TABLE OF CONTENTS

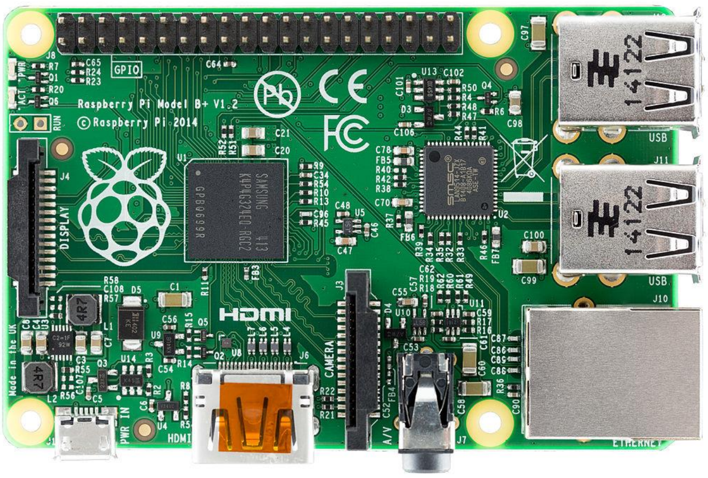
INTRODUCTION

Diabetes has become a significant public health problem with the increasing number of patients every year in the world. According to estimation of WHO the number of diabetic patient will reach 366 million by 2030 [1]. Diabetic retinopathy (DR) disease is one which is provoked by diabetic's mellitus complications. It occurs due to leakage of blood and protein from small diseased vessels into retina for prolonged period of hyper glycaemia which can damage the vision of patient severely. Diabetic retinopathy can be classified into two types: Non-Proliferative diabetic retinopathy and Proliferative diabetic retinopathy. Non proliferative retinopathy is the early stage when capillaries start bleeding with the accumulation of proteins and lipids in the retina. Proliferative retinopathy is the advance stage when new abnormal vessels start to develop in the retina [2]. Since diabetic retinopathy disease cannot be cured and patients are unaware of symptoms until loss of vision starts. The detection of diabetic retinopathy at early stage can be helpful for prevention of major vision loss. Therefore the prevention of disease requires the annually examination of diabetic patients for diabetic retinopathy so that patient can receive treatment in time [3]. It leads to the development of computer aided diagnosis system for the treatment of huge number of ophthalmologic patients. Moreover, analysis of retinal blood vessels for variations helps the ophthalmologists for the diagnosis of several other diseases such as glaucoma, obesity and hypertension [4]. The extraction of retinal vascular tree can also be used for biometric identification since every individual has unique pattern [5]. The manual segmentation of blood vessel from retinal images is time consuming and laborious task which requires training and skills. Therefore, researchers start to develop robust automated methods for the extraction of blood vessels, removal of optical disk (OD) and estimation of vessel diameter from digitized retinal images. A retinal image consists of interior surface of eye whose acquisition is done with the help of complex optical system, known as fundus camera.

In this paper, an effort is made for industrial application of the detection of 'Blood vessels from fundus images of the eye'. Pre-existing algorithms, from reviewed literature have been modified and morphed with new ones, to create new, efficient and basic algorithms which have been developed on the Matlab 2013R environment and then subsequently converted into respective High level Languages compatible with the Raspberry Pi microcontroller and then run on the Raspberry Pi.

Raspberry Pi

- This is a mini-computer developed by Raspberry Pi Foundation in Britain primarily for school teaching in the subject of CS.

- 

- The Foundation gives Debian and Arch Linux ARM circulations for download. Apparatuses are accessible for Python as the fundamental programming dialect, with backing for BBC BASIC (by means of the RISC OS picture or the Brandy Basic clone for Linux) C, C++, Java, Perl and Ruby.

- The SoC utilized as a part of the original Raspberry Pi is to some degree proportionate to the chip utilized as a part of more established cell phones, (for example, iPhone/3G/3GS). The Raspberry Pi is taking into account the Broadcom BCM2835 framework on a chip (SoC),[1] which incorporates a 700 MHz ARM1176JZF-S processor, VideoCore IV GPU,[7] and RAM. It has a Level 1 store of 16 KB and a Level 2 reserve of 128 KB. The Level 2 store is utilized basically by the GPU. The SoC is stacked underneath the RAM chip, so just its edge is noticeable.

<u>Peripherals</u>

- Mouse and keyboard are used as the input devices for the computer. Any USB Keyboard and mice are acceptable.
- Ethernet connection is available and is necessary for the purpose of upgrading and updating the system.
- HDMI peripheral is available which is used to connect to a HDMI Display such as TV or LCD Screens.
- Micro SD slot is available in Raspi 2 with a maximum capacity of 64 GB.

