ELL715
Assignment 8 Report

Ankit Kumar (2017MT10727) , Naman Jhunjhunwala (2017MT10737)

May 12, 2020

1 Original Image

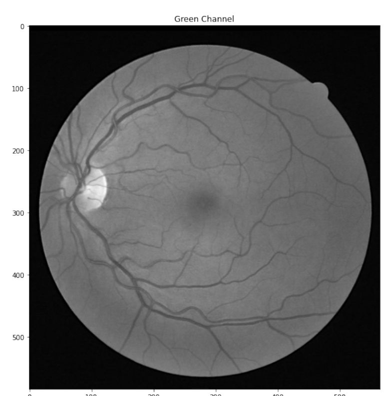
Unless otherwise stated, all the results shown in this document correspond the input image below:

2 Intensity based Thresholding

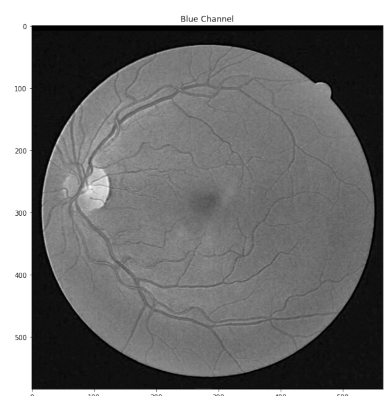
2.1 Channel Extraction

The image has three channels: R, G and B.

(a) Red Channel



(b) Green Channel



(c) Blue Channel

Figure 1: Images showing different channels corresponding to the input image

The green channel gives the best result in the contrast of blood vessels for extracting darker blood vessels on a bright background. Hence, we use the green channel in our further analysis.

2.2 Thresholding

In mean adaptive thresholding, the threshold value is calculated for smaller regions and therefore, there will be different threshold values for different regions. Thus, first, we apply mean adaptive thresholding, with parameters $(11, 2)$ on the green channel. The output of the same is:

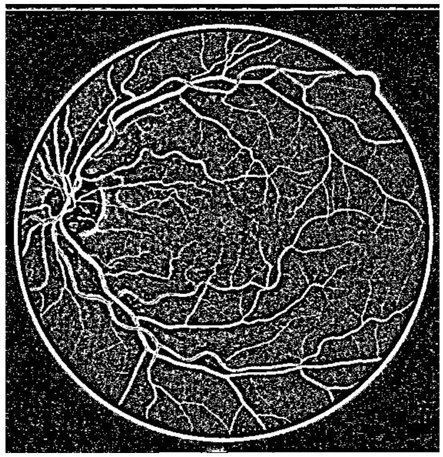


Figure 2: Output of applying mean adaptive thresholding on the green channel

2.3 Area Cleaning

Now, note that in the above output, there are some isolated regions which are misclassified as vessel pixels. These misclassification can be removed by calculating the area of each connected region and considering that if the area is less than 50 (a hyper-parameter) then it can be marked as non-vessel, and hence removed. Output of applying above area-thresholding on the thresholding output:

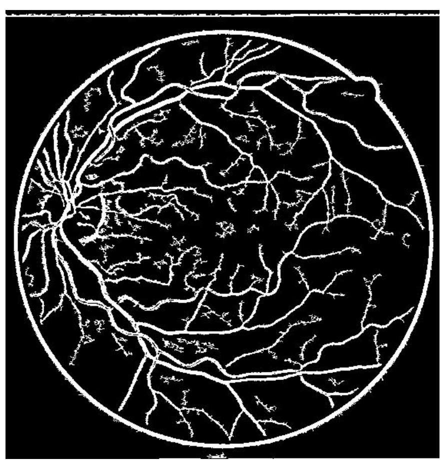


Figure 3: Output of area-based thresholding the output image

2.4 Median Filtering

Next, we use median filtering with a $3 * 3$ block. It helps us solve both the problems: smoothing the vessel map by filling unwanted gaps (holes) and removing spurious branches in a single step. The output of median filtering is:

