# Segmentation of Vessels in Retinal Images

Team Members: Shiyu Wang, Enhao He, Jiacheng Ma, Joe Wang

# 1 Introduction

# 2 Related Work

# 3 Methods

## 3.1 Data Loading and Preprocessing

The DRIVE dataset is loaded by reading the retinal images, manually segmented vessel masks, and background mask data using libraries such as `tifffile`, `PIL`, and `numpy`.
Preprocessing part mainly focuses on the **Image Augmentation**. We enhance images using adjustments to color, brightness, and contrast (implemented in `augment_color`) to mitigate uneven illumination and improve vessel feature clarity.
Besides, we also remove the background from the original image to focus on the non-background pixels by using the background mask in the dataset.

## 3.2 K-means Clustering

K-means clustering is applied to segment the retinal images based on their RGB values. However, this approach introduces several challenges:

- **Determining the Optimal Cluster Number (`n_clusters`)**: Choosing too few clusters may lead to the merging of vessel pixels with surrounding tissue, whereas an excessive number of clusters can fragment vessels, causing a discontinuous representation of the blood vessels.
- **Variations in Illumination and Noise**: Differences in image illumination and inherent noise can further complicate the segmentation task, making it challenging to distinguish between vessel and non-vessel regions without additional preprocessing (such as image augmentation techniques).
- **Limited Feature Representation**: Relying solely on RGB values can be inadequate due to the often subtle contrast between vessels and the background. This limitation might require integrating additional features (such as gradient or texture information) to capture more features of vessel morphology.

### 3.2.1 Find the Best "n_clusters"

According to the structure of normal retinal photographs, the optimal number of clusters should be **4**, including: macula (dark shadow in the center), optic disc (bright shadow on the side), blood vessels, and others. But in computer's language, more detailed exploration is still needed.
The approach we used involves evaluating different numbers of clusters (ranging from 2 to 10) by computing the AUC value for each image and taking the mean value. The cluster number should be at least 2, representing the vessel pixels and other tissue pixels.

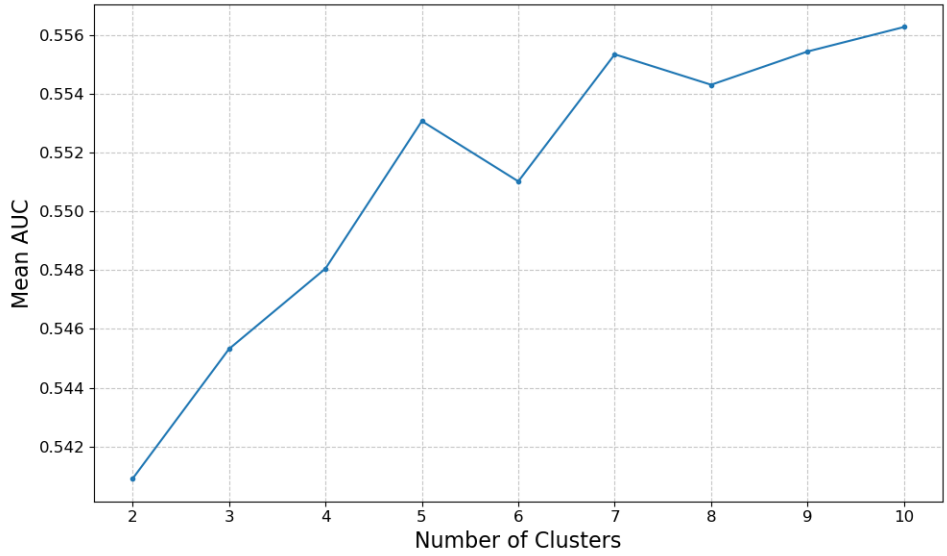



Figure 1: Mean AUC values for Different Number of Clusters.

As shown in Figure 1, the mean AUC value **increases** with the cluster number, but the growth is gradually slowing down.
Then, we try to make a further decision through visualizing the segmentation results for all images with different cluster numbers, which are shown below in Figure 2, 3, and 4.

