

# ELL715

## Assignment 10 Report

Ankit Kumar (2017MT10727)

May 31, 2020

## 1 Dataset

Please note that the images that were given to us didn't have ground truth labels. Thus, I have used the DRIVE dataset obtained from here: <https://drive.grand-challenge.org/Download/>

### 1.1 Strategy

- The dataset has 20 images with known ground truth (let's call it set A), and other with unknown ground truth (set B).
- First, I have worked only with set A
  - I follow leave-one-out strategy.
  - First, I take image 21 as test. I train on image 22 – 38. The other two images, 39 and 40, are then used as validation.
  - Thus, I keep 17 images for training, 2 for validation, and 1 for test.
  - For each such run, I have **5 epochs**.
- Then, as an "extra-credit" part, I have used all the images in set A as training set and all the images in set B as the test set. Since we are not given the ground truth labels for set B, I have attached all the output images near the very end of the document, but without any evaluation metric, of course.

## 2 Pre-processing (Feature Creation)

Retinal vessel segmentation can be seen to be a "binary classification" task. Given a fundus image as input, my neural network should predict the pixels to be vessel pixels or not. Before training such a network, I pre-processed my images as explained here:

- The training of the neural network is performed on sub-images (patches) of the pre-processed full images.
- Each patch is of dimension 96\*96, and is obtained by randomly selecting its center in the pre-processed "full" image.
- For every given image, I randomly extracted 500 patches.

### Advantage of using patches

- The area covered by one patch is 9216 sq units. On the other hand, the area of given images is approx 332k sq units. Thus, the size of patch is only roughly 0.03 times the image.
- Due to such structure, it makes it easy for the neural network to easily differentiate between:
  - Optic Disc boundary from vessel pixels
  - Dark edges of abnormalities from vessels
  - Low contrast, thin vessels from background pixels
- Due to the "small" area covered by the patch, the neural net can apply appropriate (possibly non-linear) function to the patches, and thus making it easy to get appropriate differences.