Software and Architecture

- The Raspberry Pi fundamentally utilizes Linux-piece based working frameworks.
- The ARM11 chip at the heart of the Pi (original models) is taking into account form 6 of the ARM. The present arrivals of a few well known renditions of Linux, including Ubuntu, won't keep running on the ARM11. It is unrealistic to run Windows on the first Raspberry Pi, however the new Raspberry Pi 2 will have the capacity to run Windows 10.[68] The Raspberry Pi 2 right now just backings Ubuntu Snappy Core, Raspbian, OpenELEC and RISC OS.

GNU Octave

Development

- The venture was imagined around 1988. At first it was proposed to be a partner to a concoction reactor plan course. Genuine improvement was begun by John W. Eaton in 1992. The primary alpha discharge goes back to January 4, 1993 and on February 17, 1994 form 1.0 was discharged. Variant 4.0.0 was discharged on May 29, 2015.
- The system is named after Octave Levenspiel, a previous teacher of the vital creator. Levenspiel is known for his capacity to perform brisk back-of-the-envelope calculations.

Technical Details

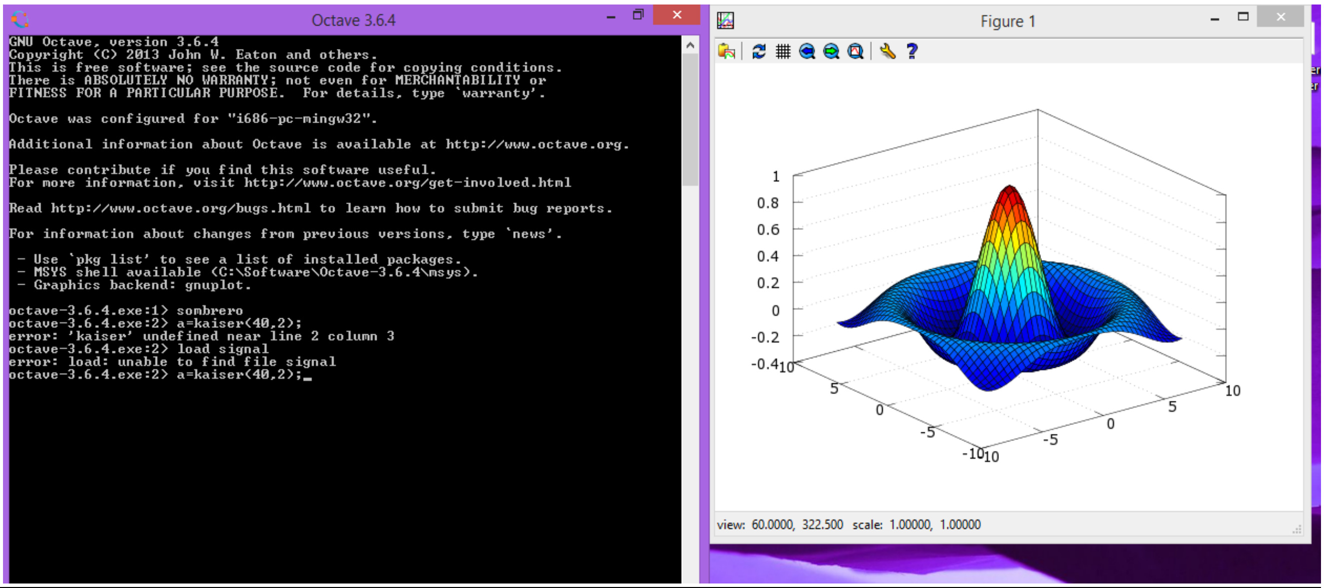
- Octave is composed in C++ utilizing the C++ standard library.
- Octave utilizes a mediator to execute the Octave scripting dialect.
- Octave is extensible utilizing progressively loadable modules.
- Octave mediator meets expectations with gnuplot and Grace programming to make plots, diagrams, and graphs, and to spare or print them.
- Octave forms 3.8.0 and later incorporate a Graphical User Interface (GUI) notwithstanding the customary Command Line Interface (CLI).

