MULTIMEDIA UNIVERSITY

CDS6334 Visual Information Processing

Trimester 2430

ASSIGNMENT (20%)

Lecturer: Dr. Loh Yuen Peng

Lecture Session: TC1L

Tutorial Session: TT2L

Student ID: 1211103024

Name: Yap Jack

Abstract

This report presents an algorithm for automated retinal vessel segmentation using a dataset of 80 fundus images with 20 secondary fundus images, each accompanied by their respective ground truth data. The proposed method employs grayscale conversion, intensity normalization, Gaussian smoothing, and Difference of Gaussian (DoG), edge subtraction, morphological operations, and connected component filtering to segment the vessels. The algorithm's performance was evaluated using a dataset of 80 fundus images, with the adapted rand error, precision, recall, and Intersection over Union (IoU) used as metrics. Results showed that the algorithm performed best on sharp, well-lit images, with adapted rand error of 31%, precision of 73%, recall of 65%, and IoU of 53%, respectively. Further testing on an additional dataset demonstrated similar performance, with slight variations in adapted rand error, precision, recall, and IoU. Despite the challenges posed by noisy, blurred, or poorly lit images, the algorithm showed promise in segmenting large vessels and identified areas for potential improvement, including the use of advanced techniques such as CLAHE, bilateral filtering, and unsharp masking to enhance vessel detection.

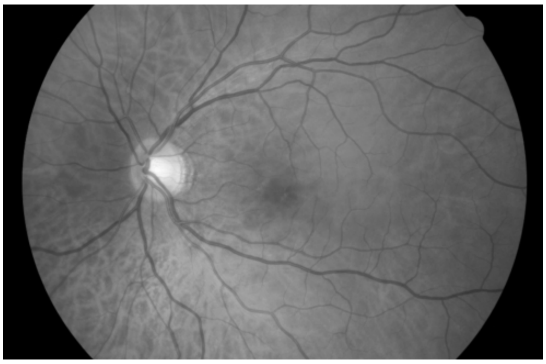
Introduction

Retinal vessel segmentation is a crucial task in medical imaging. This task utilizes a collection of 80 fundus images along with 20 secondary fundus images, each accompanied by their respective ground truth data. The primary objective of this assignment is to design an algorithm capable of automatically segmenting retinal vessels from fundus images. It involves identifying and extracting blood vessels from fundus images. The motivation behind this task stems from the growing demand for automated and accurate diagnostic tools that can assist healthcare professionals in diagnosing eye diseases quickly and reliably. Traditional methods of analyzing retinal images often require manual intervention, which is time-consuming and prone to human error. By automating the segmentation of retinal vessels, the process becomes more efficient and scalable, enabling more widespread screening in clinical settings.

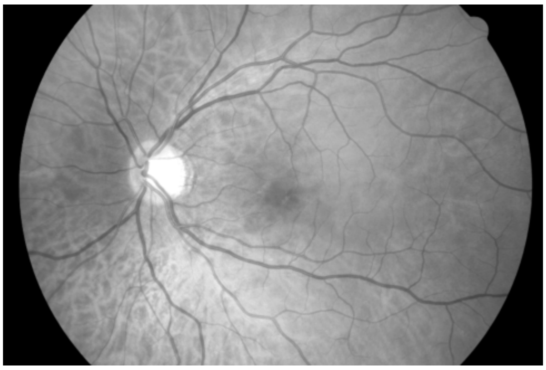
Furthermore, Retinal vessel segmentation has significant medical applications, including early disease detection, monitoring disease progression, and guiding surgical interventions like laser treatments and retinal surgeries. It also aids research into vascular health, providing insights into systemic diseases such as cardiovascular conditions, diabetes, and hypertension. Accurate segmentation helps identify abnormalities in blood vessels, enabling better diagnosis and treatment planning.

