Segmentation of Retinal Blood Vessels
A comparison between different segmentation techniques

BIOM 5202 - Group 1:
Saman Arianpour
Nihal Kulkarni
Shahrzad Baghestani
Guilherme Contesini

***Abstract*—Retinal images reveal many signs of early diseases not only related to eye-sight but to many other diseases that are unrelated to human vision. Many of these signs are observed in the blood vessels located near the retina. Automated triage has long been hinted at as one alternative that would increase public health agencies' range and capabilities. Image processing is one of the triage steps that contribute directly to better results diagnosis. In this project, we investigate the quality of different segmentation methods when segmenting blood vessels. We used the open dataset assemble and provided by the STARE project, a well-established peer-review dataset that contains the raw retinal images, diagnose annotations as well as hand-labelled segmented images. Our results show that a combination of empirical procedures produced the best results. The quality of the results was addressed using metrics derived from the confusion matrix and the MSE statistic. We found that the presence of artifacts was the main source of error and variability within and across each of the methods.**

I. Introduction

The Canadian Council for the blind estimate that 8 million Canadians suffer from an eye disease that can lead to vision loss. Furthermore, eyesight morbidities cost roughly 9.5 billion annually to Canada's public healthcare system [1]. Signs of other diseases such as diabetes, high cholesterol, high pressure, and Multiple Sclerosis also appear in the retinal exam, making retinal exams even more comprehensive to the public health system [2]. An often-sought solution to this problem is to increase the number of retinal exams and diagnoses. This alternative is plausible but requires complex coordination between different technological areas of engineering, such as automation, image processing and image classification. Here, we expect to contribute to the triage phase by implementing image processing methods to segment crucial components of the retinal scan images, particularly the blood vessels. We believe that doing so will aid computer classification algorithms and medical professionals in diagnosing earlier signs of diseases. Image processing and its broad scope of techniques have already been proven effective and are widely used to assist and complement healthcare systems, increasing their precision and capacity.

The analysis and characterization of blood vessels in the retina are the principal indicators of eyesight diseases [1]. Their characterization involves a description of their colour, diameter, shape, connectivity, and contrast. In this project, our objective is to segment the blood vessels of retinal images, more specifically to identify and separate the blood vessels, maintaining their core morphological characteristic, such as shape, diameter, and connectivity, while removing undesired elements. For this project, we used the open image dataset maintained by the *STructured Analysis of the Retina* (*STARE*) project [3]. The *STARE* project has similar goals as ours, and the available images are ideal for the purpose of our project.

In this project, we implement multiple segmentation, and edge detector methods, applying them to the available retinal images in the *STARE* project dataset mentioned previously. We then analyze their performance and quality through the confusion matrix metrics and Mean Standard Error. We organized this report as follows. In the first section, Image Dataset, we present the *STARE* dataset in detail, showing previous works performed by Dr. Adam Hoover, what kind of image is in the dataset, their numbers, format and the hand-labelled information that accompanies each image. In the following section, Methods, we present and discuss the segmentation and edge detection procedures used to segment the retinal blood vessels. Later, in the section Results, we introduce and discuss the metrics suitable for quantifying our results and the challenges that hindered the overall quality of our results. Following, in the section Discussion, we speak about the challenges that we encountered and the knowledge that we acquired during the course of this project. Finally, in the section Conclusion, we close our report by discussing alternatives to our approaches. We

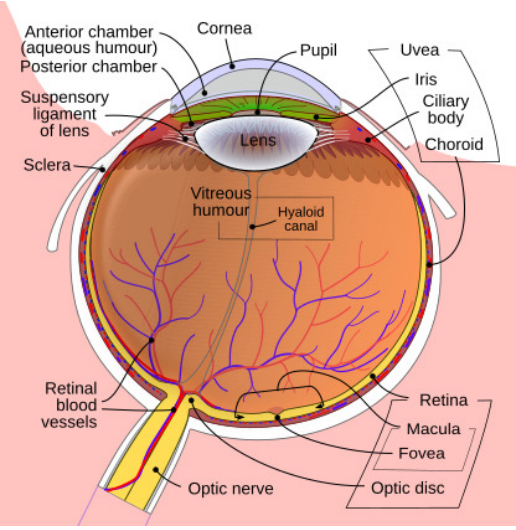

Fig. 1: Cross section schematics of the human eye. The schematics shows a comprehensive map of all the major structures that forms the human eye. Figure obtain from [7]