Figure 1: Image patch

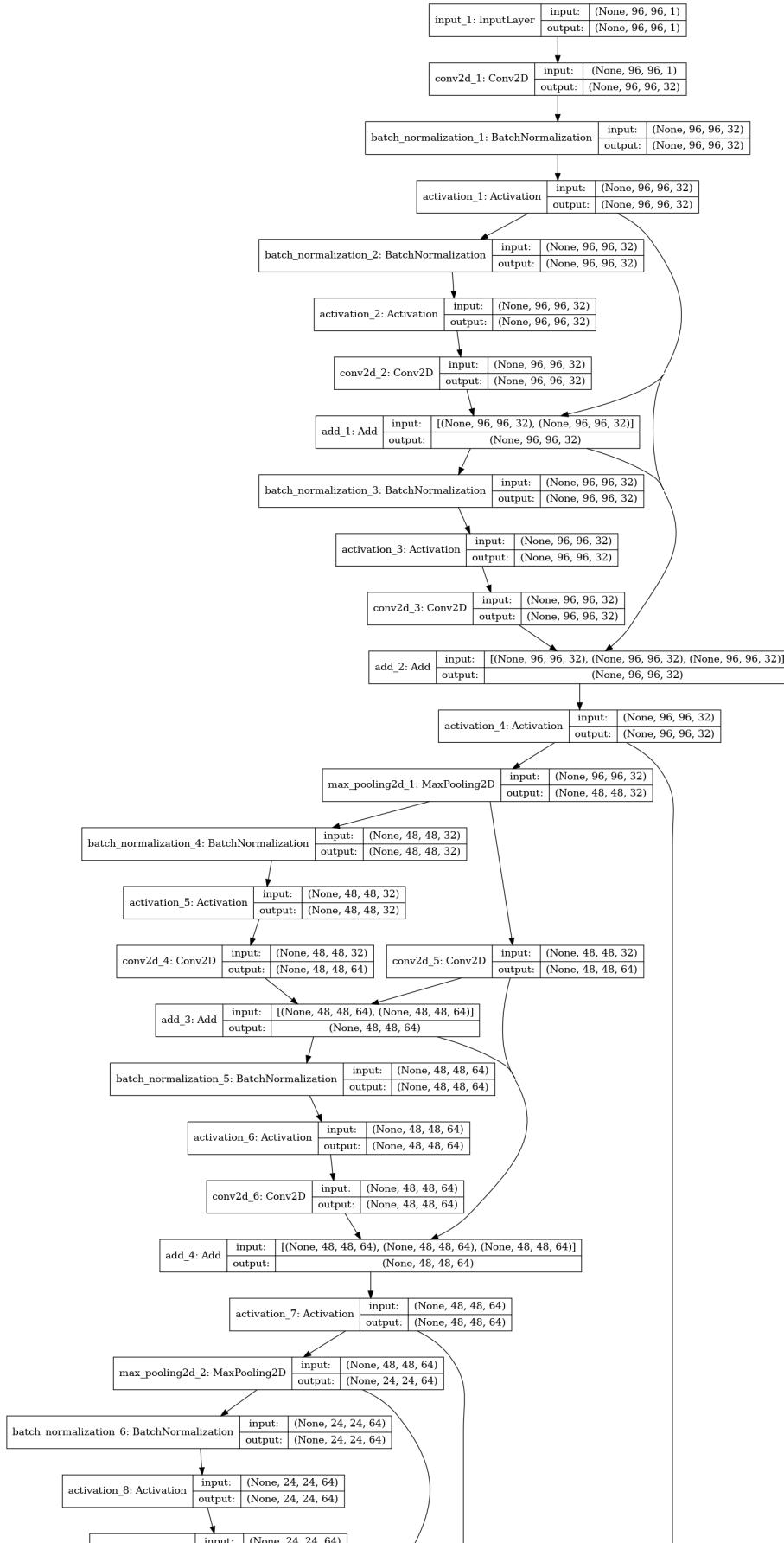


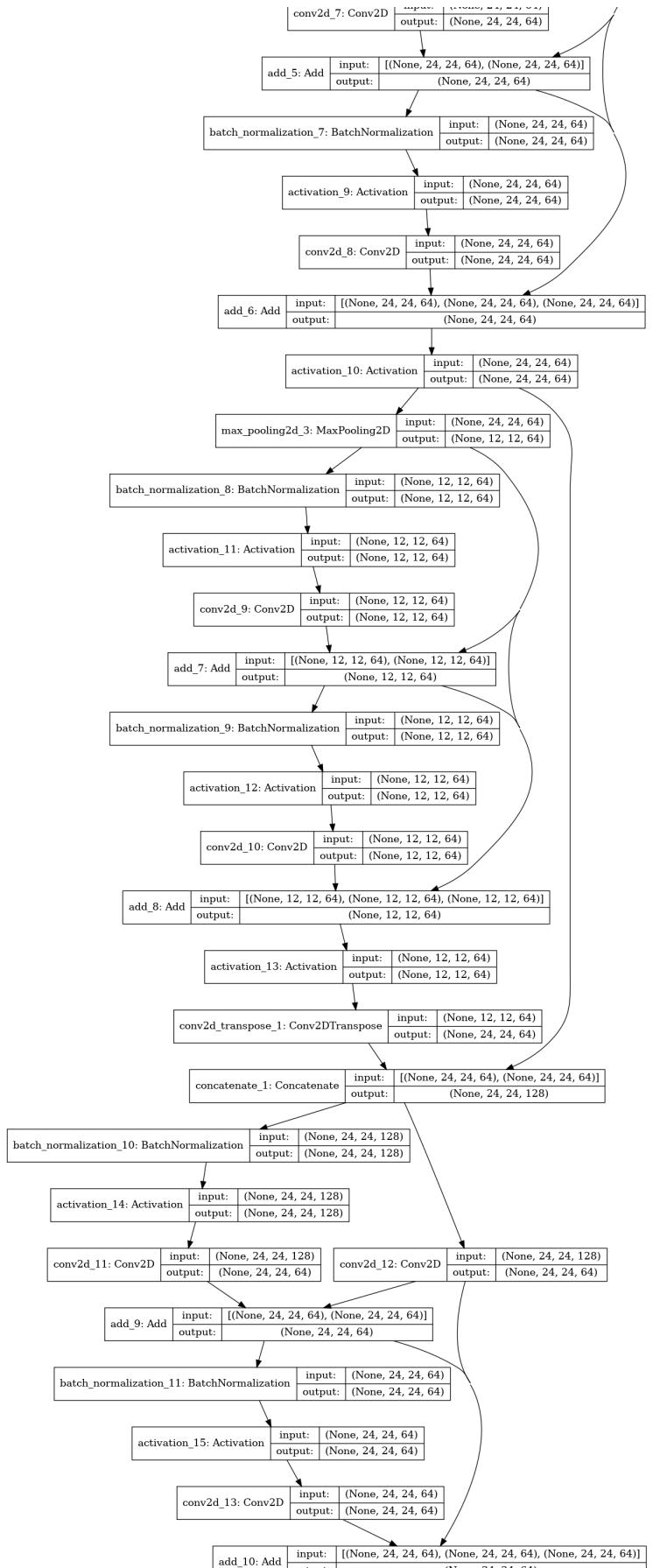
Figure 2: Ground truth patch

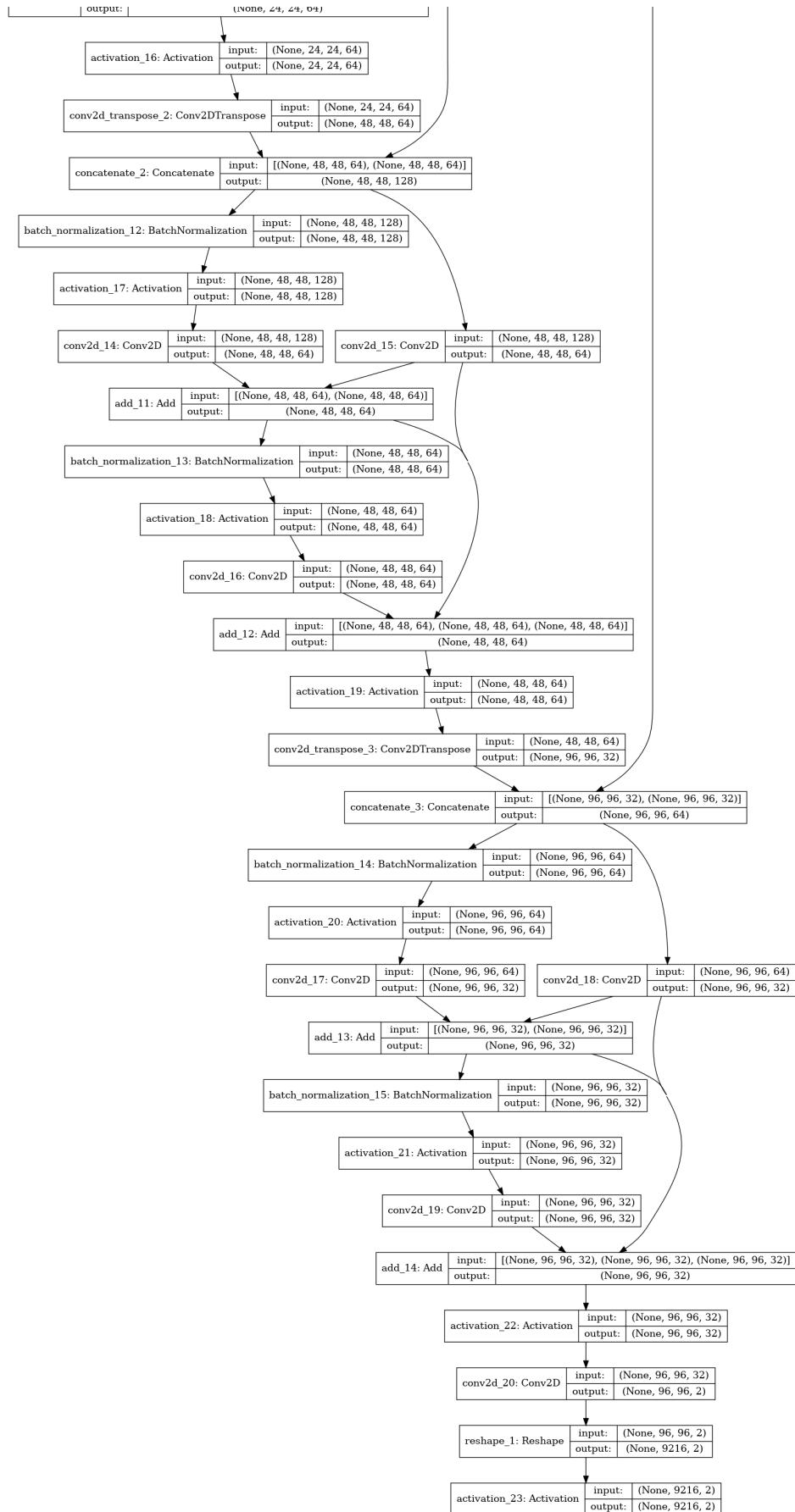
For every given image, I generate 20 such patched-image. One such patched-image has 25 patches, and hence I get 500 patches per image. The figure 1 above shows patches obtained from grayscale version of input image. Figure 2 shows the patches obtained from the ground truth of the corresponding patches.