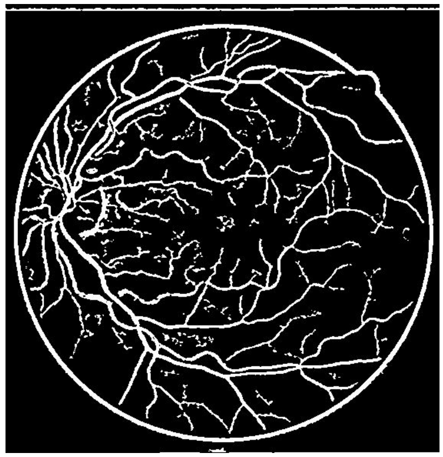


Figure 4: Output of median filtering on the area-thresholded image

2.5 Disk Removal

As the next step, we remove the following extras:

- The outer circular disk
- The extra noisy lines that occur at the top and bottom

It is done in two steps:

- To remove the extra lines, we remove all the pixels in the image that lie in $[0, 20]$ and [image.shape[1]-20, image.shape[1]] (in the vertical direction)
- Now, since the lines are removed, we know that the outer circle is the first component we will hit if we start from the top of the image. Thus, it is easy to (automatically) get a pixel of the outer circle if we start from top. Using that pixel, we remove all the pixels present in the same connected component as the outer circle.

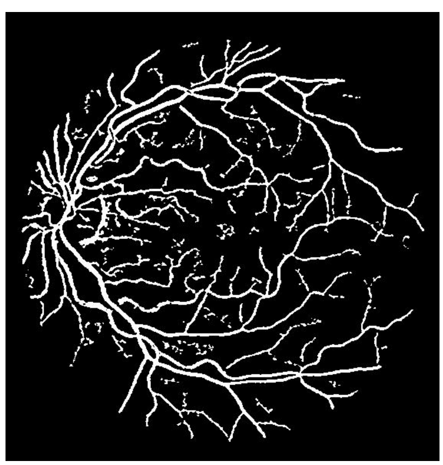


Figure 5: Output of line and circle removal

2.6 Area Cleaning

Finally, we again do area-cleaning, but now with the hyper-parameter 60, thus further removing a few more misclassifications. Thus, the final output of our algorithm is:

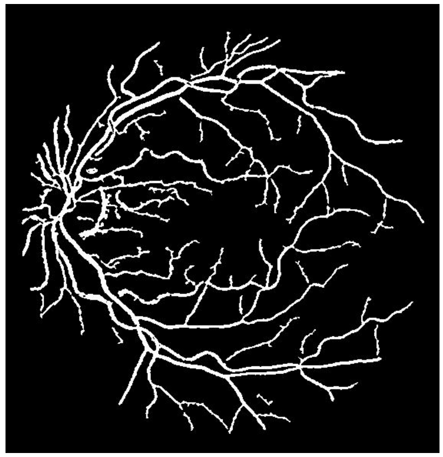


Figure 6: Final output obtained after another area based thresholding

3 Region Growing

3.1 Algorithm

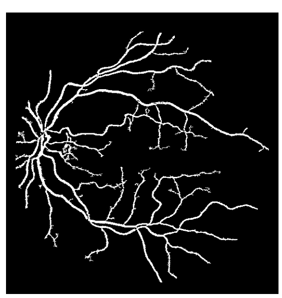
At the start of the algorithm, we manually choose the first seed point. Then, using Breadth First Search (bfs), we find the part of image which is connected to that seed point, where we had defined the appropriate definition of connectivity based on intensity values. Then, if the given image contains all the retinal vessels, we stop the algorithm. Otherwise, we have choose another seed point and generate part which is connected to that seed point, and taken union with the previous output. This process is repeated until we have found all the blood vessels.

After this, there are a few instances where a few noisy components remain attached to the image. If that is the case with the current output image, we use an appropriate seed along point, and with some area thresholding, we remove that noisy component.