Comments

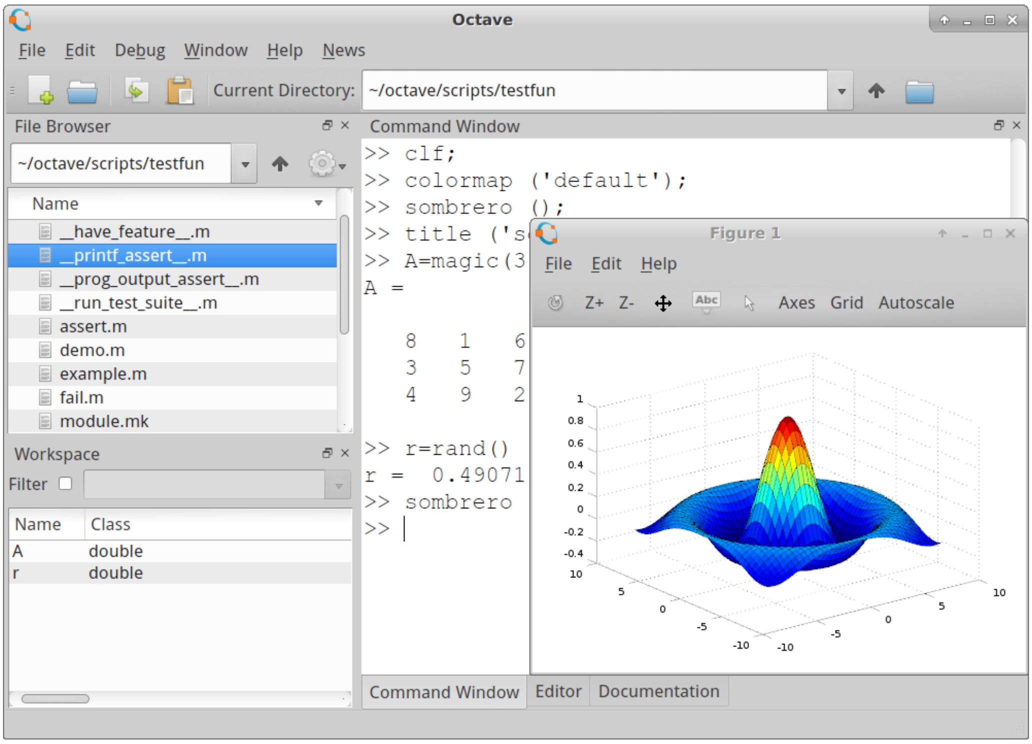
- One of the main features of MATLAB is that it can be compiled only on the Intel’s x86 Architecture and not on any other.

- Raspberry Pi uses ARM Architecture and hence MATLAB can't run on this architecture. Moreover, Raspi doesn't have that much of RAM available to run the software.
- The best alternative to MATLAB is GNU Octave which can run on Raspberry Pi hardware. Most of the codes work directly on Octave.
- The main difference between MATLAB and Octave versions before 3.8 is that Octave only has a command line interface whereas MATLAB has a great GUI.
- But the recent version of Octave 4.0.0 has an editor window like MATLAB.
- Another distinction between Octave and MATLAB is that MATLAB comes preinstalled with most libraries whereas it has to be downloaded in Octave. The versions of the packages differ substantially in their functions. Ex., Image package of 2.2 doesn't have many functions such as strel, regionprops, etc. Whereas, Image 2.4.0 has these and can only run alongside Octave 4.0

Octave without editor (Command Line)



Octave with editor window(4.0.0)



OpenCV Python

Image processing and computer vision (CV) has been driven by development in C/C++ and the usage of MATLAB software[1]. Although MATLAB offers an efficient high level platform for prototyping and testing algorithms, its performance does not compete with a well designed and optimised C/C++ implementation. Recently, potential and valuable solutions have emerged for developing image processing and computer vision algorithms in Python[2]. For image processing or computer vision development, two Python libraries are prominently used: NumPy/SciPy and OpenCV with a Python wrapper. This paper looks at using basic computer vision routines from OpenCV and SciPy.

Tools

NumPy NumPy gives strongly typed N-dimensional array support to Python[3]. The library is well recognised and offers an easier approach for multidimensional array manipulation than in the C programming language. A large part of the low level algorithms are implemented in C resulting in very fast and optimised raw data processing and iterating. NumPy can be found online at http://numpy.scipy.org

SciPy SciPy[4] is a set of Python libraries and tools for scientific and mathematical work built on top of NumPy [5]. SciPy offers many different modules including routines such as numerical integration, optimisation, signal processing and image processing/computer vision functions. Two major tools often used in conjunction with SciPy are very useful for computer vision development; Matplotlib and IPython. Matplotlib[6] is an array and image plotting library, and IPython[7] is an improved interactive shell for Python. SciPy can be downloaded from http://www.scipy.org

MATLAB

MATLAB (**mat**rix **lab**oratory) is a multi-paradigm numerical computing environment and fourth-generation programming language. A proprietary programming language developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, Fortran and Python.

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing capabilities. An additional package, Simulink, adds graphical multi-domain simulation and model-based design for dynamic and embedded systems.