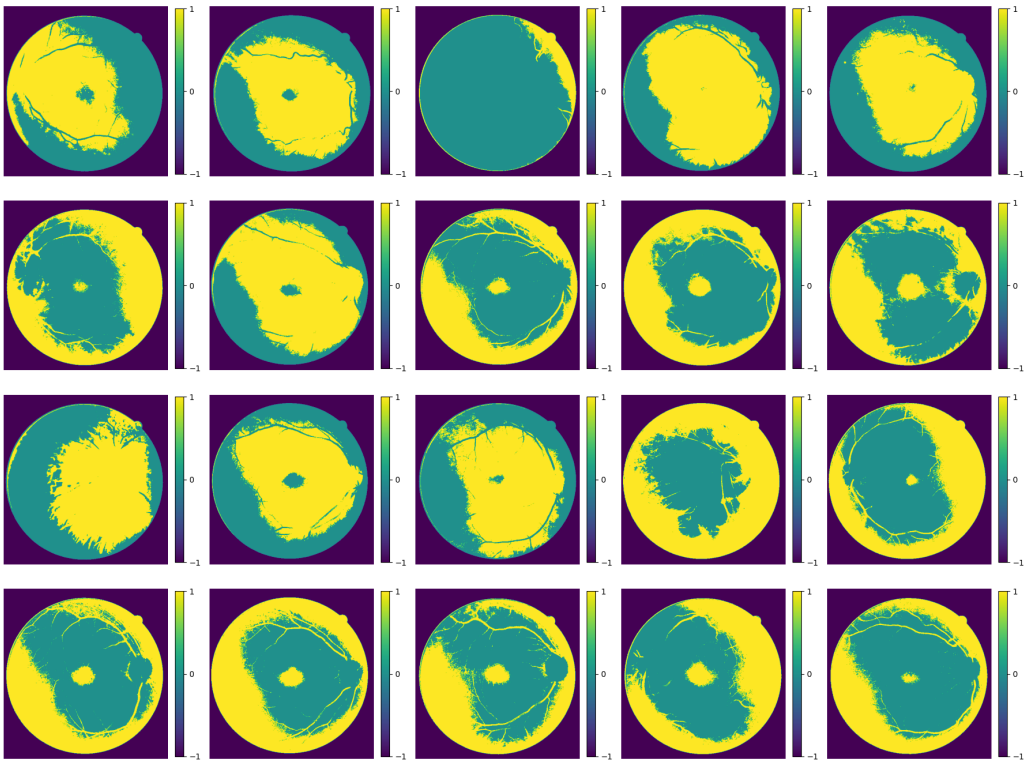


Figure 2: Segmentation Results (`n_clusters`=2).