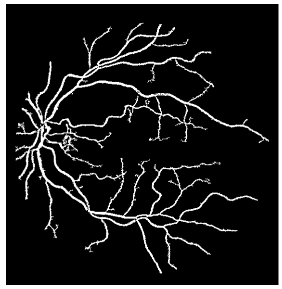
3.2 Output Images

Below, we show output images obtained iteratively after selecting various seed points on the input image:

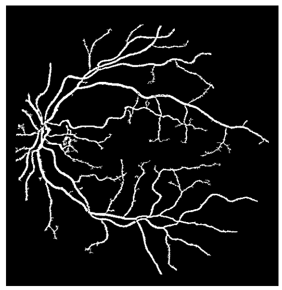
- First seed point: $(137, 181)$. Output image:



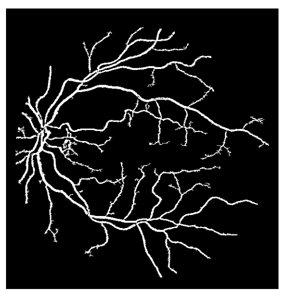
- Second seed point: $(174, 208)$. Output image:



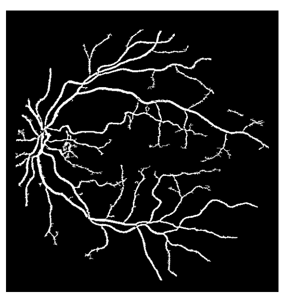
- Third seed point: $(174, 110)$. Output image:



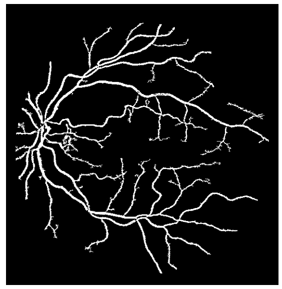
- Fourth seed point: $(457, 266)$. Output image:



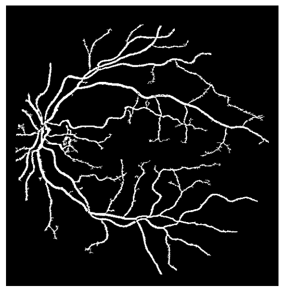
- Fifth seed point: $(521, 233)$. Output image:



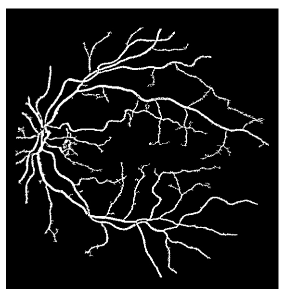
- Sizth seed point: $(473, 211)$. Output image:



- Seventh seed point: $(499, 222)$. Output image:



- Eighth seed point: $(460, 199)$. Output image:



3.3 Final Image

We finally obtained the final output of region growing after the step 2, i.e., removing a few unwanted pixels, again via region growing.

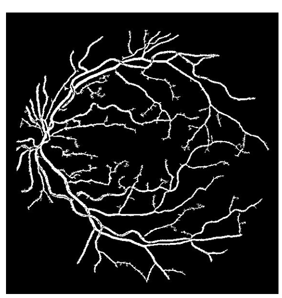


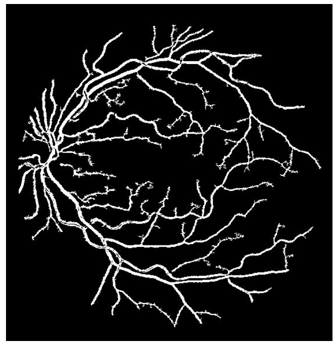
Figure 7: Final region growing output

4 Image Matting

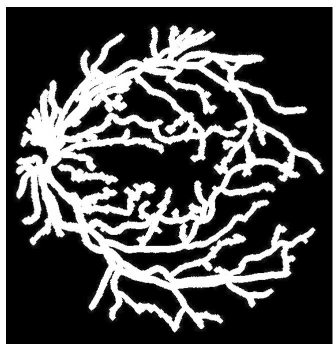
4.1 Trimap Generation

The very first step is trimap generation. We do it in the following steps:

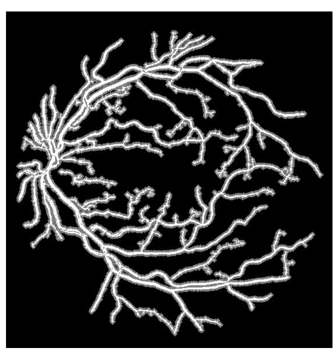
- Image 1: Start with region growing segmentation output:



- Image 2 : Apply dilation with disk kernel of size $3 * 3$



- The white pixels present in the initial image (Image 1) are marked 255. The white pixels present in the dilation output (Image 2) but not present in Image 1 are marked 127. The rest of the pixels are marked 0.



Thus, we have separated the pixels into foreground (255 intensity), background (0 intensity) and unknown pixels (127 intensity).

4.2 Matting

It is a two step process:

$$L_{ij} = \sum_{k|(i,j)\in w_k} \left(\delta_{ij} - \frac{1}{|w_k|}(1 + (I_i - \mu_k)(\Sigma_k + \frac{\epsilon}{|w_k|}I_3)^{-1}(I_j - \mu_k) \right)$$

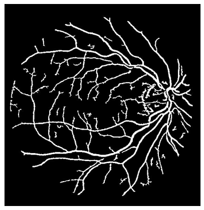
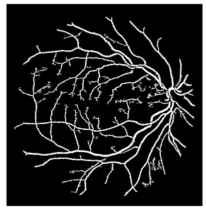
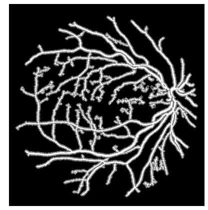
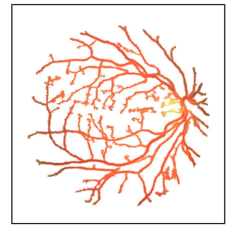
- Calculate the matrix L based on image I which describes how similar neighboring pixels are:
- Solve the linear system for alpha using a constraint matrix D and vector b to fix the known alpha values:

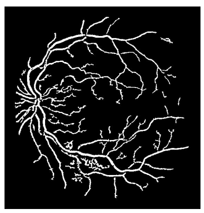
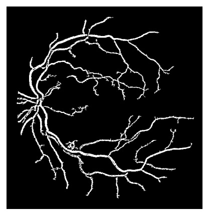
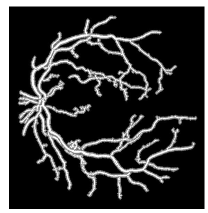
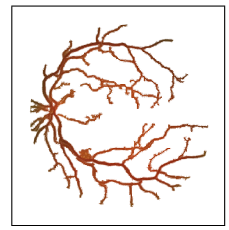
$$\alpha = (L + \lambda D)^{-1}\lambda b$$

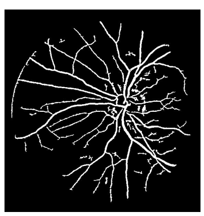
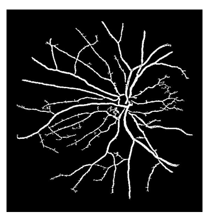
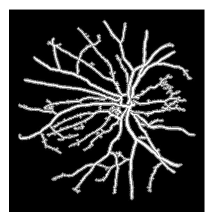
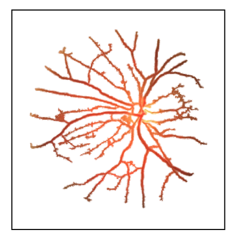
We get the following output after applying matting on the input image and trimap image:

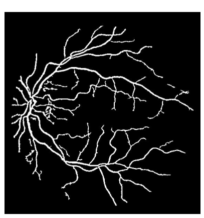
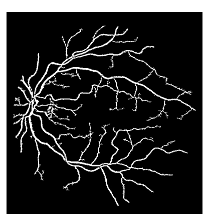
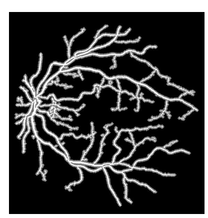
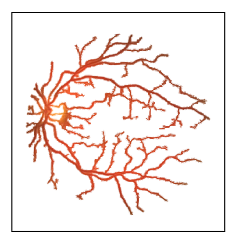
Figure 8: Output of applying matting on the above trimap and input image

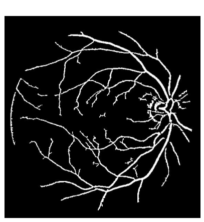
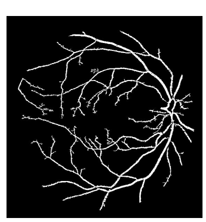
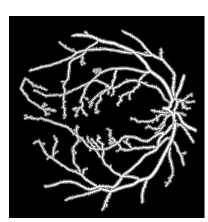
5 Additional Images: Extra Credits


(a)

(b)

(c)

(d)

(e)

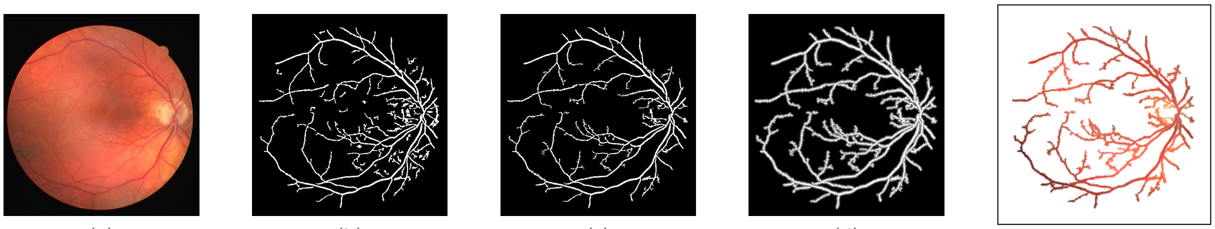

(a)

(b)

(c)

(d)

(e)


(a)

(b)

(c)

(d)

(e)

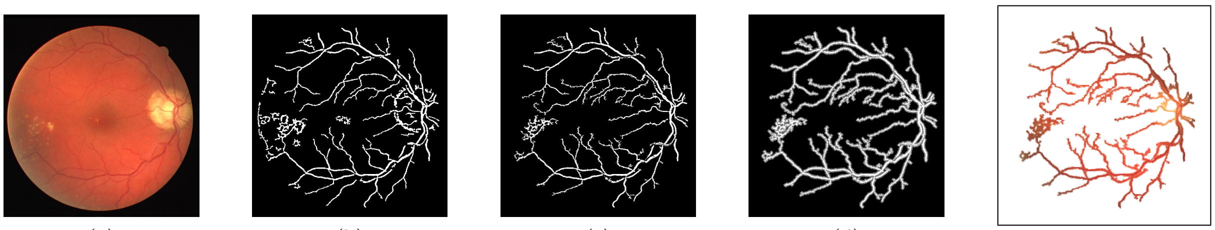

(a)

(b)

(c)

(d)

(e)


(a)

(b)

(c)

(d)

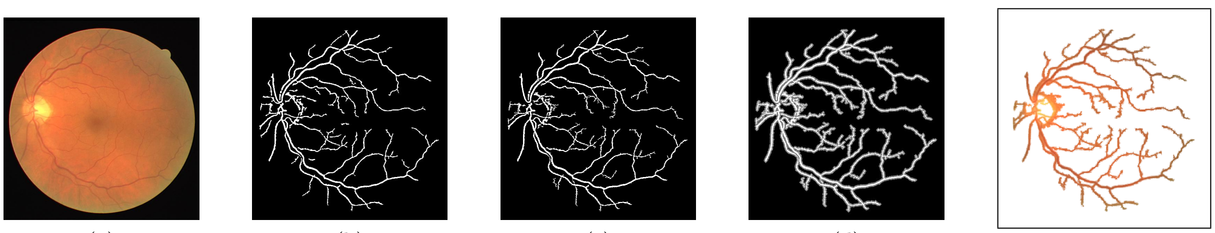
(e)