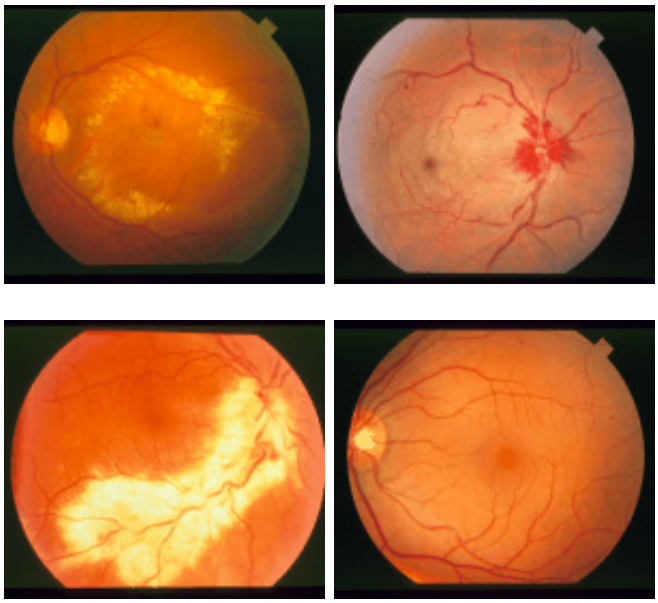

Fig. 2: Sample of images found in the dataset of the STARE project. All these presented images were drawn from the subset of hand labelled images, and are were chosen because they have a correspondent hand labelled version. These images can be find in STARE project website [3]

also provide suggestions and the experience gained while working with retinal images.

II. Database

The *STructured Analysis of the Retina* (*STARE*) project was originally proposed by Michael Goldbaum, M.D. in 1975 at the University of California. The project was funded by the U.S. National Institutes of Health, and had been used by different projects [3]. The *STARE* project contains a collection of 400 images of the human retina. These images were acquired, and provided by the Shiley Eye Center at the University of California, and resulted in at least two articles by Dr. Adam Hoover [4][5].

The image acquisition is noninvasive and it is relatively simple to perform compared to other medical image acquisition procedures such as MRI, ultrasound imaging, CT, and PET-scan. According to available documentation, all of the retinal images were acquired using a TopCon TRV-50 Mydriatic Retinal Camera [6]. In this retinal imaging technique, an optical camera is placed in front of the eye, aligned with the pupil to capture an image of the retinal surface, the rear inner surface of the eye. The image reveals the hidden structures of the retina, such as the optical nerve, the blood vessels, and the macula fovea 1. The image is captured in three color channels which reveals the color of each structure. Medical specialists compare this image to multiple references, to produce a diagnosis of the eye.

From the 400 images that the STARE project provides, a subset of 40 raw images contain hand-labelled annotations. These annotations indicate the "True" segmentation of the blood vessels and given by two medical professionals, Dr. Adam Hoover and Dr. Valentina Kouznetsova. This subset contains 120 images in total, of which 40 images are raw camera images, and 90 are black and white draws with the same dimension as the original correspondent camera images. From this subset, we selected 5 images that we judged to be representative of the wide variety of the raw images, suming up to 15 images processed. We also observed a significant variability in the medical hand labelling of the blood vessel. Although this was observed visually, we do not quantify this variability in this work.

III. Method

Image analysis of natural retina images always poses a challenge, since many of the images present different artifacts, a high variability, noise, morphological disparity and many other intricacies that hinder the direct application of segmentation methods. In addition, many of the important features appear faint or have characteristics that are very similar to unwanted features. Hence, the segmentation of blood vessels in this class of images is not performed directly, but rather it requires a compilation of sequential image processing techniques, that aims to address each of these undesirable

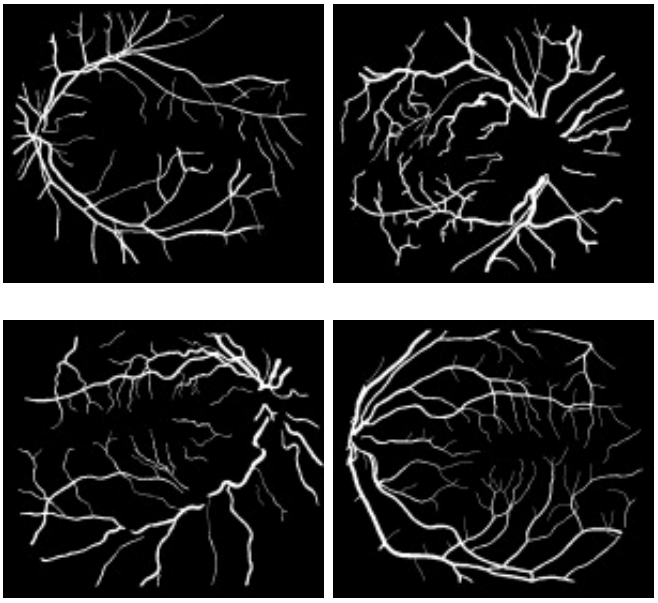

Fig. 3: Sample of images hand labelled by Dr. Adam Hoover. These segmented images are in respectively correspondence to the previous sample show in figure 2. All of these images can be find in STARE project website [3]