Description of Methods

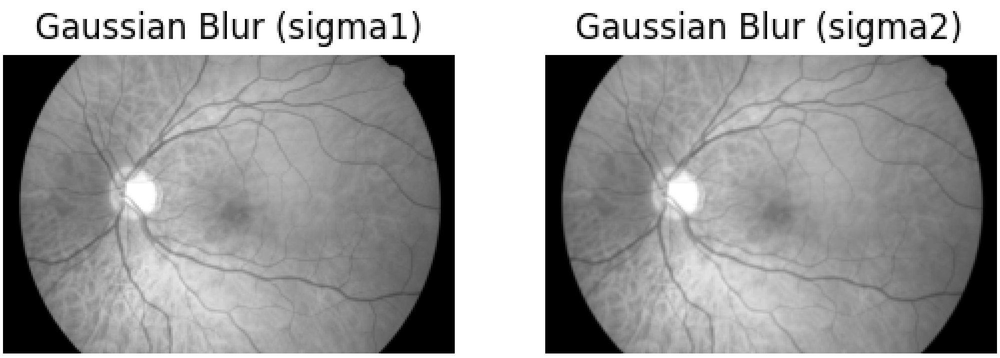
1. **Grayscale Conversion**
   The algorithm begins by converting the input image from the BGR color space to grayscale using OpenCV. This reduces the complexity of the image by focusing on intensity values.
   Output: 06.png



2. **Intensity Normalization**
   Then, the intensity of the grayscale image is adjusted. This process is based on the empirical rule. In this process, the mean and standard deviation of the pixel intensity values are calculated, and the intensities are clipped to the range of ±2σ. It removes extreme outliers, improves contrast, and normalizes the intensity.
   Output: 06.png

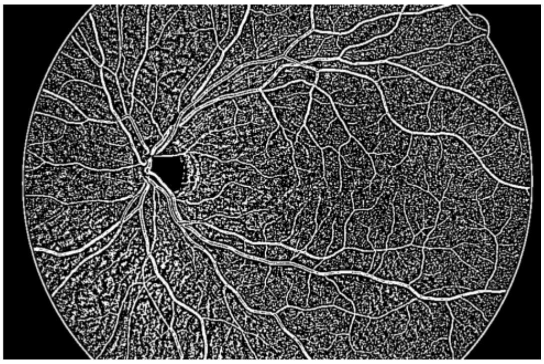


3. **Gaussian Smoothing**
   Afterward, two Gaussian blurs are applied to the normalized image with different kernel sizes and standard deviations. After excessive experimentation, the kernels should not be set to 0, as this resulted in an ineffective vessel segmentation process. Therefore, gaussian1 is set to a smaller sigma = 1 (with 3x3 kernel) to retain finer details in the image, while gaussian2 is set to a bigger sigma = 2 (with 5x5 kernel) for smoothing broader structures and suppressing further noise. This helps to smooth the image and reduce noise while preserving the edges.
   Output: 06.png

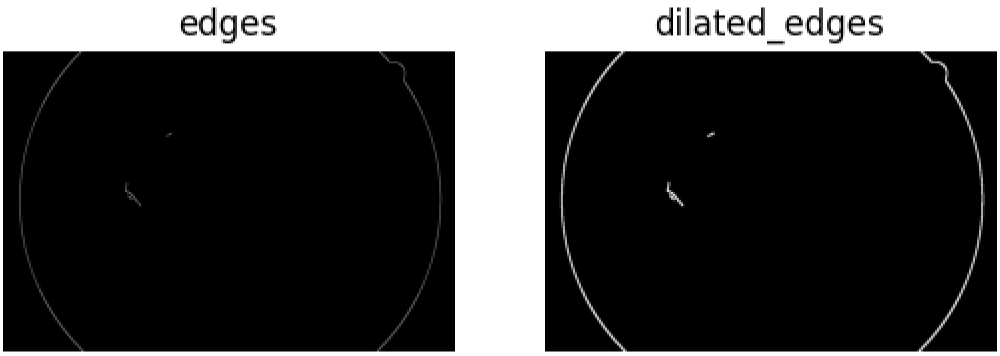


4. **Difference of Gaussian (DoG)**

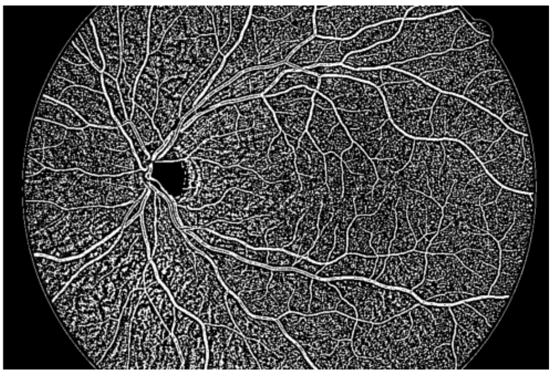
Next, the two Gaussian-blurred images are subtracted to create an edge-enhanced image. This technique highlights regions with rapid intensity changes, which are often associated with object boundaries.
Output: 06.png



