Contour Based Blood Vessel Segmentation in Digital Fundus Images

SACHIN M B

1. INTRODUCTION

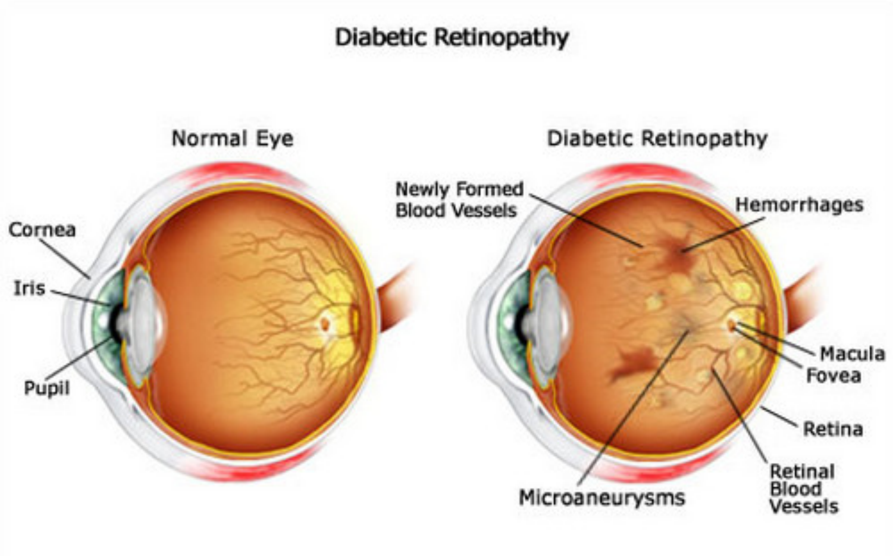
1.1 IMAGE SEGMENTATION

Image segmentation is the process of partitioning in digital images into multiple segments (sets of pixels). The primary objective image segmentation is to change the representation of an image into something that is much easier to analyse. Image segmentation is typically used to locate objects and boundaries such as lines, curves etc. in images. The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image. Each of the pixels in a region are similar with respect to some characteristic or computed property, such as colour, intensity or texture. Several general-purpose algorithms and techniques have been developed for image segmentation. To be useful, these techniques must typically be combined with a domain's specific knowledge in order to effectively solve the domain's segmentation problems.

1.2 DIABETIC RETINOPATHY

Diabetics will lead people to have an eye disease called diabetic retinopathy [11-12]. It results when sugar level in the blood increases and cause damage to blood vessels in the retina. It can be identified by the swelling and leakage of blood vessels. It also sometimes results in abnormal growth of new blood vessels on the retina. People having diabetics for 20 years or more are likely to have Diabetic Retinopathy. When proper treatment is undergone at the early stage, 90% of the time it can cured by continuous monitoring of the retina. Diabetic Retinopathy severity is based on the number of years a person has diabetics.

Diabetic retinopathy affects retinal blood vessels which is linked to the back of the eye. When Diabetic Retinopathy prevails for a longer, it leads to either blindness or vision impairment among adults above 40 ages. The first stage is called non-proliferative diabetic retinopathy (NPDR) and the signs are not visible to the eye. The only available and popular way to identify NPDR is by taking fundus photographs of retina and detecting microaneurysms from them. Diabetic Retinopathy is shown in the Fig. 1.a.

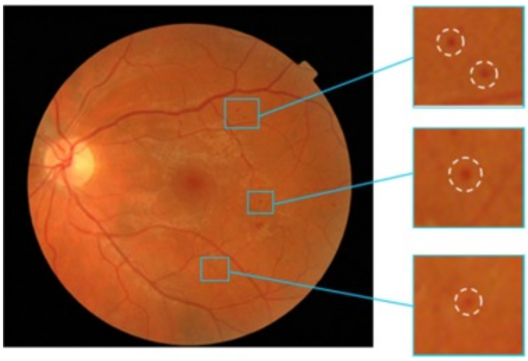


1.a. Diabetic Retinopathy

1.3 MICROANEURYSM

Microaneurysms are small saccular outpouchings that involve capillaries of many vascular districts such as the heart, kidney and eye. Example of Microaneurysm is shown in the Fig. 1.b. Ophthalmologists know that although they occur in several pathologic conditions such as hypertension, venous occlusion and hemorheological diseases, including methemoglobinemia and sickle-cell disease, they are the hallmark of Diabetic Retinopathy. Their importance is underscored by the fact that they are the first clinically evident sign of NPDR.

So, the recognition of microaneurysms can be the first step in secondary prevention of Diabetic Retinopathy progression to the proliferative stage and consequent severe visual loss.



1.b. Microaneurysm

2. LITERATURE SURVEY

2.1 DETECTION OF RETINAL BLOOD VESSELS AND REDUCTION OF FALSE MICROANEURYSMS

Rahul Chauhan et al [1] has proposed a method to detect all the blood vessels in the retina and find all the correct microaneurysms that lead to Diabetic Retinopathy without any false microaneurysms. The diameter of the blood vessel has to measured accurately to properly diagnose Diabetic Retinopathy. Detecting the total number of microaneurysms is useful to identify the severity stage of the Diabetic Retinopathy.

In this method, fundus photography technique is used to take fundus images of the retina. Then the images are pre-processed using spatial low pass filter. The low pass filter removes high spatial frequency noise from the digital image. Then the image is passed through s Gabor filter to extract various feature values. Gabor filter helps in filter out noise while enhance the contrast of image. Gabor filters are act like band pass filter and its use in various application of image processing like feature extraction and texture analysis. After pre-processing, the method of segmentation is carried out by thresholding techniques. Global and local thresholding are used for segmentation.

Morphological operations are used to enhance the shape of the image. Operations mainly erosion and dilation are used to extract the vessel structure from the fundus image. This helps to extract the different part of same vessel in order to measure the width of vessel. More morphological operations such as extended minima transform and Skeletonization are used for further extraction. After morphological operations, the width of the blood vessels is measured using Euclidian distance mathematical technique. Finally, the microaneurysm detection is done with Gabor filter with thresholding. The selection of sigma involves a trade-off larger values more robust to noise but more likely to create spurious rings and smaller values less likely produce spurious rings but less effective in removing noise.

In this proposed work an efficient algorithm of detection of microaneurysm in retinal fundus image proposed and shows the efficient results. Gabor filter for feature extraction provide the better segmentation. Further global thresholding for segmentation of vessel segmentation followed by the process of erosion in order to remove the noise and information extraction of vessel in retinal fundus image. Vessels width are measured accurately and false microaneurysms are reduced.