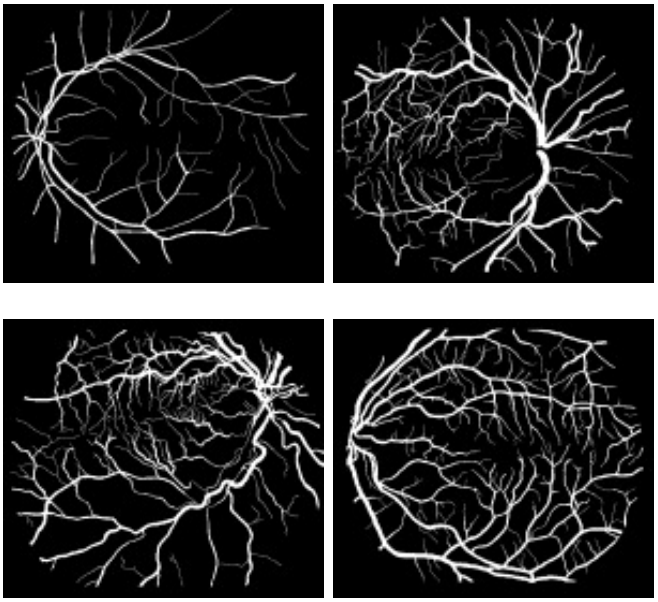

Fig. 4: Sample of images hand labelled by Dr. Valentina Kouznetsova. Similarly to 4 these segmented images are in direct correspondence to the previous sample show in figure 2. These images were can be found at STARE project website [3]

elements and artifacts separately. These sequential steps are often referred to as pipelines. These pipelines are often designed and based on the engineer's experience. Therefore is not unusual to see drastic differences in the pipelines, especially when multiple engineers collaborate on the same project. Here, we design our procedures and pipelines based on the techniques that we judged to be adequate for the task.

A. Hessian Matrix

The Hessian matrix is the first method that we implemented in our project. It belongs to a larger class called Ridge Filter, which are particularly efficient in identifying mountain ridges, rivers, pathways, contours or tubular structures. This characteristic make them ideal for detecting the boundaries of the blood vessel [9]. Although there are many different ridge filters, we focus on the Hessian Matrix since it provides the most information.

$$\begin{bmatrix} I_{xx} & I_{xy} \\ I_{yx} & I_{yy} \end{bmatrix} \tag{1}$$

The Hessian has a fairly simple form given by equation (1). It consist of the second derivatives, in respective to the x and y coordinates in the main diagonal, the partial cross derivatives in its off diagonal. Furthermore, the trace of the Hessian matrix, consisting of the sum of the terms from the main diagonal, equates to the Laplacian filter of the image. Meanwhile, the determinant of the Hessian Matrix contains information about the critical points, which, in an image correspond to the local maxima, minima and saddle points. One of the most important information that the Hessian matrix provides resides in the value of its eigenvalues. By applying an eigenvalues decomposition we can obtain information about which regions correspond to a corner, and edge or if it belong to a flat surface "a pixel belonging to a vessel region will be signalled by λ_1 being small (ideally zero), and λ_2 of a large magnitude and equal sign (the sign is an indicator of brightness/darkness)."[page - 132] [9].

The procedure for the Hessian matrix method follows these steps. First, we load the image and apply a Gaussian filter. Then we calculate the Hessian matrix alongside its eigenvalues and eigenvector. Having the eigenvalues, we normalize them and compute the similarity. Our resulting image is the outcome of these later steps [17]. Figure 5 shows the results of the Hessian method that we implemented.

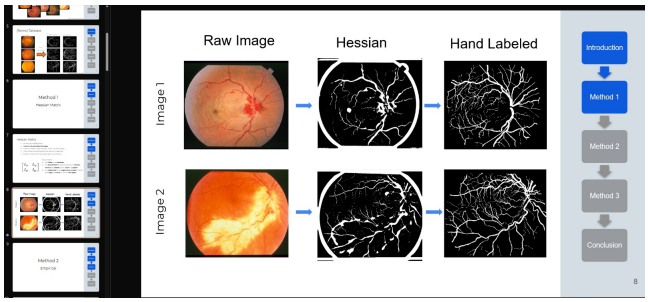

Fig. 5: Results from the Hessian matrix method.

B. Empirical Approach

The second method that we implemented followed an "empirical approach". It consists of an ensemble of steps from the different subjects within the image processing area. Therefore, it does not follow a central idea but rather combines known effects from different procedures, such as morphological operations, noise filtering, direct subtraction, and segmentation by thresholding.

The empirical method is described as the following. First, we load the image and convert it to grayscale. Later we apply a histogram equalization, concurrent with this process we filter the grayscale image with an average filter. Following these steps, we subtract the averaged filter from the image with an equalized histogram. In the following step, we apply a thresholding segmentation where the threshold value is the average of the mean above the average of the histogram, with the mean below the average of the histogram. Finally, we apply a morphological opening which gives us the final image [18].

C. 3D Segmentation

The active contour method is the last segmentation method we explored in this project. It belongs to the class of adaptive semi-automated level-set methods, where the user manually provides an initial, approximate area where the desired region lies. Here we create a procedure that improves the quality of the active contour method. We preprocess the image, filtering and masking it in the spatial and frequency domain. Our procedure also utilizes empirical knowledge to fine-tune our results.