5. **Canny Edge Detection**
The canny edge detection is applied to the original grayscale to remove the edge of the retina. It uses thresholds 1 of 200 and 2 of 210 so that it only detect the sharp edges (retina edges). The threshold cannot be set too low, otherwise actual vessels will be subtracted from the results. therefore, a 3x3 kernel dilation (iteration = 1) expands the edge. After that, the dilated edge is subtracted from the DoG output.
Output: 06.png



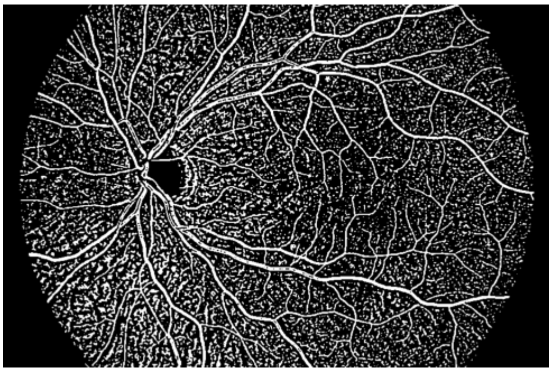
As you can see, the edge is subtracted.



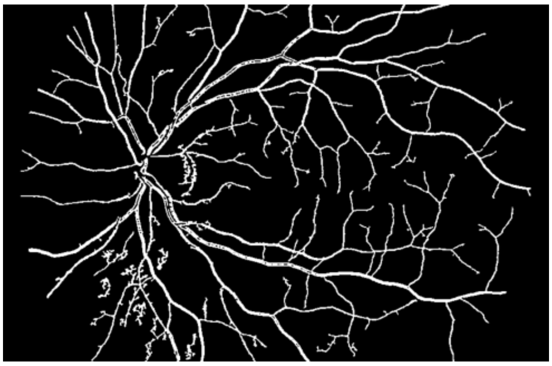
6. **DoG Normalization**
The DoG result is normalized to scale the output to a fixed range (0–1) for consistent contrast enhancement.

7. **Noise Reduction**

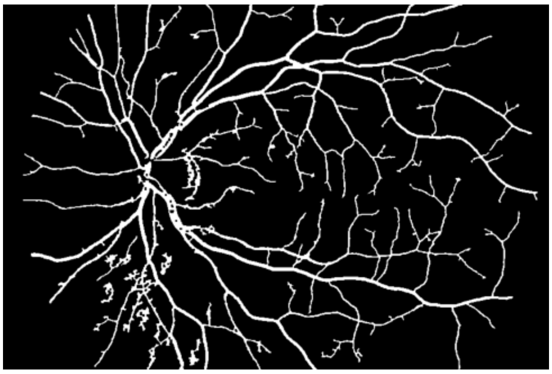
As you can see, the retina edge is broken and not fully subtracted. For that reason, the normalized DoG is processed using a median filter to remove small noise artifacts. A kernel of size 3 × 3 is used for the filter.
Output: 06.png



8. **Component Filtering**
   Additionally, component filtering is used to remove all components smaller than 200 pixels as they are treated as noise. Theoretically, a median filter could be considered redundant before component filtering. However, in practice applying a median filter before component filtering increased overall IoU. Hence, connected components analysis is performed to identify all components of the image. Then, each component's size is calculated and stored in a 1D array. After that, components larger or equal to 200 pixels are inserted into a mask. Each pixel in the mask is checked to see if they are valid and the mask is converted into a binary image.
   Output: 06.png



