A Mini Project report on

"RGB CHANNEL ANALYSIS FOR GLAUCOMA DETECTION IN RETINAL FUNDUS IMAGE"

Submitted

In partial fulfillment of the Requirement for the award of the Degree of

BACHELOR OF TECHNOLOGY

In

ELECTRONICS AND COMMUNICATION ENGINEERING

By

J. HARSHINI	19W91A082
K. SRIHITHA	19W91A04A2
G. NAVEEN KUMAR	20W95A0410

Under the Esteemed Guidance of

MISS. JHANSI RANI

Associate Professor

Department of Electronics & Communication Engineering

MALLA REDDY INSTITUTE OF ENGINEERING AND TECHNOLOGY

Approved by A.I.C.T.E. - New Delhi, Affiliated to J.N.T. University Hyderabad

Maisammaguda, Dhulapally, Secunderabad-500 100

December-2022

Malla Reddy Institute of Engineering and Technology

(Sponsored by Malla Reddy Educational Society)

ISO 9001-2008 Certified institution, Affiliated to JNTU, Hyderabad

Maisammaguda, Dhulapally (Post via Hakimpet), Sec'Bad - 500 100.

Department of Electronics and Communication Engineering

CERTIFICATE

This is to certify that the mini project entitled "RGB CHANNEL ANALYSIS FOR GLAUCOMA DETECTION IN RETINAL FUNDUS IMAGE" that is being submitted by J. HARSHINI (19W91A082),K. SRIHITHA (19W91A04A2),G. NAVEEN KUMAR (20W95A0410) under the guidance of MISS. JHANSI RANI for the award of B.Tech Degree in ELECTRONICS AND COMMUNICATION ENGINEERING from the MALLAREDDY INSTITUTE OF ENGINEERING & TECHNOLOGY, Maisammaguda (Affiliated to JNTU Hyderabad) is a record of Bonafide work carried out by themunder our guidance and supervision. The results embodied in this mini project have not been submitted to any other university or institute for the award of any degree.

Project Guide	HOD
MISS. JHANSI RANI	Dr. P. SAMPATH KUMAR
Department of ECE	Department of ECE

EXTERNAL EXAMINER

PRINCIPAL

Dr. M Ashok

Malla Reddy Institute of Engineering and Technology

(Sponsored by Malla Reddy Educational Society)

ISO 9001-2008 Certified institution, Affiliated to JNTU, Hyderabad

Maisammaguda, Dhulapally (Post via Hakimpet), Sec'Bad - 500 100.

Department of Electronics and Communication Engineering

DECLARATION

We, J. HARSHINI (19W91A082), K. SRIHITHA (19W91A04A2), G. NAVEEN KUMAR (20W95A0410) hereby declare that the mini project entitled"RGB CHANNEL ANALYSIS FOR GLAUCOMA DETECTION IN RETINAL FUNDUS IMAGE" is bonafide work done and submitted under the guidance of MISS. JHANSI RANI in partial fulfillment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in ELECTRONICS AND COMMUNICATION ENGINEERING.

DEPARTMENT OF ECE

J. HARSHINI	19W91A082
K. SRIHITHA	19W91A04A2
G. NAVEEN KUMAR	20W95A0410

ACKNOWLEDGEMENT

We are very much thankful to Director, **Shri. P. PRAVEEN REDDY** for giving us this opportunity to do this mini project. We express our deep sense of gratitude to him for his constant guidance and inspiring words.

We express our profound thanks to our Principal, **Dr. M. ASHOK,** for extending all the college facilities for the completion of the mini project.

We would like to thank **Dr. P. SAMPATH KUMAR** Professor and Head of the Department of Electronics and Communication Engineering & **Dr. RAJSREE RAO** Dean of Academics for having provided the freedom to use all the facilities available in the department, especially the laboratories and the library, at anytime.

We feel highly obliged to our project coordinator **Mr. SHAIK SOHEL PASHA (Ph.D)**, Asst Professor and project guide **MISS. JHANSI RANI** Associate Professor Department of Electronics and Communication Engineering for their constant encouragement and moral support. They have been a source of valuable guidance, suggestions and kindness during the course of the project work. We find no words to express our gratitude and thanks to them.

We sincerely thank all the staff of the Department of Electronics and Communication Engineering, for their timely suggestions, healthy criticism and motivation during the course of our study. We would also like to thank our friends for always being there to provide required help or support. With great respect and affection, we thank our parents who were the backbone behind our deeds.

Finally, we express our immense gratitude with pleasure to one and all who have either directly or indirectly contributed to our need at right time for the development and execution of project work.

J. HARSHINI 19W91A082

K. SRIHITHA 19W91A04A2

G. NAVEEN KUMAR 20W95A0410

CONTENTS

LIST OF FIGURES

RGB Channel Analysis for Glaucoma Detection in Retinal Fundus Image

ABSTRACT

The research that used digital image processing to detect glaucoma has been known as one of the popular methods. The beginning step of the research is to choose the best channel among red, green, or blue (RGB) so it can ease the glaucoma segmentation. In choosing the channel, it is important to analyze deeply the used retinal images. Choosing of best component can affect the accuracy of glaucoma diagnosis results. In this research, the most suitable component will be analyzed to detect glaucoma based on the visual, MSE (Mean Square Error) value, and PSNR (Peak Signal to Noise Ratio). In this research, we used 85 images of glaucoma from the DRISTHI-GS database and 101 normal images from the RIM-ONE database. From the visualization, it showed that the red component had high brightness level so it can differ the optic disc and other parts of retinal eyes. The green component still has vessel blood so it will make it more difficult to segment the images. Blue component results in very dark of retinal image. From MSE and PNSR values, it showed that the green component had the smallest MSE value while the blue component has the biggest MSE value. PSNR value was obtained from the green component. Both red and blue components had PSNR value which had a small difference. From these results, it can be concluded that the MSE and PSNR values do not guarantee visual results. So that for further research, it is expected that the MSE and PSNR values will be obtained from the part that we want to observe

Keywords—Glaucoma, RGB component, segmentation, MSE, PSNR.

CHAPTER 1
INTRODUCTION

Glaucoma is one of disease which can attack retinal eyes and cause blindness without earlier treatment. In many cases, some patients did not find any symptoms until they came into the blindness phase. Glaucoma is caused by increasing intraocular pressure of eyes which results in the destruction of nerve tissue in optics so the capacity of sight will decrease and cause blindness (in serious case). Glaucoma aggression can be delayed through treatment. That is why some people who potentially attacked by glaucoma because of the heredity factor can be treated routinely. Some treatments such as air-puff intraocular pressure (IOP) measurement, visual field test, and optic nerve head (ONH) assessment can be implemented but those tests are very expensive and can be found in hospitals only. So, the earlier detection of glaucoma that is has high accuracy and cheap is important to develop.

The research that used digital image processing to detect glaucoma has been known as one of the popular methods. The beginning step of the research is to choose the best channel among red, green, or blue (RGB) so it can ease the glaucoma segmentation. In choosing the channel, it is important to analyze deeply the used retinal images.

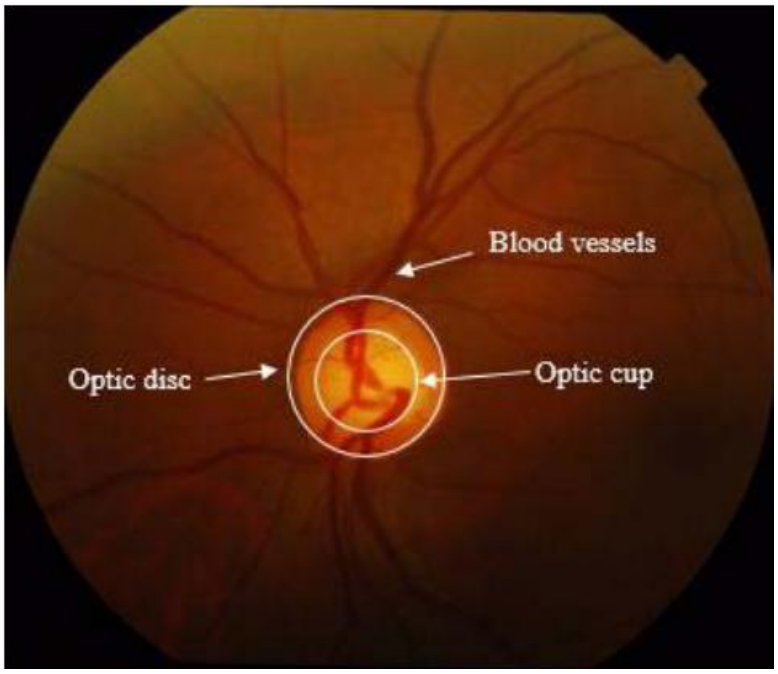


Fig 1.1. Retinal fundus image, optic disc and optic cup

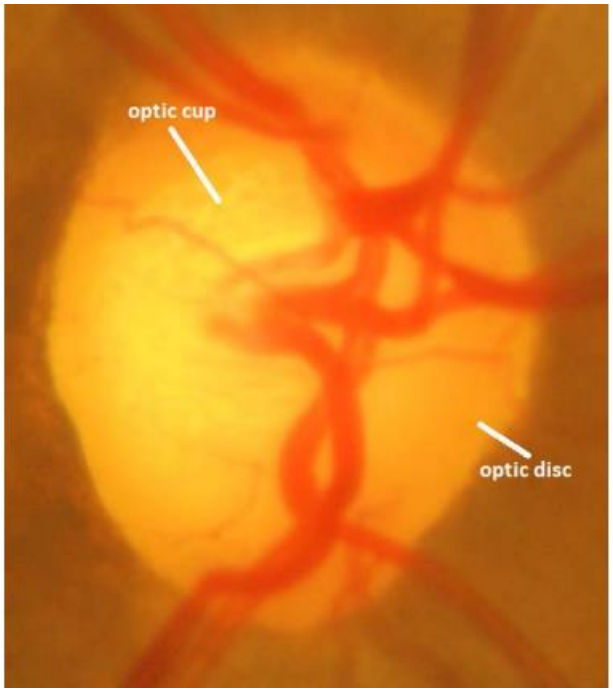


Fig 1.2. Optic disc and optic cup in detail aspect

Issac et al began his research by extracting the red component and green component for optic cup. Gaussian smoothing was used to equalize the distribution of pixel intensity in both histograms of the optic cup and optic disc. For segmentation optic disc, it will be used the obtained threshold from the calculation of size, deviation standard of Gaussian window, and the deviation standard of the imae which was obtained from the red component. Meanwhile, the mean value and deviation standard was used in the segmentation of optic cup. In this research, the Gaussian method can only be used at retinal data set which consist of noise that can be filtered by Gaussian filtering. Akhade et al. segmented optic disc by using the algorithm of principal component analysis (PCA), mathematics morphology, and Watershed Transform. Preprocessing was begun by analyzing the red component of the retina image and erase blood vessels using mathematics morphology especially closing operation. Nayak et al. Diagnosed glaucoma using three features including the ratio between optic disc and optic cup, the distance between optic disc center and the head of optic nerve, and the ratio of ISNT inferior, superior, nasal, and temporal. In the ratio between optic disc and optic cup, Nayak et al used red component for optic disc while green component for the optic cup. The Morphological operation was used to remove part of the blood vessel and deviation standard of the image was used for segmentation. From this research, there was not a detailed explanation of why the red component, blue component and green component was used. Choosing the best component can affect the accuracy of glaucoma diagnosis results. In this research, the most suitable component will be analyzed to detect glaucoma based on the visual, MSE (Mean Square Error) value, and PSNR (Peak Signal to Noise Ratio).

Age-related macular degeneration (AMD) is a typical medical condition, which can cause vision loss in elderly people. The retinal image is the key factor for AMD examination.

However, due to the quality of machines or the operator level, the poor quality retinal image may appear . These retinal images can affect the diagnosis of doctors. Recently, various retinal image analyzing studies are done using deep learning methods to classify retinal images to aid diagnosis and monitor AMD disease progression. However, the performance of those automatic image system can also decrease by the poor quality retinal images. Many previous studies were able to improve the quality of retinal images by adjusting the illumination or contrast. While these methods can remove the dark regions and give less blurry images, the pathological information is completely ignored. Therefore, the pathological information, which is the most important, maybe lost during the enhancement. In AMD, the main factor is drusen. The drusen are the small yellow or white areas in the retinal images. The doctors usually focus on drusen to detect and monitor the level of AMD. To solve that problem, we propose the deep-learning based method for AMD retinal image enhancement. We first make a drusen segmentation model and then produce drusen segmentation masks as additional information for our proposed enhancement model. By this setting, our method can keep the medical characteristic of retinal images during the enhancement process. Because of the importance of retinal images in eye examination, there are many works on retinal image analysis and improvement. The previous enhancement methods usually applied histogram equalization and matching. It can improve the contrast of retinal images but the outputs are not realistic. On the other hand, deep-learning based methods provide many advantages in the image-to-image translation task. Therefore, researchers also apply generative adversarial networks (GANs) for the enhancement task. The normalized convolution is used in while Zhao et al. exploited the CycleGAN model. Although GANs can produce more realistic images, the preservation of pathological information is not guaranteed.

CHAPTER 2
LITERATURE SURVEY

Computer vision and image processing techniques play an important role in all fields of medical science and are especially relevant to modern Ophthalmology. Medical imaging has revolutionized the field of medicine by providing cost-effective healthcare and efficient diagnosis in all major disease areas. Medical imaging allows scientists and physicians to understand potential life-saving information using less invasive techniques. Applications that can interpret an image are being developed, which in turn can aid a physician in detecting possible subtle abnormalities. The computer indicates places in the image that require extra attention from the physician because they could be abnormal. These technologies known as Computer Aided Diagnosis (CAD) systems show that CAD can be helpful to improve diagnostic accuracy of physicians and lighten the burden of increasing workload.

The influence and impact of digital images on modern society, science, technology and art are tremendous. Image processing has become such a critical component in contemporary science and technology that many tasks would not be attempted without it. Digital image processing is an interdisciplinary subject that draws from synergistic developments involving many disciplines and is used in medical imaging, microscopy, astronomy, computer vision, geology and many other fields. The rapid and continuing progress in computerized medical imaging, the associated developments in methods of analysis and computer-aided diagnosis, have propelled medical imaging into one of the most important sub-fields in scientific imaging. Medical image analysis is an area of research that attracts intensive interests of scientists and physicians and covers image processing, pattern recognition and computer visualization. Medical image processing involves the study of digital images with the objective of providing computational tools which will assist the quantification and visualization of interesting pathology and anatomical structures. The progress achieved in this field in recent years has significantly improved the type of medical care that is available to the patients.

The application of digital imaging to ophthalmology has now provided the possibility of processing retinal images to assist clinical diagnosis and treatment. Automated diagnosis of retinal fundus images using digital image analysis offers huge potential benefits. Due to advances in computer technology, medical diagnosis can be benefited from computers which will assist doctors to analyze medical data and images with improved accuracy. Designing and developing computer-aided diagnostic tools or systems for medical images is a fast growing area in recent years. Development of an automated system for analyzing the images of the retina will facilitate computer

aided diagnosis of eye diseases.

The interest towards automatic detection of glaucoma and diabetic retinopathy has been increasing along with the rapid development of digital imaging and computing power. However, the most important single event that attracted the wider attention of medical research community has been the decision to recognize digital imaging as an accepted modality to document eye fundus. This introductory chapter presents some background information on the anatomy of the eye, ocular diseases like glaucoma need for screening.

Glaucoma is a chronic disease which if not detected in early stages can lead to permanent blindness. The medical techniques used by ophthalmologists like HRT and OCT is costly and time consuming. Hence there is a need to develop automatic computer aided system which can detect glaucoma efficiently and in less time. Optic disk and optic cup are prime features which help in diagnosing glaucoma. Thus, proper segmentation of optic disk and optic cup plays an important role in detecting the disorder.

In this paper an adaptive threshold-based method which is independent of image quality and invariant to noise is used to segment optic disk, optic cup, Neuroretinal rim and cup to disk ratio is calculated to screen glaucoma. Another ocular parameter, rim to disk ratio is also considered which in combination with CDR gives more reliability in determining glaucoma and makes the system more robust. Further an SVM classifier has been used to categorize the images as glaucomatic or non glaucomatic. The experimental results obtained are compared with those of ophthalmologist and are found to have high accuracy of 90%. Also in addition, the proposed method is faster having low computational cost.

R.Manjula Sri, Ch.Madhubabu, and KMM Rao:

Lab VIEW based assessment of CDR for the detection of Glaucoma In this paper, Image processing and analysis has great significance in the field of medicine, especially in non-invasive treatment and clinical study. Glaucoma is a group of diseases, which damage eye's optic nerve that leads to blindness.

Glaucoma usually has no early symptoms, and by the time people experience problems with their vision, they usually have lost a significant amount of their sight. The structural change observed in the retinal fundus images of Glaucoma affected individuals

is enlargement of cup. Normally fundus images are manually graded by specially trained clinicians in a time-consuming and resource-intensive process. The ratio of the diameters or areas of the cup and disc (CDR) is widely used indicator for the assessment of glaucoma. The authors proposed a novel algorithm to detect glaucoma from color fundus images. The algorithm locates cup and disc on fundus images using circular fitting techniques and measures the diameter and area of cup and disc of the retina. To simulate the algorithm vision assistant in LabVIEW software is used.

R.Manjula Sri, V.AnushaSree, T.L.N.Rohitha, and K.M.M. Rao:

Measurement of Retinal Blood Vessel Diameter for the Detection of Eye Diseases using LabVIEW In this paper, Automated analysis of retinal images can assist in the diagnosis and management of blinding retinal diseases, such as diabetic retinopathy, age-related macular degeneration (AMD), and glaucoma. For evaluating and imaging patients with retinal diseases, clinical photographers usually capture color images of the retina using a specialized fundus camera. Subsequently, a fluorescein dye is injected into a vein in the subject's arm, and as the dye propagates through the retinal blood vessels, and series of (over a 5–10 min period) pictures of the retina are taken. Retinal fluorescein images at two different stages of Angiogram and a color fundus image of the same patient are used as input images for lesion diagnosis. From the two fluorescein images, vessel extraction is done, they are aligned and fused to identify the region of abnormality and lesion growth. From the color image, the 3D plot is simulated to identify the area of abnormality of the eye and is used as reference to the fused fluorescein image. In this paper we present a novel technique for diagnosis of lesions through Fluorescein Angiographic Images using Virtual Instrumentation (VI).

Medha V. Wyawahare and Dr. Pradeep M. Patil: Extraction of Optic Cup in Retinal

Fundus Images:

A Comparative approach In this paper, Glaucoma which is a leading cause of blindness in the world is not a single disease but a group of disorders with diverse clinical manifestations. If not controlled at an early stage, it causes irreversible damage to vision. Careful evaluation of Optic nerve head structure and its documentation is extremely important for diagnosis of the disease and to monitor its progression. The optic nerve head comprises of the optic disc and the optic cup. 'Optic cup' is one of the imperative and crucial factors in disease diagnosis and monitoring. The aim of the research is to compare various existing optic cup segmentation methods and report their comparative performance. A modification in calculating threshold has been suggested which improves the accuracy.