Our pipeline consists of the following steps. First, we load the RGB images and manually select an initial region. This step gives us local information about the pixels near the blood vessels and the values for the background. We then convert the image from RGB colour to grayscale. With the grayscale image, we apply a "log" mask. Note that we tested different edge detection masks to compare which performed the best. We tested the traditional masks the "log", the "Sobel", "Prewitt" and

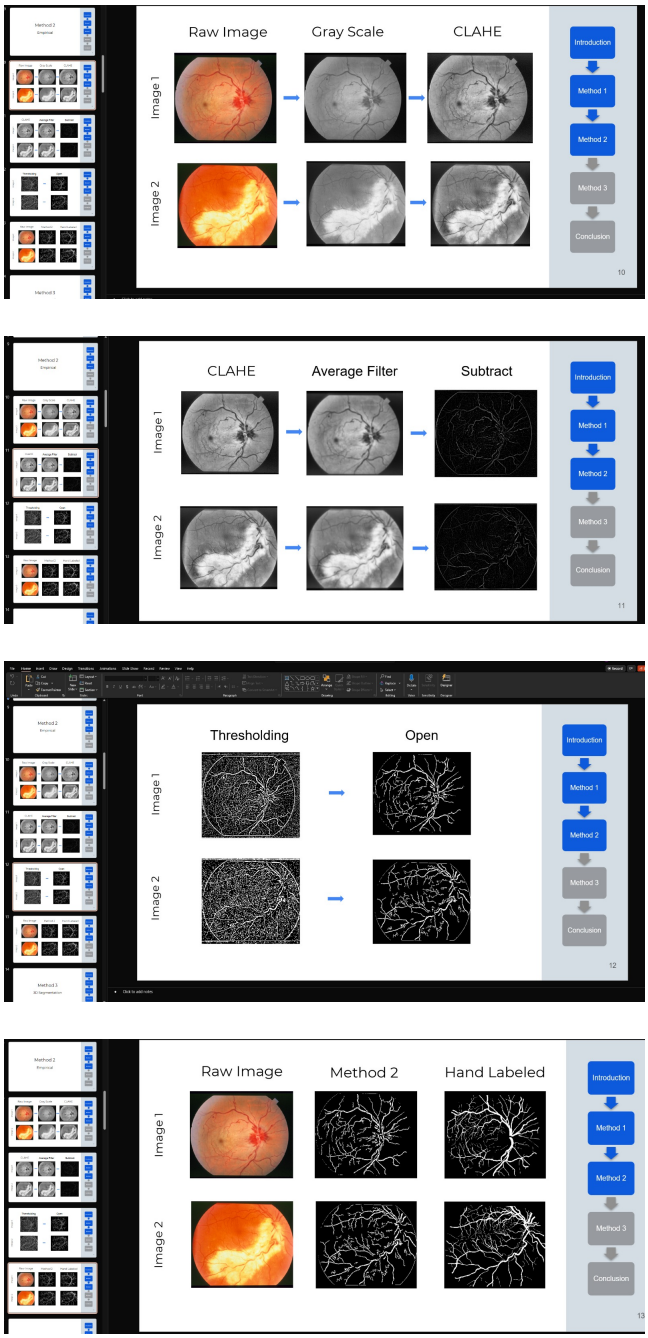

Fig. 6: Results from the Hessian matrix method.

the "Canny" [13]. These masks aim to identify the first edges and corners of the image. We then remove small artifacts using the bwareaopen [16], followed by the active contour method[15]. Continuing in the pipeline, we binarize our image using an empirical threshold of 0.1[12]. The best threshold value was obtained after applying a forward greed search algorithm. Later we apply a Fourier Transform followed by a lowpass filter and an Inverse Fourier Transform, bringing the filtered image back to the spatial domain [14]. The results of this method can be seen in figure 7.

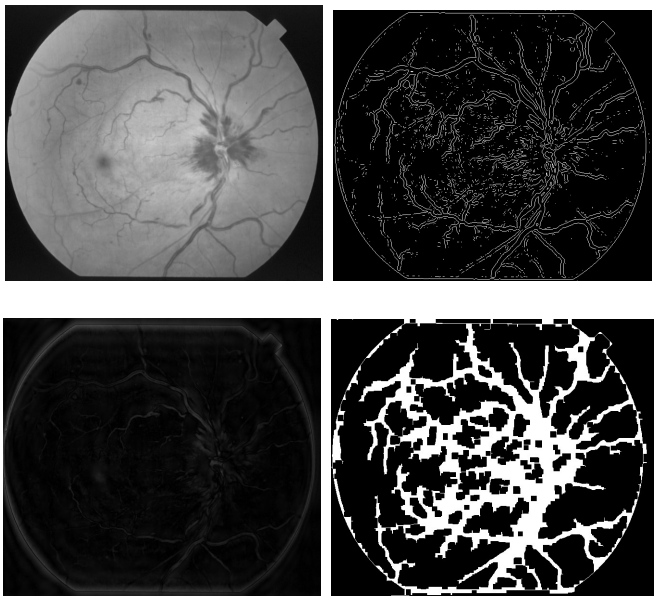

Fig. 7: Results of each step of the 3d segmentation produce develop for this project. (First row, Left) Original image converted to Grayscale without any modification. (First row, Right) Resulting edges and corners from the "log" mask. (Second row, Left) Resulting image after applying a lowpass filter. (Second row, Right) Final image showing the segmented blood vessels.

IV. Results

The old saying says that "a picture is worth a thousand words". Although this might be useful sometimes in our daily life, it is not valid when working on image processing. Qualitative descriptions of images solely are insufficient when describing the results obtained. Hence, we are required to extract quantitative measurements from our resulting image. There is an enormous amount of metrics that, when calculated, quantitatively describe the changes that a resulting image had. In our project, we chose the Confusion Matrix and the Mean Square Error (MSE) to quantify our results since we are interested in comparing how far the resulting images are from a hand-labelled images.