BLOOD VESSELS DETECTION

PRE-PROCESSING STAGE

Image Acquisition All digital retinal images are taken from patients using the non-mydriatic retinal fundus camera. The images are stored in JPEG image format file (.jpg). The original (RGB) image is transformed into appropriate colour space for further processes. And then, filtering technique is used to reduce the effect of noise. After using the filtering technique, the noise such as salt and pepper noise are removed from the image. Then contrast-limited adaptive histogram equalization (CLAHE) is used for image enhancement. Unlike histogram, it operates on small data regions rather than the entire image. This function uses a contrast-enhancement method that work significantly better than regular histogram equalization for most images.

Filtering Techniques

Noise can cause the trouble in the detection of disease. The noise contains in the image is reduced by using the filtering technique such as median filter, averaging filter and wiener filter.

Median Filter

The median filter is a non-linear filter type and which is used to reduce the effect of noise without blurring the sharp edge. The operation of the median filter is – first arrange the pixel values in either the ascending or descending order and then compute the median value of the neighbourhood pixels.

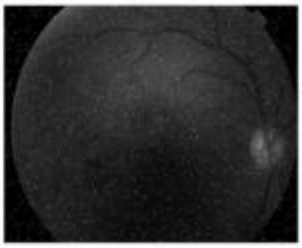


Fig- 4: (a) Image with noise

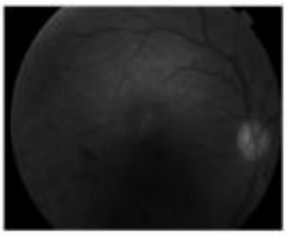


Fig-4: (b) Result of median filter

Averaging Filter

Averaging filter is useful for removing grain noise from a photograph. Each pixel gets set to the average of the pixels in its neighbourhood. The result of the averaging filter is shown in Fig.5 (b).

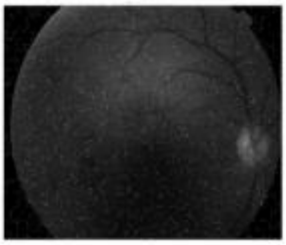


Fig-5: (a) Image with noise

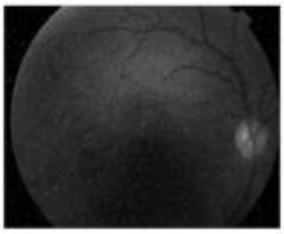


Fig-5: (b) Result of averaging filter

Wiener Filter

The wiener filter is used to minimize the mean square error between input and output image. But the wiener filter requires knowing the power spectral density of the original image which is unavailable in practice. The result of the wiener filter is shown in Fig.6 (b).

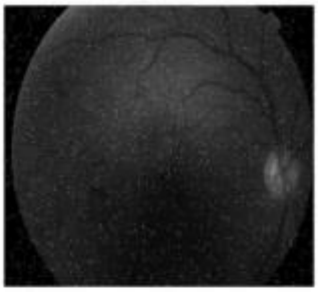


Fig-6: (a) Image with noise

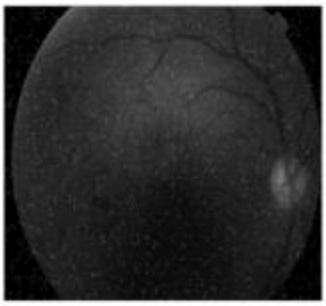


Fig-6: (b) Result of wiener filter

The median filter is the best suit to reduce the effect the noise. And also, it can reduce the noise without blurring the edge. Therefore, the median filter is chosen for the filtering purpose.

Image Enhancement

The result image of the median filter is enhanced by using the histogram equalization technique. The histogram equalization technique is used to overcome the uneven-illumination case. There are two methods to enhance the image: Histogram equalisation and Adaptive histogram equalisation.

Histogram Equalisation

It enhances the contrast of the images by transforming the values in an intensity image. The procedures of the histogram equalisation are- (i) Find the running sum of the pixel values (ii) Normalise the values by dividing the total number of pixels (iii) Multiply by the maximum gray-level value and round the value The result of the histogram equalization is shown in Fig.7 (b)

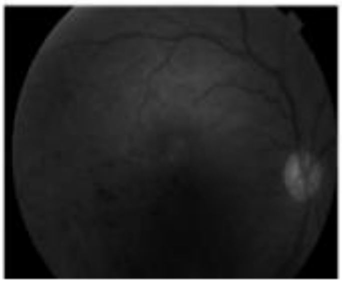


Fig: 7- (a) Original Image

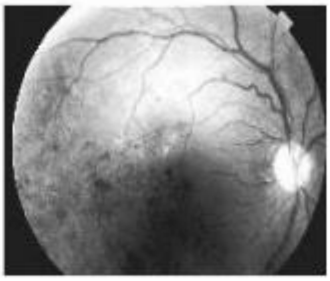


Fig-7: (b) Result of histogram Equalisation

Adaptive Histogram Equalisation

Unlike histogram, it operates on small data regions (tiles) rather than the entire image. And also contrast enhancement can be limited in order to avoid amplifying the noise which might be presented in the image. So, Adaptive histogram equalisation technique works significantly better than regular histogram equalization for most images.