### 3 Architecture

The next 3 pages describe the CNN architecture used pictorially.







## 4 Results

### 4.1 Detailed Results

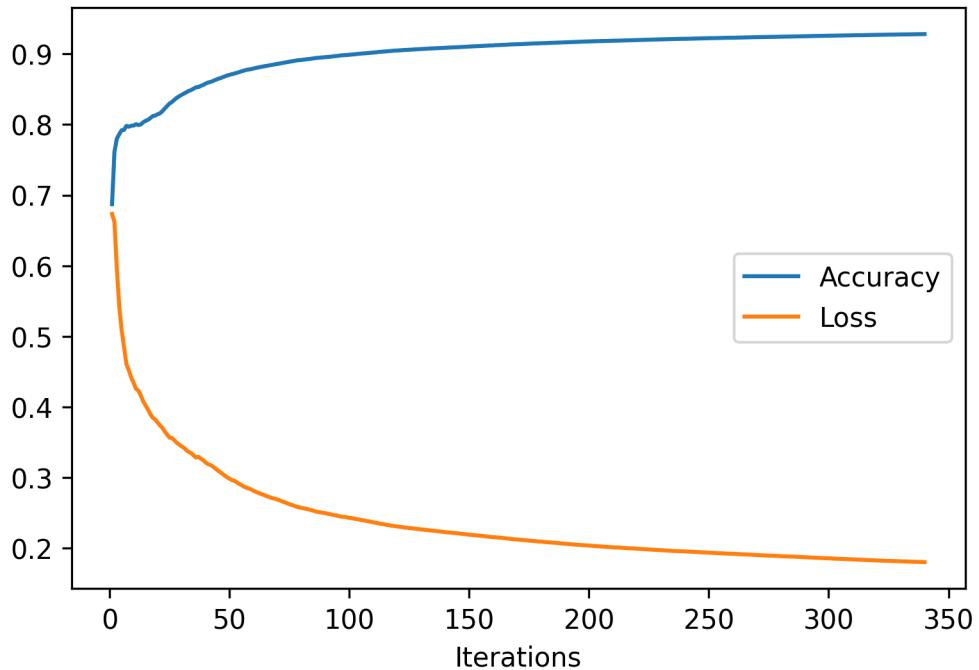
The results in this subsection correspond to this image (#21):



Since such an analysis would be rather tedious for all the 20 images, I'll skip some of the graphs for the rest 19 images.

#### 4.1.1 Iteration VS Accuracy/Loss

The following plot shows the number of iterations V/S the training accuracy (in blue label) and the training loss (in red label)



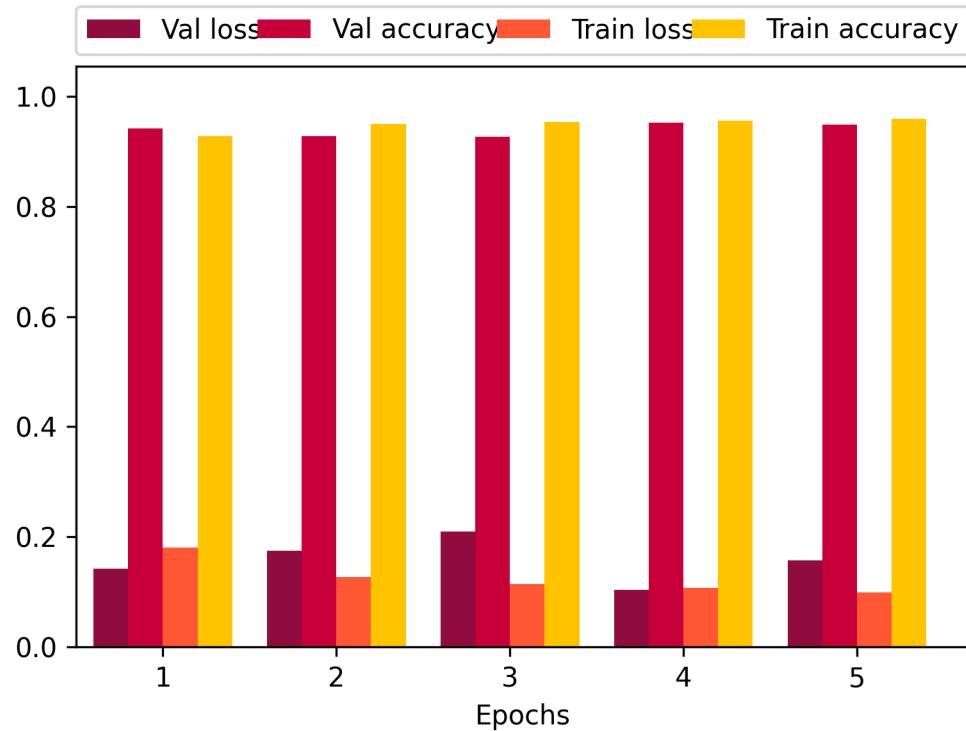
#### 4.1.2 Epochs VS Accuracy/Loss

The following plot shows:

- Training accuracy
- Training loss
- Validation accuracy

- Validation loss

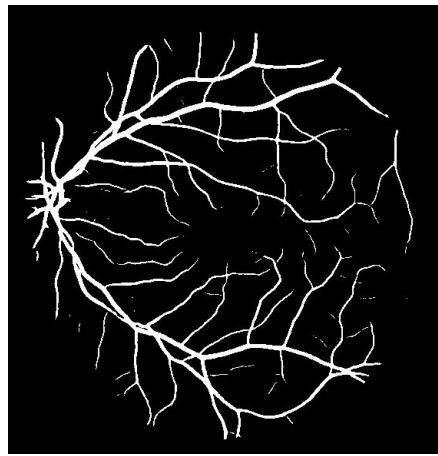
V/S the number of epochs (1 – 5)



Minimum validation loss is for epoch 4, and hence those weights are used for testing the model.

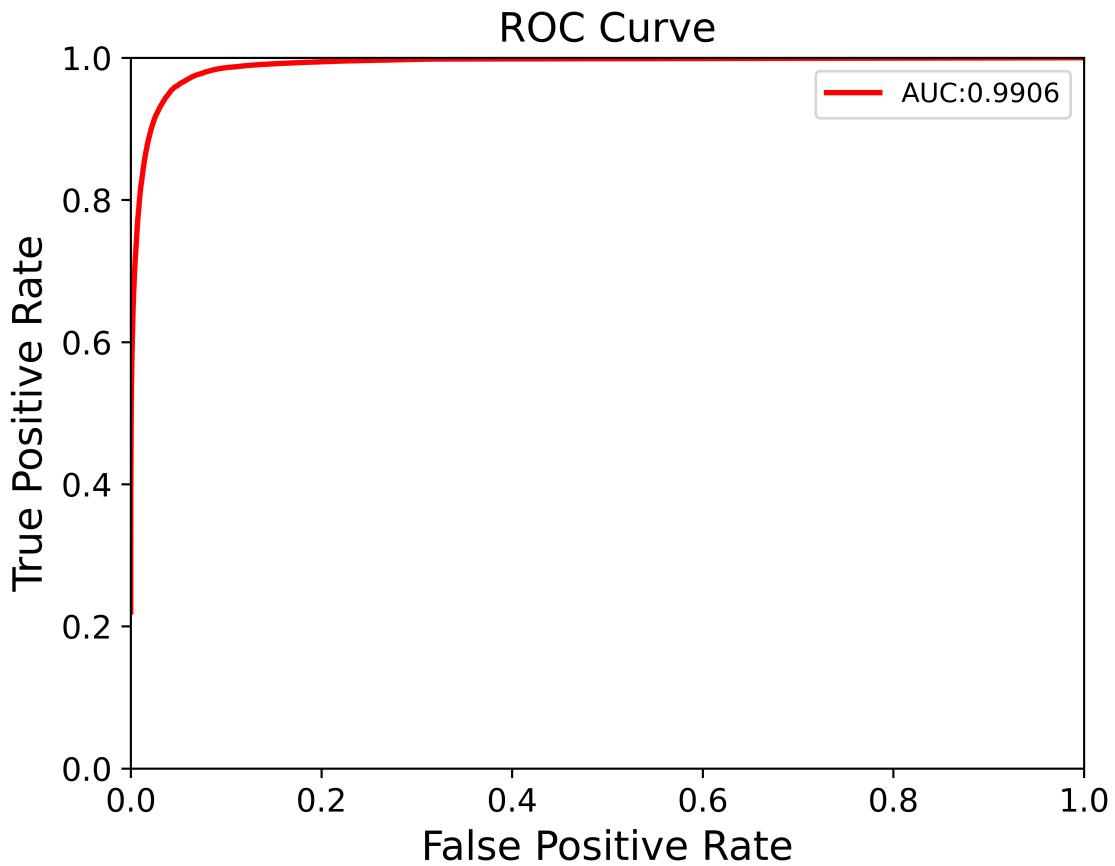
#### 4.1.3 Prediction (Segmented Output)

The output obtained after prediction on the image shown in section 4.1 is:



#### 4.1.4 Performance Evaluation

- Accuracy: 0.88
- AUC under ROC: 0.9906

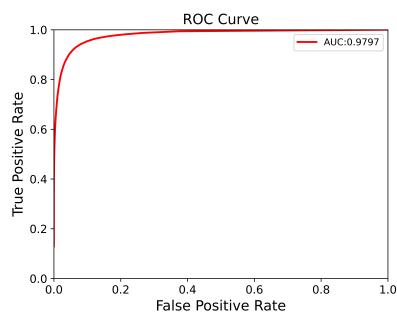
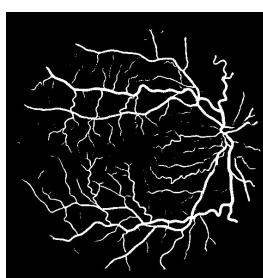


- Precision: 0.9
- Sensitivity: 0.38
- Specificity: 0.99

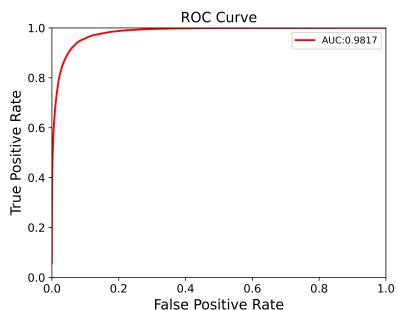
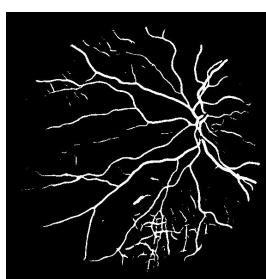
## 4.2 Other Results

This subsection contains relatively "shortened" results. The format followed is:

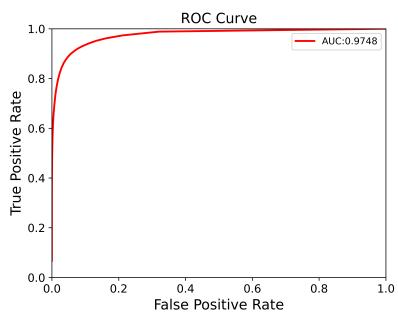
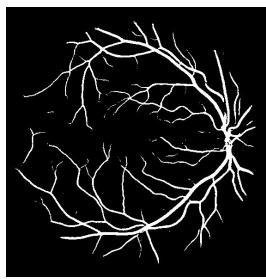
{Input Image}    {Segmented Vessel Map}    {ROC Curve}    {Other Performance Metrics}



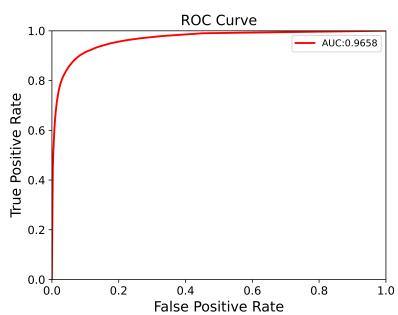
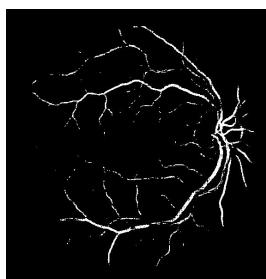
Sensitivity: 0.38  
 Specificity: 0.98  
 Precision: 0.84  
 Accuracy: 0.86



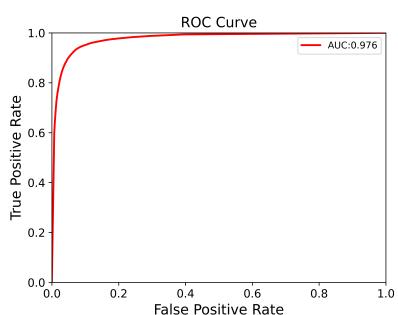
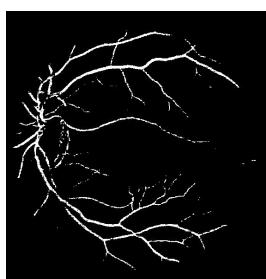
Sensitivity: 0.34  
Specificity: 0.99  
Precision: 0.85  
Accuracy: 0.88



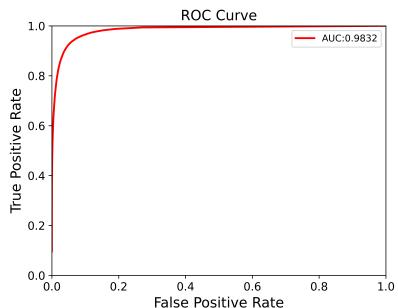
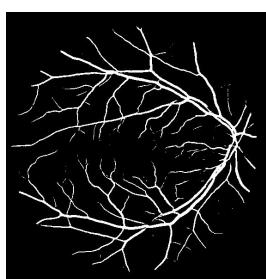
Sensitivity: 0.49  
Specificity: 0.97  
Precision: 0.76  
Accuracy: 0.88



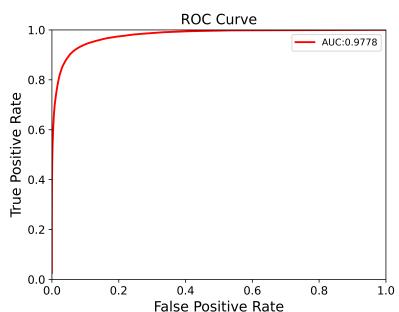
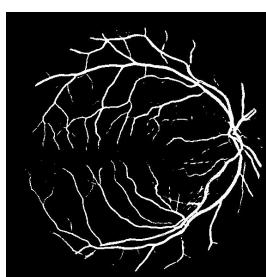
Sensitivity: 0.49  
Specificity: 0.95  
Precision: 0.54  
Accuracy: 0.9



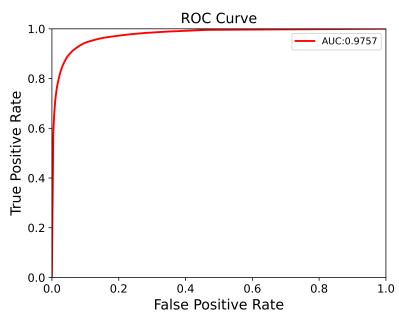
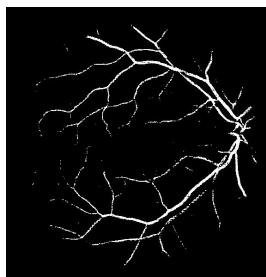
Sensitivity: 0.46  
Specificity: 0.97  
Precision: 0.65  
Accuracy: 0.91



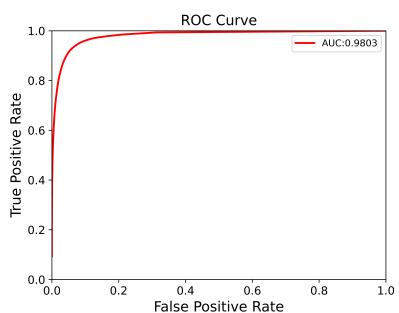
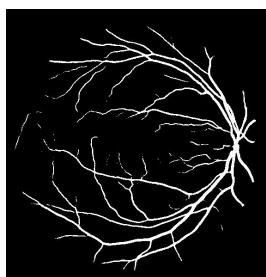
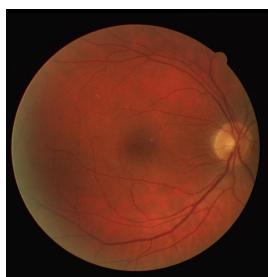
Sensitivity: 0.39  
Specificity: 0.98  
Precision: 0.83  
Accuracy: 0.87