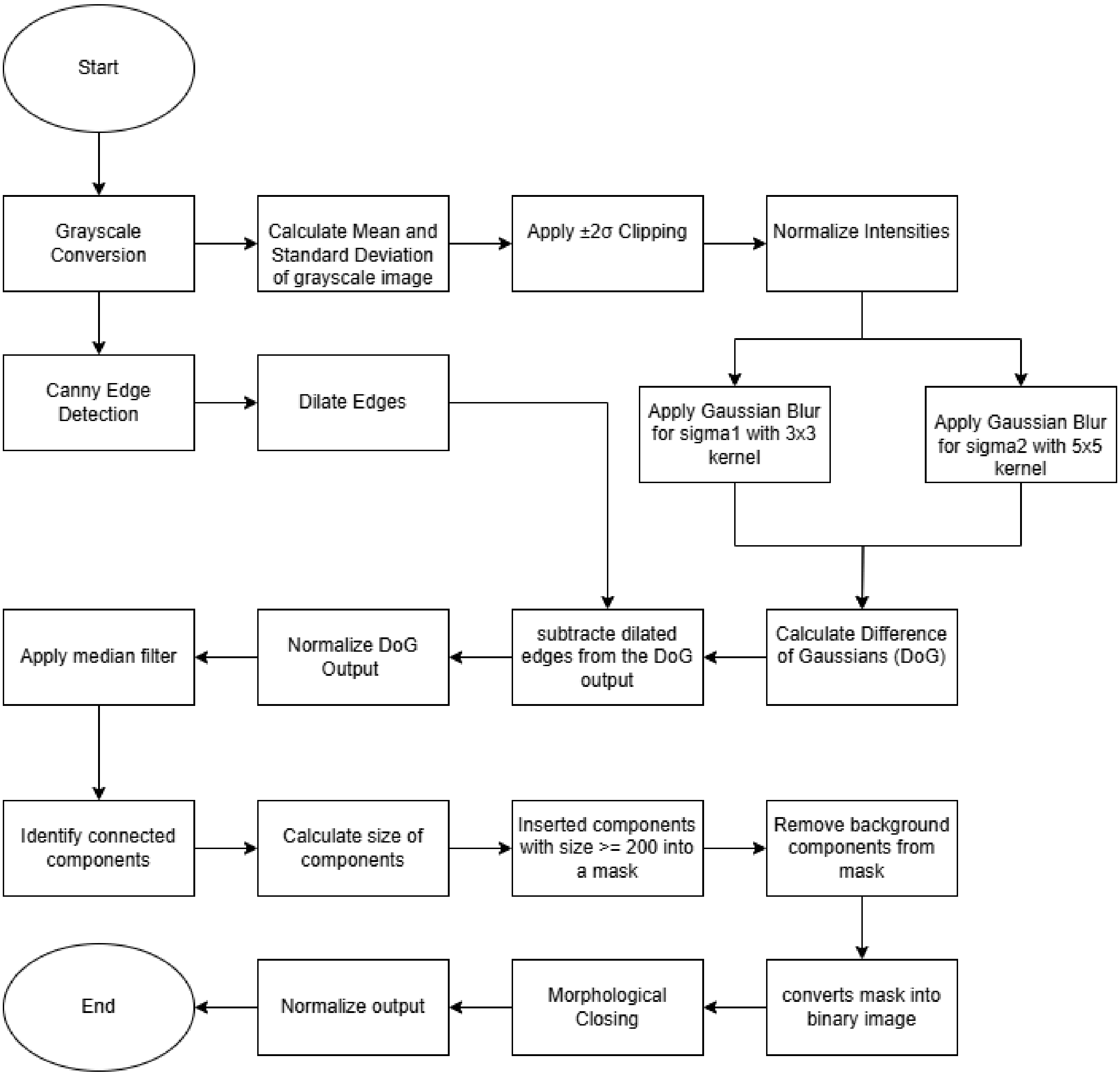
9. **Morphological Closing**
   Morphological closing is applied once to fill in the holes and connect some of the vessels. A kernel of size 3 × 3 is used for this operation.
   Output: 06.png



10. Output Normalization

Final normalization scales the cleaned segmentation mask to the range [0–1], preparing it for visualization or further analysis. This step ensures consistency in the output format.

The flowchart below shows the entire process.