Arturo Aquino, Manuel Emilio Gegúndez-Arias, and Diego Marin: Detecting the Optic Disc Boundary in Digital Fundus Images Using Morphological, Edge Detection, and Feature Extraction Techniques In this paper, presents a new template-based methodology for segmenting the OD from digital retinal images. This methodology uses morphological and edge detection techniques followed by the Circular Hough Transform to obtain a circular OD boundary approximation. It

requires a pixel located within the OD as initial information. For this purpose, a location methodology based on a voting-type algorithm is also proposed. The algorithms were evaluated on the 1200 images of the publicly available MESSIDOR database.

The location procedure succeeded in 99% of cases, taking an average computational time of 1.67 s. with a standard deviation of 0.15 s. On the other hand, the segmentation algorithm rendered an average common area overlapping between automated segmentations and true OD regions of 86%. The average computational time was 5.69 s with a standard deviation of 0.55 s.

Prakash.H.Patil1, V.Kamkhedka:

Analysis of Human Retinal Images for Automated Glaucoma Screening In this paper, Current tests which are used to detect Glaucoma using intraocular pressure (IOP) are not sensitive enough for population based glaucoma screening. The assessment of Optic nerve head damage in retinal fundus images is more promising and superior. The manual examination of optic disc (OD) is a standard procedure used for detecting glaucoma. In this paper we proposes a system of automatic optic cup and optic disk segmentation using super pixel classification for glaucoma screening. The SLIC (Simple Linear Iterative Clustering) algorithm is incorporated to segment the fundus retinal image into compact and nearly uniform super pixels. It divides an image into a grid of regular pixels, as super pixels have the important property of preserving local boundaries. For optic disk & optic cup segmentation K-means clustering pixel technique, Gabor wavelet transform & thresholding is used. Then segmented optic disc and optic cup are used to compute the cup to disc ratio for glaucoma screening. The Cup to Disc Ratio (CDR) of the color retinal fundus camera image is the primary identifier to confirm Glaucoma for a given patient.

Jayanthi Sivaswamy, S.R.Krishnadas, Arunava Chakravarty, Gopal Datt Joshi, and Ujjwal:

A Comprehensive Retinal Image Dataset for the Assessment of Glaucoma from the Optic Nerve Head Analysis In this paper, Optic nerve head (ONH) segmentation problem is of interest for automated glaucoma assessment. Although various segmentation methods have been proposed in the recent past, it is difficult to evaluate and compare the p individual methods due to a lack of a benchmark dataset. The assessment involves segmentation of optic disk and cup region within the ONH. In this paper, we present a comprehensive dataset of retinal images of both normal and glaucomatous eyes with manual segmentations from multiple human experts. The dataset also provides expert opinion on an image representing a normal or glaucomatous eye and on the presence of notching in an image. Several state of the art methods are assessed against this dataset

using cup to disc diameter ratio (CDR), area and boundary-based evaluation measures. These are presented to aid benchmarking of new methods. A supervised, notch detection method based on the segmentation results is also proposed and its assessment results are included for benchmarking.

Eleesa Jacob, R.Venkatesh:

A Method of Segmentation for Glaucoma Screening Using Superpixel Classification In this paper, an optic disc and optic cup segmentation is used to identify the glaucoma disease in time. In optic disc and optic cup segmentation, super pixel classification for glaucoma screening is proposed. In optic disc segmentation, histograms and centre surround statistics are used to classify each super pixel as disc or non-disc. A self-assessment reliability score is computed to evaluate the quality of the automated optic disc segmentation. In optic cup segmentation, the location information is also included into the feature space for better performance in addition to the histograms and centre surround statistics. The segmented optic cup and optic disc is then used to compute the cup to disc ratio for glaucoma screening. From the cup to disc ratio, analysis is performed to identify whether the given image is glaucomatous or not. The segmentation can be analyzed using the MATLAB.

Sheeba O, Jithin George, Rajin P. K., Nisha Thomas, and Sherin George:

Glaucoma Detection Using Artificial Neural Network In this paper, the nerve that transmits visual images to the brain. Here the detection of glaucoma is done by image processing. The screening of patients for the development of glaucoma potentially reduces the risk of blindness in these patients by 50%. Here neural network is trained to recognize the

parameters for the detection of different stages of the disease. The neuron model has been developed using feed forward back propagation network. Here the program is developed using Matlab. The images acquired using medical imaging techniques are analysed in Matlab. Matlab provide variety of options for image processing that enable us to extract the required features and information from the images. The software can be used to detect the early stages of glaucoma.

Li Xiong, Huiqi Li; Yan Zheng: Automatic detection of glaucoma in retinal images In this paper, which is based on principle components analysis (PCA) and Bayes classifier. Firstly, optic disc center is located using the combination of thresholding and distance transformation. Eigenvector spaces of normal set and glaucoma set are obtained respectively using PCA. A test image is projected onto these two spaces and the distance between projection and each template is calculated. Finally, decision is made according to Bayes classifier. The success rate of optic disk localization is 95.3% and 89.9% for normal set and glaucoma set respectively. The glaucoma detection algorithm was tested by over three hundred retinal images and the success rate is 78%.

Dey, N, Roy, A.B.; Das, A.; Chaudhuri, S.S:

Optical cup to disc ratio measurement for glaucoma diagnosis using Harris corner In this paper, Glaucoma is physiologically described as the deterioration of optic nerve cells, and is characterized

by alterations in the optic nerve head and visual field. The measurement of neuro-retinal optic cup-to-disc ratio (CDR) is an important index of Glaucoma, as the increased excavation of the optic cup occurs because of Glaucomatous neuropathy increasing the CDR. Currently, CDR evaluation is manually performed by ophthalmologists. Due to the interweavement of blood vessels with the surrounding tissues around the cup, automatic calculation of optic cup boundary thus being challenging. In this paper, an automatic method for CDR determination using Harris Corner is proposed.

CHAPTER 3

SOFTWARE INTRODUCTION

3.1. Introduction to MATLAB:

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most uses of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M – files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

3.2 The MATLAB system:

The MATLAB system consists of five main parts

- **Development Environment:**

  This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

- **The MATLAB Mathematical Function Library:**

  This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigen values, Bessel functions, and fast Fourier transforms.

- **The MATLAB Language:**

  This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create large and complex application programs.

- **Graphics:**

  MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

- **The MATLAB Application Program Interface (API):**

  This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

Various toolboxes are there in MATLAB for computing recognition techniques, but we are using IMAGE PROCESSING toolbox.

3.3 GRAPHICAL USER INTERFACE (GUI):

MATLAB's Graphical User Interface Development Environment (GUIDE) provides a rich set of tools for incorporating graphical user interfaces (GUIs) in M-functions. Using GUIDE, the processes of laying out a GUI (i.e., its buttons, pop-up menus, etc.)and programming the operation of the GUI are divided conveniently. Into two easily managed and relatively independent tasks. The resulting graphical M-function is composed of two identically named (ignoring extensions) files:

- A file with extension .fig, called a FIG-file that contains a complete graphical description of all the function's GUI objects or elements and their spatial arrangement. A FIG-file contains binary data that does not need to be parsed when he associated GUI-based M-function is executed.

- A file with extension .m, called a GUI M-file, which contains the code that controls the GUI operation. This file includes functions that are called when the GUI is launched and exited, and callback functions that are executed when a user interacts with GUI objects for example, when a button is pushed.

To launch GUIDE from the MATLAB command window, type
guide filename

Where filename is the name of an existing FIG-file on the current path. If filename is omitted, GUIDE opens a new (i.e., blank) window.

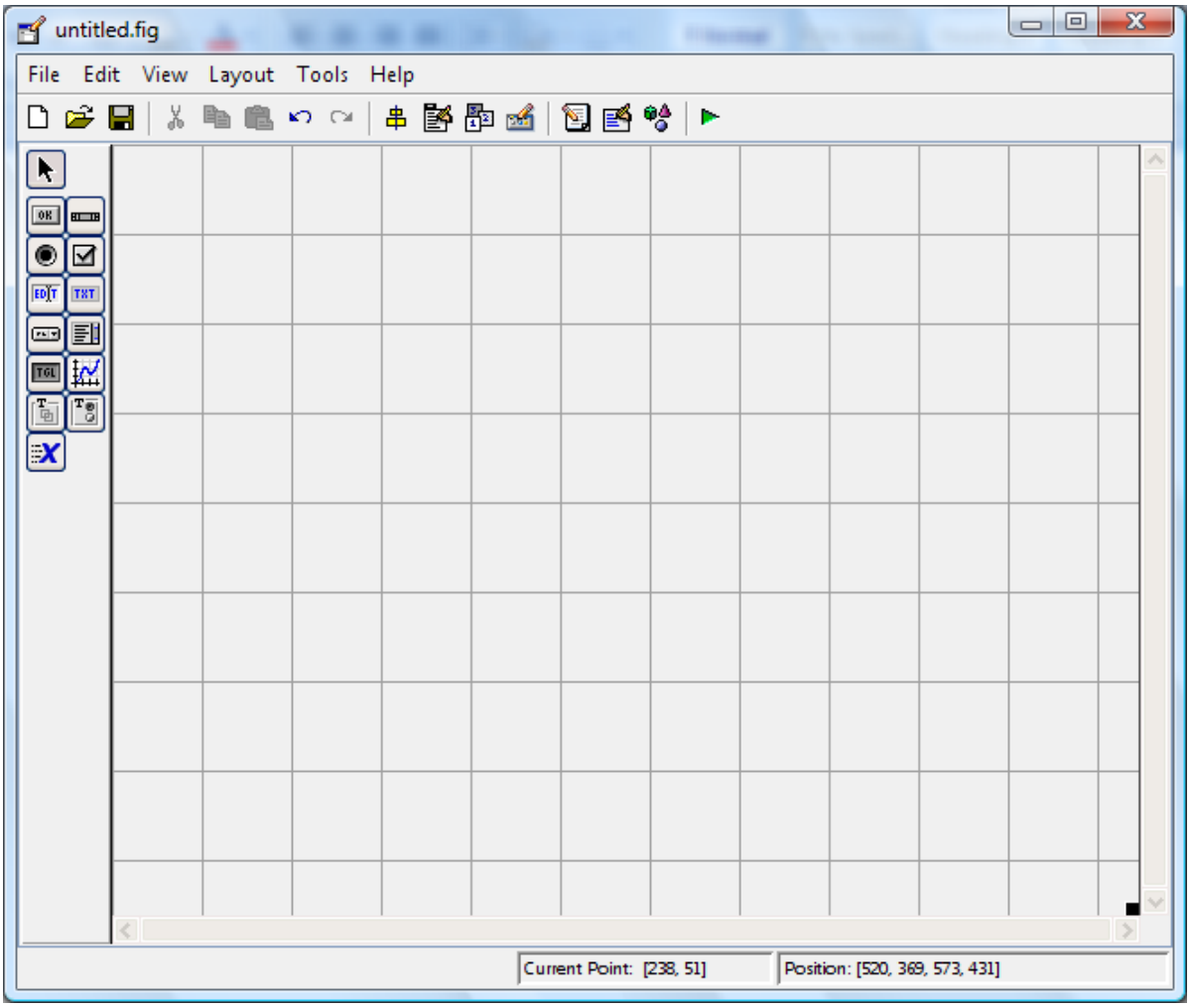


FIG 3.4: A graphical user interface (GUI)

A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed.

GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders just to name a few. GUIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots.

3.4 Getting Started:

If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the: (colon) operator, and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos.

At the heart of MATLAB is a new language you must learn before you can fully exploit its power. You can learn the basics of MATLAB quickly, and mastery comes shortly after. You will be rewarded with high productivity, high-creativity computing power that will change the way you work.

3.4.1 Introduction: - describes the components of the MATLAB system.

3.4.2 Development Environment: - introduces the MATLAB development environment, including information about tools and the MATLAB desktop.

3.4.3 Manipulating Matrices: - introduces how to use MATLAB to generate matrices and perform mathematical operations on matrices.

3.4.5 Graphics: - introduces MATLAB graphic capabilities, including information about plotting data, annotating graphs, and working with images.

3.4.5 Programming with MATLAB: - describes how to use the MATLAB language to create scripts and functions, and manipulate data structures, such as cell arrays and multidimensional arrays.

3.5 DEVELOPMENT ENVIRONMENT:

3.5.1Introduction

This chapter provides a brief introduction to starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics under Development Environment in the MATLAB documentation, which is available online as well as in print.

Starting and Quitting MATLAB

3.5.2 Starting MATLAB:

On a Microsoft Windows platform, to start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop.On a UNIX platform, to start MATLAB, type matlab at the operating system prompt. After starting MATLAB, the MATLAB desktop opens - see MATLAB Desktop.

You can change the directory in which MATLAB starts, define startup options including running a script upon startup, and reduce startup time in some situations.

3.5.3 Quitting MATLAB:

To end your MATLAB session, select Exit MATLAB from the File menu in the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish.m script.

3.5.4 MATLAB Desktop:

When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with MATLAB.The first time MATLAB starts, the desktop appears as shown in the following illustration, although your Launch Pad may contain different entries.

You can change the way your desktop looks by opening, closing, moving, and resizing the tools in it. You can also move tools outside of the desktop or return them back inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts.

You can specify certain characteristics for the desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for Command Window text. For more information, click the Help button in the Preferences dialog box.

3.5.5 Desktop Tools:

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools. The tools are:

- Current Directory Browser
- Workspace Browser
- Array Editor
- Editor/Debugger
- Command Window
- Command History
- Launch Pad
- Help Browser

Command Window:

Use the Command Window to enter variables and run functions and M-files.

Command History:

Lines you enter in the Command Window are logged in the Command History window. In the Command History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

Running External Programs:

You can run external programs from the MATLAB Command Window. The exclamation point character! is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other programs without quitting MATLAB. On Linux, for example,!emacs magik.m invokes an editor called emacs for a file named magik.m. When you quit the external program, the operating system returns control to MATLAB.

Launch Pad:

MATLAB's Launch Pad provides easy access to tools, demos, and documentation.

Help Browser:

Use the Help browser to search and view documentation for all your Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents.

To open the Help browser, click the help button in the toolbar, or type helpbrowser in the Command Window. The Help browser consists of two panes, the Help Navigator, which you use to find information, and the display pane, where you view the information.

Help Navigator:

Use to Help Navigator to find information. It includes:

Product filter :- Set the filter to show documentation only for the products you specify.

Contents tab :- View the titles and tables of contents of documentation for your products.

Index tab :- Find specific index entries (selected keywords) in the MathWorks documentation for your products.

Search tab: - Look for a specific phrase in the documentation. To get help for a specific function, set the Search type to Function Name.

Favorites tab :- View a list of documents you previously designated as favorites.

Display Pane

After finding documentation using the Help Navigator, view it in the display pane. While viewing the documentation, you can:

Browse to other pages :- Use the arrows at the tops and bottoms of the pages, or use the back

and forward buttons in the toolbar.

Bookmark pages :- Click the Add to Favorites button in the toolbar.

Print pages - Click the print button in the toolbar.

Find a term in the page :- Type a term in the Find in page field in the toolbar and click Go. Other features available in the display pane are: copying information, evaluating a selection, and viewing Web pages.

Current Directory Browser:

MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.

Search Path:

To determine how to execute functions you call, MATLAB uses a search path to find M-files and other MATLAB-related files, which are organized in directories on your file system. Any file you want to run in MATLAB must reside in the current directory or in a directory that is on the search path. By default, the files supplied with MATLAB and MathWorks toolboxes are included in the search path.

Workspace Browser:

The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace by using functions, running M-files, and loading saved workspaces.

To view the workspace and information about each variable, use the Workspace browser, or use the functions who and whos.

To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function.

The workspace is not maintained after you end the MATLAB session. To save the workspace to a file that can be read during a later MATLAB session, select Save Workspace As from the File menu, or use the save function. This saves the workspace to a binary file called a MAT-file, which has a .mat extension. There are options for saving to different formats. To read in a MAT-file, select Import Data from the File menu, or use the load function.

Array Editor:

Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array

Editor to view and edit a visual representation of one- or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace.

Editor/Debugger:

Use the Editor/Debugger to create and debug M-files, which are programs you write to run MATLAB functions. The Editor/Debugger provides a graphical user interface for basic text editing, as well as for M-file debugging. You can use any text editor to create M-files, such as Emacs, and can use preferences (accessible from the desktop File menu) to specify that editor as the default. If you use another editor, you can still use the MATLAB Editor/Debugger for debugging, or you can use debugging functions, such as dbstop, which sets a breakpoint.

If you just need to view the contents of an M-file, you can display it in the Command Window by using the type function

3.6 MANIPULATING MATRICES:

3.6.1 Entering Matrices:

The best way for you to get started with MATLAB is to learn how to handle matrices. Start MATLAB and follow along with each example.

You can enter matrices into MATLAB in several different ways:

- Enter an explicit list of elements.
- Load matrices from external data files.
- Generate matrices using built-in functions.
- Create matrices with your own functions in M-files.

Start by entering Dürer's matrix as a list of its elements. You have only to follow a few basic conventions:

- Separate the elements of a row with blanks or commas.
- Use a semicolon, ; , to indicate the end of each row.
- Surround the entire list of elements with square brackets, [ ].

To enter Dürer's matrix, simply type in the Command Window

A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 5 15 15 1]

MATLAB displays the matrix you just entered.

A =

16 3 2 13

5 10 11 8

9 6 7 12

5 15 15 1

This exactly matches the numbers in the engraving. Once you have entered the matrix, it is automatically remembered in the MATLAB workspace. You can refer to it simply as A

3.6.2 Expressions:

Like most other programming languages, MATLAB provides mathematical expressions, but unlike most programming languages, these expressions involve entire matrices. The building blocks of expressions are:

- Variables
- Numbers
- Operators
- Functions

Variables

MATLAB does not require any type declarations or dimension statements. When MATLAB encounters a new variable name, it automatically creates the variable and allocates the appropriate amount of storage. If the variable already exists, MATLAB changes its contents and, if necessary, allocates new storage. For example,

num_students = 25

Creates a 1-by-1 matrix named num_students and stores the value 25 in its single element.

Variable names consist of a letter, followed by any number of letters, digits, or underscores. MATLAB: uses only the first 31 characters of a variable name. MATLAB is case sensitive; it distinguishes between uppercase and lowercase letters. A and a are not the same variable. To view the matrix assigned to any variable, simply enter the variable name.

Numbers:

MATLAB provides a large number of standard elementary mathematical functions, including abs, sqrt, exp, and sin. Taking the square root or logarithm of a negative number is not an error; the appropriate complex result is produced automatically. MATLAB also provides many more advanced mathematical functions, including Bessel and gamma functions. Most of these functions accept complex arguments. For a list of the elementary mathematical functions, type help elfun,

For a list of more advanced mathematical and matrix functions, type help specfun help elmat

Some of the functions, like sqrt and sin, are built-in. They are part of the MATLAB core so they are very efficient, but the computational details are not readily accessible. Other functions, like gamma and sinh, are implemented in M-files. You can see the code and even modify it if you want. Several special functions provide values of useful constants.

