825024 | 0.956997 | 0.796466 | 0.980411 |
| 17 | 0.109056 | 0.9812 | 0.826139 | 0.956952 | 0.803512 | 0.979331 |
| 18 | 0.108771 | 0.981059 | 0.826091 | 0.957055 | 0.801308 | 0.979771 |
| 19 | 0.108427 | 0.981204 | 0.827335 | 0.95697 | 0.809907 | 0.978419 |
| 20 | 0.108075 | 0.981065 | 0.826878 | 0.956942 | 0.807842 | 0.978689 |
| 21 | 0.107787 | 0.981228 | 0.825986 | 0.957021 | 0.80136 | 0.979724 |
| **22** | **0.107427** | **0.981297** | **0.826557** | **0.957246** | **0.800351** | **0.980129** |
| 23 | 0.107217 | 0.980927 | 0.825871 | 0.956955 | 0.801958 | 0.979561 |
| 24 | 0.106966 | 0.980955 | 0.827072 | 0.956787 | 0.811853 | 0.977926 |
| 25 | 0.106759 | 0.981075 | 0.828667 | 0.956981 | 0.817308 | 0.977353 |

**Table 1** The training metric for every epoch. Highlighted epoch showcases the best epoch with AUC\_ROC = 0.981297

The patches after testing are spliced to form a full image. This image from Figure 9 looks remarkably close to the mask standing testimony to the accuracy of the model. For both, training on DRIVE dataset as well as CHASEDB1, the area under the AUC curve is above 0.98000. The binary segmentation was generated using a threshold of 0.5.

**DRIVE:**

For DRIVE dataset, this model exceeded all performance expectations as shown in Figure 9-10.