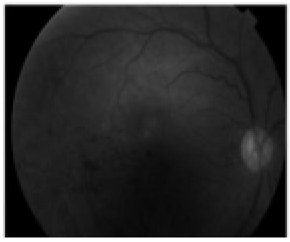


Fig-8: (a) Original Image

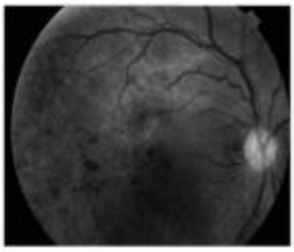


Fig-8: (b) Result of adaptive histogram equalization

According to results, the adaptive histogram equalisation technique is used for image enhancement purpose.

Mathematical Morphology

The basic mathematical morphology operators include the following:

- Dilation
- Erosion

- Closing
- Opening

Dilation adds pixels to the boundaries of objects in an image. Erosion removes pixels on object boundaries. The morphological open operation is an erosion followed by a dilation, using the same structuring element for both operations.

$A \circ B = (A \ominus B) \oplus B$

The closing operator is a dilation followed by erosion.

$A \bullet B = (A \oplus B) \ominus B$

Thresholding

The Otsu's thresholding technique is applied to the image to detect the desire area.

Equations of Otsu algorithm are

$$\sigma^2_{Between}(T)=w_B(T)w_o(T)[\mu_B(T)-\mu_o(T)]^2$$

$$w_B(T)=\sum_{i=0}^{T-1} p(i) \quad , \mu_B=\sum_{i=0}^{T-1}\left(\frac{ip(i)}{p(i)}\right)$$

$$w_0(T)=\sum_{i=T}^{L-1} p(i) \quad , \mu_O=\sum_{i=T}^{L-1}\left(\frac{ip(i)}{p(i)}\right)$$

- $\sigma^2_{Between}(T)$= Between-class variance
- w=weight, B=background of the image, o=object of image
- μ= combined mean,
- T= threshold value

The blood vessels detection is important for further process because the blood vessels are the normal features of the images. The general flow chart of the blood vessels detection is shown in Fig. To detect the blood vessels, first the input image is converted into grayscale image due to strengthen the appearance of the blood vessels. Then the median filtering and the CLAHE techniques are used for reducing noise and image enhancement purposes. Then, the closing and the filling operators are used to close the same intensity values and fill the holes in the vessels.

I(Difference)=φ (B1)(I)-fill(I)

Where, B1 is the morphological structuring element.

Then the Otsu's thresholding technique is applied to the result image to obtain the vessels area. The blood vessels detected area is shown in Fig. 8(e).

I(vessels)=Thresholding(I(Difference))

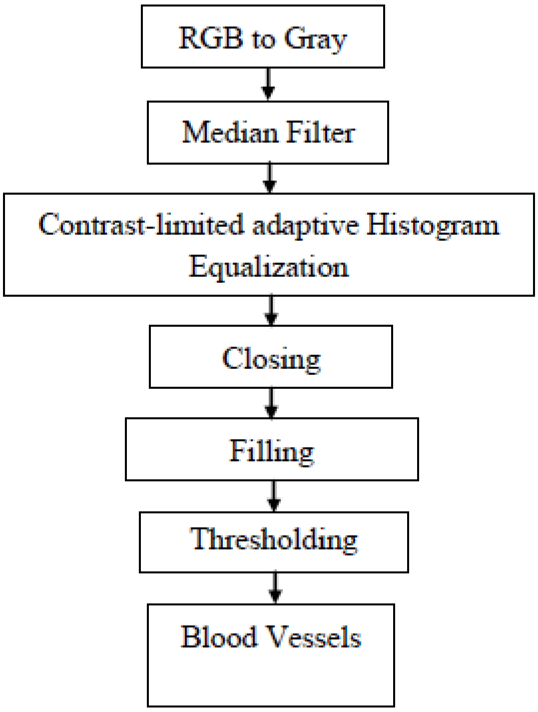


Fig-7: Flow Chart of Blood Vessels Detection

MATLAB Code

```
t=('image.jpg');
 i=imread(t);
 figure, imshow(i)
 q=rgb2hsv(i);
 t=rgb2gray(i);
 figure, imshow(t)
 k=medfilt2(t);
 figure, imshow(k)
 u=adapthisteq(k);
 se=strel('disk',10);
 c=imclose(u,se);
 figure, imshow(c)
 d=imfill(u,'holes');
 figure, imshow(d)
 z=imabsdiff(c,d);
 figure, imshow(z)
 level=graythresh(z);
 BW=im2bw(z,level);
 imshow(BW)
```