Pi	3.15159265...
I	Imaginary unit, $\sqrt{-1}$
I	Same as i
Eps	Floating-point relative precision, 2^{-52}
Realmin	Smallest floating-point number, 2^{-1022}
Realmax	Largest floating-point number, $(2-\varepsilon)2^{1023}$
Inf	Infinity
NaN	Not-a-number

3.7 GUI:

A graphical user interface (GUI) is a user interface built with graphical objects, such as buttons, text fields, sliders, and menus. In general, these objects already have meanings to most computer users. For example, when you move a slider, a value changes; when you press an OK button, your settings are applied and the dialog box is dismissed. Of course, to leverage this built-in familiarity, you must be consistent in how you use the various GUI-building components.

Applications that provide GUIs are generally easier to learn and use since the person using the application does not need to know what commands are available or how they work. The action that results from a particular user action can be made clear by the design of the interface.

The sections that follow describe how to create GUIs with MATLAB. This includes laying out the components, programming them to do specific things in response to user actions, and saving and launching the GUI; in other words, the mechanics of creating GUIs. This documentation does not attempt to cover the "art" of good user interface design, which is an entire field unto itself. Topics covered in this section include.

3.7.1 Creating GUIs with GUIDE:

MATLAB implements GUIs as figure windows containing various styles of uicontrol objects. You must program each object to perform the intended action when activated by the user of the GUI. In addition, you must be able to save and launch your GUI. All of these tasks are simplified by GUIDE, MATLAB's graphical user interface development environment.

3.7.2 GUI Development Environment:

The process of implementing a GUI involves two basic task.

- Laying out the GUI components
- Programming the GUI components

GUIDE primarily is a set of layout tools. However, GUIDE also generates an M-file that contains code to handle the initialization and launching of the GUI. This M-file provides a framework for the implementation of the callbacks – the functions that execute when users activate components in the GUI.

The Implementation of a GUI

While it is possible to write an M-file that contains all the commands to lay out a GUI, it is easier to use GUIDE to lay out the components interactively and to generate two files that save and launch the GUI:

A FIG-file – contains a complete description of the GUI figure and all of its children (uicontrols and axes), as well as the values of all object properties.

An M-file - contains the functions that launch and control the GUI and the callbacks, which are defined as subfunctions. This M-file is referred to as the application M-file in this documentation.

Note that the application M-file does not contain the code that lays out the uicontrols; this information is saved in the FIG-file.

The following diagram illustrates the parts of a GUI implementation.

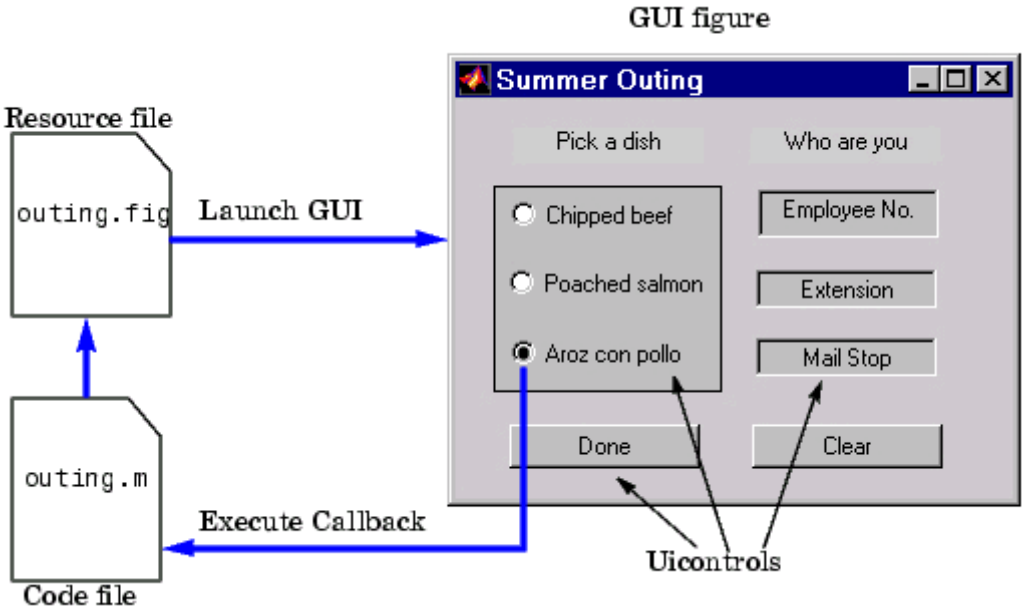


FIG 3.7.2 graphical user blocks

3.7.3 Features of the GUIDE-Generated Application M-Fil:

GUIDE simplifies the creation of GUI applications by automatically generating an M-file framework directly from your layout. You can then use this framework to code your application M-file. This approach provides a number of advantages: The M-file contains code to implement a number of useful features (see Configuring Application Options for information on these features). The M-file adopts an effective approach to managing object handles and executing callback routines (see Creating and Storing the Object Handle Structure for more information). The M-files provides a way to manage global data (see Managing GUI Data for more information).

The automatically inserted subfunction prototypes for callbacks ensure compatibility with future releases. For more information, see Generating Callback Function Prototypes for information on syntax and arguments.

You can elect to have GUIDE generate only the FIG-file and write the application M-file yourself. Keep in mind that there are no uicontrol creation commands in the application M-file; the layout information is contained in the FIG-file generated by the Layout Editor.

3.7.5 Beginning the Implementation Process:

To begin implementing your GUI, proceed to the following sections:

Getting Started with GUIDE - the basics of using GUIDE.

Selecting GUIDE Application Options - set both FIG-file and M-file options.

Using the Layout Editor - begin laying out the GUI.

Understanding the Application M-File - discussion of programming techniques used in the application M-file.

Application Examples - a collection of examples that illustrate techniques which are useful for implementing GUIs.

Command-Line Accessibility

When MATLAB creates a graph, the figure and axes are included in the list of children of their respective parents and their handles are available through commands such as findobj, set, and get. If you issue another plotting command, the output is directed to the current figure and axes.

GUIs are also created in figure windows. Generally, you do not want GUI figures to be available as targets for graphics output, since issuing a plotting command could direct the output to the GUI In contrast, if you create a GUI that contains an axes and you want commands entered in the command window to display in this axes, you should enable command-line access.

3.7.5 User Interface Control:

The Layout Editor component palette contains the user interface controls that you can use in your GUI. These components are MATLAB uicontrol objects and are programmable via their Callback properties. This section provides information on these components.

- Push Buttons
- Sliders
- Toggle Buttons
- Frames
- Radio Buttons
- Listboxes
- Checkboxes
- Popup Menus
- Edit Text
- Axes
- Static Text
- Figures

Push Buttons

Push buttons generate an action when pressed (e.g., an OK button may close a dialog box and apply settings). When you click down on a push button, it appears depressed; when you release the mouse, the button's appearance returns to its nondepressed state; and its callback executes on the button up event.

Properties to Set

String - set this property to the character string you want displayed on the push button.

Tag - GUIDE uses the Tag property to name the callback subfunction in the application M-file. Set Tag to a descriptive name (e.g., close_button) before activating the GUI.

Programming the Callback

When the user clicks on the push button, its callback executes. Push buttons do not return a value or maintain a state.

Toggle Buttons

Toggle buttons generate an action and indicate a binary state (e.g., on or off). When you click on a toggle button, it appears depressed and remains depressed when you release the mouse button, at which point the callback executes. A subsequent mouse click returns the toggle button to the nondepressed state and again executes its callback.

Programming the Callback

The callback routine needs to query the toggle button to determine what state it is in. MATLAB sets the Value property equal to the Max property when the toggle button is depressed (Max is 1 by default) and equal to the Min property when the toggle button is not depressed (Min is 0 by default).

From the GUIDE Application M-File

The following code illustrates how to program the callback in the GUIDE application M-file.

```
function varargout = togglebutton1_Callback(h,eventdata,handles,varargin)
button_state = get(h,'Value');
if button_state == get(h,'Max')
   % toggle button is pressed
elseif button_state == get(h,'Min')
   % toggle button is not pressed
end
```

Adding an Image to a Push Button or Toggle Button

Assign the CData property an m-by-n-by-3 array of RGB values that define a truecolor image. For example, the array a defines 16-by-128 truecolor image using random values between 0 and 1 (generated by rand).

```
a(:,:,1) = rand(16,128);
a(:,:,2) = rand(16,128);
a(:,:,3) = rand(16,128);
set(h,'CData',a)
```

Radio Buttons

Radio buttons are similar to checkboxes, but are intended to be mutually exclusive within a group of related radio buttons (i.e., only one button is in a selected state at any given time)

. To activate a radio button, click the mouse button on the object. The display indicates the state of the button.

Implementing Mutually Exclusive Behavior

Radio buttons have two states - selected and not selected. You can query and set the state of a radio button through its Value property:

Value = Max, button is selected.

Value = Min, button is not selected.

To make radio buttons mutually exclusive within a group, the callback for each radio button must set the Value property to 0 on all other radio buttons in the group. MATLAB sets the Value property to 1 on the radio button clicked by the user.

The following subfunction, when added to the application M-file, can be called by each radio button callback. The argument is an array containing the handles of all other radio buttons in the group that must be deselected.

```
function mutual_exclude(off)
set(off,'Value',0)
```

Obtaining the Radio Button Handles.

The handles of the radio buttons are available from the handles structure, which contains the handles of all components in the GUI. This structure is an input argument to all radio button callbacks.

The following code shows the call to mutual_exclude being made from the first radio button's callback in a group of four radio buttons.

```
function varargout = radiobutton1_Callback(h,eventdata,handles,varargin)
off = [handles.radiobutton2,handles.radiobutton3,handles.radiobutton5];
mutual_exclude(off)
% Continue with callback
   .
   .
   .
```

After setting the radio buttons to the appropriate state, the callback can continue with its implementation-specific tasks.

Checkboxes

Check boxes generate an action when clicked and indicate their state as checked or not checked. Check boxes are useful when providing the user with a number of independent choices that set a mode (e.g., display a toolbar or generate callback function prototypes).

The Value property indicates the state of the check box by taking on the value of the Max or Min

property (1 and 0 respectively by default):

Value = Max, box is checked.

Value = Min, box is not checked.

You can determine the current state of a check box from within its callback by querying the state of its Value property, as illustrated in the following example:

```
function checkbox1_Callback(h,eventdata,handles,varargin)
if (get(h,'Value') == get(h,'Max'))
   % then checkbox is checked-take approriate action
else
   % checkbox is not checked-take approriate action
end
```

Edit Text

Edit text controls are fields that enable users to enter or modify text strings. Use edit text when you want text as input. The String property contains the text entered by the user.

To obtain the string typed by the user, get the String property in the callback.

```
function edittext1_Callback(h,eventdata, handles,varargin)
user_string = get(h,'string');
% proceed with callback...
```

Obtaining Numeric Data from an Edit Test Component

MATLAB returns the value of the edit text String property as a character string. If you want users to enter numeric values, you must convert the characters to numbers. You can do this using the str2double command, which converts strings to doubles. If the user enters non-numeric characters, str2double returns NaN.

You can use the following code in the edit text callback. It gets the value of the String property and converts it to a double. It then checks if the converted value is NaN, indicating the user entered a non-numeric character (isnan) and displays an error dialog (errordlg).

```
function edittext1_Callback(h,eventdata,handles,varargin)
user_entry = str2double(get(h,'string'));
if isnan(user_entry)
```

```
    errordlg('You must enter a numeric value','Bad Input','modal')
end
% proceed with callback...
```

Triggering Callback Execution

On UNIX systems, clicking on the menubar of the figure window causes the edit text callback to execute. However, on Microsoft Windows systems, if an editable text box has focus, clicking on the menubar does not cause the editable text callback routine to execute. This behavior is consistent with the respective platform conventions. Clicking on other components in the GUI execute the callback.

Static Text

Static text controls displays lines of text. Static text is typically used to label other controls, provide directions to the user, or indicate values associated with a slider. Users cannot change static text interactively and there is no way to invoke the callback routine associated with it

Frames

Frames are boxes that enclose regions of a figure window. Frames can make a user interface easier to understand by visually grouping related controls. Frames have no callback routines associated with them and only uicontrols can appear within frames (axes cannot).

Placing Components on Top of Frames

Frames are opaque. If you add a frame after adding components that you want to be positioned within the frame, you need to bring forward those components. Use the Bring to Front and Send to Back operations in the Layout menu for this purpose.

List Boxes

List boxes display a list of items and enable users to select one or more items.

The String property contains the list of strings displayed in the list box. The first item in the list has an index of 1.

The Value property contains the index into the list of strings that correspond to the selected item. If the user selects multiple items, then Value is a vector of indices. By default, the first item in the list is highlighted when the list box is first displayed. If you do not want any item highlighted, then set the Value property to empty.

The ListboxTop property defines which string in the list displays as the top most item when the list box is not large enough to display all list entries. ListboxTop is an index into the array of strings defined by the String property and must have a value between 1 and the number of strings. Noninteger values are fixed to the next lowest integer.

Single or Multiple Selection

The values of the Min and Max properties determine whether users can make single or multiple selections:

If Max - Min > 1, then list boxes allow multiple item selection.

If Max - Min <= 1, then list boxes do not allow multiple item selection.

Popup Menus

Popup menus open to display a list of choices when users press the arrow. The String property contains the list of string displayed in the popup menu. The Value property contains the index into the list of strings that correspond to the selected item. When not open, a popup menu displays the current choice, which is determined by the index contained in the Value property. The first item in the list has an index of 1.

Popup menus are useful when you want to provide users with a number of mutually exclusive choices, but do not want to take up the amount of space that a series of radio buttons requires.

Programming the Popup Menu

You can program the popup menu callback to work by checking only the index of the item selected (contained in the Value property) or you can obtain the actual string contained in the selected item. This callback checks the index of the selected item and uses a switch statement to take action based on the value. If the contents of the popup menu is fixed, then you can use this approach.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)
val = get(h,'Value');
switch val
case 1
% The user selected the first item
case 2
% The user selected the second item
% etc.
```

This callback obtains the actual string selected in the popup menu. It uses the value to index into the list of strings. This approach may be useful if your program dynamically loads the contents of the popup menu based on user action and you need to obtain the selected string. Note that it is necessary to convert the value returned by the String property from a cell array to a string.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)
val = get(h,'Value');
string_list = get(h,'String');
selected_string = string_list{val}; % convert from cell array to string
% etc.
```

Enabling or Disabling Controls

You can control whether a control responds to mouse button clicks by setting the Enable property. Controls have three states:

on - The control is operational

off - The control is disabled and its label (set by the string property) is grayed out.

inactive - The control is disabled, but its label is not grayed out.

When a control is disabled, clicking on it with the left mouse button does not execute its callback routine. However, the left-click causes two other callback routines to execute: First the figure WindowButtonDownFcn callback executes.

Then the control's ButtonDownFcn callback executes. A right mouse button click on a disabled control posts a context menu, if one is defined for that control. See the Enable property description for more details.

Axes

Axes enable your GUI to display graphics (e.g., graphs and images). Like all graphics objects, axes have properties that you can set to control many aspects of its behavior and appearance. See Axes Properties for general information on axes objects.

Axes Callbacks

Axes are not uicontrol objects, but can be programmed to execute a callback when users click a mouse button in the axes. Use the axes ButtonDownFcn property to define the callback.

3.7.6 Plotting to Axes in GUIs:

GUIs that contain axes should ensure the Command-line accessibility option in the Application Options dialog is set to Callback (the default). This enables you to issue plotting commands from callbacks without explicitly specifying the target axes.

GUIs with Multiple Axes

If a GUI has multiple axes, you should explicitly specify which axes you want to target when you issue plotting commands. You can do this using the axes command and the handles structure. For example,

```
axes(handles.axes1)
```

makes the axes whose Tag property is axes1 the current axes, and therefore the target for plotting commands. You can switch the current axes whenever you want to target a different axes. See GUI with Multiple Axes for and example that uses two axes.

CHAPTER 4
INTRODUCTION TO IMAGE PROCESSING

INTRODUCTION

4.1 IMAGE:

An image is a two-dimensional picture, which has a similar appearance to some subject usually a physical object or a person.

Image is a two-dimensional, such as a photograph, screen display, and as well as a three-dimensional, such as a statue. They may be captured by optical devices—such as cameras, mirrors, lenses, telescopes, microscopes, etc. and natural objects and phenomena, such as the human eye or water surfaces.

The word image is also used in the broader sense of any two-dimensional figure such as a map, a graph, a pie chart, or an abstract painting. In this wider sense, images can also be rendered manually, such as by drawing, painting, carving, rendered automatically by printing or computer graphics technology, or developed by a combination of methods, especially in a pseudo-photograph.

Fig 4.1 General image

An image is a rectangular grid of pixels. It has a definite height and a definite width counted in pixels. Each pixel is square and has a fixed size on a given display. However different computer monitors may use different sized pixels. The pixels that constitute an image are ordered as a grid (columns and rows); each pixel consists of numbers representing magnitudes of brightness and color.

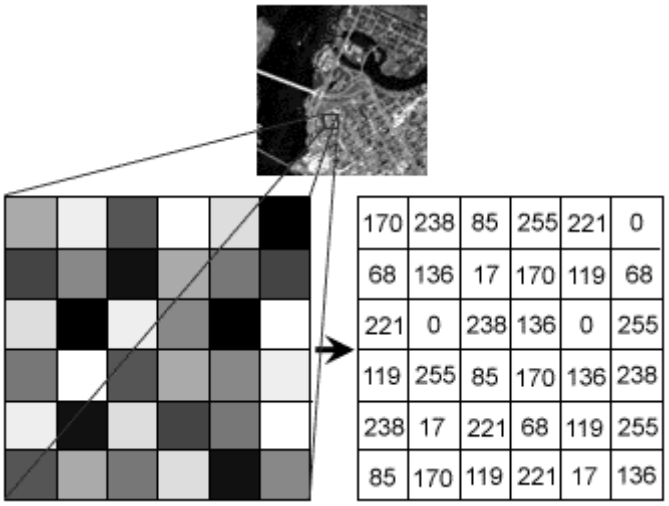


Fig 4.2 Image pixel

Each pixel has a color. The color is a 32-bit integer. The first eight bits determine the redness of the pixel, the next eight bits the greenness, the next eight bits the blueness, and the remaining eight bits the transparency of the pixel.

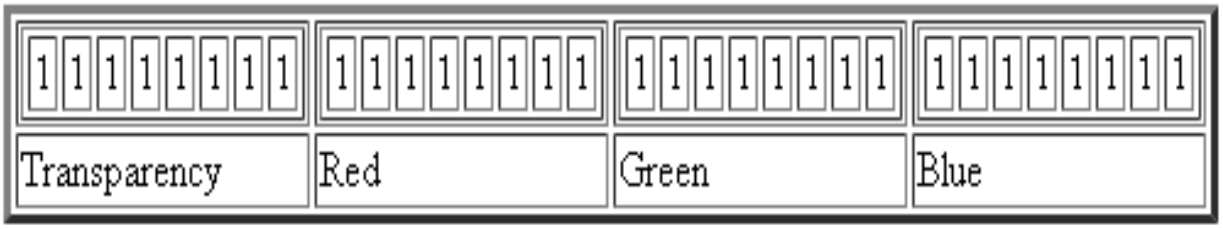


Fig4.3 Transparency image

4.2 IMAGE FILE SIZES:

Image file size is expressed as the number of bytes that increases with the number of pixels composing an image, and the color depth of the pixels. The greater the number of rows and columns, the greater the image resolution, and the larger the file. Also, each pixel of an image increases in size when its color depth increases, an 8-bit pixel (1 byte) stores 256 colors, a 25-bit pixel (3 bytes) stores 16 million colors, the latter known as true color.