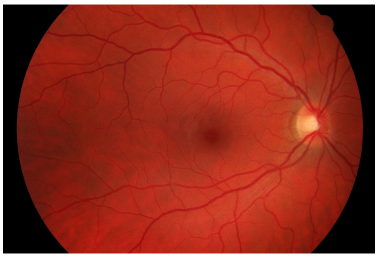
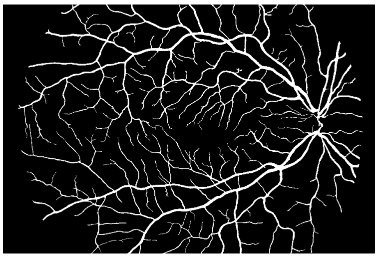
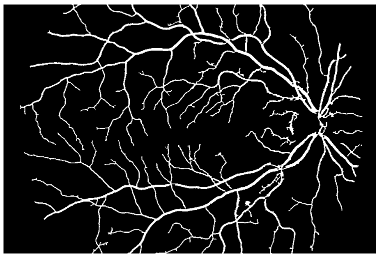
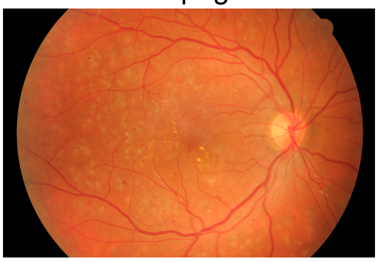
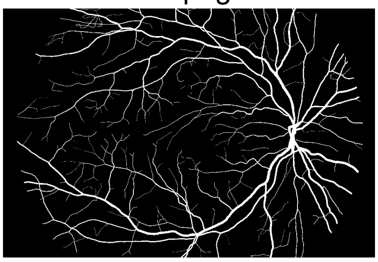
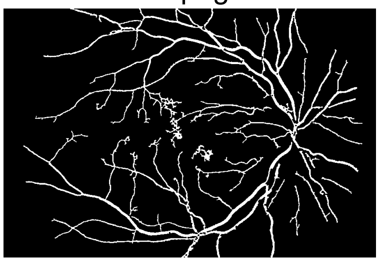
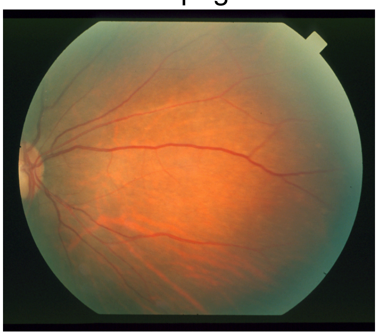
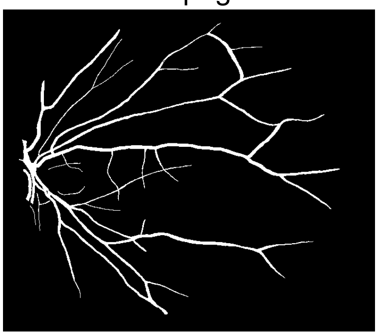
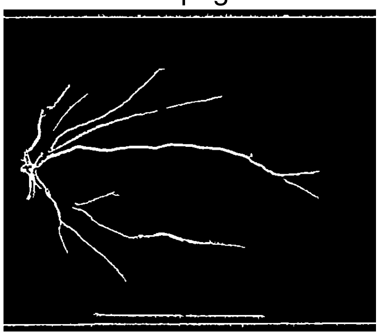


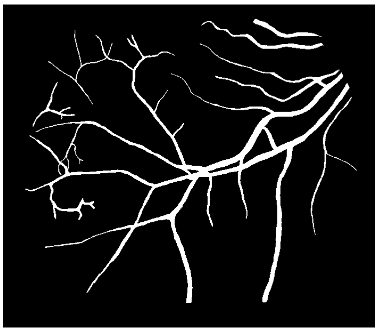
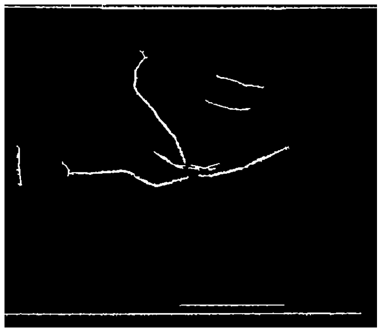
Results & Analysis