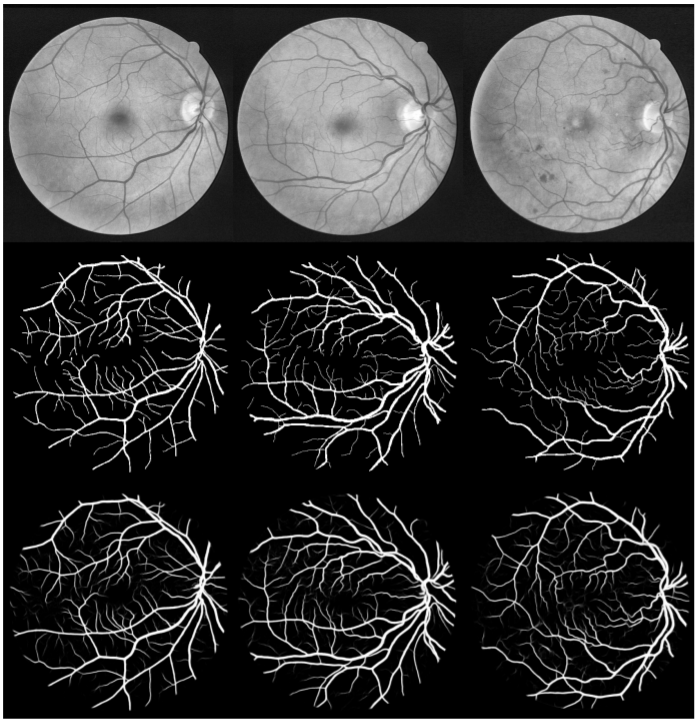
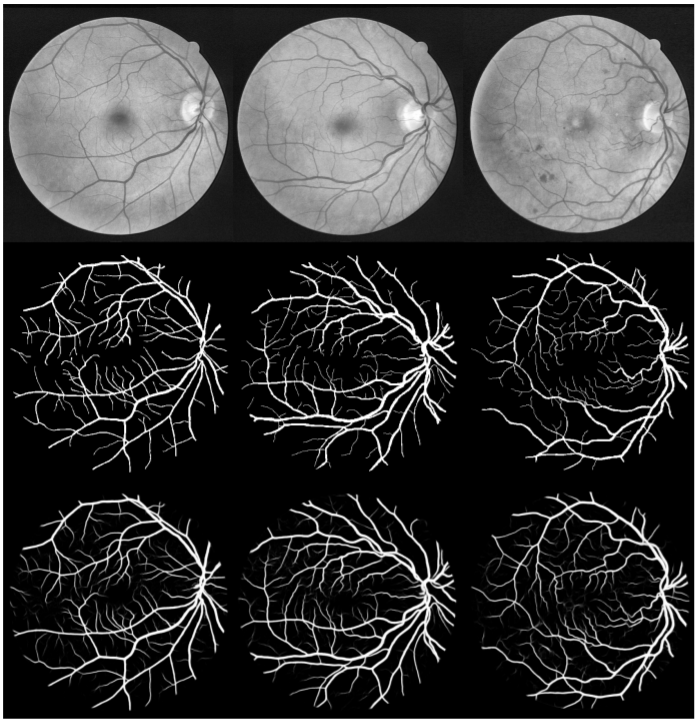
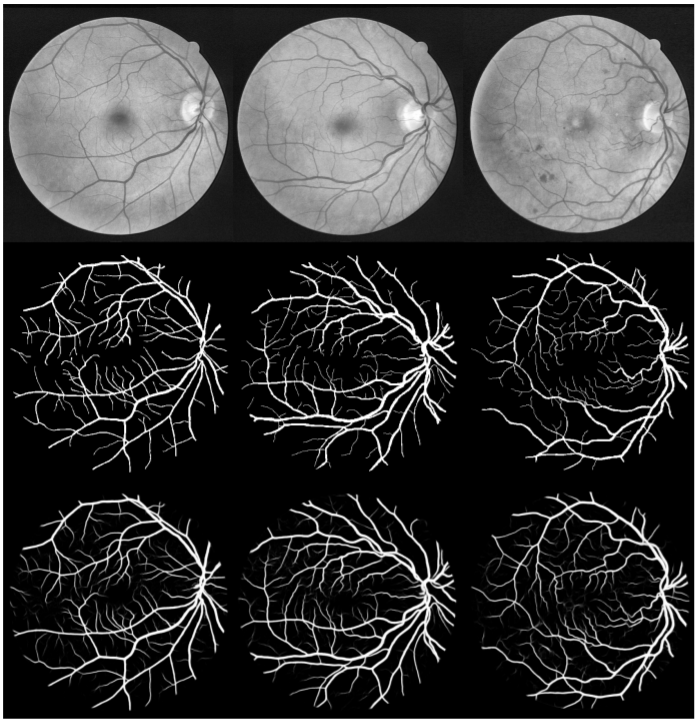


1. (b) (c) (d)

**Figure 9**. (a) Original Image (b) Probability (c) Binary (d) Ground truth.

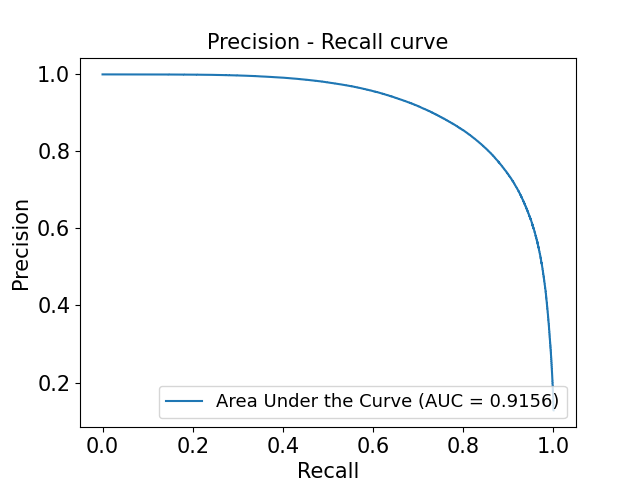
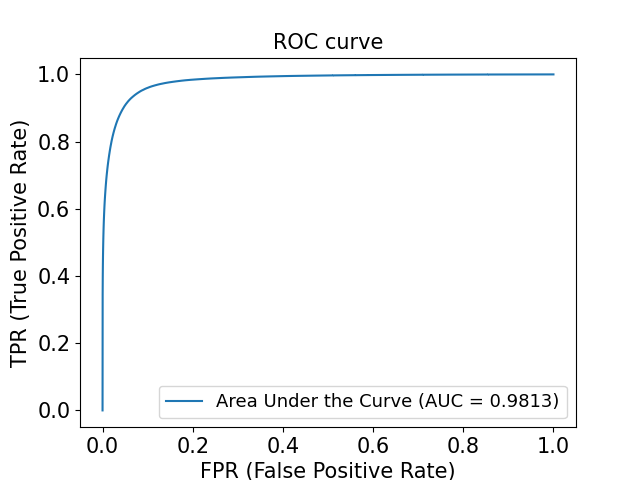
**Test results on DRIVE dataset.**