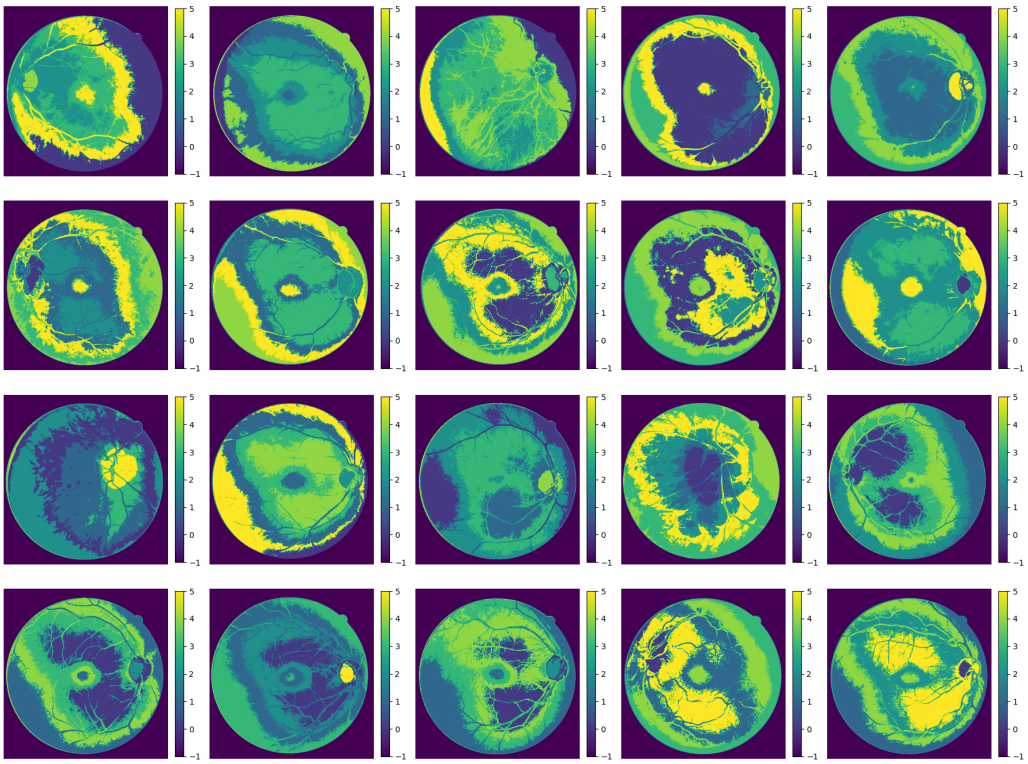


Figure 3: Segmentation Results (`n_clusters`=6).

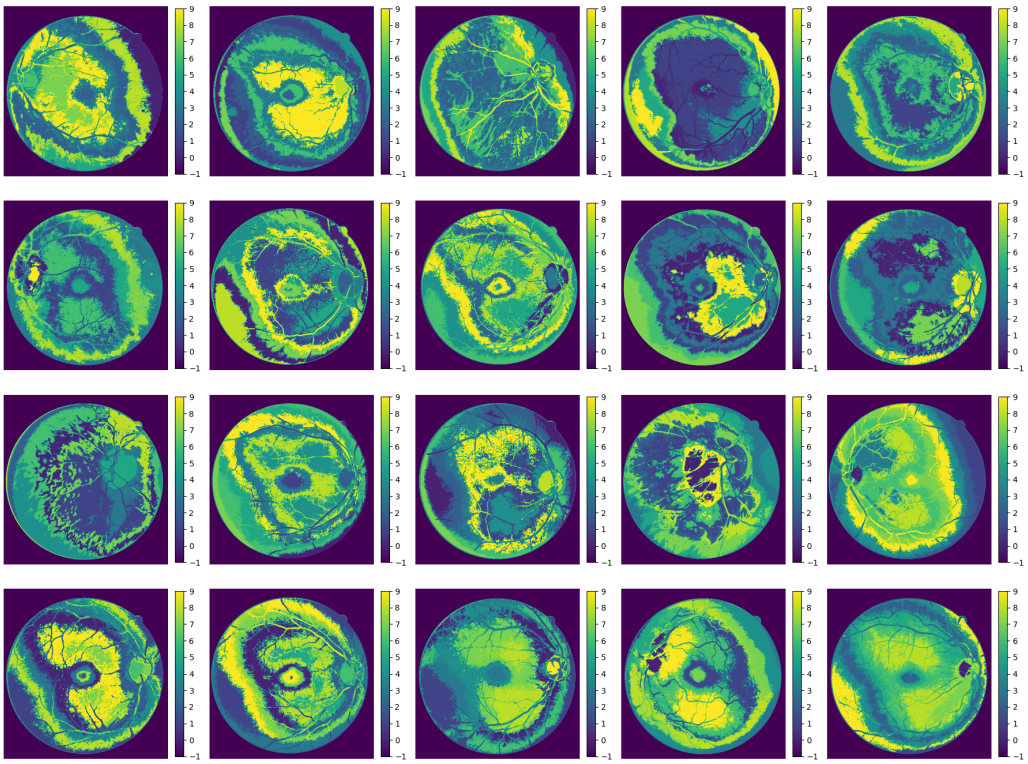


Figure 4: Segmentation Results (`n_clusters`=10).

There are much clearer blood vessel shapes can be observed with the increase of cluster number. However, the vessel pixels are fragmented into multiple clusters. Thus, the AUC value for a single cluster in each image is reduced and not sufficient to evaluate the segmentation performance.
Therefore, we decided to set `n_clusters` = 10 as an optimal compromise with an **optimized K-means algorithm**, which can handle the merging of clusters and improve the segmentation performance.
To match the biological explanation at the beginning of this section, we merge the top 3 clusters with the highest AUC values to form the final vessel segmentation mask (since $10/3 \sim 4$).