1. dataset

```
Adapted Rand Error: 31%
Precision: 73%
Recall: 65%
IoU: 52%
```

As you can see from the dataset results, the Adapted Rand Error is 31%. This suggests that 31% of the pixel pairs are incorrectly classified compared to the ground truth. Moreover, the precision (73%) is higher than the recall (65%). This indicates that the algorithm is generally good at correctly predicting positive cases, but it may miss some actual positive cases. Furthermore, the IoU is 52%. This means that there is a 52% overlap between the predicted components and the ground truth components.

Test Images	Groundtruth Images	Output Images	Results
33.png 	33.png 	33.png 	Error: 0.1887 Precision: 0.8394 Recall: 0.785 IoU: 0.6825
22.png 	22.png 	22.png 	Error: 0.3174 Precision: 0.655 Recall: 0.7126 IoU: 0.5182
78.png 	78.png 	78.png 	Error: 0.5066 Precision: 0.5933 Recall: 0.4223 IoU: 0.3275
80.png	80.png	80.png	Error: 0.8021 Precision: 0.4133 Recall: 0.1301 IoU: 0.1098

The table above shows images from the dataset folder. We can see that, the algorithm performs best on sharp fundus images with well-lit and noiseless backgrounds. 33.png is an example of a sharp image with a well-lit and clean background, while 22.png is an example of a sharp image with a well-lit background containing specks of noise. 78.png is an example of a partially blurry image with an inadequately lit background, and 80.png is an example of a blurry image.

Based on the evaluation, 33.png has the best results. This is because the vessels are very clear and defined, making them easily distinguishable for the algorithm. Moreover, the background is free from concentrated noise, such as white spots, light streaks, and speckles. However, there are some blurry black sections in the background. These sections cause the algorithm to overlook minor vessel branches.

In 22.png, the results are not as high as in 33.png. This is due to the Intense brightness directed onto the retina, which causes small vessels to blend in with the surrounding background, making them harder for the algorithm to detect. Moreover, the bright white dots in the retina also caused difficulties, since the algorithm mistakenly segmented them as vessels. With these in mind, the algorithm still segmented out all the large and moderate-sized vessels promisingly.

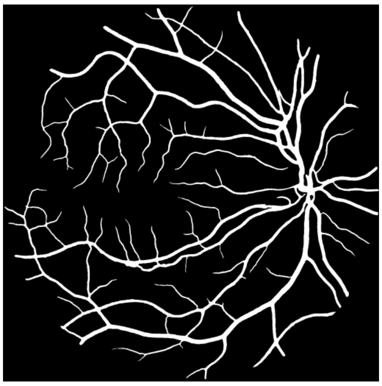
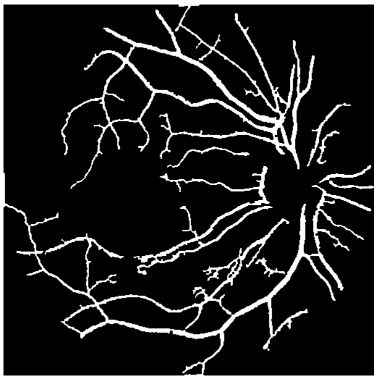
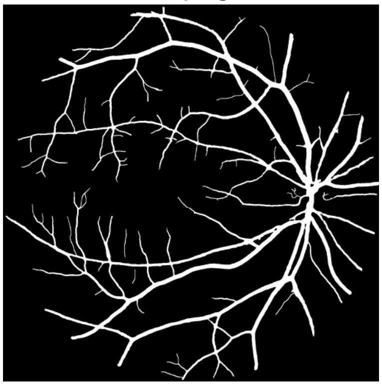
In 78.png, we can see that only large vessels in the middle are segmented. Firstly, the small and moderate-sized vessels are less visible due to insufficient brightness in the image, making them harder to distinguish from the background. Additionally, vessels near the edge are not visible due to complete darkness.