The old saying says that "a picture is worth a thousand words". Although this might be useful sometimes in our daily life, it is not valid when working on image processing. Qualitative descriptions of images solely are insufficient when describing the results obtained. Hence, we are required to extract quantitative measurements from our resulting image. There is an enormous amount of metrics that, when calculated, quantitatively describe the changes that a resulting image had. In our project, we chose the Confusion Matrix and the Mean Square Error (MSE) to quantify our results since we are interested in

		Predicted	
		Negative	Positive
Actual Values	Negative	TN	FP
	Positive	FN	TP

TABLE I: Template of the confusion matrix calculate between predicted images and actual images. We reiterate that in our project we assumed that the hand-labelled images by Dr. Adam Hoover correspond to the actual blood vessels. Note that the values of True Positives and True Negative is inverted compared with the traditional version of the confusion matrix. This inversion is due to the interpretation of the MATLAB function that considers black pixel "0", as being true values, and white pixel "1" as being false values.

comparing how far the resulting images are from hand-labelled images. Additionally, we calculate the variance and the standard deviation within each method to establish how stable they are when dealing with the wide variety of images within the *STARE* dataset.

The confusion matrix is not a metric but rather a rich structure from which several statistical metrics are calculated [10]. It is a type contingency table composed of the values of True Positives, True Negatives, False Positive and False Negatives, which captures the amount of correct and erroneous predictions compared to the actual values. It usually represented as the following I table.

In the case of image analysis, we take each pixel's brightness in each resulting image as the predicted values and the respective hand-labelled segmented image provided by a specialist as the true value. For each pair of pixels of this image pair, if the brightness of the pixels matches, we increment the value for true positive and true negative. Moreover, a similar white pixel gives us a true positive and a similar black pixel a true negative. In the case where the pixel's brightness disagrees, it sums up to the values of false positive, when a white pixel is expected, and of false negative, when a black pixel is expected.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

The statistical metrics derive from the ratios of the values found in the confusion matrix. Precision is fairly common metric which use the ratio of how many correctly predicted positives appear given all the expected in all predicted positives. Similarly, Recall is the ratio

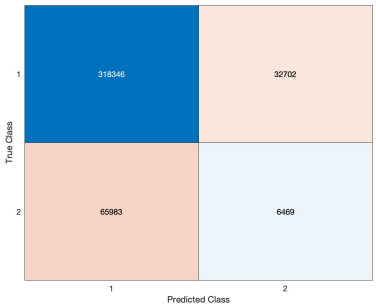

Fig. 8: This confusion matrix results from the processed images obtained when using the Hessian method. One of the confusion matrix obtained when comparing the resulting image to the true image. Same pair of images was used to across the different methods.

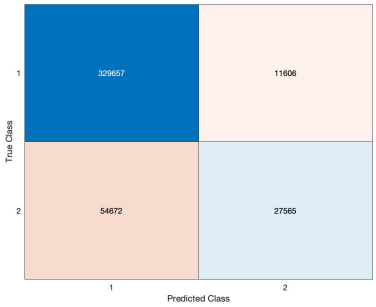

Fig. 9: Confusion matrix calculate similarly to the 8. This confusion matrix results from the Empirical method. Same pair of images was used to across the different methods.

of how many correctly predicted white pixels occurred given the total amount of actual white pixels.

For every pair of images we calculated their respective confusion matrix. This protocol was performed for every pair predicted-hand-labelled image. Shown bellow is one example of the confusion matrix obtain for each of the methods that we implemented.

The following table summarizes the values for precision, and recall calculate using each the previously shown confusion matrices.

The Mean Squared Error (MSE) is another metric often used to address how far the predicted results are

	Recall	Precision
Hessian Matrix	0.38%	0.25%
Empirical approach	35.51%	70.00%
3D segmentation	8.93%	16.51%

TABLE II: Table summarizing the resulting metrics for each of the methods.

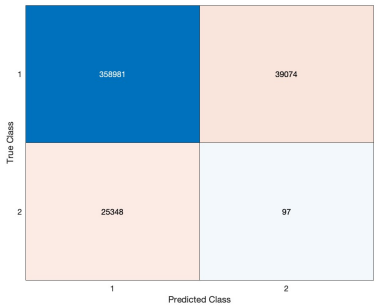

Fig. 10: Confusion matrix resulting from the 3D segmentation method. Same pair of images was used to across the different methods.

from an actual value. It is a measurement of variability employed during statistical analysis. In short, the MSE quantifies how far apart two images are from each other. The MSE is computed by adding the squared pixel-by-pixel difference between the predicted image and the actual, expected image. This calculation is performed for every pixel in the image and normalized by the total number of pixels. A low value of MSE indicates that the resulting predicted image is very similar to the expected image and interpreted as a good result. Larger values of MSE, on other hand, indicate that the predicted image is distant from the true image, meaning that the results were poor. A single value of the MSE does not suggest much but is commonly computed across different methods and procedures. In our project, we calculated the MSE for each of the methods 4.