### 3.2.2 Optimized K-means Algorithm

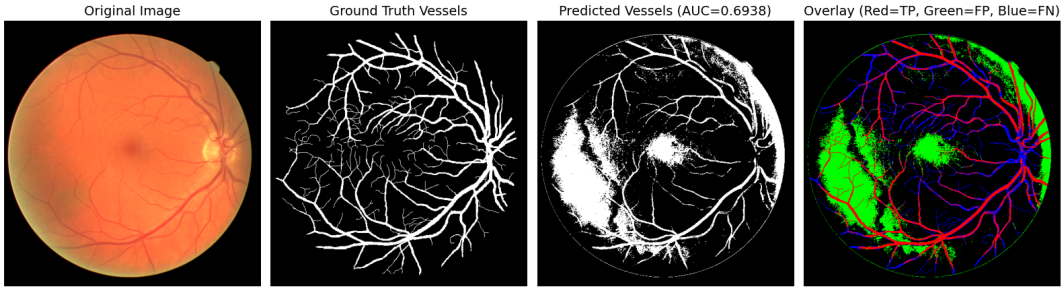
To address the challenge of correctly identifying vessel clusters, an optimized version of K-means is used. This method involves:

1. Running standard K-means on non-background pixels.

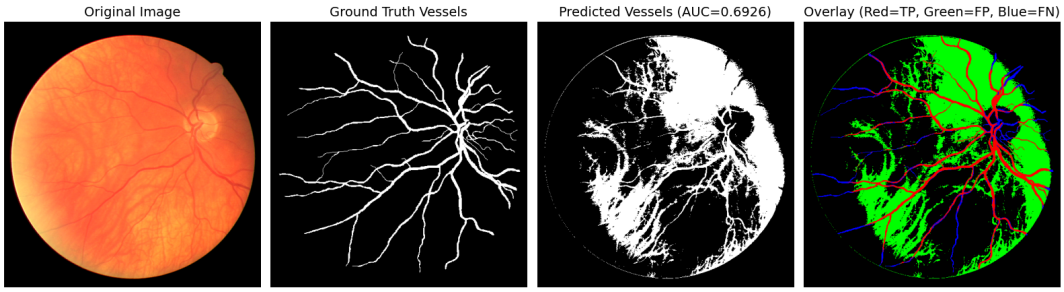
2. Evaluating each cluster by computing the AUC value between the binary prediction for that cluster and the ground truth.

3. Merging the top-K clusters with the top 3 AUC values to form the final vessel segmentation mask.

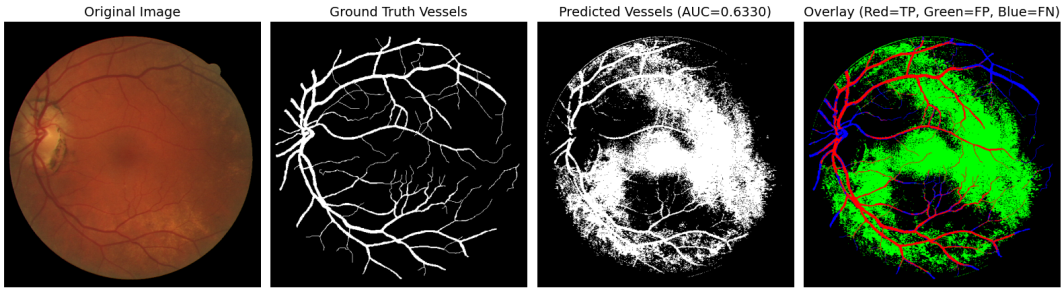
Figure 5 depict the three best segmentation examples using this approach, including the original image, the ground truth, the segmentation result with AUC values, and the segmentation overlay.