2.2 A MORPHOLOGICAL HESSIAN BASED APPROACH FOR RETINAL BLOOD VESSELS SEGMENTATION AND DENOISING USING REGION BASED OTSU THRESHOLDING

Khan Bahadar Khan et al [2] has proposed a method to extract the blood vessels using Hessian based approach and Otsu thresholding is used to denoise the images after segmentation.

In the proposed work, CLAHE (Contrast Limited Adaptive Histogram Equalization) a pre-processing technique and morphological filters are used for vessel contrast enhancement and low frequency noise/geometrical objects removal respectively. The main advantage of using CLAHE is it does not discard the part of the histogram that exceeds the clip limit but to redistribute it equally among all the histogram bins. The modified top-hat transform is used as a morphological filter.

After pre-processing, Hessian matrix and eigenvalues transformation is applied in a modified form at two different scales to extract wide and thin vessels enhanced images, separately. The second derivative of the image is computed at two different scales and the wide and thin vessels are isolated. The eigenvalues of hessian matrix and the difference between them have been used for further contrast enhancement and suppression of non-vasculature structure.

Global and local Otsu thresholding is utilized in a modified way to classify vessel and non-vessel pixels from wide and thin vessel enhanced images, respectively. This approach is applied in a modified way to suppress the unwanted noise and geometrical objects based on vessel structure. Global thresholding is applied on wide vessels enhanced images and fused its resultant image into thin vessel enhanced image. After obtaining a single enhanced image, local thresholding (vessel-based thresholding) is applied.

Post processing steps are used for eliminating background noise, undesired segments and erroneously detected vessel pixels. Less than or equal to 30 pixels are removed based on the connectivity of the retinal vessels as a non-vessel or a part of the background noise. The vessel pixels are labelled as ‘1’ or ‘0’ to obtain the final binary image.

2.3 AUTOMATED MICROANEURYSM DETECTION METHOD BASED ON EIGENVALUE ANALYSIS USING HESSIAN MATRIX IN RETINAL FUNDUS IMAGES

Tsuyoshi Inoue et al [3] has proposed an automated microaneurysm method based on the eigenvalues approach on hessian matrices. In the proposed method, the images are first pre-processed and then hessian matrix is formed from the green channelled retinal fundus images. After that, approximately 126 features are calculated for each identified candidate and after some filtering processes 25 features were finalised. Artificial Neural Networks (ANN) has been used to classify the microaneurysms using feature analysis to remove the false positive microaneurysms from the real ones.

Microaneurysm appears as a dark spot in the retinal fundus images. TO reduce the adverse effect on the image, pre-processing methods such as brightness correction, gamma correction and contrast enhancement has been applied.

Low-pass filter is used to reduce the noise on the green-channel components. The blood vessels are excluded from the candidate regions by combining double-ring filter and black-top-hat morphological transform.

The microaneurysm candidate regions were detected using eigenvalue analysis based on hessian matrix and are modified by re-thresholding the eigenvalues. The candidates included many false positives and therefore it was classified using feature analysis and Artificial Neural Network (ANN). Here, 126 features were extracted based on pixel value, shape and texture analysis and finalised 25 features were fed into three-layered ANN.

The main advantage of this proposed work is that it has a better true positive rate candidates being microaneurysms, the microaneurysm image is emphasized in this method, usage of double-ring filter emphasized the irregularly shaped spots with high contrast, shape-index of the proposed method is able to classify the shape of intensity curve surface and it is very effective in detecting dark lesion in the retinal fundus images.

DISADVANTAGES OF EXISTING SYSTEM

- The existing system pre-processing methods produces images with noise and not an improved image for the blood vessel extraction.
- The existing systems also considers the thin blood vessels for microaneurysm detection but they are present only in the thick blood vessels.
- Identifies many false positive microaneurysms.
- If central pixel of microaneurysm is not dark, it goes undetected.
- The microaneurysms with faint colour are not detected in the existing systems.
- Microaneurysms with abnormal shape and size are not detected in the existing systems.

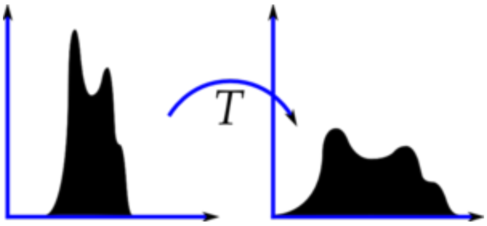
In the proposed system, the image is pre-processed by CLAHE to get a better image for segmenting the blood vessels. Alternate sequential filtering and Top hat transform is applied to remove the noises from the pre-processed image. Then, Otsu thresholding is applied to group the relative pixels for easier segmentation. Finally, contour detection method is deployed to segment the blood vessels depending upon the size and shape of the blood vessels. Only the thick blood vessels are extracted and the thin blood vessels are removed using post processing methods such as erosion and dilation [4].

3. SYSTEM ARCHITECTURE

Blood vessel segmentation is important for retinal image analysis. It helps to give useful information for the diagnosis and monitoring of eye diseases such as Diabetic Retinopathy (DR), haemorrhage, glaucoma etc. The propose work is first pre-processed using CLAHE then minor noisy structures are removed using morphological filters [5] and Otsu thresholding and blood vessels are segmented using contour detection.

3.1. Contrast Limited Adaptive Histogram Equalization (CLAHE)

We used green channel of the retinal image for analysis and segmentation of blood-vessel structure. In green channel, blood vessels are more differentiated than the background as compared to red channel and blue channel. The datasets can be taken from DRIVE and STARE datasets which images are used for analysis. Generally, histogram equalization techniques can be used for contrast improvement. We used the CLAHE method to get a local contrast enhanced retinal image. The concept of CLAHE is shown in the Fig. 3.a. CLAHE uses a user-defined value (clip-limit) to constrain enhancement by clipping the histogram. The clipping level specifies noise level to be smoothen and contrast level to be enhanced in histogram. For our image, clip limit is used from 0 to 0.05.



3.a. CLAHE

3.2. MORPHOLOGICAL FILTERS

Vessel structure appears in more prominent contrast than background intensity variations. However, a more local investigation of vessel intensities can indicate changes that influence the whole vessel extraction process. To overcome this, we used a morphological filter known as modified top-hat transform which has been applied on the green channel retinal image.

We used morphological top-hat transformation to find out difference between the input and the opened image. Closed image followed by the opened image to obtain inverse image. Top-hat transformation has two types black top-hat and white top-hat.