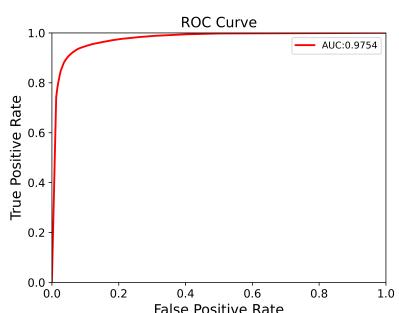
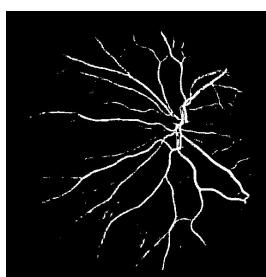
Sensitivity: 0.43  
Specificity: 0.98  
Precision: 0.8  
Accuracy: 0.88



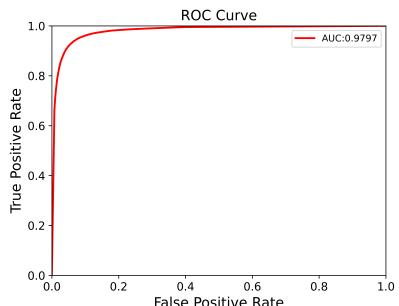
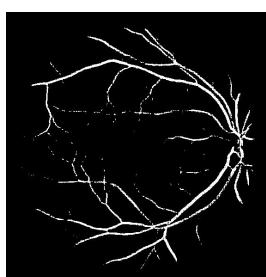
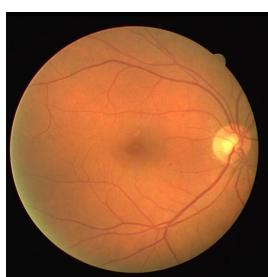
Sensitivity: 0.45  
Specificity: 0.97  
Precision: 0.64  
Accuracy: 0.9



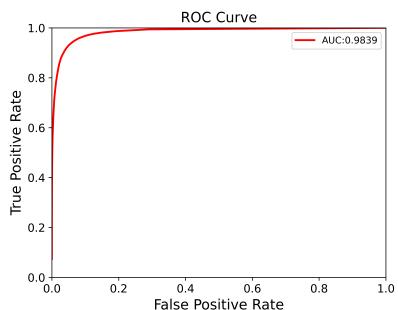
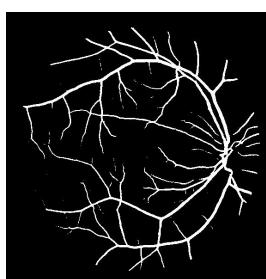
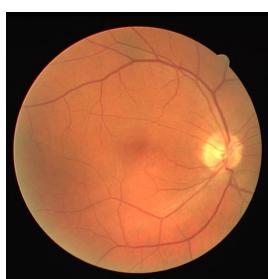
Sensitivity: 0.4  
Specificity: 0.98  
Precision: 0.82  
Accuracy: 0.89



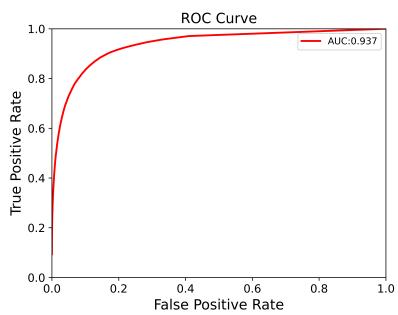
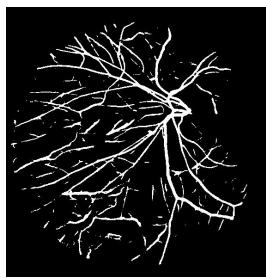
Sensitivity: 0.38  
Specificity: 0.98  
Precision: 0.75  
Accuracy: 0.91



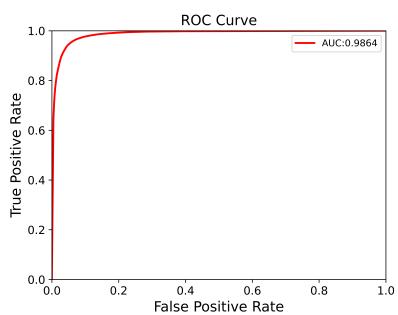
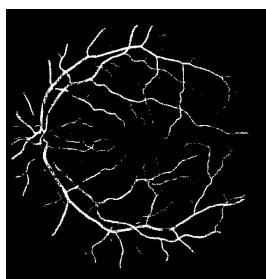
Sensitivity: 0.45  
Specificity: 0.97  
Precision: 0.69  
Accuracy: 0.9



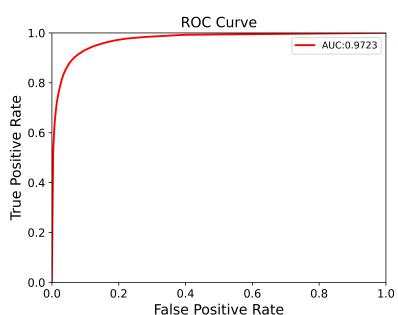
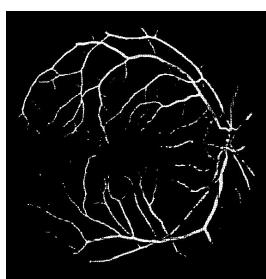
Sensitivity: 0.41  
Specificity: 0.98  
Precision: 0.83  
Accuracy: 0.89



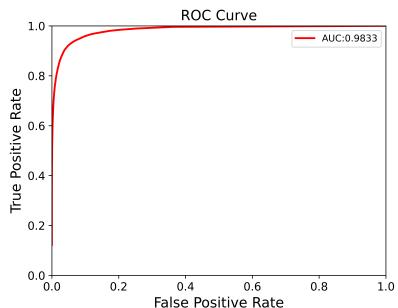
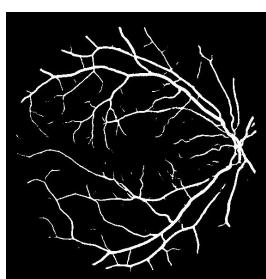
Sensitivity: 0.35  
Specificity: 0.97  
Precision: 0.74  
Accuracy: 0.84