**CHASEDB1:**

**Figure 11**. Top – to – bottom (a) Original Image (b) Ground Truth (c) Prediction

**Test results on CHASEDB1 dataset.**



1. (b)

**Figure 10.** (a) ROC curve (b) Precision recall curve on DRIVE dataset



1. (b)

**Figure 12.** (a) ROC Curve (b) Precision recall curve on CHASEDB1

**Figure 11-12** showcase the test results on CHASEDB1 dataset.

**STARE:**

Figure 13. shows the test results on STARE database

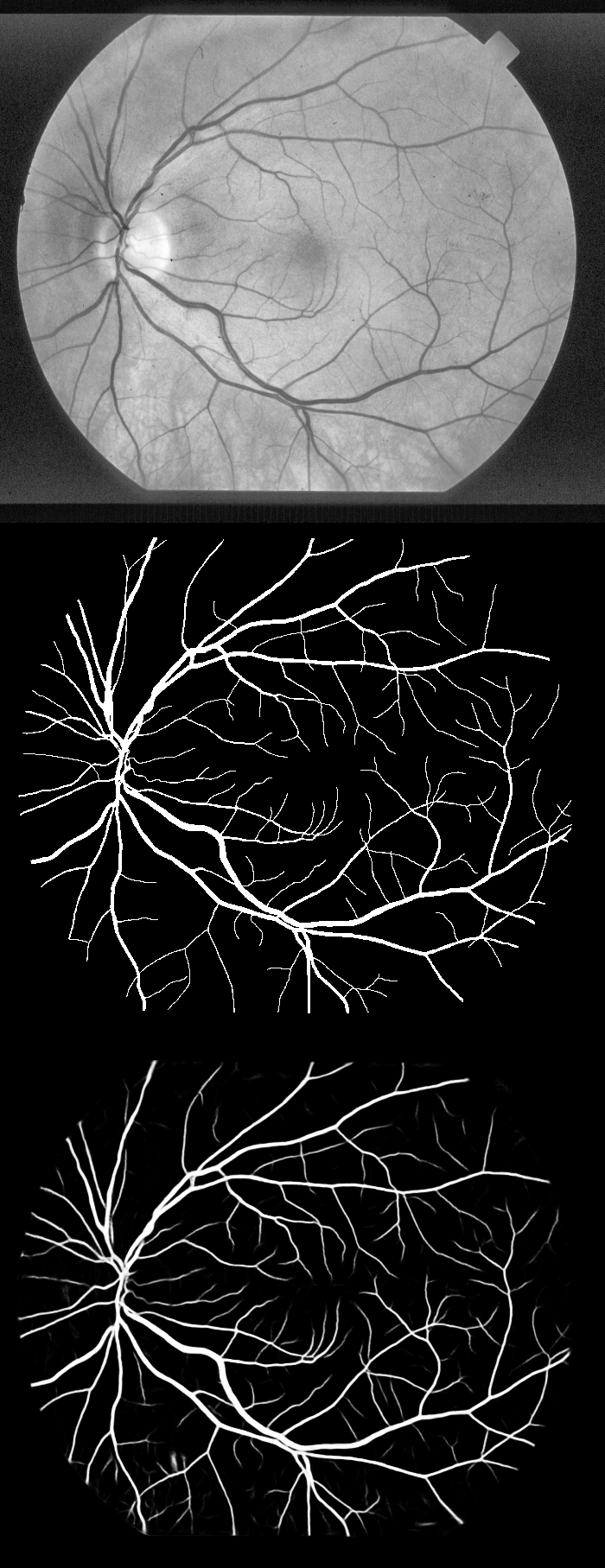
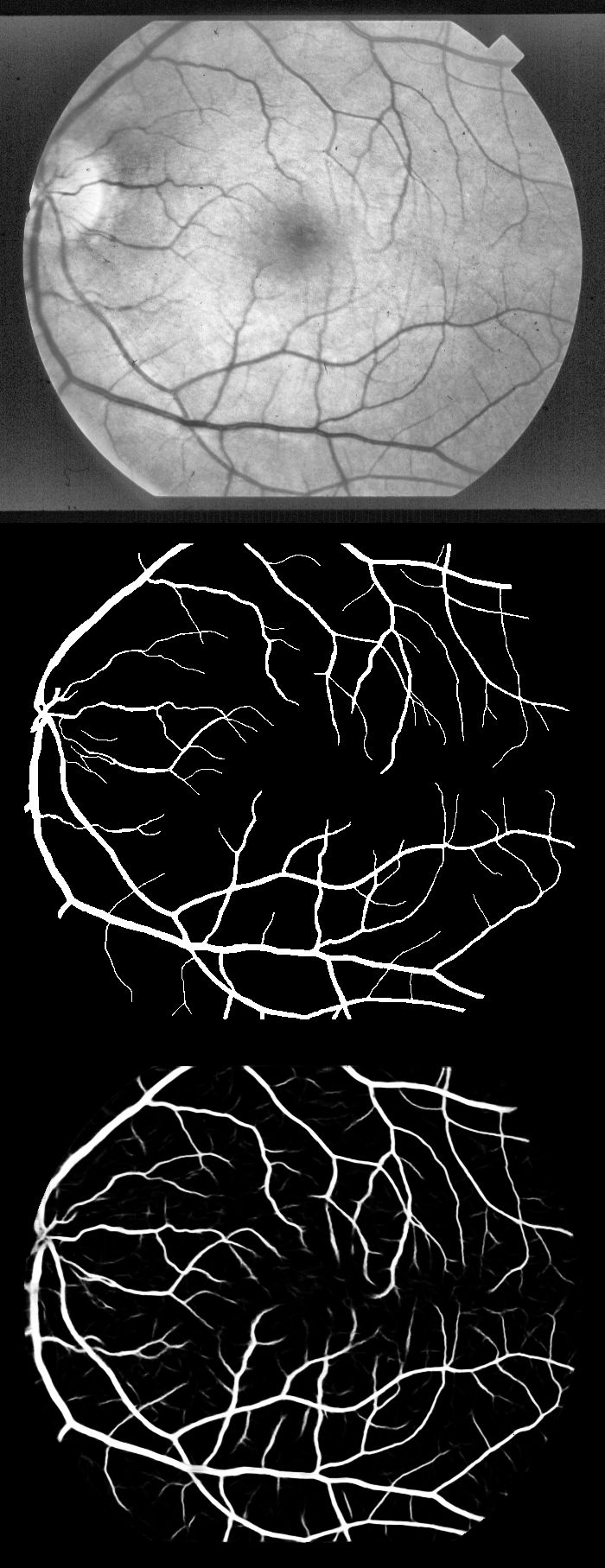
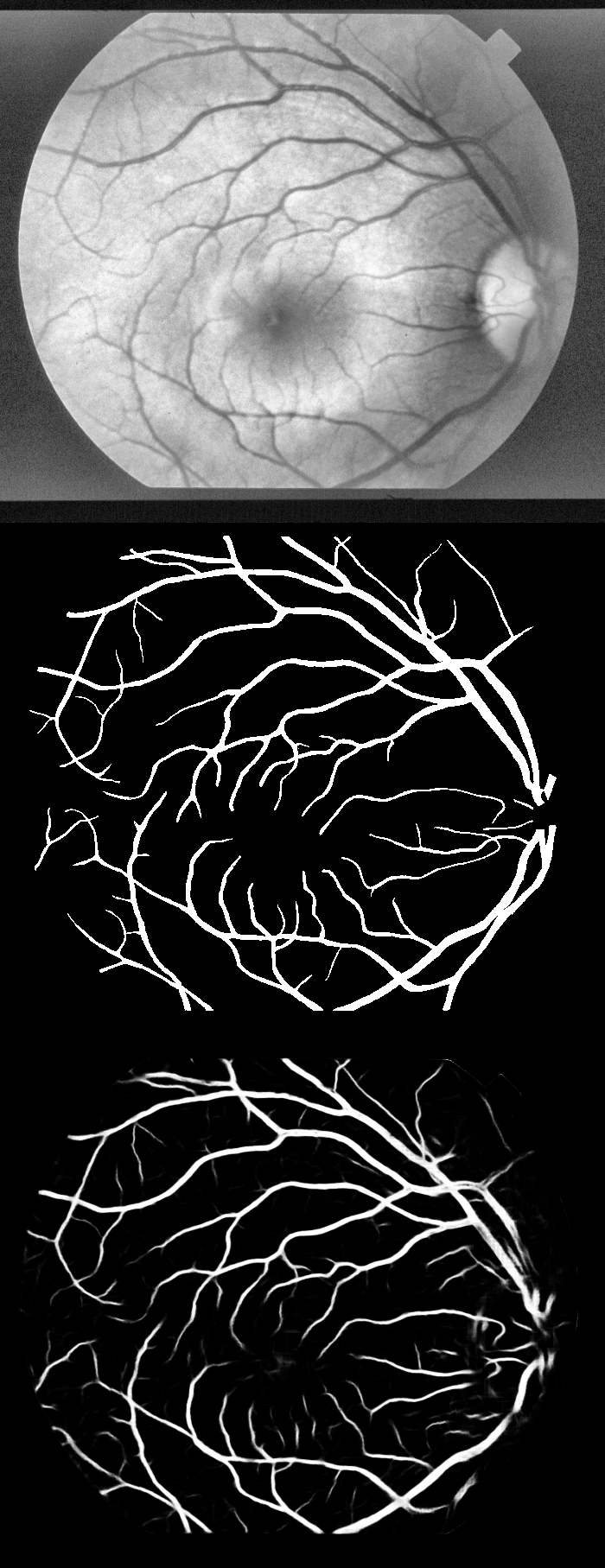
  