OpenCV-Python Code

```
import cv2
import numpy as np
img=cv2.imread('/home/pi/Desktop/eye.jpg&f=1', 0)
j=cv2.medianBlur(img, 3)
clahe=cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))
k=clahe.apply(j)
cv2.imwrite('clahe.jpg', k)
kernel=np.ones((5,5), np.uint8)
closing=cv2.morphologyEx(k, cv2.MORPH_CLOSE, kernel)
h,w=img.shape[:2]
mask=np.zeros((h+2, w+2), np.uint8)
seed_pt=None
flooded=k.copy()
cv2.floodFill(flooded, mask, seed_pt, 220, 0, 3)
z=cv2.absdiff(closing,flooded)
ret, th1=cv2.threshold(z, 10, 255, cv2.THRESH_BINARY)
cv2.imwrite('threshold.jpg', th1)
cv2.imshow('image', th1)
cv2.waitKey(0)
```

GNU Octave code

```
clear all:
close all;
clc
[kk1, pathname] = uigetfile({'*.*';'*.*'},'Select a image File from image
folder');
 k2=strcat(pathname,kk1);
 im = imread(k2);

%img = imread('C:\MATLAB7\work\MF_DOF\DRIVE\test\images\01_test.tif');
[m n]=size(im);
im_gray=rgb2gray(im );
figure,imshow(im_gray)
[x y]=size(im_gray);
display(x);
display(y);
maskForGDRange = zeros(x,y);
mask= zeros([x y]);
for i=1:x
    for j=1:y
        if   (im_gray(i,j)>10)
```

```
            mask(i,j)=255;
        else
            maskForGDRange(i,j) = 1;
        end
    end
end

se = strel('disk',5);
     mask = imerode(mask,se);
%     im_name = strcat('E:\Computer Vision\Copy of MatchFilter\stare-images\im',num2str(k,'%04d'),'.ppm');
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%im_name = strcat('D:\oruami\Sem 8\FinalYearProject_5Dec13\MF_DOF\DRIVE\test\images\',num2str(k,'%02d'),'_test','.tif');
%     im_name = strcat('C:\Documents and Settings\Administrator\My Documents\MATLAB\DRIVE\DRIVE\test\images\',num2str(k,'%02d'),'_test.tif');
    %im = imread(im_name);
%     mask = bwselect(im2bw(im(:,:,1), 20/255), size(im(:,:,1),1)/2, size(im(:,:,1),2)/2);
    se = strel('disk',2);
    mask=imerode(mask,se);
    im = double(im(:,:,2));
%     man_name = strcat('E:\Computer Vision\Copy of MatchFilter\stare-images\labels-ah\im',num2str(k,'%04d'),'.ah.ppm');
%    man_name = strcat('D:\matlab_final\CBM_Bob_code\all-images\im',num2str(k,'%04d'),'.ppm');
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% man_name = strcat('C:\MATLAB7\work\MF_DOF\DRIVE\test\1st_manual\',num2str(k,'%02d'),'_manual1.gif');
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% man = imread(man_name);
        [vess1] = MatchFilterWithGaussDerivative([],1,im,2,2,9,5,41,201,8,mask,maskForGDRange,3,40);
        [vess2] = MatchFilterWithGaussDerivative([],2,im,1,1,9,5,21,101,8,mask,maskForGDRange,2,40);
     vess = vess1 | vess2;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% out_name = strcat('C:\MATLAB7\work\MF_DOF\DRIVE\test\output\',num2str(k,'%02d'),'_out_n','.gif');
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   imwrite(vess,out_name);
  %out_name = strcat('C:\MATLAB7\work\MF_DOF2\output1\',num2str(kk,'%02d'),'_out_n','.tif');
    % imwrite(vess,out_name);
    figure,imshow(vess);

    % yd=vess1-vess2;
     %imshow(yd)



%BW2 = edge(yd,'canny');
```