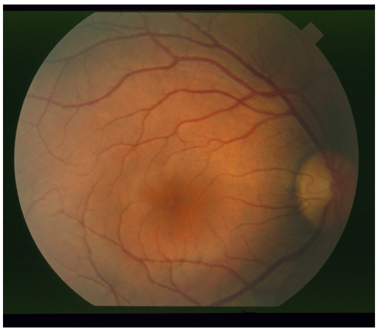
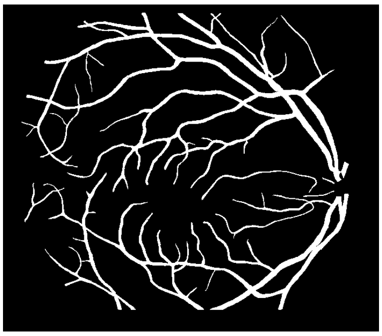
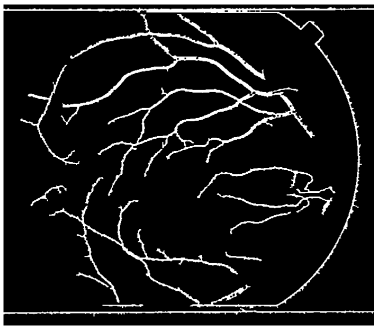
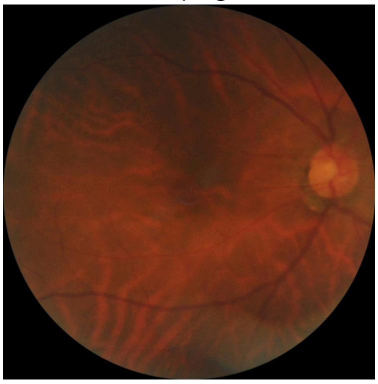
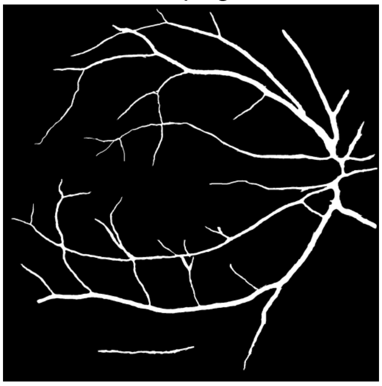
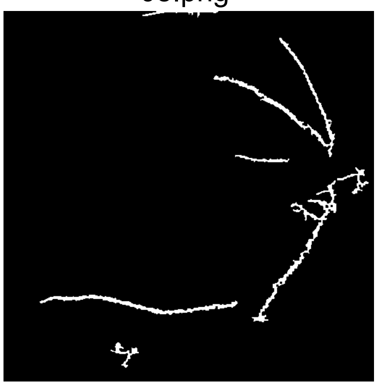
From the evaluation, 80.png has the worst results. This is due to the blurriness of the vessels in the image. The lack of sharpness reduces the contrast between the vessels and the background, making it challenging for the algorithm to accurately segment the vessels. The algorithm completely misses small and moderate-sized vessels. Even though the vessels in the edge region are large, they were missed due to the severe blurriness. Only a few vessels in the middle are segmented, but the segmentation is incomplete, with large vessels appearing thinner or fragmented.

2. **add_dataset**

```
Adapted Rand Error: 33%
Precision: 78%
Recall: 59%
IoU: 50%
```

As you can see from the add_dataset results, the Adapted Rand Error is 33%, which is a slight increase from the dataset result of 31%. Moreover, the precision of the add_dataset (78%) is higher than that of the dataset (73%), indicating that the algorithm better identified positive cases in the add_dataset. On the other hand, the recall of the add_dataset (58%) is lower than the dataset recall (65%), suggesting that more actual positive cases were missed. Furthermore, the IoU of add_dataset is 50%, which only decreased by 2% from the dataset IoU of 52%. These differences are not significant and indicates that the algorithm's overall performance remains consistent. Overall, the trade-offs observed highlight the algorithm's adaptability to different datasets while maintaining a consistent level of accuracy in vessel segmentation.