Image compression uses algorithms to decrease the size of a file. High resolution cameras produce large image files, ranging from hundreds of kilobytes to megabytes, per the camera's resolution and the image-storage format capacity. High resolution digital cameras record 12 megapixel (1MP = 1,000,000 pixels / 1 million) images, or more, in true color. For example, an image recorded by a 12 MP camera; since each pixel uses 3 bytes to record true color, the uncompressed image would occupy 36,000,000 bytes of memory, a great amount of digital storage for one image, given that cameras must record and store many images to be practical. Faced with large file sizes, both within the camera and a storage disc, image file formats were developed to store such large images.

4.3 IMAGE FILE FORMATS:

Image file formats are standardized means of organizing and storing images. This entry is about digital image formats used to store photographic and other images. Image files are composed of either pixel or vector (geometric) data that are rasterized to pixels when displayed (with few exceptions) in a vector graphic display. Including proprietary types, there are hundreds of image file types. The PNG, JPEG, and GIF formats are most often used to display images on the Internet.

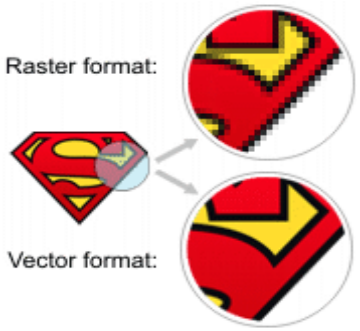


Fig4.4 Resolution image

In addition to straight image formats, Metafile formats are portable formats which can include both raster and vector information. The metafile format is an intermediate format. Most Windows applications open metafiles and then save them in their own native format.

4.3.1 RASTER FORMATS:

These formats store images as bitmaps (also known as pixmaps)

- **JPEG/JFIF:**

JPEG (Joint Photographic Experts Group) is a compression method. JPEG compressed images are usually stored in the JFIF (JPEG File Interchange Format) file format. JPEG compression is lossy compression. Nearly every digital camera can save images in the JPEG/JFIF format, which supports 8 bits per color (red, green, blue) for a 25-bit total, producing relatively small files. Photographic images may be better stored in a lossless non-JPEG format if they will be re-edited, or if small "artifacts" are unacceptable. The JPEG/JFIF format also is used as the image compression algorithm in many Adobe PDF files.

- **EXIF:**

The EXIF (Exchangeable image file format) format is a file standard similar to the JFIF format with TIFF extensions. It is incorporated in the JPEG writing software used in most cameras. Its purpose is to record and to standardize the exchange of images with image metadata between digital cameras and editing and viewing software. The metadata are recorded for individual images and include such things as camera settings, time and date, shutter speed, exposure, image size, compression, name of camera, color information, etc.

When images are viewed or edited by image editing software, all of this image information can be displayed.

- **TIFF:**

The TIFF (Tagged Image File Format) format is a flexible format that normally saves 8 bits or 16 bits per color (red, green, blue) for 25-bit and 58-bit totals, respectively, usually using either the TIFF or TIF filename extension. TIFFs are lossy and lossless. Some offer relatively good lossless compression for bi-level (black & white) images. Some digital cameras can save in TIFF format, using the LZW compression algorithm for lossless storage. TIFF image format is not widely supported by web browsers. TIFF remains widely accepted as a photograph file standard in the printing business. TIFF can handle device-specific color spaces, such as the CMYK defined by a particular set of printing press inks.

- **PNG:**

The PNG (Portable Network Graphics) file format was created as the free, open-source successor to the GIF. The PNG file format supports true color (16 million colors) while the GIF supports only 256 colors. The PNG file excels when the image has large, uniformly colored areas. The lossless PNG format is best suited for editing pictures, and the lossy formats, like JPG, are best for the final distribution of photographic images, because JPG files are smaller than PNG files. PNG, an extensible file format for the lossless, portable, well-compressed storage of raster images. PNG provides a patent-free replacement for GIF and can also replace many common uses of TIFF. Indexed-color, grayscale, and true color images are supported, plus an optional alpha channel. PNG is designed to work well in online viewing applications, such as the World Wide Web. PNG is robust, providing both full file integrity checking and simple detection of common transmission errors.

- **GIF:**

GIF (Graphics Interchange Format) is limited to an 8-bit palette, or 256 colors. This makes the GIF format suitable for storing graphics with relatively few colors such as simple diagrams, shapes, logos and cartoon style images. The GIF format supports animation and is still widely used to provide image animation effects. It also uses a lossless compression that is more effective when large areas have a single color, and ineffective for detailed images or dithered images.

- **BMP:**

The BMP file format (Windows bitmap) handles graphics files within the Microsoft Windows OS. Typically, BMP files are uncompressed, hence they are large. The advantage is their simplicity and wide acceptance in Windows programs.

4.3.2 VECTOR FORMATS:

As opposed to the raster image formats above (where the data describes the characteristics of each individual pixel), vector image formats contain a geometric description which can be rendered smoothly at any desired display size.

At some point, all vector graphics must be rasterized in order to be displayed on digital monitors. However, vector images can be displayed with analog CRT technology such as that used in some electronic test equipment, medical monitors, radar displays, laser shows and early video games. Plotters are printers that use vector data rather than pixel data to draw graphics.

- **CGM:**

CGM (Computer Graphics Metafile) is a file format for 2D vector graphics, raster graphics, and text. All graphical elements can be specified in a textual source file that can be compiled into a binary file or one of two text representations. CGM provides a means of graphics data interchange for computer representation of 2D graphical information independent from any particular application, system, platform, or device.

- **SVG:**

SVG (Scalable Vector Graphics) is an open standard created and developed by the World Wide Web Consortium to address the need for a versatile, scriptable and all purpose vector format for the web and otherwise. The SVG format does not have a compression scheme of its own, but due to the textual nature of XML, an SVG graphic can be compressed using a program such as gzip.

4.4 IMAGE PROCESSING:

Digital image processing, the manipulation of images by computer, is relatively recent development in terms of man's ancient fascination with visual stimuli. In its short history, it has been applied to practically every type of images with varying degree of success. The inherent subjective appeal of pictorial displays attracts perhaps a disproportionate amount of attention from the scientists and also from the layman. Digital image processing like other glamour fields, suffers from myths, mis-connect ions, mis-understandings and mis-information. It is vast umbrella under which fall diverse aspect of optics, electronics, mathematics, photography graphics and computer technology. It is truly multidisciplinary endeavor ploughed with imprecise jargon.

Several factor combine to indicate a lively future for digital image processing. A major factor is the declining cost of computer equipment. Several new technological trends promise to further promote digital image processing. These include parallel processing mode practical by low cost microprocessors, and the use of charge coupled devices (CCDs) for digitizing, storage during processing and display and large low cost of image storage arrays.

4.5 FUNDAMENTAL STEPS IN DIGITAL IMAGE PROCESSING:

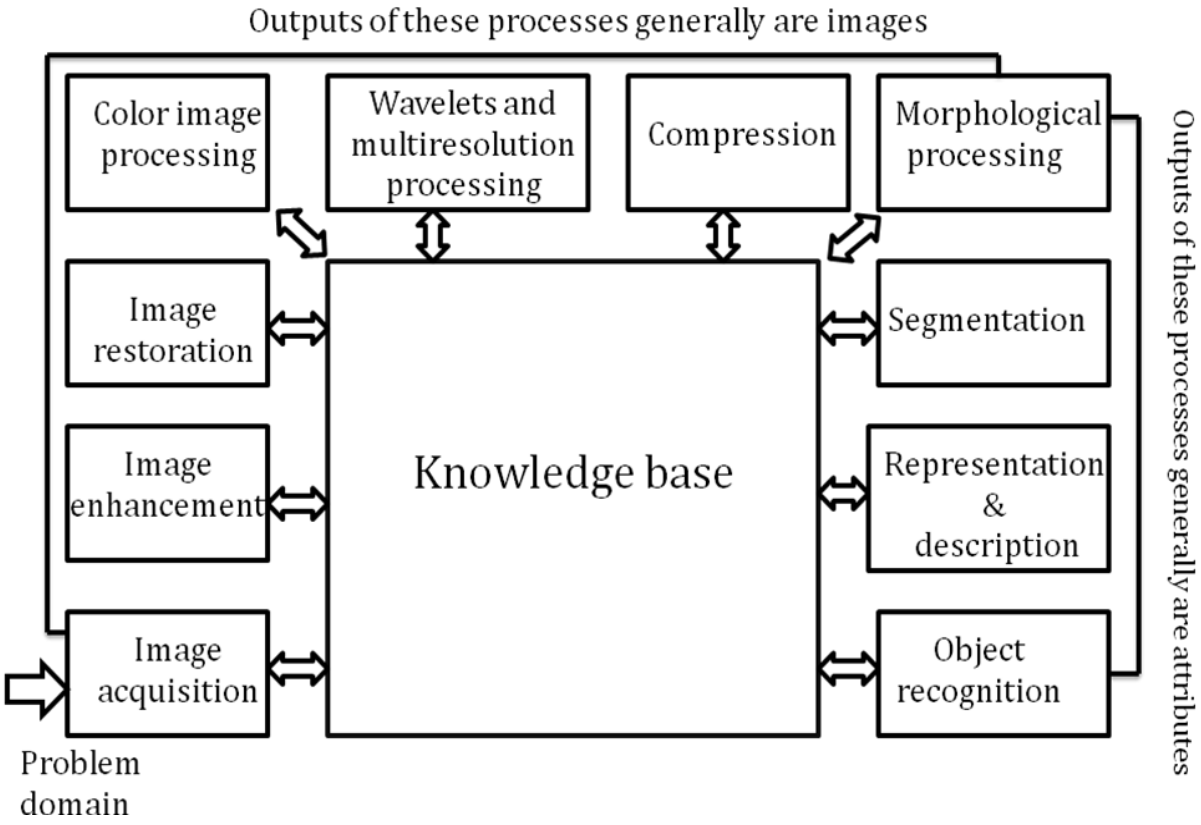


Fig 4.5 Image fundamental

4.5.1 Image Acquisition:

Image Acquisition is to acquire a digital image. To do so requires an image sensor and the capability to digitize the signal produced by the sensor. The sensor could be monochrome or color TV camera that produces an entire image of the problem domain every 1/30 sec. the image sensor could also be line scan camera that produces a single image line at a time. In this case, the objects motion past the line.

Fig 4.5.1 Digital camera image

Scanner produces a two-dimensional image. If the output of the camera or other imaging sensor is not in digital form, an analog to digital converter digitizes it. The nature of the sensor and the image it produces are determined by the application.

Fig 5.5.2 digital camera cell

4.5.2 Image Enhancement:

Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interesting an image. A familiar example of enhancement is when we increase the contrast of an image because “it looks better.” It is important to keep in mind that enhancement is a very subjective area of image processing.

Fig 4.5.3 Image enhancement

4.5.3 Image restoration:

Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.

Fig 4.4.5 Image restoration

Enhancement, on the other hand, is based on human subjective preferences regarding what constitutes a "good" enhancement result. For example, contrast stretching is considered an enhancement technique because it is based primarily on the pleasing aspects it might present to the viewer, where as removal of image blur by applying a deblurring function is considered a restoration technique.

4.5.4 Color image processing:

The use of color in image processing is motivated by two principal factors. First, color is a powerful descriptor that often simplifies object identification and extraction from a scene. Second, humans can discern thousands of color shades and intensities, compared to about only two dozen shades of gray. This second factor is particularly important in manual image analysis.

Fig 4.5.4 Color & Gray scale image

4.5.5 Wavelets and multiresolution processing:

Wavelets are the formation for representing images in various degrees of resolution. Although the Fourier transform has been the mainstay of transform based image processing since the late1950's, a more recent transformation, called the wavelet transform, and is now making it even easier to compress, transmit, and analyze many images. Unlike the Fourier transform, whose basis functions are sinusoids, wavelet transforms are based on small values, called Wavelets, of varying frequency and limited duration.

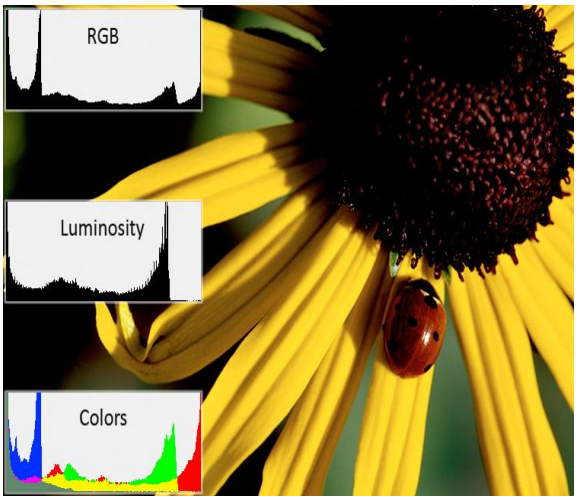


Fig 4.5.5 rgb histogram image

Wavelets were first shown to be the foundation of a powerful new approach to signal processing and analysis called Multiresolution theory. Multiresolution theory incorporates and unifies techniques from a variety of disciplines, including sub band coding from signal processing, quadrature mirror filtering from digital speech recognition, and pyramidal image processing.

4.5.6 Compression:

Compression, as the name implies, deals with techniques for reducing the storage required saving an image, or the bandwidth required for transmitting it. Although storage technology has improved significantly over the past decade, the same cannot be said for transmission capacity. This is true particularly in uses of the Internet, which are characterized by significant pictorial content. Image compression is familiar to most users of computers in the form of image file extensions, such as the jpg file extension used in the JPEG (Joint Photographic Experts Group) image compression standard.

4.5.7 Morphological processing:

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape. The language of mathematical morphology is set theory. As such, morphology offers a unified and powerful approach to numerous image processing problems. Sets in mathematical morphology represent objects in an image. For example, the set of all black pixels in a binary image is a complete morphological description of the image.

Fig 4.5.7 blur to deblur image

In binary images, the sets in question are members of the 2-D integer space Z2, where each element of a set is a 2-D vector whose coordinates are the (x,y) coordinates of a black(or white) pixel in the image. Gray-scale digital images can be represented as sets whose components are in Z3. In this case, two components of each element of the set refer to the coordinates of a pixel, and the third corresponds to its discrete gray-level value.

4.5.8 Segmentation:

Segmentation procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward successful solution of imaging problems that require objects to be identified individually.

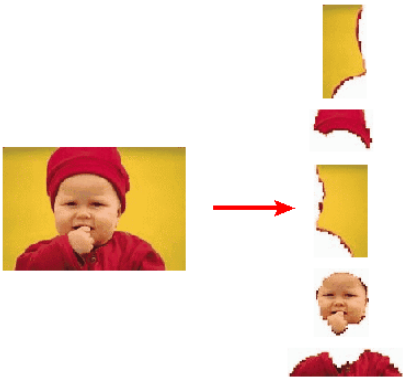


Fig 4.5.8 Image segmentation

On the other hand, weak or erratic segmentation algorithms almost always guarantee eventual failure. In general, the more accurate the segmentation, the more likely recognition is to succeed.

4.5.9 Representation and description:

Representation and description almost always follow the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region (i.e., the set of pixels separating one image region from another) or all the points in the region itself. In either case, converting the data to a form suitable for computer processing is necessary. The first decision that must be made is whether the data should be represented as a boundary or as a complete region. Boundary representation is appropriate when the focus is on external shape characteristics, such as corners and inflections.

Regional representation is appropriate when the focus is on internal properties, such as texture or skeletal shape. In some applications, these representations complement each other. Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing. A method must also be specified for describing the data so that features of interest are highlighted. Description, also called feature selection, deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another.

4.5.10 Object recognition:

The last stage involves recognition and interpretation. Recognition is the process that assigns a label to an object based on the information provided by its descriptors. Interpretation involves assigning meaning to an ensemble of recognized objects.

4.5.11 Knowledgebase:

Knowledge about a problem domain is coded into image processing system in the form of a knowledge database. This knowledge may be as simple as detailing regions of an image when the information of interests is known to be located, thus limiting the search that has to be conducted in seeking that information. The knowledge base also can be quite complex, such as an inter related to list of all major possible defects in a materials inspection problem or an image data base containing high resolution satellite images of a region in connection with change deletion

application. In addition to guiding the operation of each processing module, the knowledge base also controls the interaction between modules.

The system must be endowed with the knowledge to recognize the significance of the location of the string with respect to other components of an address field. This knowledge glides not only the operation of each module, but it also aids in feedback operations between modules through the knowledge base. We implemented preprocessing techniques using MATLAB.

Although large-scale image processing systems still are being sold for massive imaging applications, such as processing of satellite images, the trend continues toward miniaturizing and blending of general-purpose small computers with specialized image processing hardware. Figure 1.25 shows the basic components comprising a typical general-purpose system used for digital image processing. The function of each component is discussed in the following paragraphs, starting with image sensing.

- **Image sensors:**

With reference to sensing, two elements are required to acquire digital images. The first is a physical device that is sensitive to the energy radiated by the object we wish to image. The second, called a digitizer, is a device for converting the output of the physical sensing device into digital form. For instance, in a digital video camera, the sensors produce an electrical output proportional to light intensity. The digitizer converts these outputs to digital data.

- **Specialized image processing hardware:**

Specialized image processing hardware usually consists of the digitizer just mentioned, plus hardware that performs other primitive operations, such as an arithmetic logic unit (ALU), which performs arithmetic and logical operations in parallel on entire images. One example of how an ALU is used is in averaging images as quickly as they are digitized, for the purpose of noise reduction. This type of hardware sometimes is called a front-end subsystem, and its most distinguishing characteristic is speed. In other words, this unit performs functions that require fast data throughputs (e.g., digitizing and averaging video images at 30 frames) that the typical main computer cannot handle.

- **Computer:**

The computer in an image processing system is a general-purpose computer and can range from a PC to a supercomputer. In dedicated applications, sometimes specially designed computers are used to achieve a required level of performance, but our interest here is on general-purpose image processing systems. In these systems, almost any well-equipped PC-type machine is suitable for offline image processing tasks.

- **Image processing software:**

Software for image processing consists of specialized modules that perform specific tasks. A well-designed package also includes the capability for the user to write code that, as a minimum, utilizes the specialized modules. More sophisticated software packages allow the integration of those modules and general-purpose software commands from at least one computer language.