(a) First Best Example (AUC=0.6938)




(b) Second Best Example (AUC=0.6926)




(c) Third Best Example (AUC=0.6330)

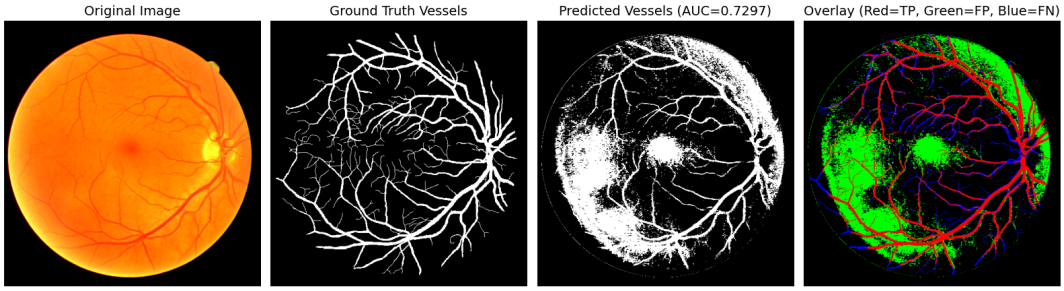
Figure 5: Optimized K-means: Top 3 Examples

As we can see, the biggest problem is that there are a lot of false positive pixels (FP, Green) in the segmentation results, which means that their colors are difficult to distinguish from blood vessels.
Therefore, we apply **image augmentation** to enhance the color differences, which may help to improve the segmentation performance.

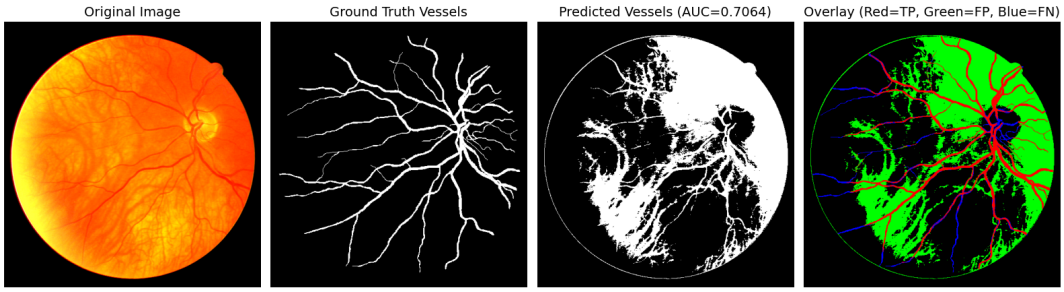
### 3.2.3 Image Augmentation with Optimized K-means

Image augmentation was further applied to enhance the quality of vessel segmentation. By pre-enhancing the input images, the clustering algorithm produces improved segmentation

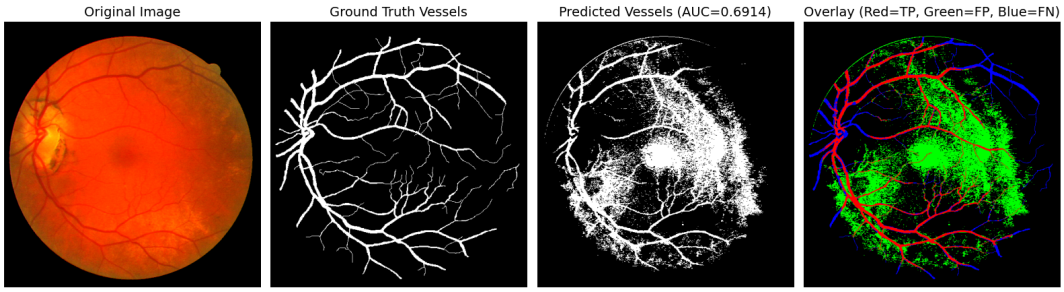
results. The results corresponding to the 3 best examples above after applying image augmentation are shown in Figure 6 below.