RESULTS

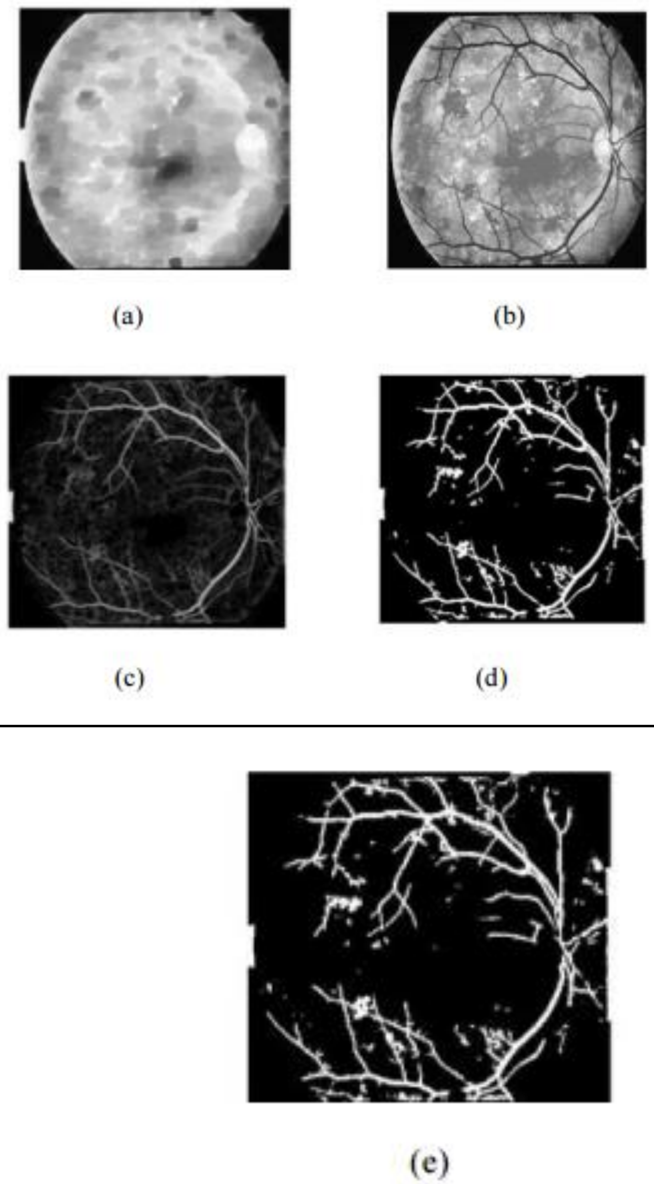


Fig-8:(a) Closing (b) Filling (c) Difference (d) Thresholding
(e) Detected Blood Vessels

The results of the blood vessels detection are shown in Fig.8 (a), (b), (c), (d) and (e).The closing and the filling operators are used to close the same intensity values and fill the holes in the vessels. The result of the closing and the filling of the images are shown in Fig.8 (a) and 8 (b). To get the blood vessels area, Otsu algorithm is applied to the difference image between the closing and the filling images. The result images are shown in Fig.8 (c) and (d). The blood vessels detected area is shown in Fig.8 (e).

APPLICATIONS AND FUTURE IMPLICATIONS

- Analysis of retinal blood vessels for variations helps the ophthalmologists for the diagnosis of several other diseases such as glaucoma, obesity and hypertension [4].
- The extraction of retinal vascular tree can also be used for biometric identification since every individual has unique pattern [5].

The manual segmentation of blood vessel from retinal images is time consuming and laborious task which requires training and skills. Therefore, researchers start to develop robust automated methods for the extraction of blood vessels, removal of optical disk (OD) and estimation of vessel diameter from digitized retinal images. A retinal image consists of interior surface of eye whose acquisition is done with the help of complex optical system, known as fundus camera. With the onset of the digital revolution, new technologies in forms of optics, hardware and software has opened up new fronts for applications in this subject.

OBSERVATIONS AND CONCLUSION

From above explanations, it can be understood as to how small but important part of medical image processing is the above stated topic. Applications of the topic 'Implementation of Blood Vessels Detection in Fundus Images via Raspberry Pi' are immense ranging from medical ailments detection to increasing prowess in the fields of biometric detection, image processing, etc.

During the scope of the project, several crucial observations were made. Between Octave and OpenCV-Python, though the latter is a more powerful programming tool, the implementation of the former is easier and more efficient on the Raspberry Pi. The present image processing functions that are available in OCV-Python are not exactly equitable to their named counterparts developed on the Matlab platform and thus errors sneak into the final output which is not satisfactory. Thus, GNU Octave presents a more elegant option, providing the developer to focus on the evolution of the algorithms.

REFERENCES

[1]. W. Hsu, P.M.D.S Pallawala, Mong Li Lee et al., “The Role of Domain Knowledge in the Detection of Retinal Hard Exudates,” In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2, 2001, pp.II-246 - II-251.