- **Mass storage:**

Mass storage capability is a must in image processing applications. An image of size 1025*1025 pixels, in which the intensity of each pixel is an 8-bit quantity, requires one megabyte of storage space if the image is not compressed. When dealing with thousands, or even millions, of images, providing adequate storage in an image processing system can be a challenge. Digital storage forimage processing applications fall into three principal categories: (1) short-term storage for use during processing, (2) on-line storage for relatively fast recall, and (3) archival storage, characterized by infrequent access. Storage is measured in bytes (eight bits), Kbytes (one thousand bytes), Mbytes (one million bytes), Gbytes (meaning giga, or one billion, bytes), and Tbytes (meaning tera, or one trillion, bytes)

One method of providing short-term storage is computer memory. Another is by specialized boards, called frame buffers that store one or more images and can be accessed rapidly, usually at video rates. The latter method allows virtually instantaneous image zoom, as well as scroll (vertical shifts) and pan (horizontal shifts). Frame buffers usually are housed in the specialized image processing hardware unit shown in Fig. 1.25. Online storage generally takes the form of magnetic disks or optical-media storage. The key factor characterizing on-line storage is frequent access to the stored data. Finally, archival storage is characterized by massive storage requirements but infrequent need for access. Magnetic tapes and optical disks housed in “jukeboxes” are the usual media for archival applications.

- **Image displays:**

Image displays in use today are mainly color (preferably flat screen) TV monitors. Monitors are driven by the outputs of image and graphics display cards that are an integral part of the computer system. Seldom are there requirements for image display applications that cannot be met by display cards available commercially as part of the computer system. In some cases, it is necessary to have stereo displays, and these are implemented in the form of headgear containing two small displays embedded in goggles worn by the user.

- **Hardcopy:**

Hardcopy devices for recording images include laser printers, film cameras, heat-sensitive devices, inkjet units, and digital units, such as optical and CD-ROM disks. Film provides the highest possible resolution, but paper is the obvious medium of choice for written material. For presentations, images are displayed on film transparencies or in a digital medium if image projection equipment is used. The latter approach is gaining acceptance as the standard for image presentations.

- **Network:**

Networking is almost a default function in any computer system in use today. Because of the large amount of data inherent in image processing applications, the key consideration in image transmission is bandwidth. In dedicated networks, this typically is not a problem, but communications with remote sites via the Internet are not always as efficient. Fortunately, this situation is improving quickly as a result of optical fiber and other broadband technologies.

CCM:

In music

- Contemporary Christian music, a genre of popular music which is lyrically focused on matters concerned with the Christian faith
- CCM Magazine, a magazine that covers Contemporary Christian music
- University of Cincinnati College-Conservatory of Music, the performing arts college of the University of Cincinnati
- In the context of MIDI: control change message.

In cryptography

- CCM mode, a mode of operation for cryptographic block ciphers
- Combined Cipher Machine, a common cipher machine system for securing Allied communications during World War II.

In politics

- Chama Cha Mapinduzi, the ruling political party of Tanzania
- Crown Council of Monaco, a seven-member administrative body which meets at least twice annually to advise the Prince of Monaco on various domestic and international affairs
- Convention on Cluster Munitions, is an international treaty that prohibits the use of cluster bombs, a type of explosive weapon which scatters submunitions ("bomblets") over an area.

In religion

- Catholic Campus Ministry, a Catholic student organization on many college campuses

- Council of Churches of Malaysia, an ecumenical body in Malaysia comprising mainline Protestant churches and Oriental Orthodox Church
- Christian Compassion Ministries, a mission organisation in the Philippines.

In sports

- CCM (The Hockey Company), a manufacturing company of Canada
- CCM (cycle), a manufacturing company of Canada
- Central Coast Mariners FC, an Australian professional football (soccer) team based on the Central Coast of New South Wales, Australia

In technology

- Continuous Controls Monitoring describes techniques of continuously monitoring and auditing an IT system
- Continuous Current Mode, operational mode of DC-DC converters
- Cisco CallManager, a Cisco product

In transportation

- CCM (cycle), a cycle manufacturer
- CCM Airlines, a regional airline based in Ajaccio, Corsica, France
- Clews Competition Motorcycles, a British motorcycle manufacturer based in Blackburn, England
- Cabin Crew Member, another name for flight attendant.

In military

- Center for Countermeasures, a United States military center based at White Sands Missile Range, New Mexico
- Command Chief Master Sergeant, a position in the United States Air Force

Integrated color and intensity co-occurrence matrix:

We propose to capture color and intensity variation around each pixel in a two-dimensional matrix called Integrated Color and Intensity Co-occurrence Matrix (ICICM). This is a generalization of the Grayscale Co-occurrence Matrix and the Color Co-occurrence Matrix techniques. For each pair of neighboring pixels, we consider their contribution to both color perception as well as gray level perception to the human eye. Some of the useful properties of the HSV color space and their relationship to human color perception are utilized for extracting this feature. In the next sub-section, we briefly explain relevant properties of the HSV color space. In the subsequent subsection, we describe how the properties can be effectively used for generating

ICICM.

HSV color space:

HSV Color space: Basically there are three properties or three dimensions of color that being hue, saturation and value HSV means Hue, Saturation and Value. It is important to look at because it describes the color based on three properties. It can create the full spectrum of colors by editing the HSV values. The first dimension is the Hue. Hue is the other name for the color or the complicated variation in the color. The quality of color as determined by its dominant wavelength. This Hue is broadly classified into three categories. They are primary Hue, Secondary Hue and Teritiary Hue. The first and the foremost is the primary Hue it consists of three colors they are red, yellow and blue. The secondary Hue is formed by the combination of the equal amount of colors of the primary Hue and the colors of the secondary Hue which was formed by the primary Hue are Orange, Green and violet. The remaining one is the teritiary Hue is formed by the combination of the primary Hue and the secondary Hue. The limitless number of colors are produced by mixing the colors of the primary Hue in different amounts.

Saturation is the degree or the purity of color. Then the second dimension is the saturation. Saturation just gives the intensity to the colors. The saturation and intensity drops just by mixing the colors or by adding black to the color. By adding the white to the color in spite of more intense the color becomes lighter. Then finally the third dimension is the Value. The value is the brightness of the color. When the value is zero the color space is totally black with the increase in the color there is also increase in the brightness and shows the various colors. The value describes the contrast of the color. That means it describes the lightness and darkness of the color. As similar to the saturation this value consists of the tints and shades. Tints are the colors with the added white and shades are the colors with the added black.

Properties of the HSV color space:

Sensing of light from an image in the layers of human retina is a complex process with rod cells contributing to scotopic or dim-light vision and cone cells to photopic or bright-light vision (Gonzalez and Woods, 2002). At low levels of illumination, only the rod cells are excited so that only gray shades are perceived. As the illumination level increases, more and more cone cells are excited, resulting in increased color perception. Various color spaces have been introduced to represent and specify colors in a way suitable for storage, processing or transmission of color information in images.

Out of these, HSV is one of the models that separate out the luminance component (Intensity) of a pixel color from its chrominance components (Hue and Saturation). Hue represents pure color, which is perceived when incident light is of sufficient illumination and contains

a single wavelength. Saturation gives a measure of the degree by which a pure color is diluted by white light. For light with low illumination, corresponding intensity value in the HSV color space is also low.

The HSV color space can be represented as a Hexa cone, with the central vertical axis denoting the luminance component, I (often denoted by V for Intensity Value). Hue, is a chrominance component defined as an angle in the range [0,2p] relative to the red axis with red at angle 0, green at 2p/3, blue at 5p/3 and red again at 2p. Saturation, S, is the other chrominance component, measured as a radial distance from the central axis of the hexacone with value between 0 at the center to 1 at the outer surface. For zero saturation, as the intensity is increased, we move from black to white through various shades of gray. On the other hand, for a given intensity and hue, if the saturation is changed from 0 to 1, the perceived color changes from a shade of gray to the most pure form of the color represented by its hue. When saturation is near 0, all the pixels in an image look alike even though their hue values are different.

As we increase saturation towards 1, the colors get separated out and are visually perceived as the true colors represented by their hues. Low saturation implies presence of a large number of spectral components in the incident light, causing loss of color information even though the illumination level is sufficiently high. Thus, for low values of saturation or intensity, we can approximate a pixel color by a gray level while for higher saturation and intensity, the pixel color can be approximated by its hue. For low intensities, even for a high saturation, a pixel color is close to its gray value. Similarly, for low saturation even for a high value of intensity, a pixel is perceived as gray. We use these properties to estimate the degree by which a pixel contributes to color perception and gray level perception.

One possible way of capturing color perception of a pixel is to choose suitable thresholds on the intensity and saturation. If the saturation and the intensity are above their respective thresholds, we may consider the pixel to have color dominance; else, it has gray level

dominance. However, such a hard thresholding does not properly capture color perception near the threshold values. This is due to the fact that there is no fixed level of illumination above which the cone cells get excited. Instead, there is a gradual transition from scotopic to photopic vision. Similarly, there is no fixed threshold for the saturation of cone cells that leads to loss of chromatic information at higher levels of illumination caused by color dilution. We, therefore, use suitable weights that vary smoothly with saturation and intensity to represent both color and gray scale perception for each pixel.

NON-INTERVAL QUANTIZATION:

Due to the large range for each component by directly calculating the characteristics for the retrieval then the computation will be very difficult to ensure rapid retrieval. It is essential to quantify HSV space component to reduce computation and improve efficiency. At the same time, because the human eye to distinguish colors is limited, do not need to calculate all segments. Unequal interval quantization according the human color perception has been applied on H , S ,V components.

Based on the color model of substantial analysis, we divide color into eight parts. Saturation and intensity is divided into three parts separately in accordance with the human eyes to distinguish. In accordance with the different colors and subjective color perception quantification, quantified hue(H), saturation(S) and value(V)

In accordance with the quantization level above, the H, S, V three-dimensional feature vector for different values of with different weights to form one dimensional feature vector and is given by the following equation:

$$G = Qs*Qv*H+Qv*s+V$$

Where Qs is the quantized series of S and Qv is the quantized series of V. And now by setting Qs = Qv = 3, Then G = 9H+3S+V

$$H=\begin{cases}0 \ if \ h\in[316,20]\\ 1 \ if \ h\in[21,40]\\ 2 \ if \ h\in[41,75]\\ 3 \ if \ h\in[76,155]\\ 4 \ if \ h\in[156,190]\\ 5 \ if \ h\in[191,270]\\ 6 \ if \ h\in[271,295]\\ 7 \ if \ h\in[296,315]\end{cases} \qquad S=\begin{cases}0 \ if \ s\in[0,0.2)\\ 1 \ if \ s\in[0.2,0.7)\\ 2 \ if \ s\in[0.7,1)\end{cases}$$

$$V=\begin{cases}0 \ if \ v\in[0,0.2)\\ 1 \ if \ v\in[0.2,0.7)\\ 2 \ if \ v\in[0.7,1)\end{cases}$$

In this way three component vector of the HSV from one dimensional vector, Which quantize the whole color space for the 72 kinds of the main colors. So we can handle 72 bins of one dimensional histogram. This qualification is effective in reducing the images by the effect of the light intensity, but also reducing the computational time and complexity.

IMAGE RETRIEVAL:

Image retrieval is nothing but a computer system used for browsing searching and retrieving images from a large database of digital images. Most traditional and common methods of image retrieval use some method of adding metadata by captioning, Keywords or the descriptions to the images so that the retrieval can be performed. Manual image

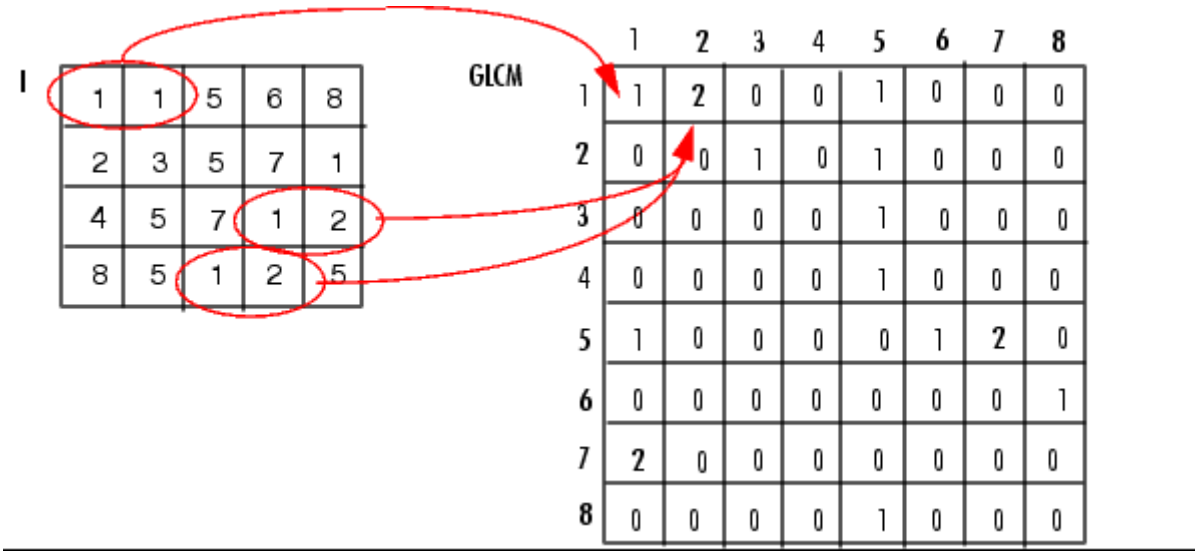
annotation is time consuming, expensive and laborious. For addressing this there has been a large amount of research done on automatic image annotation. It is crucial to understand the scope and nature of the image data in order to determine the complexity of the image search system design. The design is also largely dependent on the factors. And some of the factors include archives, Domain specific collection, Enterprise collection, Personal collection and web etc..,.

An image retrieval system designed to serve a personal collection should focus on features such as personalization, flexibility of browsing, and display methodology. For example, Google's Picasa system [Picasa 2005] provides a chronological display of images taking a user on a journey down memory lane. Domain specific collections may impose specific standards for presentation of results. Searching an archive for content discovery could involve long user search sessions. Good visualization and a rich query support system should be the design goals. A system designed for the Web should be able to support massive user traffic. One way to supplement software approaches for this purpose is to provide hardware support to the system architecture. Unfortunately, very little has been explored in this direction, partly due to the lack of agreed-upon indexing and retrieval methods. The notable few applications include an FPGA implementation of a color-histogram-based image retrieval system [Kotoulas and Andreadis 2003], an FPGA implementation for sub image retrieval within an image database [Nakano and Takamichi 2003], and a method for efficient retrieval in a network of imaging devices [Woodrow and Heinzelman 2002].

OVERVIEW OF TEXTURE:

We all know about the term Texture but for defining it is a hard time. One can differentiate the two different Textures by recognizing the similarities and differences. Commonly there are three ways for the usage of the Textures:

Based on the Textures the images can be segmented To differentiate between already segmented regions or to classify them.We can reproduce Textures by producing the descriptions. The texture can be analyzed in three different ways. They are Spectral, Structural and Statistical:


I
1 1 5 6 8
2 3 5 7 1
4 5 7 1 2
8 5 1 2 5
GLCM
1 2 3 4 5 6 7 8
1 1 2 0 0 1 0 0 0
2 0 0 1 0 1 0 0 0
3 0 0 0 0 1 0 0 0
4 0 0 0 0 1 0 0 0
5 1 0 0 0 0 1 2 0
6 0 0 0 0 0 0 0 1
7 2 0 0 0 0 0 0 0
8 0 0 0 0 1 0 0 0

CHAPTER 5

EXISTING SYSTEM

5.1 Introduction

Glaucoma is the ocular disorder which leads to permanent blindness especially in aged humans. As the disorder is irreversible it is important to detect it in its early stages. The medical techniques used by ophthalmologists such as Heidelberg retinal tomography (HRT) and Ocular coherence tomography (OCT) to screen glaucoma are time consuming and requires special skill and equipment's. Hence there is a need for computer based automatic systems which make screening of glaucoma easier and faster. A digital fundus image is used for screening of glaucoma as it consumes less time, has higher accuracy and requires no skilled force. As compared to other complex devices, digital fundus camera is more economical and is frequently used in basic eye examination.

5.2 Proposed Method for Glaucoma Detection

The proposed method for the detection of glaucoma employs, RGB fundus image is used as an input. In order to detect glaucoma, the most important region of interest is optic disk. Thus, instead of processing on the whole retinal image, region around optic disk is extracted. This ROI is a small image which helps in faster processing and large automated screening of glaucoma. The flowchart of proposed method is described in Fig.5.1.

Flow chart:

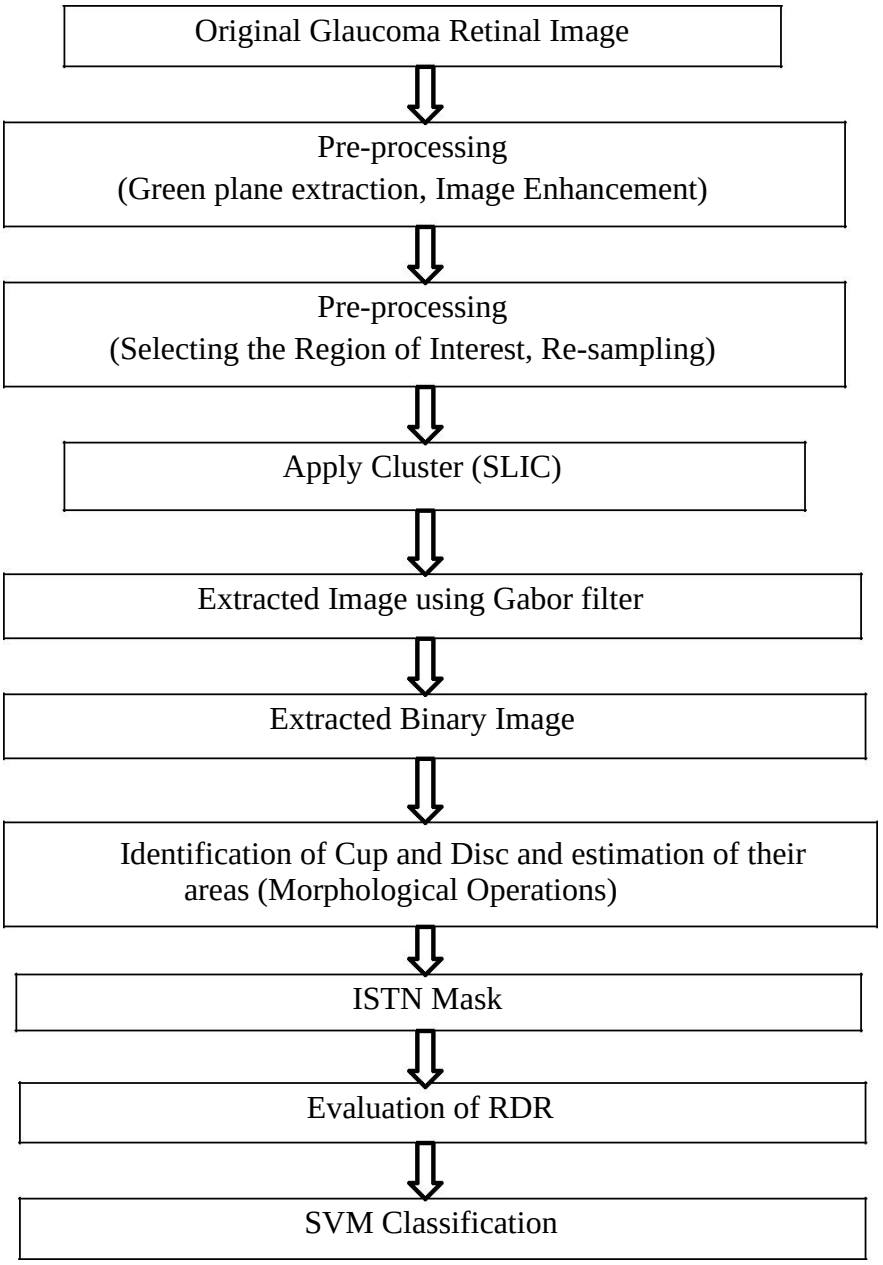


Fig.5.1 Flowchart for Glaucoma Detection

5.2.1 Region of Interest

In order to extract the region around optic disk, Centre of optic disk is first computed. A window of dimension r*column is created as shown in Fig.5.2 (a). Where r is the radius of optic disk (approximate). Then passing the window through the image spatially, for all rows and obtaining the maxima, we find the row of centre of optic disk. Another window of dimensions r*r is created as shown in Fig.5.2 (b). This window is passed through detected row for all the columns and the maxima are detected. Hence the coordinates of centre of optic cup is obtained.