Figure 13 : Top – to – bottom a) Original Image, b) Ground truth c) Predictions on STARE

The overall performance of our method and other state-of-the-art deep-learning based methods on DRIVE and CHASE DB1 are tabulated below in Table 2 for DRIVE, 3 for CHASEDB1 respectively. The results prove that our method has displayed superior performance compared to recent state-of-the-art methods.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Method** | **DRIVE** | | | | |
| **AUC** | **SENS** | **SPEC** | **F1** | **ACCU** |
| [1] | 0.9749 | 0.7276 | 0.9785 | - | 0.9466 |
| [2] | 0.9738 | 0.7569 | **0.9816** | - | 0.9527 |
| [3] - | 0.9710 | 0.7520 | 0.9806 | - | 0.9515 |
| [4] | - | 0.7603 | - | - | 0.9523 |
| [5] | 0.9714 | - | - | 0.8091 | 0.9714 |
| [6] | 0.9784 | 0.7792 | 0.9813 | 0.8171 | 0.9784 |
| [7] | 0.9794 | 0.7856 | 0.9810 | 0.8202 | **0.9793** |
| Pre-processing + Our model | **0.9813** | **0.800351** | 0.98013 | **0.8266** | 0. 957246 |

**Table 1. Comparison table of LadderNet results against other results on DRIVE dataset**

[1] Melinˇsˇcak, M., Prentaˇsi´c, P., Lonˇcari´c, S.: Retinal vessel segmentation using deep neural networks. In: 10th International Conference on Computer Vision Theory and Applications (VISAPP 2015) (2015)