Opening of an image **I** with structuring element $\boldsymbol{S_o}$ is given in equation (1)**,**

$$T_{open} = I \circ S_o \tag{1}$$

Closing of an image **I** with structuring element $\boldsymbol{S_C}$ is given in equation (2)**,**

$$T_{close} = I \cdot S_C \tag{2}$$

Modified Top-hat transform is given in equation (3),

$$TopHat = I - (I \cdot S_C) \circ S_o \tag{3}$$

3.3. OTSU THRESHOLDING

Otsu's approach to suppress the unwanted noise and geometrical objects based on vessel structure on the image. Usually Otsu's approach is used locally or globally on the entire image to find a threshold for classification of vessel and non-vessel pixels. Applying Otsu threshold on the whole image at once does not give fruitful results that's why we have applied it separately on wide and thin vessel images. We have used global threshold on wide vessels enhanced image and fused its resultant image into thin vessel enhanced image. In this way both thin and thick vessels become more prominent.

We obtained a single enhanced image on which further local thresholding has been applied. In local thresholding, we used vessel-based thresholding which depends upon vessel locality to define a new threshold. We added some offset in the global threshold to suppress the noise more effectively for vessels in the neighbourhood of wide vessels. For other regions away from wide vessels, we set lower threshold than the global by subtracting some offset from it to extract the small or thin vessels from the background having low intensity [6].

$$\sigma^2 = \sigma_w^2(t) + \sigma_b^2 \tag{4}$$

In equation (4), $\boldsymbol{\sigma^2}$ is the total variance of the image while $\boldsymbol{\sigma_w^2}$ is the within-class variance and $\boldsymbol{\sigma_b^2}$ is the between-class variance respectively.

$$\sigma_b^2 = q(t)[1 - q(t)][\mu_1(t) - \mu_2(t)]^2 \tag{5}$$

In equation (5), $\boldsymbol{q}$ is the class probabilities while $\boldsymbol{\mu_1}$ and $\boldsymbol{\mu_2}$ are the class means of the respective classes.

3.4. CONTOUR DETECTION

Contour detection includes a variety of mathematical methods that aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities. The points at which image brightness changes sharply are typically organized into a set of curved line segments termed contours.

Contour detection is a fundamental tool in image processing, machine vision and computer vision, particularly in the areas of feature detection and feature extraction.

The flow of the project is shown below in the Fig. 3.b.

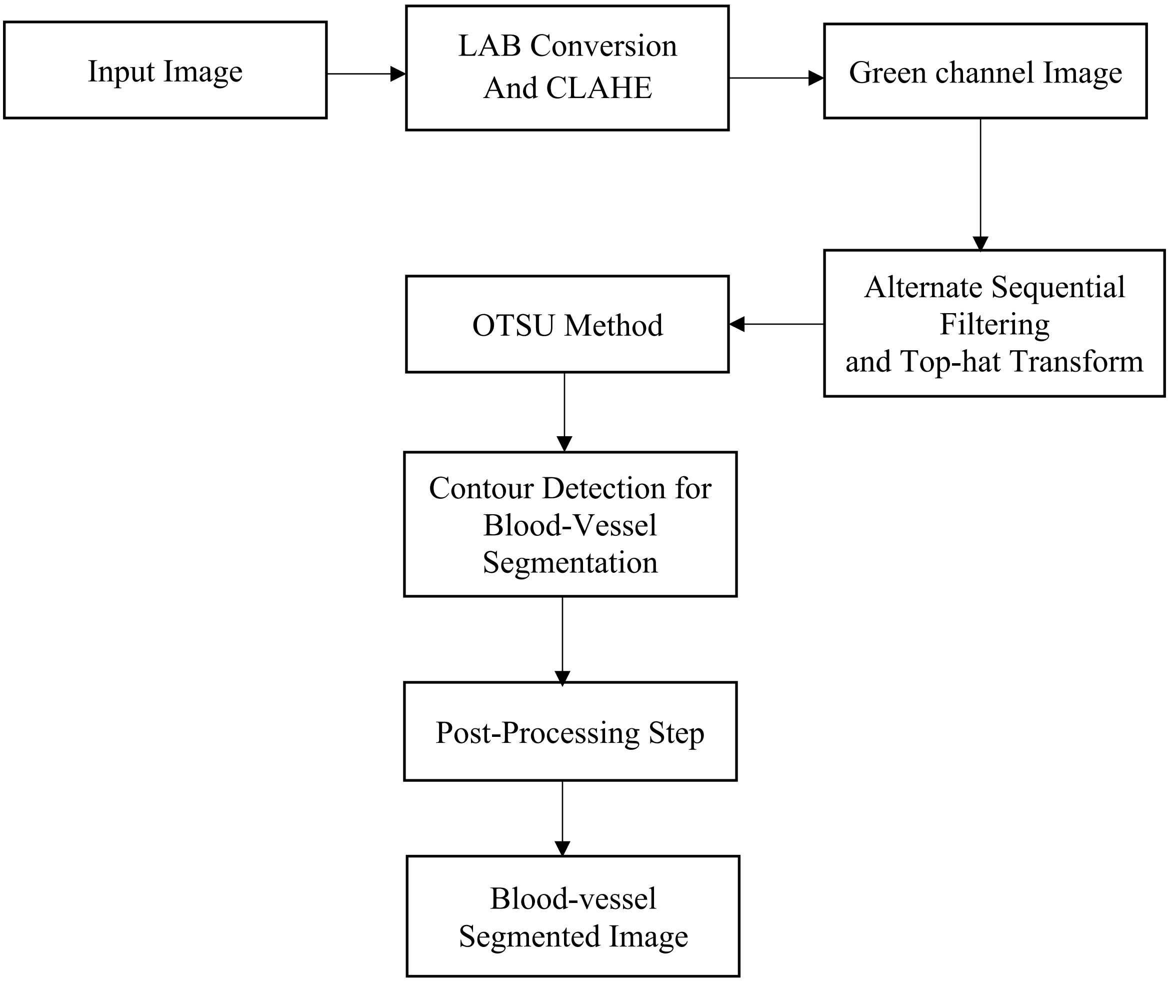

Input Image
LAB Conversion And CLAHE
Green channel Image
Alternate Sequential Filtering and Top-hat Transform
OTSU Method
Contour Detection for Blood-Vessel Segmentation
Post-Processing Step
Blood-vessel Segmented Image

Fig: 3.b Flow-chart for blood-vessel segmentation

4. REQUIREMENTS

HARDWARE REQUIREMENTS

➢	System	:	Intel CORE i3
➢	Hard Disk	:	50 GB.
➢	RAM	:	512 Mb.
➢	Processor Speed	:	800Mhz

SOFTWARE REQUIREMENTS

➢	Operating system	:	Windows 7/8/8.1/10/Linux
➢	Coding Language	:	Python
➢	IDE Tool	:	Spyder
➢	Packages	:	OpenCV, NumPy, OS, OpenPyXL

5. SYSTEM DESIGN AND IMPLEMENTATION

The proposed work is implemented in Python 3. The proposed work starts by reading the image and converting it to LAB modal. Then, the image is pre-processed using CLAHE then minor noisy structures are removed using morphological filters and Otsu thresholding and blood vessels are segmented using contour detection.