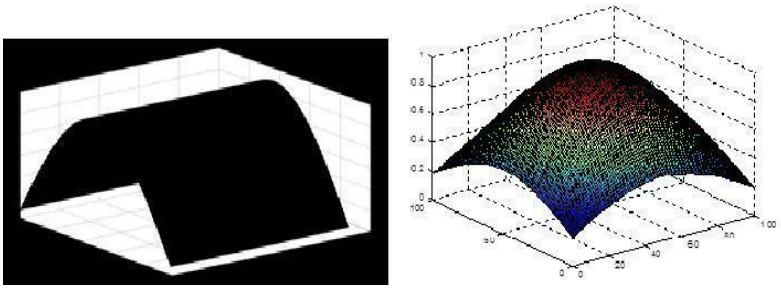
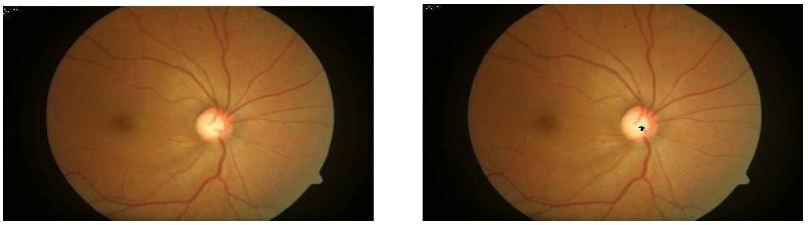
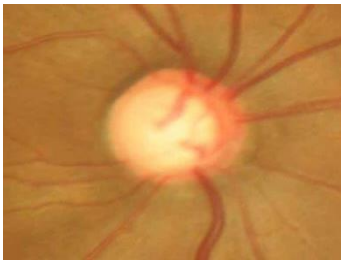


Fig.5.2 :(a) Window1 for OD Detection, (b) Window 2 for OD detection

The ROI as shown in Fig.5.3(c) is defined as a square around optic disk centre with dimensions of thrice the optic disk diameter.



(a) (b)



(c)

Fig.5.3: (a) Input RGB fundus image, (b) Fundus image with OD Centre marked on it, (c)Region of interest

5.2.2 SLIC Algorithm

SLIC is a simple and efficient method to decompose an image in visually homogeneous regions. It is based on a spatially localized version of k-means clustering. Similar to mean shift or quick shift, each pixel is associated to a feature vector.

SLIC takes two parameters: the nominal size of the regions (super pixels) region Size and the strength of the spatial regularization regularizer. The image is first divided into a grid with step region Size. The centre of each grid tile is then used to initialize a corresponding k-means (up to a small shift to avoid image edges). Finally, the k-means centre and clusters are refined, yielding the segmented image. As a further restriction and simplification, during the kmeans iterations each pixel can be assigned to only the 2 x 2 centres corresponding to grid tiles adjacent to the pixel.

After the k-means step, SLIC optionally removes any segment whose area is smaller than a threshold minRegionSize by merging them into larger ones. In SLIC, k initial cluster centres Ck are sampled on a regular grid spaced by pixels apart from the image with N pixels. The centres are first moved towards the lowest gradient position in a 3 × 3 neighbourhood. Clustering is then applied. For each Ck, SLIC iteratively searches for its best matching pixel from the neighborhood around Ck based on colour and spatial proximity and then compute the new cluster centre based on the found pixel. The iteration continues until the distance between the new centres and previous ones is small enough.

Fig 5.2.1: lab space with cluster centers

5.2.3 Optic Disk Segmentation

The extracted ROI image consists of three channels, red, green and blue as shown in Fig 5. To detect optic disk, red channel is used as in this channel, optic disk appears to be the brightest and blood vessels are also suppressed in this channel. Hence it is easier and accurate to segment optic disk in red channel of input fundus image. Otsu thresholding technique is used to segment the optic disk which makes the segmentation independent of image quality. Segmented optic disc given by

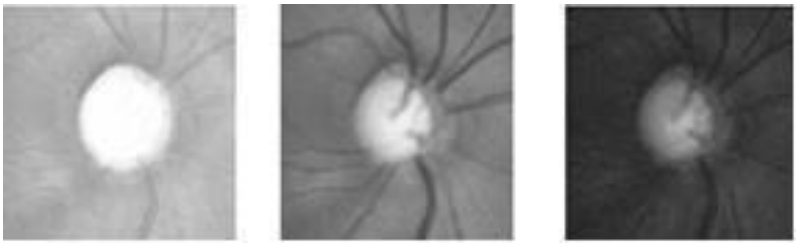


Fig.5.2.2: (a) Red channel, (b) Green channel, (c) Blue channel

5.2.4 Optic Cup Segmentation:

Optic cup is segmented using green component of ROI image shown in Fig.5.5(b). The flowchart of optic cup segmentation is shown in Fig.5.5.

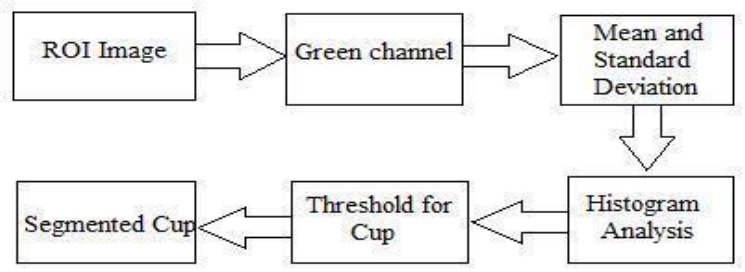


Fig.5.2.3: Flowchart for optic cup segmentation

Statistical features such as mean and standard deviation for green channel of ROI image are calculated. These features help in making the method adaptive of image quality.

Mean X and standard deviation σare obtained as:

X=1/N xi…….(1)

σ2= 1/N (xi – X) 2-----(2)

Standard deviation is defined as the square root of variance.

Histogram of green channel is examined and is shown in Fig.5.6. It is a graphical representation of number of pixels with respect to gray levels in digital image. The horizontal axis of histogram represents the tonal variation while the vertical axis of histogram represents total number of pixels in each tone.

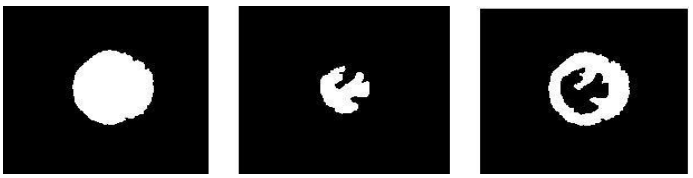
The histogram of each image is examined to determine threshold level for segmenting optic disk and optic cup. From the statistical patterns of histogram, it is known that the mean is the central tendency and the standard deviation is the dispersion from that central value. The addition of mean and standard deviation gives an intensity level that point to the highest number of pixels in the grey region of the image. After analyzing the images, it is determined

experimentally that optic cup lies at an intensity level given by Tcup =Mean+3* Standard deviation

Where Tcup is the Intensity level which is decided as a threshold value for segmenting optic cup which is shown in Fig.5.7 (b).

5.2.5 Neuroretinal Rim

Neuroretinal rim is the region located between the edge of optic disk and optic cup. After segmenting of optic disk Fig 8(a) and optic cup Fig 8(b), NRR is obtained by subtracting optic cup from optic disk. NRR is shown in Fig 8(c).



(a) (b) (c)

Fig.5.2.4: (a) Segmented optic disk, (b) Segmented optic cup,(c) Neuroretinal rim

The change in appearance of this area helps in identifying damage to the disk due to glaucoma. Thickness of the rim is an important feature to detect whether the fundus image is glaucomatic or not. The healthy optic disk is thicker in inferior portion then in superior, then nasally and thinnest temporally. In a glaucomatic eye, cup increases in its area vertically reducing the thickness of the rim in infero - temporal disk sectors. Thus rim-disk ratio for infero-temporal region of the rim can be evaluated to determine glaucoma.

Algorithm for evaluating Rim-disk ratio:

1. Mask of size of ROI image is created for each quadrant of fundus image as shown in Fig.5.8.
2. Each quadrant mask is multiplied individually with Neuroretinal. The output is shown in Fig.5.8 (b).
3. Total area of rim is calculated in inferior and temporal regions.
4. Ratio of rim area in infero-temporal region to the total disk area is taken as RDR.

(a) (b) (c) (d)

Fig.5.2.5: ISNT Mask for (a) Inferior region, (b) Superior region, (c) Temporal region, (d) Nasal regio

Fig.5.2.6: ISNT mask multiplied with Neuroretinal rim

5.2.6 Classification

Classification of images as glaucomatic or healthy is done on the basis of two parameters; Cup to disk ratio and rim to disk ratio.

5.2.6.1 Cup to disk ratio (CDR)

The first parameter that we use to detect glaucoma is CDR i.e. cup-disc ratio. CDR is defined as ratio of total segmented cup area to total segmented disk area.

CDR = Optic Cup Area/Optic Cup Area……..(5)

Cup area and disk area are obtained by summing all the white pixels in segmented cup and disk. This calculated CDR is used for screening of Glaucoma. If CDR is greater than 0.3(globally accepted value), the fundus image under test is said to be glaucomatic else it is healthy.

5.2.6.2 Rim to disk ratio (RDR)

The CDR in itself is not one of the best predictors of whether the eye is glaucomatic. Many a times a person has been diagnosed inappropriately due to large optic cup in presence of large optic disc and has healthy rim tissues. Thus Neuroretinal rim tissue plays a vital role in glaucoma detection. Rim to disk ratio is evaluated using the algorithm mentioned above. It is defined as

RDR= Rim area in infero-temporal region/ Disk Area…….. (6)

After analyzing a database of 50 images, value of 0.5 is decided as threshold for classifying images as glaucomatic stage. If RDR is less than equal to 0.5, fundus image is considered to be Normal glaucomatic.

These parameters are calculated to train the SVM classifier for detecting glaucoma so as to increase the reliability and robustness of the system.

5.3 SVM Classifier

The Support Vector Machine (SVM) is a popular data classification technique, which was proposed by Vapnik and his group at AT&T BELL Laboratories. Generally speaking, the SVM is a supervised learning method that can analyze data and recognize patterns. It has been widely used in human face recognition, handwriting recognition, and vehicle licenseplate recognition and can outperform other competing methods in most cases.

The main concept of the SVM is to construct a hyper-plane or a set of hyper-planes in a high or infinite dimensional space, and use them to classify the data. Among the possible hyper-planes, SVMs select the one where the distance of the hyper-plane from the closest data point is as large as possible. The distance between this data point and the hyper-plane is known as the margin.

Figure 5.10 shows the geometric interpretation of the SVM, the figure on the left presents a large margin whereas the image on the right displays a small margin. As a result, the hyper-plane on the left is more desirable than the one on the right for the purposes of classification.

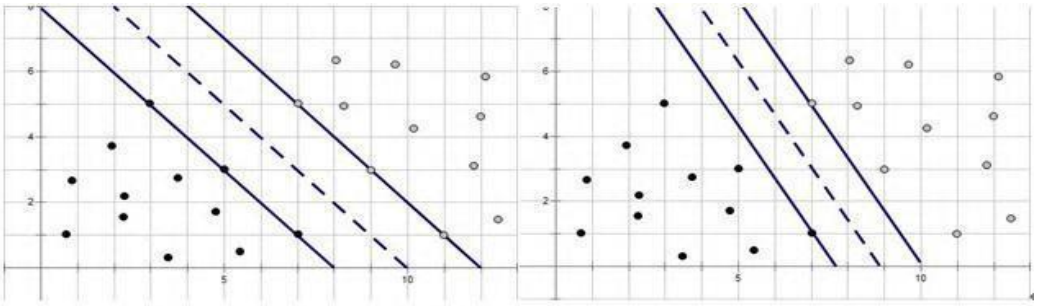


Fig.5.3.1 Geometric interpretation of SVM

The challenge in SVM classification is to find a hyper-plane with a maximal margin.

In addition to linear classification, a SVM can be applied to nonlinear classification problems. When applying a SVM in non-linear problems, nonlinear mapping is used to map the data into a high-dimensional feature space, where the data can be linearly classified. To implement the SVM for pavement segmentation, both the color as well as the textural information are used. The color information is specified by RGB values and the texture is obtained by the application of the Gabor filter as explained in the following.

The use of SVM for pavement segmentation requires an initial training phase for the estimation of the optimal parameters. The training phase involves the extraction of the features both from the background and the pavement. For the proposed approach, we have chosen two kinds of features: the color feature and the texture feature.

The RGB value is one of the dominant factors in the color image. The RGB color model is an additive color model in which red, green, and blue lights are added together in various ways to reproduce a broad array of colors. Each point has a set of values called the RGB value which is a number between 0 and 255. As a result, the RGB value has been used as the color feature.

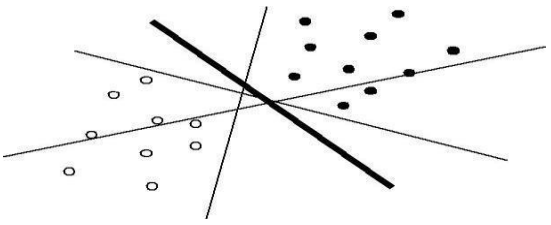


Fig.5.3.2 SVM Hyperplanes between two classes

The two planes parallel to the classifier and which passes through one or more points in the data are called ‘bounding planes’. The distance between these bounding planes is called ‘margin’. By the process of learning hyperplane, which maximizes this margin, is evaluated. The points of the corresponding class, which falls on the bounding planes, are called ‘support vectors’. These points are crucial in forming a hyperplane hence the name support vector machine. Figure 5.12 shows the

concept of support vectors, bounding planes and maximum margin.

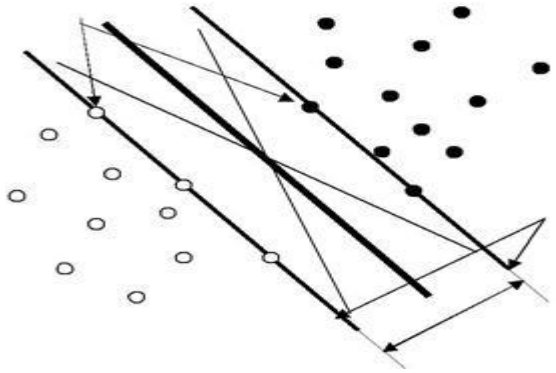


Fig 5.3.3 Bounding planes, Support vectors and Maximum Margin in SVM

In spite of taking all the required measures for classification, there are chances likely for misclassifications. SVM takes care of them, by allowing misclassifications of pixels between classes. Figure 5.13 shows two classes for classification, class A with white dots and class B with black dots. The hyperplane gives its maximum efforts in all the possible ways to classify the image with very less misclassification. The hyperplane is a straight line in this classification. It would be much more interesting if the hyperplane is a twisted line such that it surpasses the pixels of other class and classify the distinct classes without misclassification errors. Such type of classification with twisted separating boundary is known as nonlinear

SVM classification. Broadly, SVM is of two types. They are linear and nonlinear type of SVM. If the hyperplane in the SVM classification is linear in nature, it is known as linear SVM and if the hyperplane in the SVM is a nonlinear equation, it is known as nonlinear SVM. A non-linear SVM is achieved by using a kernel trick.

CHAPTER 6

PROPOSED METHODOLOGY

Since it is hard to make a paired dataset that contains different quality fundus images of the same subject. In addition, even if we can take fundus images of the same patient in different quality, there is no guarantee that those images are corresponding in the pixel-level. For that reason, the possible way is to exploit unpaired dataset. Therefore, the CycleGAN-based model [8], which can be used for image translation with unpaired dataset, is applied in our image enhancement task as the main model. Although the CycleGAN can provide the realistic images, it may fail to maintain the pathological information such as drusen. Because the drusen information is not important for generating the realistic fundus images, the model can ignore them during training. To keep the pathological information, we apply drusen mask as an additional input for training our model. The full description of our CycleGAN-based model can be found in II-C. We use a private unpaired dataset for training the main model. However, since our AMD dataset does not include the label (good and bad quality class or drusen segmentation masks).

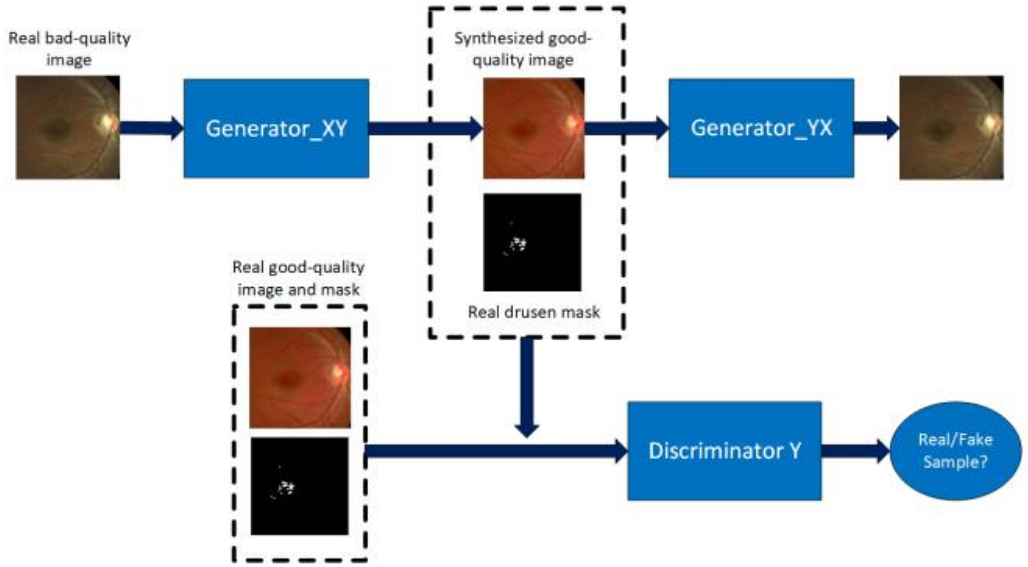


Fig. 6.1. The proposed framework for retinal image enhancement. Note that the discriminator DX is not shown for better visualization

We exploit a CycleGAN-based model for the main task. The model includes two generators GXY and GY X and two discriminators DX and DY . The goal is to translate images from blurry ones (denote as domain X) to enhanced ones (denote as domain Y). The generator GXY takes blurry images as its input and produces enhanced images while the generator GY X does the inverse task. The main difference between our model and the original CycleGAN is from the discriminators.