[2] Li, Q., Feng, B., Xie, L., Liang, P., Zhang, H., Wang, T.: A cross-modality learning approach for vessel segmentation in retinal images. IEEE transactions on medical imaging 35(1), 109–118 (2015)

[3] Liskowski, P., Krawiec, K.: Segmenting retinal blood vessels with deep neural networks. IEEE transactions on medical imaging 35(11), 2369–2380 (2016)

[4] Fu, H., Xu, Y., Lin, S., Wong, D.W.K., Liu, J.: Deepvessel: Retinal vessel segmentation via deep learning and conditional random field. In: International conference on medical image computing and computer-assisted intervention. pp. 132–139. Springer (2016)

[5] Laibacher, T., Weyde, T., Jalali, S.: M2u-net: Effective and efficient retinal vessel segmentation for resource-constrained environments. arXiv preprint arXiv:1811.07738 (2018)

[6] Alom, M.Z., Hasan, M., Yakopcic, C., Taha, T.M., Asari, V.K.: Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation. arXiv preprint arXiv:1802.06955 (2018)

[7] Zhuang, J.: Laddernet: Multi-path networks based on u-net for medical image segmentation. arXiv preprint arXiv:1810.07810 (2018)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **METHOD** | **CHASEDB1** | | | | |
| **AUC** | **SENS** | **SPEC** | **F1** | **ACCU** |
| [1] | 0.9532 | 0.7201 | 0,9824 | - | 0.9850 |
| [2] | 0.9793 | 0.7507 | 0.9793 | - | 0.9581 |
| U- Net [3] | 0.9772 | 0.8288 | 0.9709 | 0.7783 | 0.9578 |
| Recurrent – Unet [3] | 0.9803 | 0.7459 | 0.9836 | 0.7810 | 0.9622 |
| Our method | **0.9859** | 0.7978 | 0.9818 | **0.8031** | **0.9656** |

**Table 2. Comparison table of LadderNet results against other results on ChaseDB1 dataset**

[1] Sohini Roychowdhury et al., “Blood vessel segmentation of fundus images by major vessel extraction and subimage classification,” JBHI, 2015. [15] Qiaoliang Li et al., “A cross-modality learning approach for vessel segmentation in retinal images.,” IEEE Trans. Med. Imaging, 2016.

[2] Qiaoliang Li et al., “A cross-modality learning approach for vessel segmentation in retinal images.,” IEEE Trans. Med. Imaging, 2016.

[3] Md Zahangir Alom et al., “Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation,” 2018.

Table 2 and 3 tabulate the quantitative results of different methods on datasets DRIVE and CHASEDB1 respectively. It can be observed that LadderNet combined with our proposed pre-processing method generates the highest F1-score, accuracy and AUC for both tasks. It is to be noted that LadderNet also generates satisfiable SE and SP on two tasks. It is to be noted that this model surpassing existing models in AUC implies that the model is highly skilled in distinguishing between the positive and negative classes. Also, other metrics such as Accuracy and F1-score evaluate the performance of a model based on its ability to distinguish between two categories; Therefore, a higher ACCU, AUC and F1-score is more convincing than a higher SENS or SPECS. LadderNet achieves the highest ACCU, AUC and F1-score in both tasks, therefore it performs the best compared to previous models.

1. **DISCUSSION AND CONCLUSION**

In this paper, a unique model for segmentation was presented and evaluated. It is observable that, compared to UNet and most of its variants, LadderNet has more encoder-decoder pairs. However, the drawbacks associated with this phenomenon is handled through shared residual block as discussed. Top of that, the skip connections allows LadderNet to have multiple paths for information flow, and this number of paths is seen to be increasing exponentially with the increase in the number of encoder-decoder pairs.

These features present in the LadderNet allows it to enjoy the advantages of residual connection, recurrent convolution layer and dropout regularization, all while keeping the number of parameters less owing it to the shared residual block. This method can be wildly suited for other medical segmentation tasks as well such as glaucoma or brain lesion detection etc.