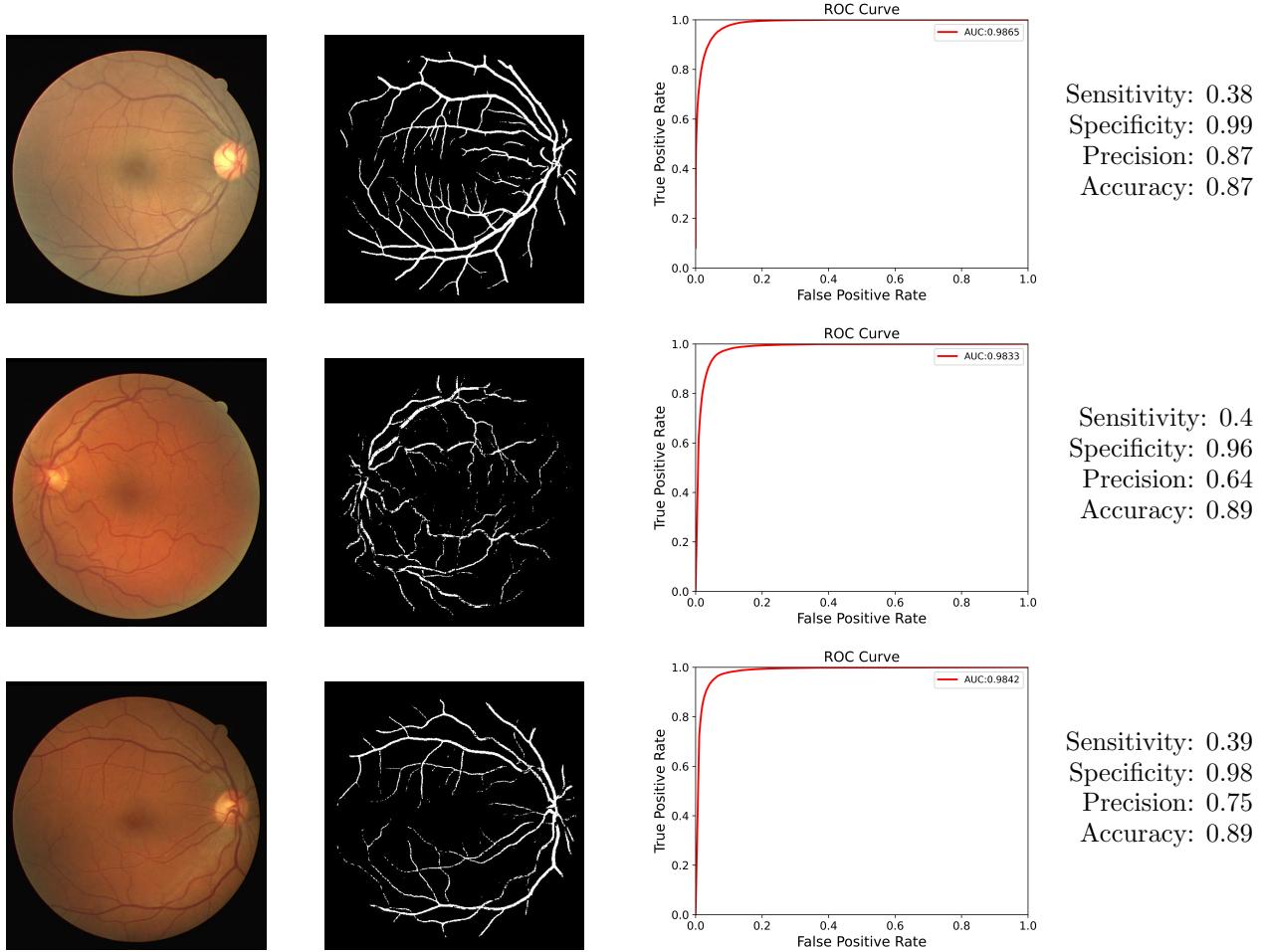
Sensitivity: 0.42  
Specificity: 0.97  
Precision: 0.69  
Accuracy: 0.89



Sensitivity: 0.45  
Specificity: 0.95  
Precision: 0.6  
Accuracy: 0.88



Sensitivity: 0.41  
Specificity: 0.98  
Precision: 0.84  
Accuracy: 0.88



### 4.3 Final Results

For obtaining the final results, I average all the values (accuracy, AUC under ROC, etc.). The results are as follows:

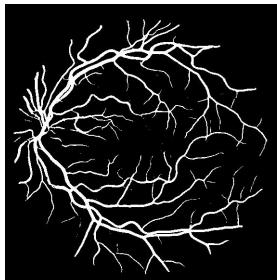
- Accuracy: 0.8845
- AUC under ROC: 0.977885
- Precision: 0.7515
- Sensitivity: 0.4125
- Specificity: 0.9745

## 5 Additional Images: Extra Credits

As mentioned earlier, I have used all the 20 images with known ground truth as training set for this section, and all the other 20 images (with unknown ground truth) as the test set



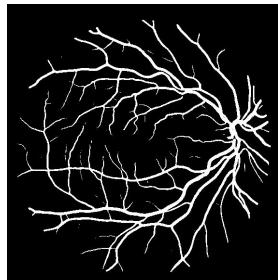
(a)



(b)



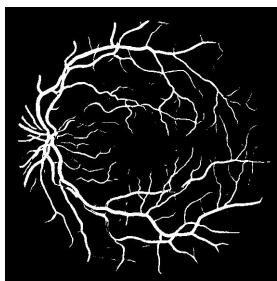
(c)



(d)



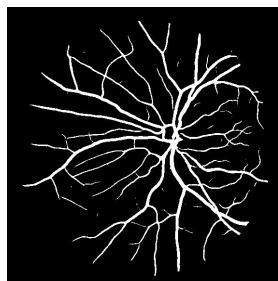
(a)