$$\text{MSE}(K, I) = \frac{1}{N \cdot M} \sum_{i=0}^{M} \sum_{j=0}^{N} (K_{ij} - I_{ij})^2 \tag{4}$$

We also considered the resilience to image variability within each method. We used calculated the standard deviation within each method to quantify how much a method's MSE metric changes given the variability of images in the dataset. The standard deviation follows the traditional formula 5

$$\text{SD} = \sqrt{\frac{1}{N-1} \sum_{i=0}^{N} (x - \langle x \rangle)^2} \tag{5}$$

V. Discussion

Image processing is a broad discipline that includes noise reduction, artifact removal, feature and measurement extraction, image registration, and many other processes. The quality of the final result depend less

	MSE
Hessian Matrix	$(14.01 \pm 1.99) \cdot 10^3$
Empirical approach	$(7.87 \pm 2.01) \cdot 10^3$
3D segmentation	$(12.8 \pm 1.78) \cdot 10^3$

TABLE III: Mean Squared Error for each of the methods implemented. The variance within each method was calculated using a sample of 5 images. The same images were used for each method.

on the chosen method, but rather, it relies more on the choices made when build the overall procedure. Throughout the development of this project, we encountered different challenges, many of which we did not expect. We overcame them by discussing them among ourselves. Despite noise being one of the main concerns in image processing projects, in our project, it had minimal influence due to the quality of the original raw images. Instead, our major challenge was identifying and removing the numerous kinds of natural artifacts present in the images. Most of these artifacts varied largely across the images. Many of our procedures were developed using a try-and-error strategy where we vary parameters according to our experience.

VI. Conclusion

Retinal imaging is a viable alternative to diagnose early signs of eye-sight disease and even diseases unrelated to human sight. Although mass triage still presents several challenges, we hope that the results of this project contribute in some way or another to one of the steps required to overcome this challenge. We believe that improvements to the image segmentation of retinal blood vessels are still achievable. Given that the image also contained a diagnose label we recommend that this dataset might be used in an image classification approach. The availability of segmented images was paramount for the development and quality assessment of the resulting images. The availability of hand-segmented images allowed us to compare how the different approaches performed, both within each method and across the different methods without the need for eye care specialists.

VII. Appendices

A. Code for method 1

```
function [Dxx,Dxy,Dyy] =\
Hessian2D(I,Sigma)
if nargin < 2, Sigma = 1; end
% Make kernel coordinates
[X,Y]    = ndgrid(-round(3*Sigma):\
round(3*Sigma));
% Build the gaussian 2nd\
derivatives filters
DGaussxx = 1/(2*pi*Sigma^4) *\
(X.^2/Sigma^2 - 1) \
.* exp(-(X.^2 + Y.^2)/(2*Sigma^2));
DGaussxy = 1/(2*pi*Sigma^6) * (X .* Y)\
.* exp(-(X.^2 + Y.^2)/(2*Sigma^2));
DGaussyy = DGaussxx';
Dxx = imfilter(I,DGaussxx,'conv');
Dxy = imfilter(I,DGaussxy,'conv');
Dyy = imfilter(I,DGaussyy,'conv');
%METHOD 1 Hessian
I = rgb2gray(imread\
('Images\im0005.ppm'));
I=double(I);
defaultoptions = struct(\
'FrangiScaleRange', [1 10],\
'FrangiScaleRatio', 2,\
'FrangiBetaOne', 0.5, \
'FrangiBetaTwo', 15,\
'verbose',true,'BlackWhite',true);
% Process inputs
if(~exist('options','var')),
    options=defaultoptions;
else
    tags = fieldnames(defaultoptions);
    for i=1:length(tags)
         if(~isfield(options,tags{i})),\
         options.(tags{i})=\
         defaultoptions.(tags{i}); end
    end
    if(length(tags)~=\
    length(fieldnames(options))),
        warning('FrangiFilter2D:\
        unknownoption',\
        'unknown options found');
    end
end
sigmas=options.FrangiScaleRange(1):\
options.FrangiScaleRatio:options.\
FrangiScaleRange(2);
sigmas = sort(sigmas, 'ascend');
beta   = 2*options.FrangiBetaOne^2;
c      = 2*options.FrangiBetaTwo^2;
% Make matrices to store all \
%filterd images
ALLfiltered=zeros([size(I)\
length(sigmas)]);
ALLangles=zeros([size(I)\
length(sigmas)]);
% Frangi filter for all sigmas
```

```
for i = 1:length(sigmas)
    % Show progress
    if(options.verbose)
        disp(['Current Frangi\
        Filter Sigma:\
        ' num2str(sigmas(i)) ]);
    end
    % Make 2D hessian
    [Dxx,Dxy,Dyy] = Hessian2D(I,\
    sigmas(i));
    % Correct for scale
    Dxx = (sigmas(i)^2)*Dxx;
    Dxy = (sigmas(i)^2)*Dxy;
    Dyy = (sigmas(i)^2)*Dyy;
% Calculate (abs sorted)\
eigenvalues and vectors
tmp = sqrt((Dxx - Dyy).^2 + 4*Dxy.^2);
v2x = 2*Dxy; v2y = Dyy - Dxx + tmp;
% Normalize
mag = sqrt(v2x.^2 + v2y.^2);
i2 = (mag ~= 0);
v2x(i2) = v2x(i2)./mag(i2);
v2y(i2) = v2y(i2)./mag(i2);
% The eigenvectors are orthogonal
v1x = -v2y;
v1y = v2x;
% Compute the eigenvalues
mu1 = 0.5*(Dxx + Dyy + tmp);
mu2 = 0.5*(Dxx + Dyy - tmp);
% Sort eigen values by absolute value\
abs(Lambda1)<abs(Lambda2)
check=abs(mu1)>abs(mu2);
Lambda1=mu1; Lambda1(check)=mu1(check);
Lambda2=mu2; Lambda2(check)=mu2(check);
Ix=v1x;
Ix(check)=v2x(check);
Iy=v1y;
Iy(check)=v2y(check);
 %[Lambda2,Lambda1,Ix,Iy]=\
 eig2image(Dxx,Dxy,Dyy);
    % Compute the direction of \
    the minor eigenvector
    angles = atan2(Ix,Iy);
    % Compute some similarity measures
    Lambda1(Lambda1==0) = eps;
    Rb = (Lambda2./Lambda1).^2;
    S2 = Lambda1.^2 + Lambda2.^2;
    % Compute the output image
    Ifiltered = exp(-Rb/beta)\
    .*(ones(size(I))-exp(-S2/c));
    % see pp. 45
    if(options.BlackWhite)
        Ifiltered(Lambda1<0)=0;
    else
        Ifiltered(Lambda1>0)=0;
    end
    % store the results in 3D matrices
    ALLfiltered(:,:,i) = Ifiltered;
    ALLangles(:,:,i) = angles;
end
% Return for every pixel the value
%of the scale(sigma) with the maximum
% output pixel value
if length(sigmas) > 1,
    [outIm,whatScale] =\
    max(ALLfiltered,[],3);
    outIm = reshape(outIm,size(I));

        Direction = reshape(\
        ALLangles((1:numel(I))'+\
        (whatScale(:)-1)*numel(I)),\
        size(I));
end
subplot(1,2,1), imshow(I,[]);
outIm = imbinarize(outIm,0.02);
subplot(1,2,2), imshow(outIm);
%subplot(1,2,2), imshow(outIm,[0 0.25]);
```