5.1. READ THE IMAGE

The retinal image will be imported from the disk to process the image to extract the blood vessels and to find the micro-aneurysm in it. The image will be imported from disk using OpenCV packages. The read image is shown in the figure 5.a.

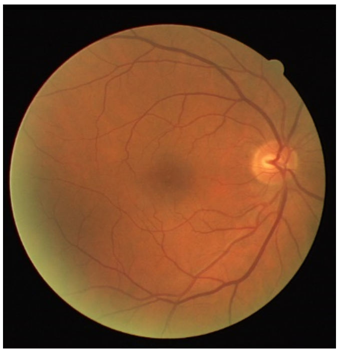


Fig: 5.a Input image

5.2. CONVERT RGB INTO LAB

The imported retinal image is in RGB colour space. Now convert the RGB colour space into LAB colour space and split it. In LAB colour space, ‘l’ component has brightness, ‘a’ component has colour ranging from green to red and ‘b’ component has colour ranging from blue to yellow. The LAB image converted from RGB is shown in figure 5.b.

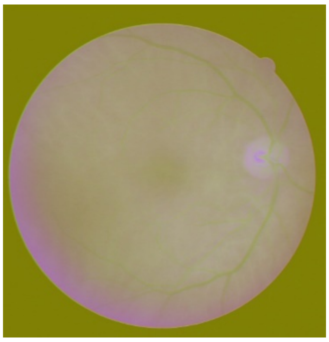


Fig: 5.b LAB image

After splitting, apply CLAHE to the ‘l’ dimension to enhance the blood vessels, to get better quality and to clip the histogram in the image. Then merge it with a & b. The CLAHE is applied on ‘l’ dimension is shown in the figure 5.c.

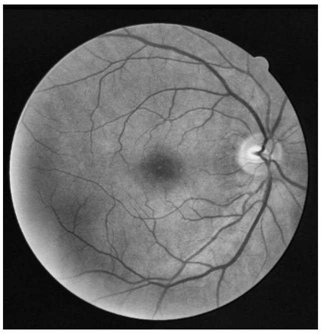


Fig: 5.c CLAHE applied on ‘l’ dimension

5.3. CONVERT LAB INTO RGB

Convert the LAB colour space into RBG colour space for the further processing to extract blood vessels. The conversion of LAB into RGB is shown in the figure 5.d.

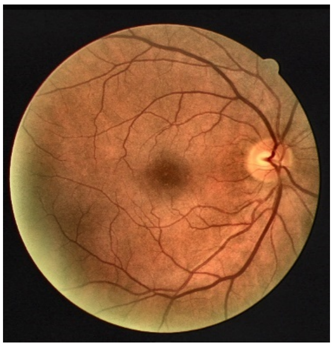


Fig: 5.d LAB into RGB

RGB colour space is split into blue, green and red channels and green-band image is taken to extract blood vessels as the blood vessels are clearly visible in the green band image. The green channel image is shown in the figure 5.e.

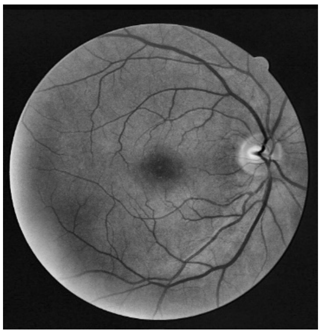


Fig:5.e Green channel Image

5.4. APPLYING MORPHOLOGICAL FILTER

5.4.1. ALTERNATE SEQUENTIAL FILTERING

Alternatively opening and closing the image sequentially, starting with a small structuring element and then proceeding with ever increasing structuring elements is the principle behind Alternate Sequential Filtering (ASF). This is done is remove the dark and bright noisy structures from the image. In this work, ASF is applied three times. In the 1st iteration, the size of the structuring element is set as (5,5). The image after 1st iteration of ASF is shown in the Fig:5.f.

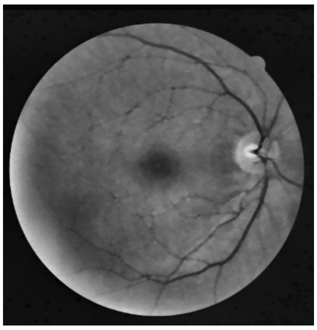


Fig:5.f Image after 1st iteration of ASF

In the 2nd iteration, the size of the structuring element is set as (11,11). The image after 2nd iteration of ASF is shown in the Fig:5.g.

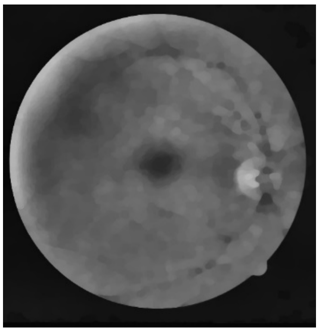


Fig:5.g Image after 2nd iteration of ASF

In the 3rd iteration, the size of the structuring element is set as (23,23). The image after 3rd iteration of ASF is shown in the Fig:5.h.

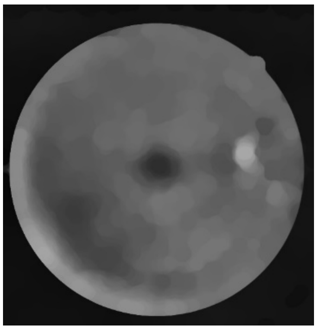


Fig:5.h Image after 3rd iteration of ASF

After applying ASF for 3 times, the resultant image is subtracted from the green channel image and CLAHE is applied to get noise-free image [7]. The ASF image after applying CLAHE is shown in the Fig:5.i.

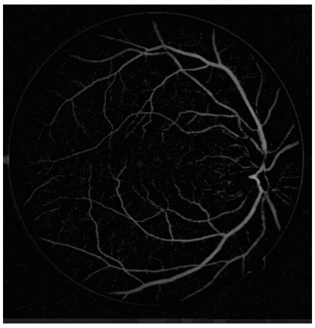


Fig:5.i CLAHE Image after ASF

5.4.2. TOPHAT MORPHOLOGICAL FILTERING

Top-hat morphological filter is applied to reduce the noise in the images. Top-hat transform extracts small elements from the given input images. This top-hat transform is used for feature extraction, image enhancement and so on. In top-hat transform, it is used to find the difference between the opening and closing and the input image. The following figure 5.j shows the resultant image of top-hat transformation.