The input of discriminator is not only the output images from the generator but also the drusen segmentation masks.The discriminator DY takes the segmentation mask and either real image from domain Y (real sample) or the output of generator GXY (fake sample) and try to distinguish whether the input is real or fake. The discriminator DX does a similar task in domain X. The additional segmentation masks help the model to focus on the drusen area while producing the output images. If the output fundus image looks realistic but it is not consistent with the drusen maks, the discriminator can easily classify it as a fake sample. By adding the drusen segmentation mask, the generator should make not only realistic output images but also accurate pathological information. The main model is described in Fig. 1. The objective function of our model is the same as the original CycleGAN, which includes the adversarial loss and cycle consistency loss. Let define x and mx are the real image and its corresponding drusen segmentation mask from domain X while y and my are from domain Y (x, mx ∈ Pdata(X) and y, my ∈ Pdata(Y)). The adversarial loss of GXY and DY with the conditional segmentation mask is described as:

$$\min_{G_{XY}} \max_{D_Y} L_{adv} = E_{y,m_y \in P_{data}(Y)}[logD_Y(y, m_y)] + E_{x,m_x \in P_{data}(X)}[1 - logD_Y(G_{XY}(x), m_x)]$$

Image Quality Measure:

IQMs (image quality measures) are divided into three groups including pixel distance-based, correlation-based, and mean square error-based . In this research, the technique used for measuring the quality was Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR). MSE refers to the difference of mean square error between the segmented image and the original image. MSE value was calculated using Equation 1. PSNR is the ratio between the maximum value of measured signal and the number of noise which affect the image. PSNR value was calculated using Equation 2 . The method of contrast enhancement are good if MSE value is smaller while PSNR value is bigger than other methods of contrast enhancement

$$MSE = \frac{1}{MN}\sum_{j=1}^{M}\sum_{k=1}^{N}[F(j,k) - H(j,k)]^2$$

$$PSNR = 20\log_{10}\left(\frac{\max value}{MSE}\right)$$

CHAPTER 7
SOURCE CODE

```
clc
clear all
close all

% READING INPUT IMAGE

[filename pathname]=uigetfile(
{'*.jpg;*.bmp;*.tif;*.png'});
I=imread([pathname filename]);
IMAG=imresize(I,[512 512]);
figure,imshow(IMAG,[]);
title('original image');
red=IMAG(:,:,1);
green=IMAG(:,:,2);
blue=IMAG(:,:,3);

figure,imshow(green,[]);
title('green gray image');
% INITIAL CONTRAST ENHANCEMENT
EQH=histeq(green);
figure,imshow(EQH,[]);
title('histogram equalized image');


% IMAGE ENHANCEMENT
% IMAGE ENHANCEMENT
for i=1:3
EQA=adapthisteq(IMAG(:,:,i));
figure,imshow(EQA,[]);
title('enhanced image');
cont(:,:,i)=imadjust(EQA);
end
figure,imshow(cont,[]);
title('clear image');

%%SLIC FOR SUPERPIXELS%%
aga=slic_full(cont,IMAG);

% K-MEANS CLUSTERING%%
kmeans_fp(aga);
```

```
%%%% ROI SEGMENTATION %%%%
ahe=adapthisteq(green);
con=imadjust(ahe);
[bw bw1 im2 ROI ROIvein]=gab_roi(con);
figure,imshow(ROIvein);
title('ROI IMAGE');

%%%% GABOR EXTRACTION %%%%

M=4;
N=3;
a=(0.4 / 0.05)^(1/(M-1));
count=1;
[JT]=tex_gabor(M,N,a,count,ROIvein);

figure,imshow(JT,[])
title('EXTRACTED IMAGE USING GABOR');
% JT2=im2bw(imresize(JT,[512 512]));
% JT1=(imresize(JT,[512 512]));
JT2=im2bw(JT);
figure,imshow(JT2,[])
title('EXTRACTED BINARY IMAGE');

% Extracting Disk from Segmented blood vessel using
Morphological Features

se=strel('disk',6);
JT3=imerode(JT2,se);
[m n]=size(JT3);

for i = 1:m
    for j=1:n
        if JT3(i,j)==1
            dis(i,j)=1;
        else
            dis(i,j)=0;
        end
    end
end
figure,imshow(dis);
title('Extracted morpho');

%     z2=1-out22;
```

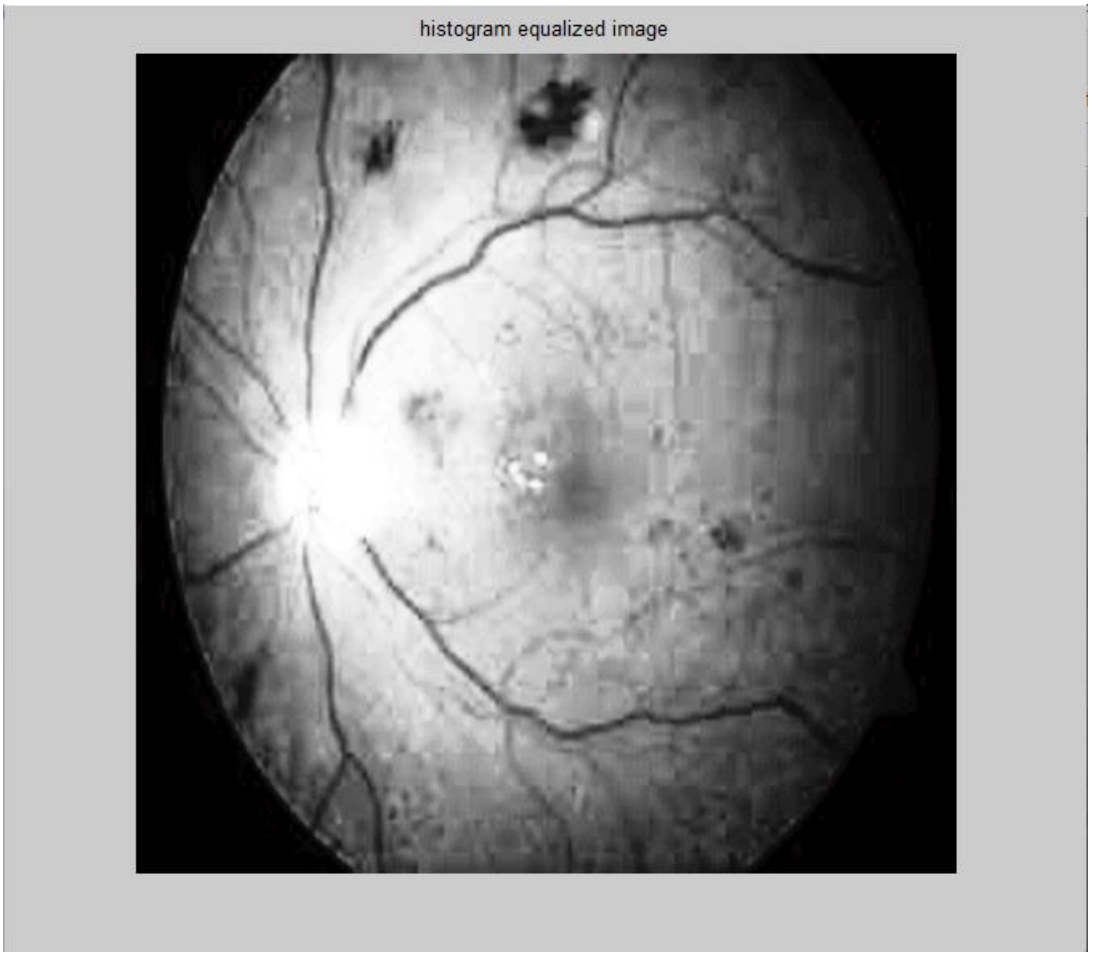
```
%     z3=bwareaopen(z2,2500);
%        z3=bwareaopen(z2,500)
%     z4=imfill(z3,'holes');
%
% AREA OF OPTIC DISK

 Area_disk=0;
  for i =1:m
      for j=1:n
          if dis(i,j)==1
              Area_disk=Area_disk+1;
          end
      end
  end

%   Extracting cup from Segmented blood vessel using
Morphological Features
  o1=imcomplement(bw1);
  disk=bwareaopen(bw,500);
  figure,imshow(disk);
  title('result Extracted disk ');

  %%%  ISNT Mask for Inferior region
  L = tril(disk);

se = strel('disk',10);
closeBW = imclose(L,se);
figure, imshow(closeBW);
% p1=bwareaopen(o1,300);
% figure,imshow(p1);
% % p=bwmorph(o1,'remove');
% p1=imfill(o1,'holes');
[s t]=size(disk);

% AREA OF OPTIC DISK

Area_cup=0;
  for i =1:s
      for j=1:t
          if disk(i,j)==1
              Area_cup=Area_cup+1;
          end
      end
  end
```

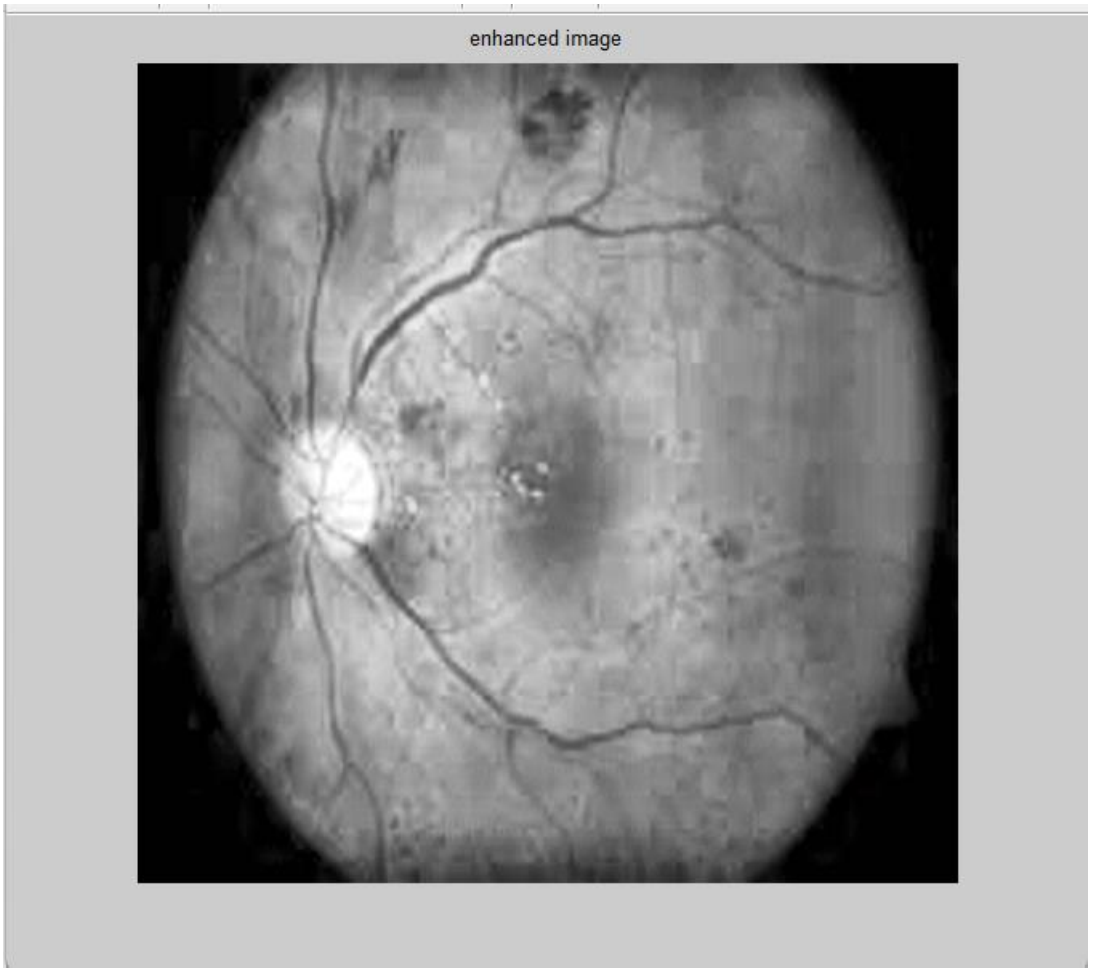
```
% CALCULATION OF CUP TO DISK RATIO;

 redpts = zeros(100,2);grnpts = redpts;
for i = 1:100
    grnpts(i,:) =
mvnrnd(redpts(randi(10),:),eye(2)*0.2);
    redpts(i,:) =
mvnrnd(redpts(randi(10),:),eye(2)*0.2);
end
figure
plot(grnpts(:,1),grnpts(:,2),'go');
hold on
plot(redpts(:,1),redpts(:,2),'ro');
hold off
cdata = [grnpts;redpts];
grp = ones(200,1);
% green label 1, red label -1
grp(101:200) = -1;
svmStruct =
svmtrain(cdata,grp,'Kernel_Function','rbf',...
'showplot',true);
[svmStruct svmStruct1]=size(svmStruct);
RDR=(Area_cup/Area_disk);

disp(RDR)

% subplot(1,2,1),imshow(dis);
%   title('Extracted Disk');
%   subplot(1,2,2),imshow(cup);
%   title('Extracted Cup');

% % % % % PARAMETER SECTION
%
%   fprintf('Area of Optic Disc is\n')
% disp(Area_disk);
%   fprintf('Area of Optic cup is\n')
% disp(Area_cup);
%   fprintf('Cup to Disc Ratio')
% disp(RDR);
% RDR=disk;

if RDR<0.4
    helpdlg('Normal condition');
elseif RDR>0.4 & RDR<25
```

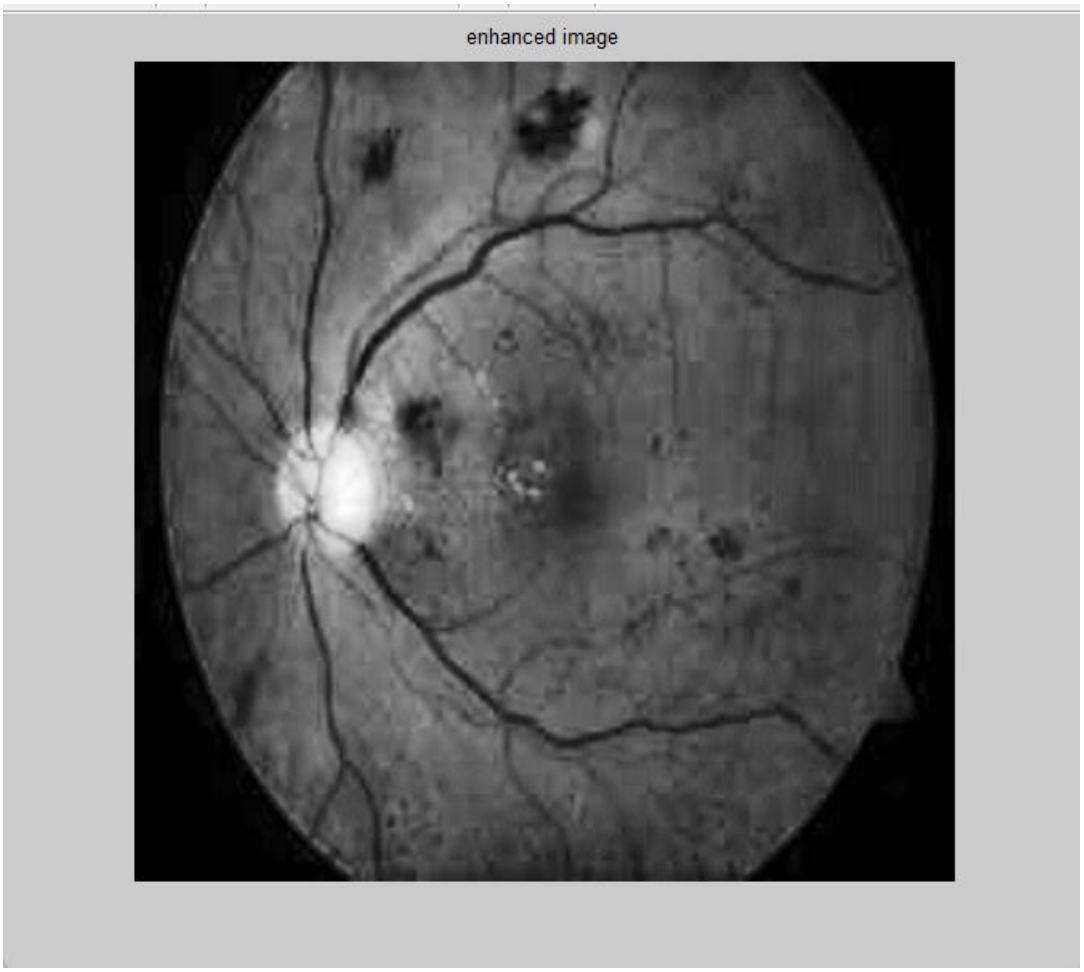
```
    helpdlg('medium condition');
elseif RDR>25
    helpdlg('abNormal condition');
end

%^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^END^^^^^^^^^^^^^^^^^^
^^^^^^^^^^^^^^^^^^
```