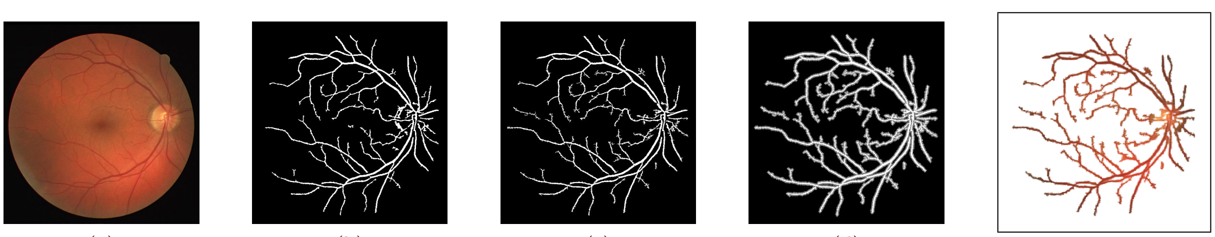
(a) (b) (c) (d) (e)



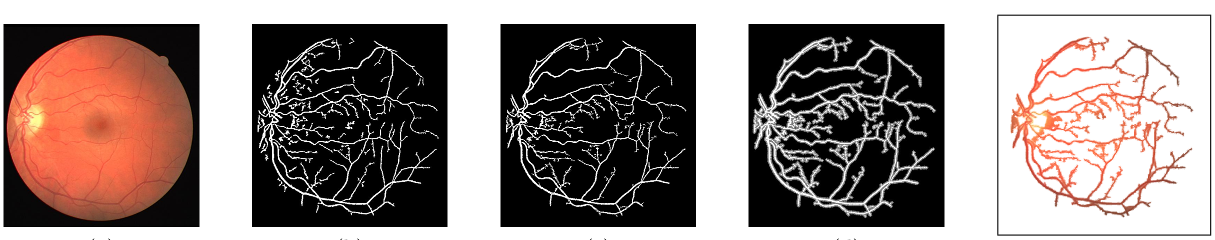
(a) (b) (c) (d) (e)



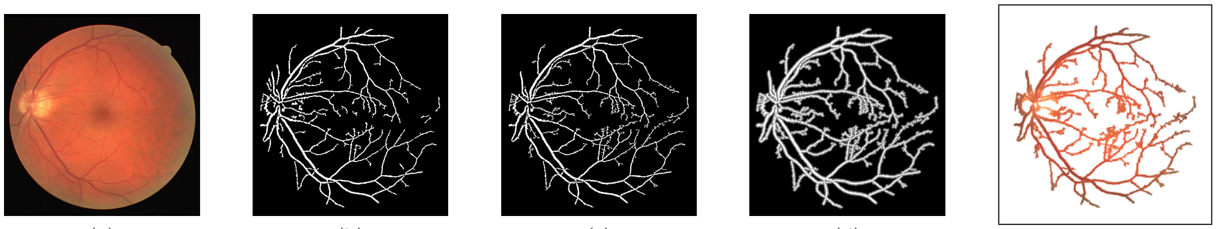
(a) (b) (c) (d) (e)