(a) First Best Example (AUC=0.6938 → 0.7297)




(b) Second Best Example (AUC=0.6926 → 0.7064)




(c) Third Best Example (AUC=0.6330 → 0.6914)

Figure 6: Optimized K-means with Augmentation: Corresponding 3 Examples

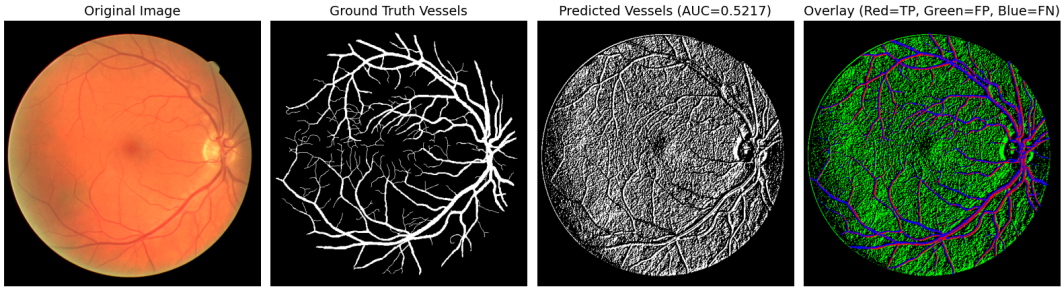
Apparently, the results are much better now. The false positive pixels (FP, Green) are significantly reduced, and the segmentation results are more accurate.

### 3.2.4 Direction-enhanced K-means Clustering

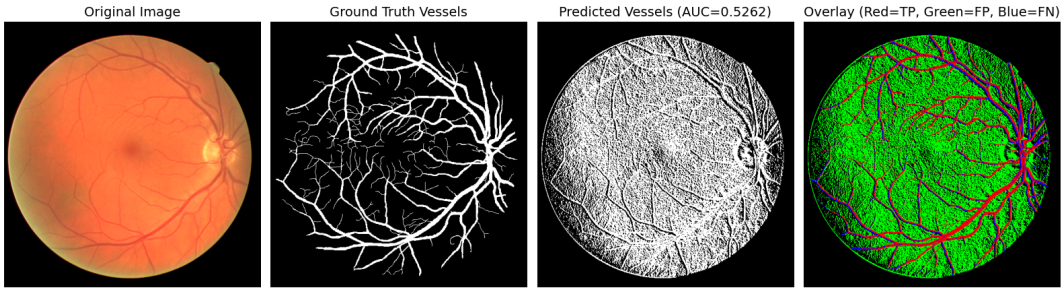
In order to further improve segmentation, we try to take advantage of the fact that blood vessels are long and thin. So we incorporate the **gradient-based features** into the clustering process.

In this approach, the original RGB features are augmented with gradient magnitude and gradient direction (obtained via the Sobel filter), forming an enhanced feature vector. Such directional information helps to better delineate vessel boundaries. Two merging strategies

(merging the top 3 and top 5 vessel clusters) were tested, as illustrated in the following figures (Figure 7).



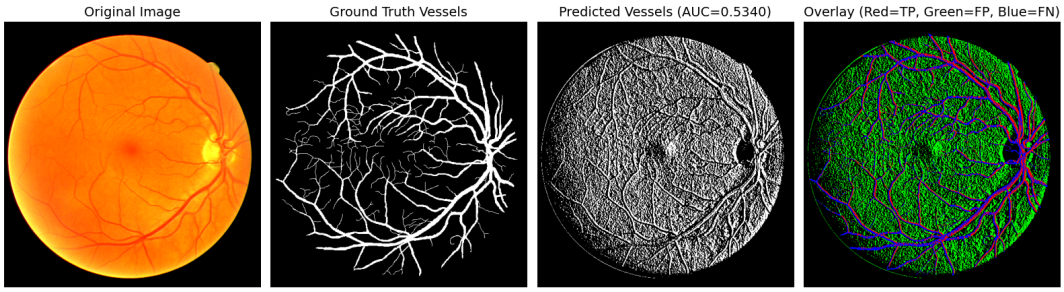

(a) Merging Top 3 Clusters (AUC=0.5217)