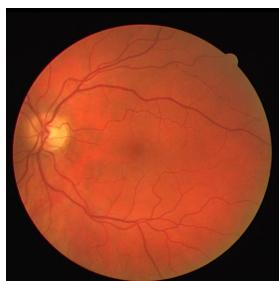
(b)



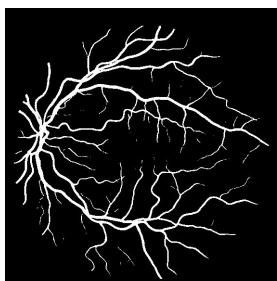
(c)



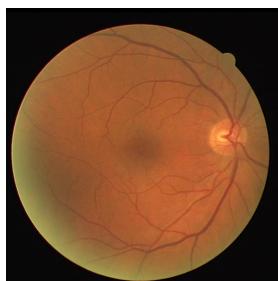
(d)



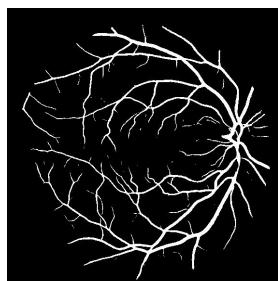
(a)



(b)



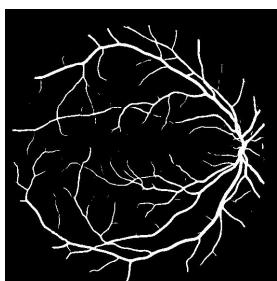
(c)



(d)



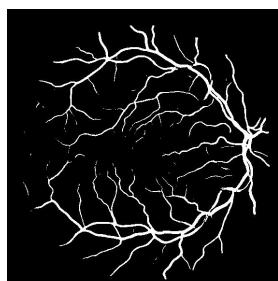
(a)



(b)



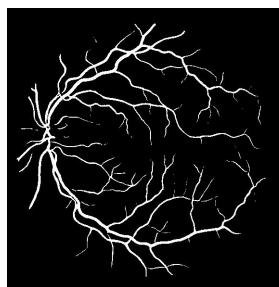
(c)



(d)



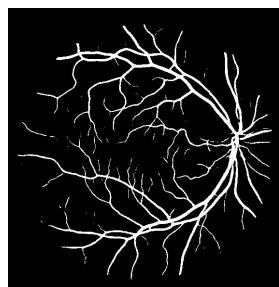
(a)



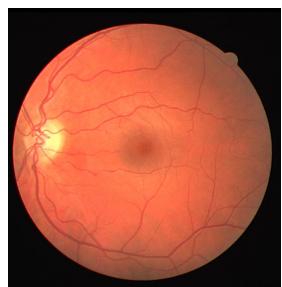
(b)



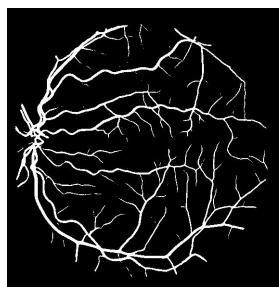
(c)



(d)



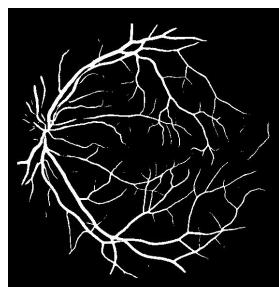
(a)



(b)



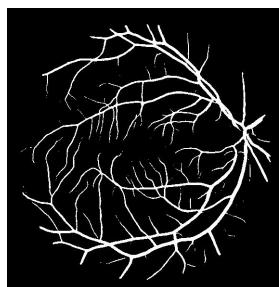
(c)



(d)



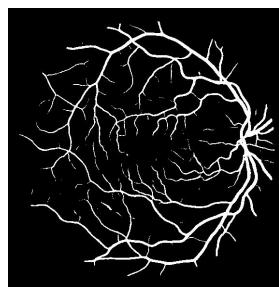
(a)



(b)



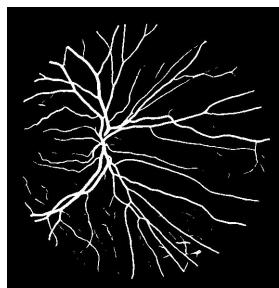
(c)



(d)



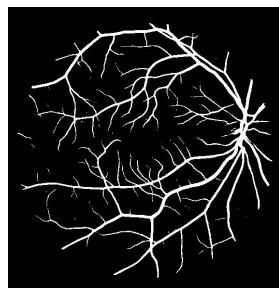
(a)



(b)



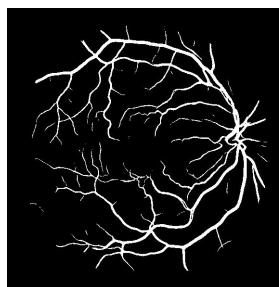
(c)



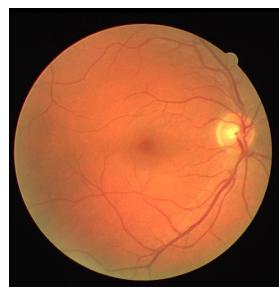
(d)



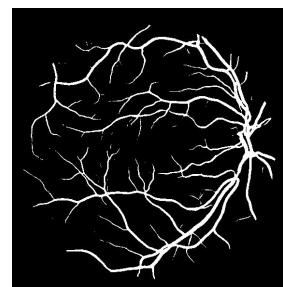
(a)



(b)



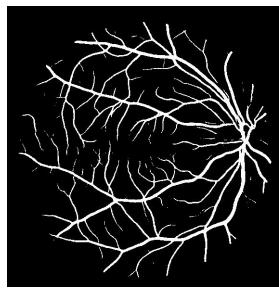
(c)



(d)



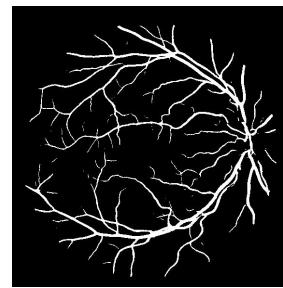
(a)



(b)



(c)



(d)

Figure 3: In the above images, (a) and (c) denote the input images, while (b) and (d) denote the corresponding segmented output images