B. Code for method 2

```
%METHOD 2
clear all;
clc;
%STEP ONE LOAD THE IMAGES
currentPath =\
convertCharsToStrings(pwd);
imgLabels =\
imageDatastore(currentPath+\
'\ImagesLabels\*.*');
imgLabels =\
readall(imgLabels);
imgRaw = imageDatastore(\
currentPath+'\Images\*.*');
imgRaw = readall(imgRaw);
%preview any image 5 & 6
imageIndex=6;
I=imgRaw{imageIndex};
Label=imgLabels{imageIndex};
subplot(341), imshow(I)
title('original')
%I = imresize(I,1);
I = rgb2gray(I);
[r,c, ch] = size(I);
```

```
B = imresize(I, [r c]);
gray = im2double(B);
% Contrast Enhancment of gray\
image using CLAHE
J = adapthisteq(gray,'numTiles',\
[8 8],'nBins',128);
% Background Exclusion
% Apply Average Filter
h = fspecial('average', [9 9]);
JF = imfilter(J, h);
% Take the difference between\
the gray image and Average Filter
Z = imsubtract(JF, J);
% Threshold using the IsoData Method
%level=isodata(Z); %
%this is our threshold level
% Convert all N-D arrays into a single\
column.  Convert to uint8 for
% fastest histogram computation.
TT = im2uint8(Z(:));
% STEP 1: Compute mean intensity\
%of image from histogram, set T=mean(I)
[counts,N]=imhist(TT);
i=1;
mu=cumsum(counts);
T(i)=(sum(N.*counts))/mu(end);
T(i)=round(T(i));
% STEP 2: compute Mean above\
%T (MAT) and Mean below T (MBT) using T from step 1
% step 1
mu2=cumsum(counts(1:T(i)));
MBT=sum(N(1:T(i))\
.*counts(1:T(i)))/mu2(end);
mu3=cumsum(counts(T(i):end));
MAT=sum(N(T(i):end)\
.*counts(T(i):end))/mu3(end);
i=i+1;
% new T = (MAT+MBT)/2
T(i)=round((MAT+MBT)/2);
Threshold=T(i);
% STEP 3 to n: repeat step 2 if T(i)~=\
T(i-1)
while abs(T(i)-T(i-1))>=1
    mu2=cumsum(counts(1:T(i)));
    MBT=sum(N(1:T(i)).\
    *counts(1:T(i)))/mu2(end);
    mu3=cumsum(counts(T(i):end));
    MAT=sum(N(T(i):end).\
    *counts(T(i):end))/mu3(end);
    i=i+1;
    T(i)=round((MAT+MBT)/2);
    Threshold=T(i);
end
% Normalize the threshold to
%the range [i, 1].
level = (Threshold - 1) / (N(end) - 1);
%END
%level = graythresh(Z)
% Convert to Binary
BW = im2bw(Z, level);
% Remove small pixels
 BW2 = bwareaopen(BW, 100);
% Overlay
BW3 = imcomplement(BW2);
out = imoverlay(B, BW3, [0 0 0]);
vv = rgb2gray(out);
binaryImage = vv > 60;
BW2 = bwmorph(binaryImage,'clean');
segimg = BW2;
%image 1,5 good
subplot(342), imshow(I)
title('grayscale')
subplot(343), imshow(J)
title('equalized image')
subplot(344), imshow(JF)
title('average filtered image')
subplot(345), imshow(Z*4)
title('substracted JF-J')
subplot(346), imshow(BW)
title('threshold binary')
subplot(347), imshow(BW2)
title('opened')
subplot(348), imshow(BW3)
title('overlay')
subplot(349), imshow(segimg)
title('segmented')
subplot(3,4,10), imshow(Label)
title('labels')
```