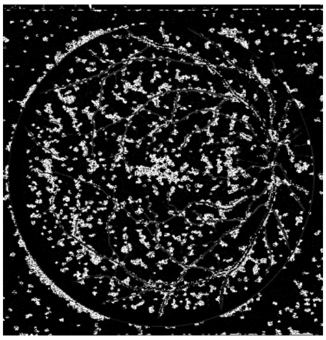


Fig:5.j Top-Hat Transform Image

5.5. APPLYING OTSU METHOD

Otsu method used to remove the unwanted noise and geometrical structures in the extracted image. It is used for the conversion of a grey-level image to a binary image. Global and local Otsu is applied on thick and thin blood vessels [9-10]. The image obtained after applying local and global Otsu thresholding is shown in the Fig:5.k.

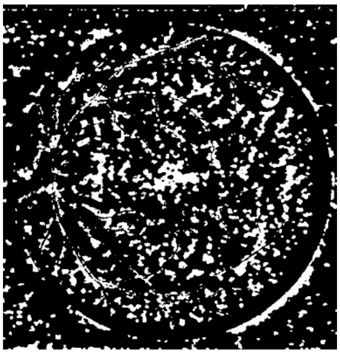


Fig:5.k Otsu image

5.6. CONTOUR DETECTION

Blood vessel can be extracted by finding the contours present in the pre-processed image. The contours are an outline representing or bounding the shape or form of the blood vessels. Initially, by finding all the contours each possible blood vessel present in the image is marked [8]. The resultant image after finding all contours is shown in the Fig:5.l.

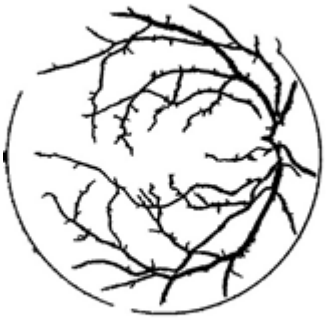


Fig: 5.l Image before removing unwanted contours

Then, the unwanted contours are removed using area parameter noise removal. The blobs with unwanted size and shape are neglected and the selected blood vessel are drawn over the mask image to get the required output. Then, the image is inverted. The image after removing the unwanted blobs is shown in the Fig:5.m.

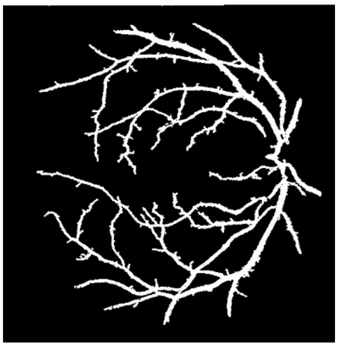


Fig: 5.m Image after removing unwanted contours

The detected blood vessel image is again eroded (morphology filter operation) to remove the minor noises present in the image. The image after post processing is shown in the Fig:5.n.

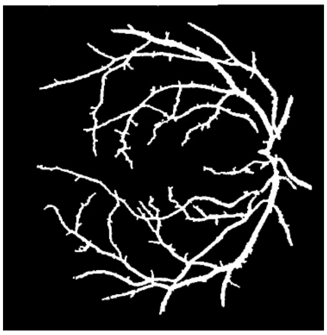


Fig: 5.n Image after post processing

6. EXPERIMENTAL RESULTS

This proposed method is tested for 40 images from the DRIVE dataset. The experimental results are calculated by comparing the output images with the ground truth images of the first observer from the DRIVE dataset. The calculated experimental results are shown below.

6.1 ACCURACY:

Accuracy means the degree to which the result of the blood vessel segmented images conforms to the correct standard by comparing with the truth images. The accuracy of the proposed work is shown in the Fig:6.a. Accuracy is calculated using the formula,

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{6}$$

The overall accuracy of this proposed work is **86.6542**

In the below graph, x-axis denotes the no. of images and y-axis denotes the accuracy for each image.

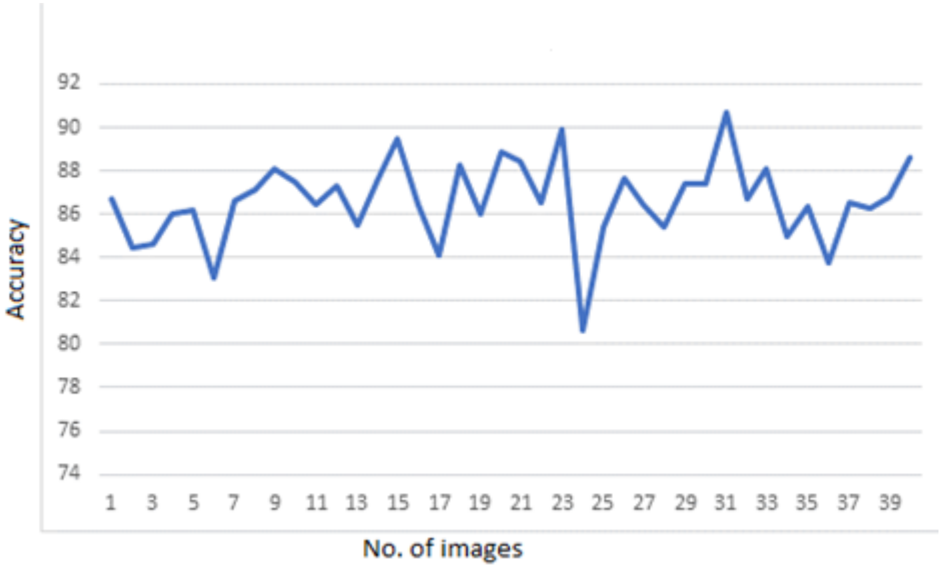


Fig:6.a No. of Images vs Accuracy Graph

6.2 SENSITIVITY:

Sensitivity is the measure that correctly identifies the proportion of positives present in the work (true positive rate). The sensitivity of the proposed work is shown in the Fig:6.b. Sensitivity is calculated by the formula,

$$Sensitivity = \frac{TP}{TP+FN} \qquad (7)$$

The overall sensitivity of this proposed work is **71.1033224**

In the below graph, x-axis denotes the no. of images and y-axis denotes the sensitivity of each image.