Test Images	Groundtruth Images	Output Images	Results
14.png 	14.png 	14.png 	Error: 0.1989 Precision: 0.8417 Recall: 0.7643 IoU: 0.6682
07.png 	07.png 	07.png 	Error: 0.317 Precision: 0.7523 Recall: 0.6255 IoU: 0.5186
20.png	20.png	20.png	Error: 0.4662 Precision: 0.6423 Recall: 0.4567 IoU: 0.3641

			
08.png 	08.png 	08.png 	Error: 0.5709 Precision: 0.8137 Recall: 0.2914 IoU: 0.2732

The table above shows images from the add_dataset folder. Similar to the dataset folder, the algorithm performs best on sharp fundus images with well-lit and noiseless backgrounds. Based on observations, add_dataset images are less well-lit, less-sharp, and they contain a significant amount of concentrated noise, such as white spots, dark sections, light streaks, intense bright areas and speckles. For example, 14.png is an example of a relatively sharp image with a partially bright and partially clean background, while some like 07.png are sharp images with partially bright areas. 20.png is an example of a Faintly blurred image with dark edges, and 08.png is an example of an extremely blurry and dark image abundant with light streaks.

Based on the evaluation,14.png has the best results. This is because the vessels are relatively clear and defined, with minimal noise. This allows the algorithm to accurately segment the vessel structures, resulting in a higher-quality segmentation compared to the other images in the add_dataset folder. However, the background is partially lit with dark areas, which causes the algorithm to miss minor vessel branches.

In 07.png, the results are lower compared to 14.png. This is because some areas of the image are only partially bright, which causes small vessels to become transparent, making them harder for the algorithm to identify. Additionally, certain bright spots on the retina caused issues, as the algorithm incorrectly identified them as vessels. The edge of the retina was also partially segmented as vessels, likely due to suboptimal image quality and contrast, which hindered the complete removal of the retina edge. However, the algorithm successfully segmented most of the larger and moderately-sized vessels, yielding relatively promising results.

In 20.png, the results are even worse. Firstly, the upper and lower edge of the image and half of the retina edge is segmented as vessels. This misclassification might be due to the lack of proper contrast between the retina and the vessels, as well as the presence of foreign structures in the image. Additionally, the lighting inconsistencies and blurred regions in the image further complicate accurate vessel segmentation. As a result, the algorithm struggled to distinguish between the actual vessels and the surrounding structures, leading to significant segmentation errors. The algorithm missed the vessel tails in the middle due to the dark spot there. Additionally, it also failed to detect vessels in the darker areas of the retina. The combination of these factors resulted in incomplete vessel segmentation, particularly in areas with poor lighting or insufficient clarity, affecting the overall performance of the algorithm.

From the evaluation, 08.png has the poorest results. However, it is noteworthy that its IoU score in the add_dataset folder (0.2732) still surpasses the lowest IoU in the dataset folder (0.1098). This underscores the critical role of vessel sharpness in the algorithm's performance, as the worst result (80.png) in the dataset folder has the blurriest vessels. In 08.png, the algorithm fails to detect small and moderate-sized vessels entirely due to their faintness and blurriness. This issue is further exacerbated by light streaks, which overwhelm and obscure these vessels, hindering accurate segmentation. Only a few larger vessels are detected, but the segmentation is incomplete, with these vessels appearing thinner or fragmented.

Suggestions for Improvement

Based on the results and analysis, we can see that there is room for improvements. Therefore, I would like to suggest improvements like the ones shown below:

1. **CLAHE (Contrast Limited Adaptive Histogram Equalization):** To adjust contrast locally to avoid over-amplification in bright areas while enhancing visibility in darker regions. This will help segmentation in sections with intense or inadequate brightness.

2. **Bilateral Filtering:** To Smooth images while preserving edges, making it ideal for reducing noise without blurring vessel boundaries. This approach is well-suited for mitigating artifacts such as white spots, light streaks, and speckles.

3. **Unsharp Masking:** To enhance the edges by subtracting a blurred version of the image from itself. This technique can tackle the blurriness that 80.png displays.

This assignment was completed in partial collaboration per course policy. I discussed concepts and shared insights with the following:

1. Yap Choo Kath Moon: Provided tips on using the canny edge detector for retina edge removal and shared advice on applying connected component analysis.
2. Lim Jun Jie: Suggested using intensity normalization for grayscale images based on the 65-95-99.7 rule.