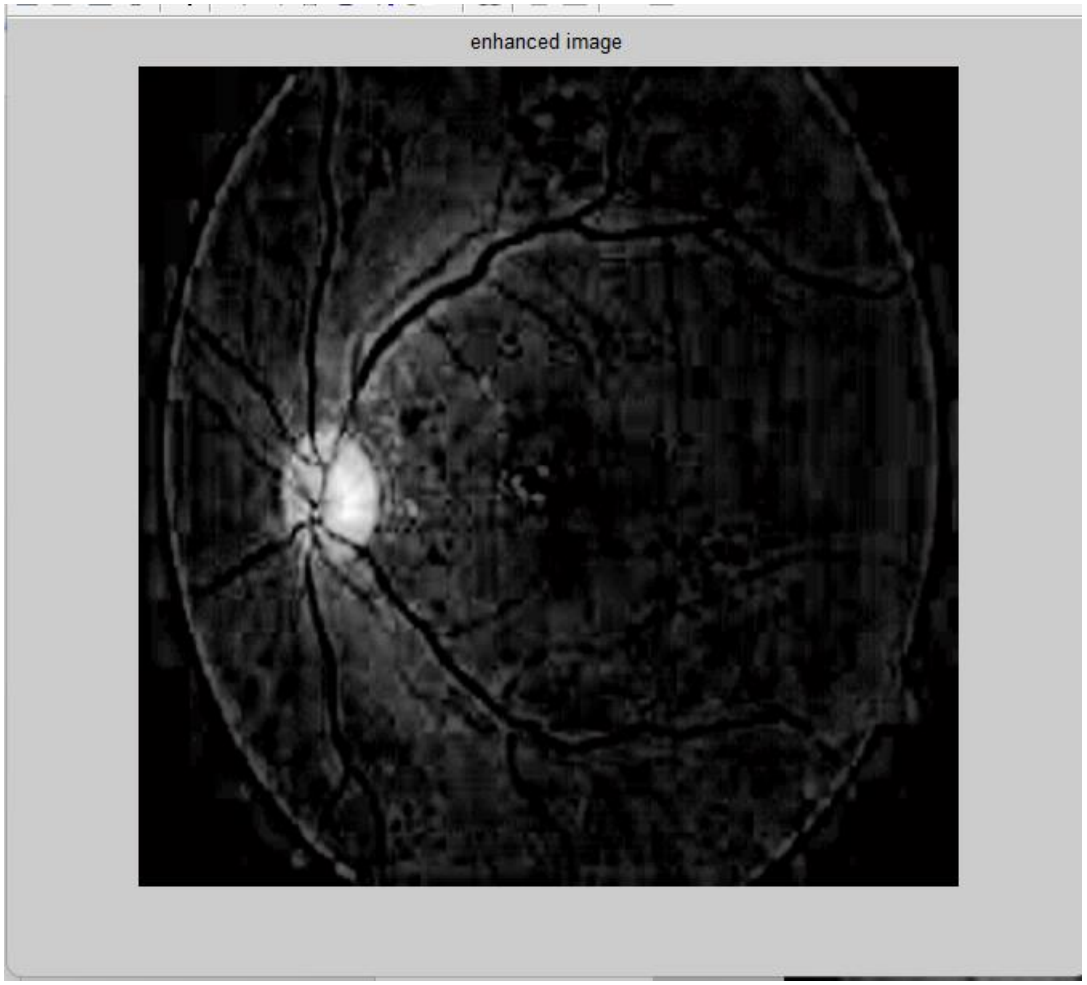
CHAPTER 8
SIMULATION RESULTS


green gray image

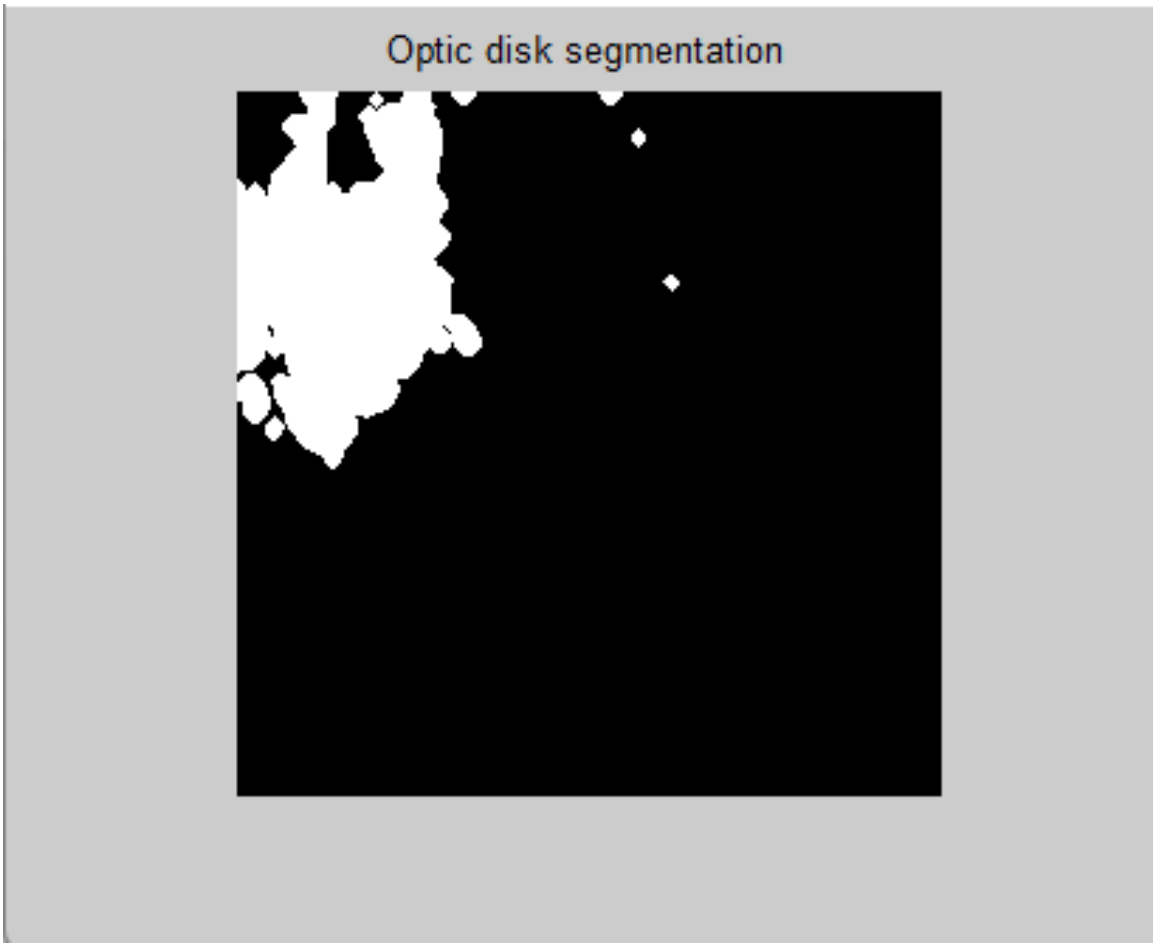

histogram equalized image

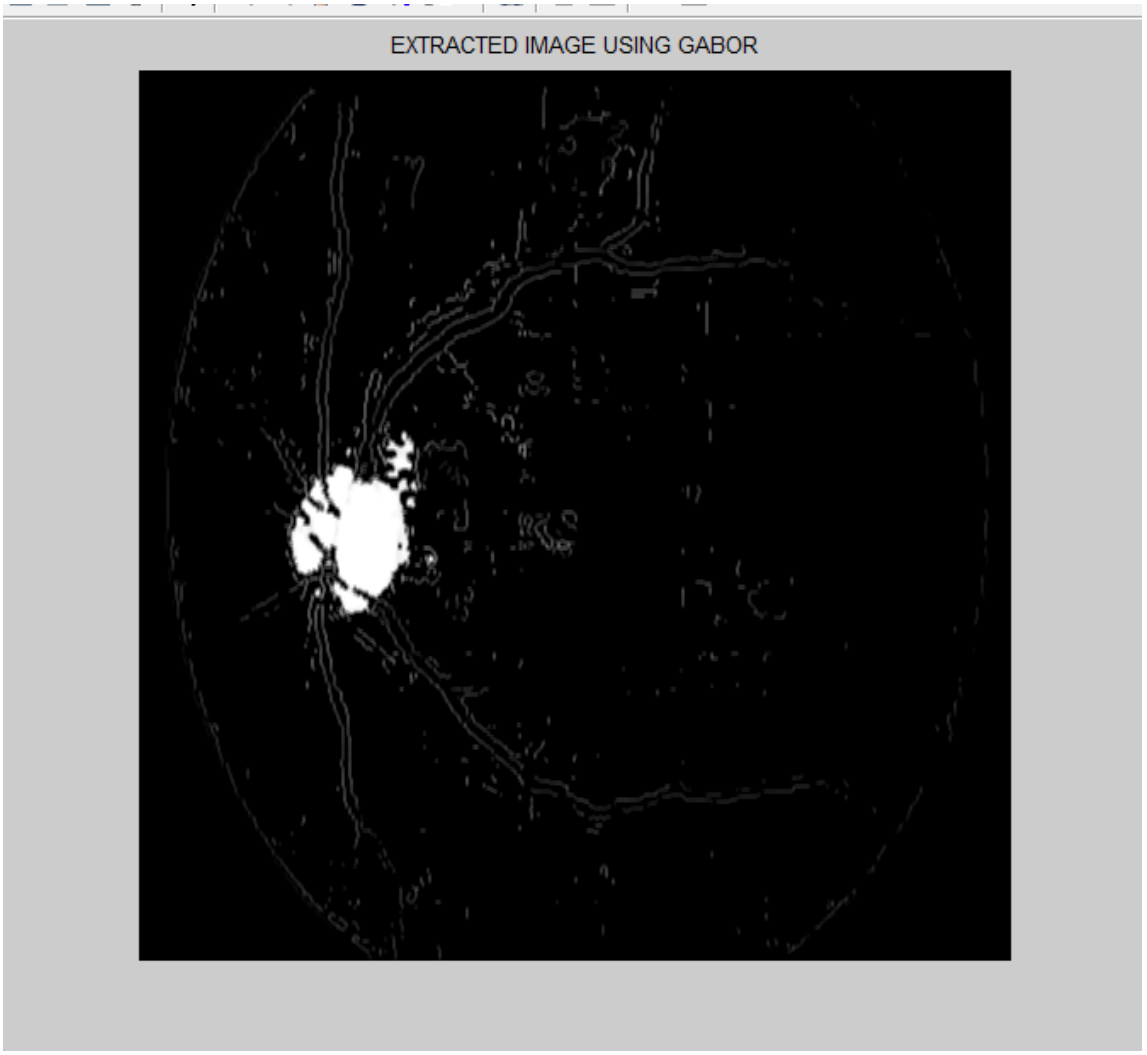

enhanced image

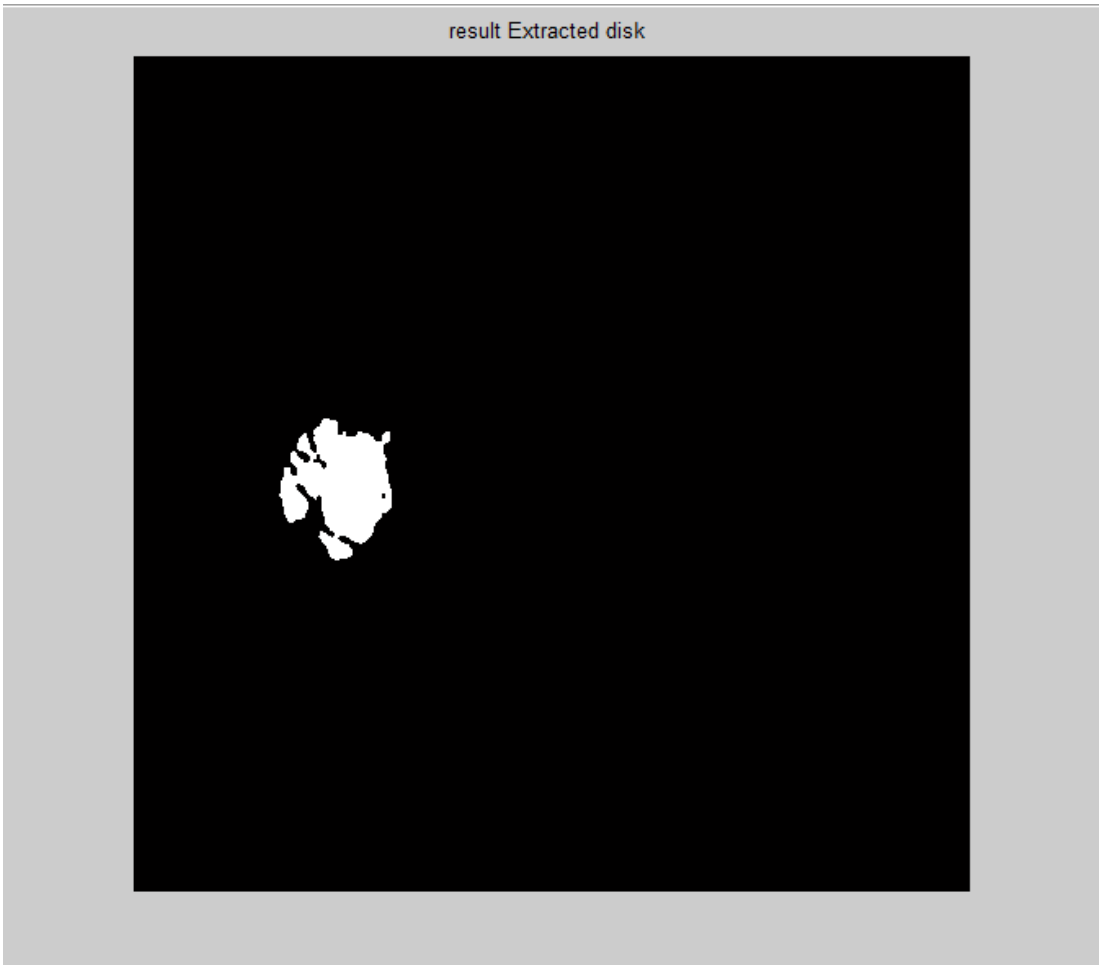

enhanced image


enhanced image


clear image


Optic disk segmentation


EXTRACTED IMAGE USING GABOR


result Extracted disk

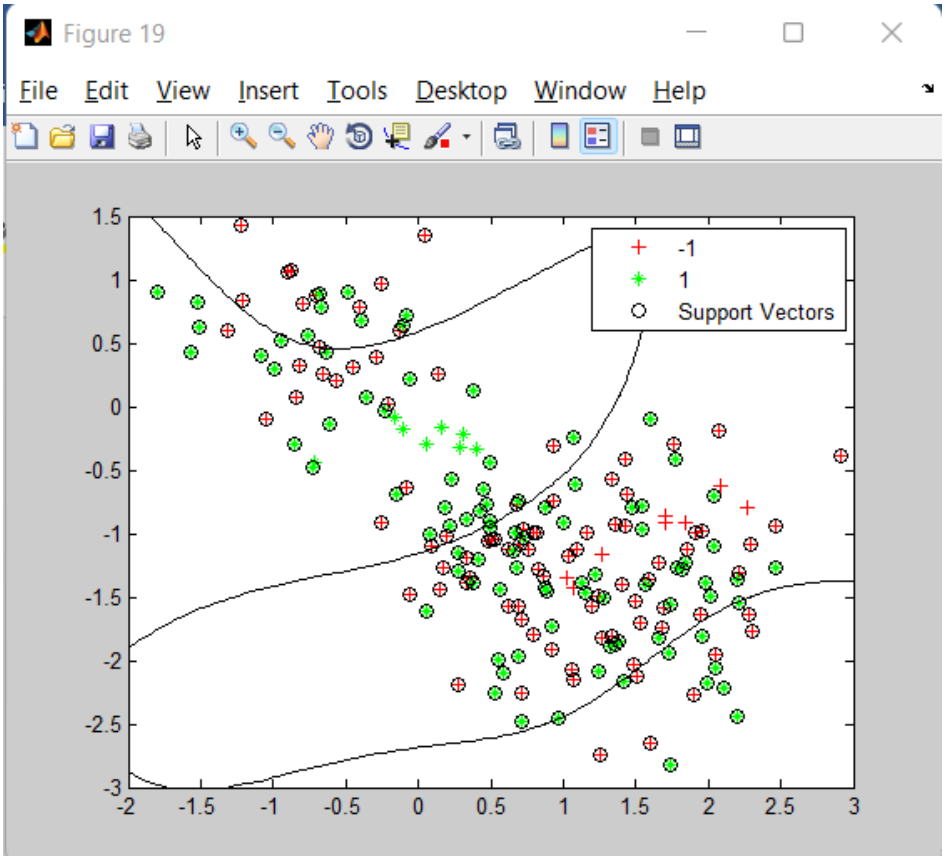
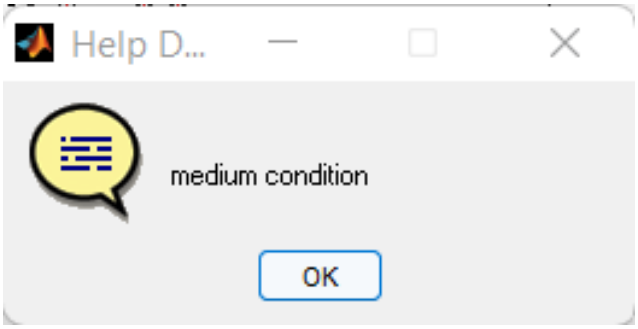


Figure 19
File Edit View Insert Tools Desktop Window Help
-1
1
Support Vectors
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
-2 -1.5 -1 -0.5 0 0.5 1 1.5 2 2.5 3


Help D...
medium condition
OK

CONCLUSION

The conclusion of this study is from the visualization, it showed that the red component had high brightness level so it can differ the optic disc and other parts of retinal eyes. The green component still has vessel blood so it will make it more difficult to segment the images. Blue component results in very dark of retinal image. From MSE and PNSR values, it showed that the green component had the smallest MSE value while the blue component of glaucoma image has the biggest MSE value. PSNR value was obtained from the green component. Both red and blue the component had PSNR value which had a small difference. From these results, it can be concluded that the MSE and PSNR values do not guarantee visual results. So that for further research, it is expected that the MSE and PSNR values will be obtained from the part that we want to observe.

REFERENCES

[1] S. Nawaldgi, “Review of Automated Glaucoma Detection Techniques,” in 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), 2016, pp. 1435–1438.

[2] J. Carrillo, L. Bautista, J. Villamizar, J. Rueda, and M. Sanchez, “GLAUCOMA DETECTION USING FUNDUS IMAGES OF THE EYE,” 2019 XXII Symp. Image, Signal Process. Artif. Vis., pp. 1–4, 2019.

[3] T. Khalil, M. U. Akram, S. Khalid, and A. Jameel, “Improved automated detection of glaucoma from fundus image using hybrid structural and textural features,” IET Image Process., vol. 11, no. 9, pp. 693–700, 2017.

[4] J. Cheng, F. Yin, D. Wing, K. Wong, D. Tao, and J. Liu, “Sparse Dissimilarity-constrained Coding for Glaucoma Screening,” IEEE Trans. Biomed. Eng., vol. X, no. X, 2015.

[5] J. Sivaswamy, S. R. Krishnadas, and A. Chakravarty, “A Comprehensive Retinal Image Dataset for the Assessment of Glaucoma from the Optic Nerve Head Analysis,” JSM Biomed Imaging Data Pap 2(1) 1004, vol. 2, pp. 1–7, 2015.

[6] S. Kavitha, S. Karthikeyan, and K. Duraiswamy, “Neuroretinal rim Quantification in Fundus Images to Detect Glaucoma,” Ijcsns, vol. 10, no. 6, p. 134, 2010.

[7] A. Issac, M. Partha Sarathi, and M. K. Dutta, “An adaptive threshold based image processing technique for improved glaucoma detection and classification,” Comput. Methods Programs Biomed., 2015.

[8] S. B. Akhade, V. U. Deshmukh, and S. B. Deosarkar, “Automatic Optic Disc Detection in Digital Fundus Images Using Image Processing,” 2014 Int. Conf. onGreen Comput. Commun. Electr. Eng. (ICGCCEE 2014), pp. 4–7, 2014.

[9] J. Nayak, R. Acharya U., P. S. Bhat, N. Shetty, and T. C. Lim, “Automated diagnosis of glaucoma using digital fundus images,” J. Med. Syst., vol. 33, no. 5, pp. 337–346, 2009.