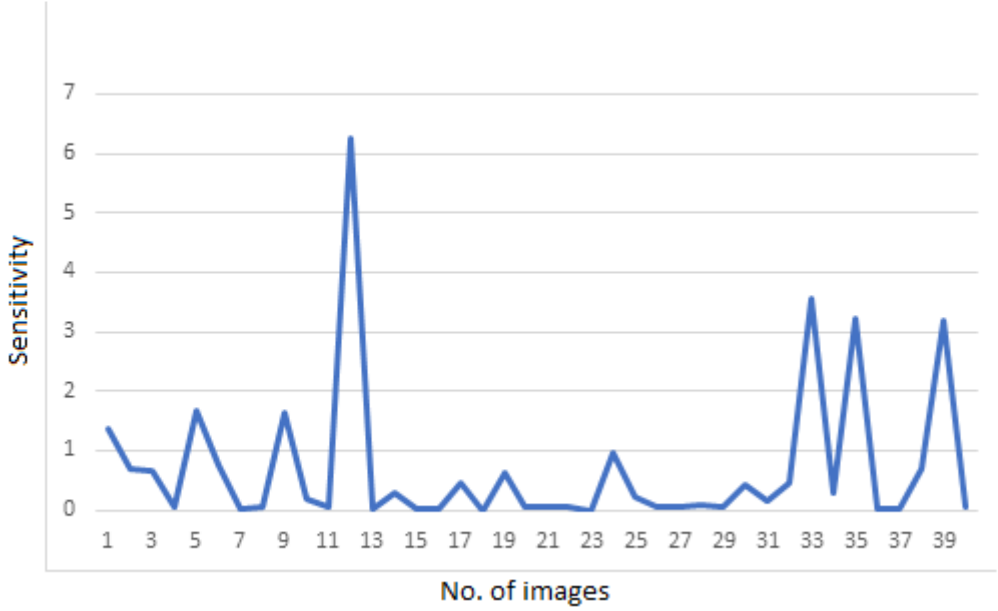


Fig:6.b No. of images vs Sensitivity Graph

6.3 SPECIFICITY:

Specificity is the measure that correctly identifies the proportion of negatives present in the work (true negative rate). The specificity of the proposed work is shown in the Fig:6.c. Specificity is calculated by the formula,

$$Specificity = \frac{TN}{TN+FP} \qquad (8)$$

The overall specificity of this proposed work is **99.13912014**

In the below graph, x-axis denotes the no. of images and y-axis denotes the specificity of each image.

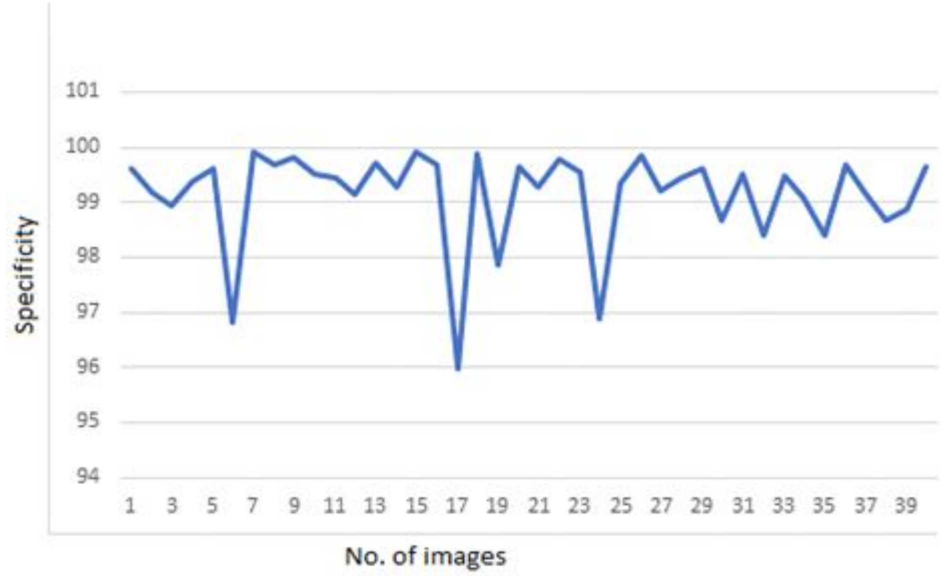


Fig:6.c No. of images vs Specificity Graph

TP – True Positive; The pixels that are correctly identified are True Positives

FP – False Positive; The pixels that are incorrectly identified are False Positives

TN – True Negative; The pixels that are correctly rejected are True Negatives

FN – False Negative; The pixels that are incorrectly rejected are False Negatives

The accuracy, sensitivity and specificity for all the 40 images in the DRIVE dataset is shown in the Table:6.1.

IMAGE	ACCURACY	SENSITIVITY	SPECIFICITY
1	86.73125968	37.0192345	99.6086472
2	84.41328595	68.0821104	99.17899089
3	84.62721162	66.7241022	98.93638613
4	85.98831367	59.4936672	99.38182517
5	86.18396652	67.1058768	99.6100564
6	83.10209989	75.2892799	96.80506483
7	86.58151171	13.3584835	99.91719163
8	87.10874934	52.9939555	99.67715561
9	88.13158143	64.7573456	99.81585513
10	87.50985997	18.8112261	99.51131676
11	86.45716742	37.5332683	99.44224241
12	87.35164411	62.4143699	99.13179685
13	85.4534995	34.2592886	99.7196587
14	87.6395524	29.1571033	99.29180424
15	89.50694519	34.1996162	99.92188305
16	86.45226496	13.4388041	99.6762668
17	84.14365048	46.3932211	95.9805867
18	88.27063312	38.3746350	99.87091193

19	86.01594573	63.6433475	97.86283746
20	88.89547433	37.1542954	99.65076318
21	88.40210827	48.7330342	99.29462131
22	86.55878211	60.5056525	99.78111815
23	89.93479653	46.1254590	99.55401363
24	80.60630046	94.8545972	96.87687799
25	85.4142798	21.2499194	99.34455761
26	87.68367457	36.5768086	99.84672645
27	86.45181928	65.7909654	99.21379415
28	85.36926627	75.5724274	99.43255511
29	87.36100336	68.7542518	99.62285391
30	87.39442924	42.2812225	98.68426966
31	90.71607089	15.6136977	99.50763195
32	86.70362761	45.4769415	98.4160967
33	88.10974319	35.4271946	99.49231085
34	84.9650359	28.7081456	99.07450685
35	86.403686	32.3676425	98.39221196
36	83.79735831	83.9411405	99.67767022
37	86.51510562	10.4832476	99.16526558
38	86.25616643	68.2621345	98.66705401
39	86.8074707	31.8877256	98.86990618
40	88.58171669	40.0842559	99.65952233

Table:6.1 Experimental Results

7. CONCLUSION

In this proposed work, a region-based approach of vessel segmentation is presented. CLAHE and alternate sequential filtering have been applied as a pre-processing step to remove the undesired noise from the input image. After pre-processing, top-hat morphological transform has been deployed in a modified way which further removes the unwanted minor structural noises. After that, Otsu thresholding has been applied to distinguish the desired blood vessels from the unwanted ones in order to easily segment the blood vessels. Then, the blood vessels have been segmented using contour detection which initially involves finding all the possible contours in the image and then removing the unwanted large contours. These series of methods have been followed correctly and the blood vessels are segmented. This method has been tested on the DRIVE dataset which consists of 40 images. The performance of the proposed method has been evaluated and achieved a sensitivity of 71.10%, specificity of 99.13% and accuracy of 86.7%. The results indicate that the proposed can accurately segment the blood vessels in the retinal images which is very much needed for DR diagnosis.