C. Code for method 3

```
clc;
clear;
%Opening an image
test_image = \
    imread("Images/im0001.ppm");
I = im2gray(test_image);
%Displaying gray scale image
figure(1);
  imshow(I)
   title("Original Image");
[x,y] = ginput(1);
%Calculate and display image gradient
```

```
using imgradient
[grad_val,grad_dir] = imgradient(I);
figure(2);
     imshow(grad_dir),
     title("Gradient Direction");
%Display edge_map using edge function
em_I1 = edge(I,"log");
em_I2 = edge(I,"sobel");
em_I3 = edge(I,"prewitt");
em_I4 = edge(I,"canny");
figure(3)
    imshow(em_I1),\
    title("Segmented Image");
% m1 = zeros(size(I));
% for i = 100:380
%     for j = 1:200
%        m1(i,j) = 1;
%     end
% end
out2 = \
imbinarize(
    I,
    'adaptive',
    'Sensitivity',0.2);
out1 = \
activecontour(
    I,
    m1,
    "edge",
    "SmoothFactor",
    0.2
);
nhood = ones(10);
out4 = imclose(em_I1,nhood);
figure(5)
    imshow(out4);
out5 = bwareaopen(out4,2000);
figure(6)
    imshow(out5);
    title("Binarized Image");

%fast fourier transform of the
%selected medical image
ft_image = fft2(I);
%shifting the image by applying
%the fourier shift to get
%frequency response of
%interested area
ft_shift = fftshift(ft_image);
%narrowing down to particular
%area using logarithmic scale
fl_fig = log(1+abs(ft_shift));
disp(size(I))
%defining a lowpass filter mask randomly
fl_mask = ones(size(I));
for i=300:320
  for j=340:420
    fl_mask(i,j)=0; %given filter value
  end
end
%applying frequency shift to the mask
ff_mask = fftshift(fl_mask);
filt_fq = ft_image.*ff_mask;
filt_img = abs(ifft2(filt_fq));
figure(10)
  imshow(uint8(8*filt_img));
  title("Vessel Detection");
```

REFERENCES

[1] The Cost of Vision Loss Report 2019 - Canadian council for the blind.

[2] https://calgaryfamilyeyedoctors.com/what-does-retinal-imaging-show/

[3] STructured Analysis of the Retina project - http://cecas.clemson.edu/~ahoover/stare/

[4] A. Hoover, V. Kouznetsova and M. Goldbaum, "Locating Blood Vessels in Retinal Images by Piece-wise Threhsold Probing of a Matched Filter Response", IEEE Transactions on Medical Imaging , vol. 19 no. 3, pp. 203-210, March 2000.

[5] A. Hoover and M. Goldbaum, "Locating the optic nerve in a retinal image using the fuzzy convergence of the blood vessels", IEEE Transactions on Medical Imaging , vol. 22 no. 8, pp. 951-958, August 2003.

[6] Topcon's Premier Mydriatic Reinal Camera https://topconhealthcare.ca/en/products/trc-50dx/

[7] Wikicommons schematics of the human eye https://commons.wikimedia.org/wiki/File:Schematic_diagram_of_the_human_eye_en.svg

[8] Sato Y., *et. al.* - "Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images" - Medical Image Analysis - Volume 2, Issue 2, June 1998, Pages 143-168 - https://doi.org/10.1016/S1361-8415(98)80009-1

[9] Alejandro F. Frangi, *et. al.* - "Multiscale vessel enhancement filtering" - Part of the Lecture Notes in Computer Science book series (LNCS,volume 1496)

[10] Snedecor, G. W. (1956). Statistical Methods: Applied to Experiments in Agriculture and Biology. United States: Iowa State University Press.

[11] Snedecor, G. W., Cochran, W. G. (1967). Statistical Methods. Iowa State University Press.

[12] https://www.mathworks.com/help/images/ref/imbinarize.html

[13] https://www.mathworks.com/help/simulink/ug/block-masks.html

[14] https://www.mathworks.com/help/matlab/ref/fft.html

[15] https://www.mathworks.com/help/images/ref/activecontour.html

[16] https://www.mathworks.com/help/images/ref/bwareaopen.html

[17] https://www.mathworks.com/matlabcentral/fileexchange/24409-hessian-based-frangi-vesselness-filter

[18] https://www.mathworks.com/matlabcentral/fileexchange/72118-reitna-blood-vessel-segmentation