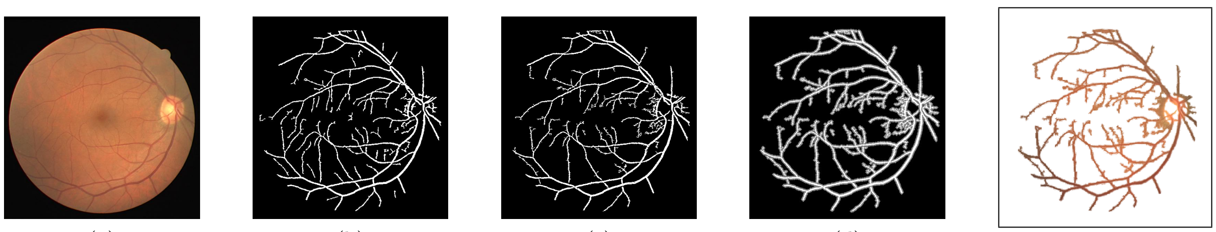
(a) (b) (c) (d) (e)



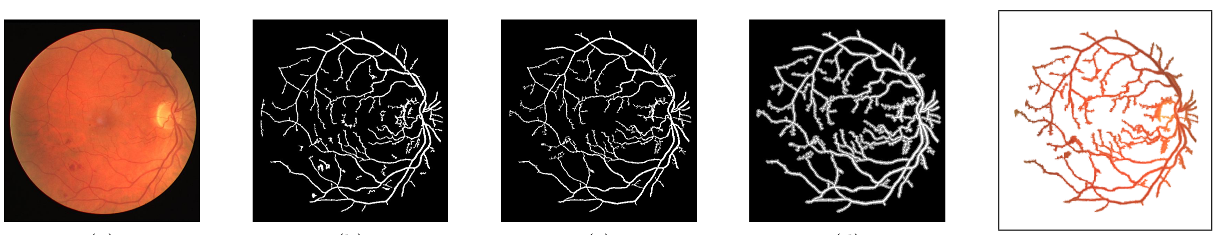
(a) (b) (c) (d) (e)



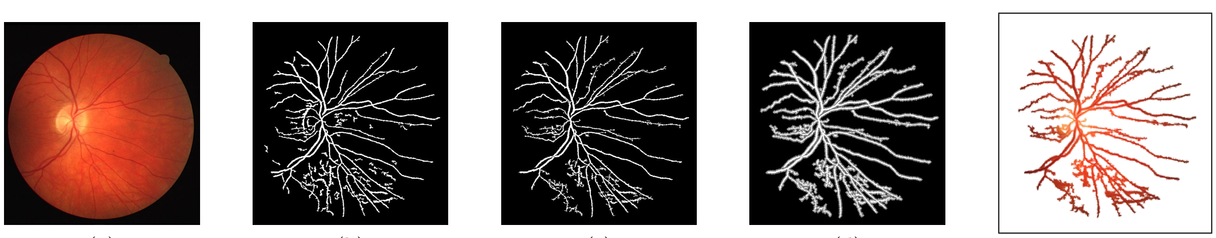
(a) (b) (c) (d) (e)



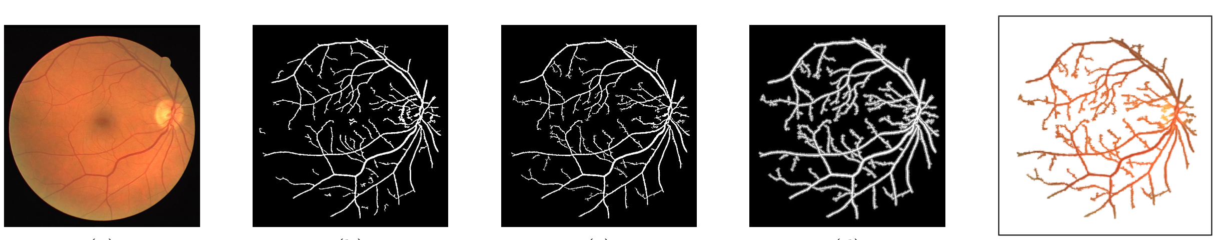
(a) (b) (c) (d) (e)



(a) (b) (c) (d) (e)



(a) (b) (c) (d) (e)



(a) (b) (c) (d) (e)