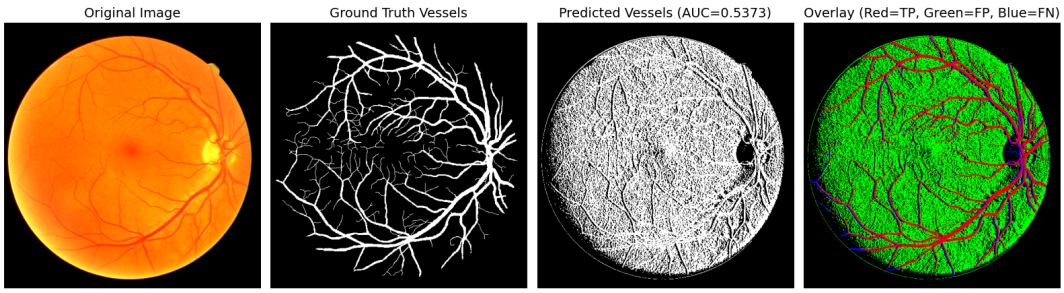

(b) Merging Top 5 Clusters (AUC=0.5262)

Figure 7: Direction-enhanced K-means

We can see that when the number of merged clusters increases, the true positive pixels (TP, Red) and false positive pixels (FP, Green) increase simultaneously. This is because the more top clusters we merge, the more TP and FP pixels are included in the final segmentation mask.
After applying image augmentation, additional improvements can be observed (see Figure 8).




(a) Merging Top 3 Clusters (AUC=0.5217 → 0.5340)




(b) Merging Top 5 Clusters (AUC=0.5262 → 0.5373)

Figure 8: Direction-enhanced K-means with Augmentation

The overall performance of the direction-enhanced K-means is not as good as previous methods, but it still provides a different perspective on the segmentation task by showing the potential of incorporating gradient features. Also, the results are shown in a different style, which is quite interesting.
Besides, we can also see the small improvement of the AUC value after applying image augmentation, which further verify the effectiveness of this technique.

### 3.2.5 Performance Evaluation and Comparison

The performance of the segmentation approaches is evaluated using accuracy, sensitivity, specificity, F1-score, and AUC value, which refer to related studies.
Table 1 summarizes the performance comparison between the classic K-means and the optimized K-means methods, both before and after applying image augmentation. (Not include the direction-enhanced K-means is because it is just a try and not the main focus of this project.)

Table 1: Performance Comparison: Before (Black) and After (Red) Image Augmentation

| Metric | Classic K-means | Optimized K-means |
|---|---|---|
| Accuracy | 0.8062 → 0.8404 | 0.6193 → 0.6938 |
| Sensitivity | 0.2228 → 0.2805 | 0.5862 → 0.6138 |
| Specificity | 0.8898 → 0.9200 | 0.6247 → 0.7057 |
| F1 Score | 0.2270 → 0.3115 | 0.2802 → 0.3393 |
| AUC value | 0.5563 → 0.6003 | 0.6054 → 0.6598 |

Through comparison of evaluation metrics, it is evident that image augmentation has a positive impact on both K-means algorithms.
Although classic K-means shows better performance in accuracy and specificity, the optimized K-means algorithm demonstrates significant advantages in key metrics such as sensitivity and AUC value, especially when combined with image augmentation techniques, providing more effective discrimination between vessel and non-vessel regions.
This suggests that the optimized K-means algorithm has stronger recognition capabilities in vessel segmentation applications, making it particularly suitable for medical image analysis tasks that require higher detection rates.

### 3.2.6 Conclusion and Future Work

For this part, we explored classical unsupervised machine learning techniques for retinal vessel segmentation. Through the use of K-means clustering and its optimized variations, including image augmentation and the incorporation of gradient features, the proposed methods demonstrated significant improvements in segmentation performance as measured by standard evaluation metrics.
For future work, the following directions are suggested:

- **Improved Image Augmentation:** In addition to traditional geometric transformations (e.g., rotation and flipping), consider advanced photometric adjustments, including saturation, hue, and white balance to mimic different lighting conditions.
- **Enhanced Feature Extraction:** Incorporating additional features beyond raw RGB values and gradients, such as texture descriptors, fractal-based features, or deep feature representations, to better capture vessel morphology.

## 3.3 Random Forest Classification

The random forest algorithm, a robust ensemble learning method, can be effectively applied to retinal image segmentation by leveraging its ability to classify pixels into anatomical or pathological regions through supervised learning. In this context, features such as pixel intensity, texture gradients, local binary patterns, and responses to edge-detection filters (e.g., Sobel or Gabor) are extracted from retinal images to capture structural details of blood vessels, lesions, or the optic disc. During training, the algorithm constructs multiple decision trees, each trained on random subsets of the data and features, to predict pixel-wise labels from annotated ground-truth masks. At inference, the ensemble aggregates predictions across trees, assigning each pixel to the most probable class (e.g., vessel vs. non-vessel), while mitigating overfitting through majority voting.