[2]. C.I. Sanchez, R. Hornero, M.I. Lopez et al., “Retinal Image Analysis to Detect and Quantify Lesions Associated with Diabetic Retinopathy,” In Proceedings of 26th IEEE Annual International Conference on Engineering in Medicine and Biology Society (EMBC) 1, 2004, pp.1624 – 1627.

[3]. C.Sinthanayothin, J. F. Boyce, H. L. Cook, and T. H.Williamson, “Automated localisation of the optic disc, fovea, and retinal blood vessels from digital colour fundus images,” Br. J. Ophthalmol., vol. 83, pp.902–910, 1999.

[4]. A. Osareh, M. Mirmehdi, B. Thomas, and R. Markham, “Automated identification of diabetic retinal exudates in digital colour images,” Br.J. Ophthalmol., vol. 87, pp. 1220–1223, 2003.

[5]. Yang, S.Huang, N.Rao M, “An automatic hybrid Method For Retinal Blood Vessel Extraction”, International Journal of Applied Math. Comput. Sci,2008, Vol18,No.3, pp 399-407,2008

[6]. A. Osareh, M. Mirmehdi, B. Thomas, and R. Markham, “Comparisonof colour spaces for optic disc localisation in retinal images,” in Proc.16th Int. Conf. Pattern Recognit., 2002, pp. 743–746.

[7]. J. Lowell, A. Hunter, D. Steel, A. Basu, R. Ryder, E. Fletcher, and L. Kennedy, “Optic nerve head segmentation,” IEEE Trans. Med. Imag., vol. 23, no. 2, pp. 256–264, Feb. 2004.

[8]. J. Xu, O. Chutatape, E. Sung, C. Zheng, and P. C. T. Kuan, “Optic disk feature extraction via modified deformable model technique for glaucoma analysis,” Pattern Recognit., vol. 40, no. 7, pp. 2063–2076, 2007.

[9]. M. Lalonde, M. Beaulieu, and L. Gagnon, “Fast and robust optic disk detection using pyramidal decomposition and Hausdorff-based template matching,” IEEE Trans. Med. Imag., vol. 20, no. 11, pp. 1193–1200, Nov. 2001.

[10]. C.Sinthanayothin, J.F.Boyce, T.Williamson, H.Cook, E.Mensah,andD.Usher, “Automated detection of diabetic retinopathy on digital fundus images”, Diabetic medicine, Vol 19, pp.105- 112, 2002.

[11]. A.Frame, P.Undrill, M.Cree, J.Olson, K.McHardy, P.Sharp and J.Forrester, “A comparision of computer based classification methods applied to the detection of microneurysms in ophthalmic fluroscein angiograms”, computer Biol. Med, Vol.28, pp.225-238, 1998

[12]. G.Gardner, D.Keating, T.Williamson, and .Elliot, “Detection of Diabetic retinopathy using neural network analysis of fundus images”, Br. J. Opthalmology., Vol.80, pp 937-948,1996.

[13]. B.M.Ege, O.K.Hejlesen, O.VLarsen, K.Moller, B.Jennings, D.Kerrr, D.A.Cavan, “Screening for diabetic retinopathy using computer based image analysis and statistical classification”, computer Methods Programs Biomed, Vol63, pp.165-175, 2000.

[14]. D.Fleming, S.Philip, K.A.Goatman, J.A.Olson, P.F.Sharp, “Automated microaneurysm detection using local contrast normalisation and local vessel detection”, IEEE transactions on Medical Imaging, vol.25, pp.1223-1232, 2006.

[15]. Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins, Digital Image Processing Using MATLAB, 2 nd Edition, Prentice-Hall, 2002.

[16]. A. Sopharak, B. Uyyanonvara, S. Barman et al., “Automatic detection of diabetic retinopathy exudates from non-dilated retinal images using mathematical morphology methods,” Computer Medical Imaging and Graphics 32(8), 2008, pp. 720-727.

[17]. http://esatjournals.org/Volumes/IJRET/2014V03/I03/IJRET20140303056.pdf

[18]. http://docs.opencv.org/opencv2refman.pdf

[19]. The Python Papers Monograph, Vol. 1 (2009) Available online at http://ojs.pythonpapers.org/index.php/tppm

[20]. https://en.wikipedia.org/wiki/MATLAB

[21]. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&tp=&arnumber=7030686

[22]. http://melab.snu.ac.kr/melab/lib/exe/fetch.php?id=research%3Apublication%3Aconference&cache=cache&media=data:2004.sy.conference.pdf

[23]. http://pdfsr.com/pdf/automatic-detection-of-optic-disc-and-blood-vessels-from-retinal-images-using-image-processing-techn

[24]. http://www.citeulike.org/user/mzabidi/tag/python