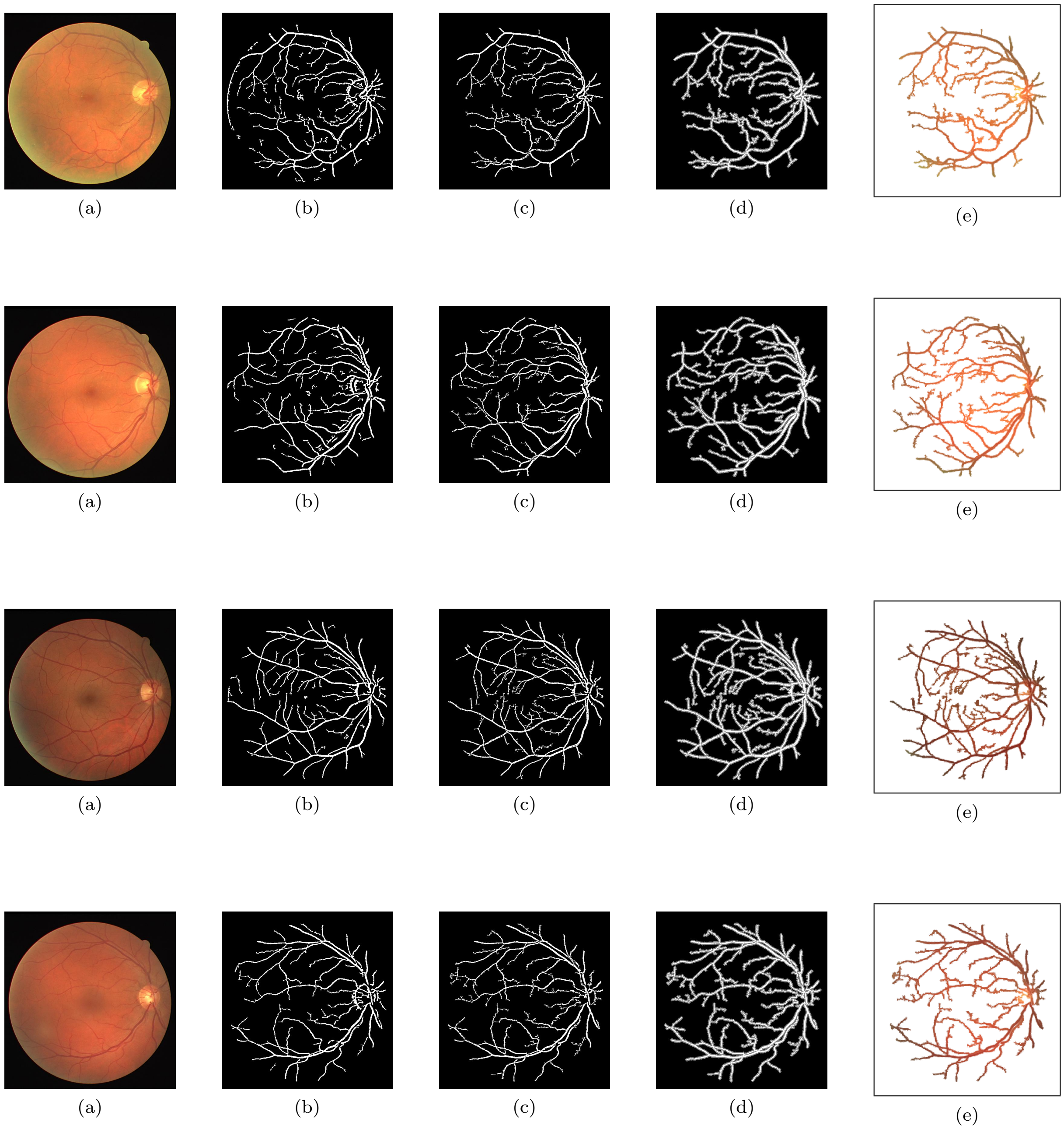


Figure 9: Images showing: (a) Input image (b) Intensity based thresholding (c) Region growing (d) Trimap generated (e) Matting output for all images