8. SCOPE FOR FUTURE WORK

- Further, the blood vessel extracted images can be used to identify Microaneurysms and detect Diabetic Retinopathy using various algorithms.
- The method can also be tested in STARE, CHASE dataset and other retinal image repositories.
- The method can be extended in the future to identify Diabetic Retinopathy from the images captured directly using mobile phones.
- The work can also be implemented using Convolutional Neural Networks (CNN) to increase the efficiency and speed of the processing.

9. APPENDIX

9.1 SOURCE CODE

```
import cv2
import numpy as np
import os
from openpyxl import Workbook

loc = 'D:/Project/test/input/'
files = os.listdir(loc)

location = 'D:/Project/test/truthh/'
files1 = os.listdir(location)
count1 = 0

#creating a new excel sheet to store the results
wb = Workbook()
ws = wb.active
ws['A1'] = "Image"
ws['B1'] = "True positive"
ws['C1'] = "True negative"
ws['D1'] = "False positive"
ws['E1'] = "Flase negative"
ws['F1'] = "Accuracy"
ws['G1'] = "Sensitivity"
ws['H1'] = "Specificity"
count = 2

for ff in files:
    file =  loc + "/" + ff
    test_image = cv2.imread(file,1)

    #test image is converted to LAB modal
    lab = cv2.cvtColor(test_image,cv2.COLOR_BGR2LAB)
    l, a, b = cv2.split(lab)

    #Contrast Limited Adaptive Histogram Equalization is applied
    clahe = cv2.createCLAHE(clipLimit=3.0)
    cl = clahe.apply(l)
    limg = cv2.merge((cl,a,b))

    #LAB modal converted back to RGB
    final = cv2.cvtColor(limg, cv2.COLOR_LAB2BGR)


    #applying alternate sequential filtering
    blue,green,red = cv2.split(final)
    r1 = cv2.morphologyEx(green, cv2.MORPH_OPEN,
cv2.getStructuringElement(cv2.MORPH_ELLIPSE,(5,5)), iterations = 1)
    R1 = cv2.morphologyEx(r1, cv2.MORPH_CLOSE,
cv2.getStructuringElement(cv2.MORPH_ELLIPSE,(5,5)), iterations = 1)
```

```
    r2 = cv2.morphologyEx(R1, cv2.MORPH_OPEN,
cv2.getStructuringElement(cv2.MORPH_ELLIPSE,(11,11)), iterations = 1)
    R2 = cv2.morphologyEx(r2, cv2.MORPH_CLOSE,
cv2.getStructuringElement(cv2.MORPH_ELLIPSE,(11,11)), iterations = 1)
    r3 = cv2.morphologyEx(R2, cv2.MORPH_OPEN,
cv2.getStructuringElement(cv2.MORPH_ELLIPSE,(23,23)), iterations = 1)
    R3 = cv2.morphologyEx(r3, cv2.MORPH_CLOSE,
cv2.getStructuringElement(cv2.MORPH_ELLIPSE,(23,23)), iterations = 1)
    f4 = cv2.subtract(R3,green)
    f5 = clahe.apply(f4)

    #tophat morphological transformation
    image1 = f5
    e_kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE,(5,5))
    closeImg = cv2.morphologyEx(image1, cv2.MORPH_CLOSE, e_kernel)
    revImg = closeImg
    topHat = image1 - revImg

    #otsu with probability and minimization function
    imge = topHat
    blur = cv2.GaussianBlur(imge,(5,5),0)

    hist = cv2.calcHist([blur],[0],None,[256],[0,256])
    hist_norm = hist.ravel()/hist.max()
    Q = hist_norm.cumsum()
    bins = np.arange(256)
    fn_min = np.inf
    thresh = -1
    for i in range(1,256):
        p1,p2 = np.hsplit(hist_norm,[i]) #probabilities
        q1,q2 = Q[i],Q[255]-Q[i] #cum sum of classes
        b1,b2 = np.hsplit(bins,[i]) #weights
        #finding means and variances
        if q1 == 0:
            q1 = 0.0000001
        if q2 == 0:
            q2 = 0.0000001
        m1,m2 = np.sum(p1*b1)/q1, np.sum(p2*b2)/q2

        v1,v2 = np.sum(((b1-m1)**2)*p1)/q1,np.sum(((b2-m2)**2)*p2)/q2

        #calculates the minimization function
        fn = v1*q1 + v2*q2
        if fn < fn_min:
            fn_min = fn
            thresh = i

    #find otsu's threshold value with OpenCV function
    ret, otsu = cv2.threshold(blur,0,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)
    #removing very small contours through area parameter noise removal
    ret,f6 = cv2.threshold(f5,15,255,cv2.THRESH_BINARY)


   mask = np.ones(f5.shape[:2], dtype="uint8") * 255
    im2, contours, hierarchy =
cv2.findContours(f6.copy(),cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
    for cnt in contours:
        if cv2.contourArea(cnt) <= 255:
            cv2.drawContours(mask, [cnt], -1, 0, -1)
    im = cv2.bitwise_and(f5, f5, mask=mask)
    ret,fin = cv2.threshold(im,15,255,cv2.THRESH_BINARY_INV)
    newfin = cv2.erode(fin, cv2.getStructuringElement(cv2.MORPH_ELLIPSE,(3,3)),
iterations=1)

    #removing blobs of unwanted size
    fundus_eroded = cv2.bitwise_not(newfin)
    xmask = np.ones(fundus_eroded.shape[:2], dtype="uint8") * 255
    x1, xcontours, xhierarchy =
cv2.findContours(fundus_eroded.copy(),cv2.RETR_LIST,cv2.CHAIN_APPROX_SIMPLE)
    for cnt in xcontours:
        shape = "unidentified";
        peri = cv2.arcLength(cnt, True);
        approx = cv2.approxPolyDP(cnt, 0.04 * peri, False)
        if len(approx) > 4 and cv2.contourArea(cnt) <= 3000 and cv2.contourArea(cnt) >= 100:
            shape = "circle";
        else:
            shape = "vessels";
        if(shape=="circle"):
            cv2.drawContours(xmask, [cnt], -1, 0 ,-1)

    finimage = cv2.bitwise_and(fundus_eroded,fundus_eroded,mask=xmask)

    blood_vessels = cv2.bitwise_not(finimage)
    kernel = np.ones((2,2), np.uint8)
    blood_vessels = cv2.subtract(255, blood_vessels)

    #eroding is done to eliminate the minor noises (post processing)
    new = cv2.morphologyEx(blood_vessels, cv2.MORPH_OPEN, kernel)
    new1 = cv2.morphologyEx(new, cv2.MORPH_CLOSE, kernel)

    #saving the blood vessel extracted image
    cv2.imwrite('D:/Project/test/bv/' + ff + 'bvextracted.jpg',new1)

    #reading the masked image
    mask_image = cv2.imread('01_test_mask.jpg',0)

    #reading the manually segmented blood vessel images for results calculation
    fil =  location + "/" + files1[count1]
    count1 = count1 + 1
    truth_image = cv2.imread(fil,0)
    prediction = new1
```

```
for i in range(0, prediction.shape[0]):
    for j in range(0, prediction.shape[1]):
        if prediction[i, j] > 127:
            prediction[i, j] = 1


        else:
            prediction[i, j] = 0

for i in range(0, truth_image.shape[0]):
    for j in range(0, truth_image.shape[1]):
        if truth_image[i, j] > 127:
            truth_image[i, j] = 1
        else:
            truth_image[i, j] = 0

for i in range(0, mask_image.shape[0]):
    for j in range(0, mask_image.shape[1]):
        if mask_image[i, j] > 127:
            mask_image[i, j] = 1
        else:
            mask_image[i, j] = 0

#initialising the four parameters
TruePositive = 0.001
TrueNegative = 0.001
FalsePositive = 0.001
FalseNegative = 0.001

for i in range(0, mask_image.shape[0]):
    for j in range(0, mask_image.shape[1]):

        pred = prediction[i, j]
        truth = truth_image[i, j]
        mask = mask_image[i, j]

        if mask == 1:

            #finding the number of matches of each parameter
            if pred == 1 and truth == 1:
                TruePositive = TruePositive + 1
            else:
                if pred == 0 and truth == 0:
                    TrueNegative = TrueNegative + 1
                else:
                    if pred == 0 and truth == 1:
                        FalseNegative = FalseNegative + 1
                    else:
                        if pred == 1 and truth == 0:
                            FalsePositive = FalsePositive + 1
    #calculating the results(accuracy, sensitivity & specificity)
    accuracy = float((TruePositive + TrueNegative)) / float((TruePositive + FalsePositive +
FalseNegative + TrueNegative))
    sensitivity = float((TruePositive)) / float((TruePositive + FalseNegative))
    specificity = float((TrueNegative)) / float((TrueNegative + FalsePositive))



    #to avoid run-time exceptions
    try:
        positivePredictiveValue = float((TruePositive)) / float((TruePositive + FalsePositive))
    except Exception:
        positivePredictiveValue = 0

    #stores the calculated results in an excel sheet
    ws['A' + str(count)] = file
    ws['B' + str(count)] = abs(TruePositive)
    ws['C' + str(count)] = abs(TrueNegative)
    ws['D' + str(count)] = abs(FalsePositive)
    ws['E' + str(count)] = abs(FalseNegative)
    ws['F' + str(count)] = accuracy
    ws['G' + str(count)] = sensitivity
    ws['H' + str(count)] = specificity
    count = count + 1

#saving the excel file
wb.save("output.xlsx")
```

10. REFERENCES

1) Rahul Chauhan, Anta Uniyal and V.P.Dubey "Detection of retinal blood vessels and reduction of false microaneurysms", Emerging Trends in Communication Technologies (ETCT) International Conference, November 2016.
2) Khan BahadarKhan, Amir A Khaliq and Muhammad Shahid "A Morphological Hessian Based Approach for Retinal Blood Vessels segmentation and denoising using region based Otsu thresholding", DICTA, 2016.
3) Tsuyoshi Inoue, Yuji Hatanaka, Susumu Okumura, Chisako Muramatsu and Hiroshi Fujita "Automated Microaneurysm detection method based on Eigen value Analysis using Hessian Matrix in Retinal fundus images ", 35th Annual International Conference of the IEEE, 7 July, 2013.
4) Toufique A Soomro, Junbin Gao, Mohammad A, U. Khan, Tariq M Khan and Manoranjan Paul "Automatic Retinal Vessel Extraction Algorithm ", Digital Image Computing: Techniques and Applications (DICTA), 2016.
5) Wei Zhou, Chengdong Wu, Dali Chen, Yugen Yi and Wenyou Du "Automatic Microaneurysm detection Using the Sparse Principal Component Analysis-Based Unsupervised Classification Method ", IEEE Access (Volume:5), March 15, 2017.
6) Balint Antal, Andras Hajdu "Evaluation of Preprocessing Methods for Microaneurysm Detection ", 8th International Symposium on Image and Signal processing analysis, September 3, 2013.
7) A.Elbalaoui, M.Fakir, K.Taifi and A.Merbouha "Automatic detection of blood vessel in retinal images", 13th International Conference Computer graphics, Imaging and Visualisation, 2016.
8) Syed Ayaz Ali Shah, Augustinus Lande, Ibrahima Faye and Tong Boon Tang "Automated Microaneurysm detection in diabetic retinopathy using curvelet transform ", Journal of Biomedical Optics, October 2016.
9) G Nagarjuna Reddy, K Durga Ganga Rao "Microaneurysm Identification using Cross Sectional Profile Analysis with optic disc removal ", Communication and Electronics Systems (ICCES) International conference, 2016.
10) Syna Sreng, Noppadol Maneerat and Kazhuhiko Hamamoto "Automated Microaneurysm detection in Fundus images using Image Segmentation ", Digital Arts, Media and Technology (ICDAMT) International Conference, 2017.

11) Features of Diabetic Retinopathy
http://www.glycosmedia.com/education/diabetic-retinopathy/diabetic-retinopathy-features-of-diabetes-microaneurysms/

12) Diabetic Retinopathy
http://www.diabeticretinopathy.org.uk/diabetic_retinopathy_mech.html