### 3.3.1 Training Process

The gist of Random Forest is to build multiple decision trees and aggregate their predictions. The training process involves the following steps:

1. **Feature Extraction**: Extract features from the training images, in our cases, the features are simply the RGB values of the pixels.

2. **Random Sampling**: Randomly select subsets of the training data and features to build each decision tree, ensuring diversity among trees.

3. **Tree Construction**: For each subset, construct a decision tree by recursively splitting the data based on feature thresholds that maximize information gain or minimize Gini impurity (which is the taken approach).

   **Gini impurity** (for class $k$):

$$G = 1 - \sum_{k=1}^{K} p_k^2$$

   where $p_k$ is the proportion of samples of class $k$ in the node.

   **Information gain** (using entropy $H$):

$$H = -\sum_{k=1}^{K} p_k \log p_k$$

$$\text{Information Gain} = H_{\text{parent}} - \sum_{\text{child}} \frac{N_{\text{child}}}{N_{\text{parent}}} H_{\text{child}}$$

4. **Prediction Aggregation**: At inference, aggregate predictions from all trees using majority voting to assign the final class label to each pixel.

Through the whole training process, totally 10 images, $10 \times 584 \times 565 = 3299600$ pixels are trained upon. The pixels are separeted into 3 classes: 0 for masked boundary, 1 for background and 2 for blood vessels. Below is a typical prediction result given by the classifier:

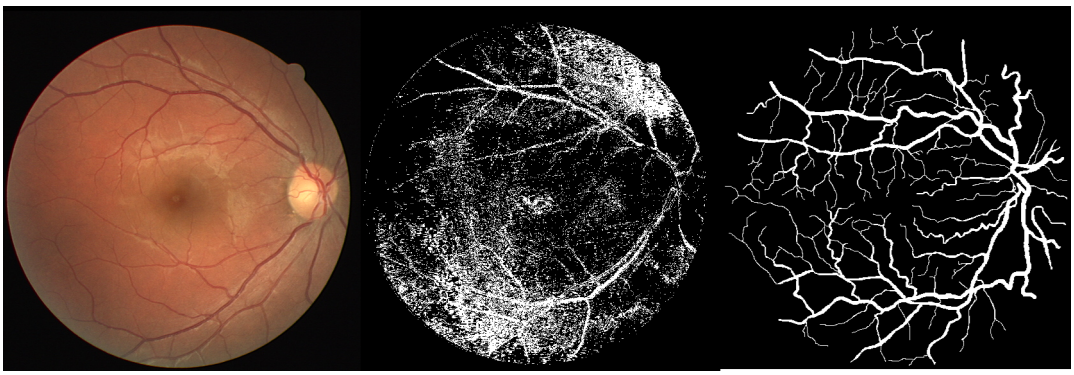


Figure 9: 1st image is the original image, 2nd image is the manually labeled image for training, 3rd image is the predicted image by the random forest algorithm.

Table 2: Metrics for the segmentation Task

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| background | 0.91 | 0.80 | 0.85 | 198755 |
| retinal | 0.24 | 0.44 | 0.31 | 29073 |
| accuracy | | | 0.75 | 227828 |
| weighted average | 0.82 | 0.75 | 0.78 | 227828 |

## 3.4 Conclusion

### 3.4.1 Reflection and Future Work

Despite our efforts, the output given by our classifier are still far from satisfactory compared with the the provided manual data used for training. We want to identify some potential reasons and improvement that can be made:

- **The limiation of RGB under different Saturation**: Both training and testing images are of different saturation, meaning that the pixels of same RGB value may be in completely different category in different images. This suggests that using only RGB value for classification is vastly insufficient. We can try to use other features such as texture, gradient, etc. to improve the performance. Further more, to make sure localness between pixels is reflected in training, we can consider including the pixel's neighbors's color in the feature set.
- **Feature Selection**:
- **Hyperparameter Tuning**: