VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI

PROJECT REPORT ON

"Detection of Malaria Using Image Processing"

Submitted in partial fulfilment of the requirements for the award of

Bachelor's Degree in Information Science and Engineering

Submitted by:

VYSHAK G	**4VV13IS062**
HARSHITH B	**4VV13IS017**
PAVITHRA R	**4VV13IS031**
VINUTHA D	**4VV13IS059**

Under the guidance of:

PROF. PREMA N S

Associate Professor

Department of Information Science and Engineering

Vidyavardhaka College of Engineering

Gokulum 3rd stage, Mysuru-570002

2016 - 2017

VIDYAVARDHAKA COLLEGE OF ENGINEERING

Gokulum 3rd stage, Mysuru-570002

Department of Information Science and Engineering

Certificate

This is to certify that the project titled **"Detection of Malarial Parasite in Blood Using Image"** is a bonafide work carried out by **VYSHAK G(4VV13IS062), HARSHITH B(4VV13IS017), PAVITHRA R(4VV13IS031)** and **VINUTHA D(4VV13IS059)** of 8th semester Information Science and Engineering as prescribed by Visvesvaraya Technological University, Belagavi during the academic year 2016-2017. It is certified that all corrections and suggestions indicated for Internal Assessment have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the Bachelor of Engineering degree.

Signature of the Guide	Signature of the HOD	Signature of the Principal
Prof. PREMA N S	Dr. RAVI KUMAR V	Dr. B SADASHIVE GOWDA
Associate Professor	HOD	Principal
Dept. of IS&E	Dept. of IS&E	VVCE, Mysuru
VVCE, Mysuru	VVCE, Mysuru	

EXTERNAL VIVA-VOCE

SL. No	Name of the examiners	Signature with date
1.		
2.		

ABSTRACT

This paper reviews image analysis studies aiming at automated diagnosis or screening of malaria infection in microscope images of thin blood film smears. In this work, we propose a algorithm consisting of colour based pixel discrimination technique and morphological operation to identify malaria parasites from thick smear images. The algorithm generated will be helpful in the area where the expert in microscopic analysis may not be available. This project curbs the human error while detecting the presence of malaria parasites in the blood sample by using image processing and automation. Evaluation of percentage of detection and efficiency shows that which algorithm i.e. colour based pixel discrimination technique and morphological operation has higher predictive rate, which is comparative study when tested on same input slide.

ACKNOWLEDGEMENT

It is our privilege to express gratitude to all those who inspired me and guided us to complete the project work. This work has been accomplished only with the direct or indirect help of many people who have guided me. We am grateful to them.

We owe gratitude to The Principal, **Dr. B. Sadashive Gowda** for providing congenial atmosphere and his whole hearted support.

We wish to express our deepest gratitude to **Dr. Ravi Kumar V**, Head of Department, Information Science and Engineering, VVCE, for his profound alacrity in our progress and his constant guidance and encouragement.

We also thank our project guide **Prof. Prema N S** Associate Professor, Information Science and Engineering, VVCE, for her immense support and help.

We also thank the project coordinator **Prof. Vinutha D C** Associate Professor, Information Science and Engineering, VVCE, for her support and encouragement.

At the end, we are anxious to offer our sincere thanks to our family members and friends for their suggestions, encouragement and support.

VYSHAK G

HARSHITH B

PAVITHRA R

VINUTHA D

CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

Malaria management is a challenging problem all over the globe particularly in Asian and African continents. Presently, even 110 years after the Nobel Prize of Ronald Ross for his work on malaria, people in the European region are also at risk from diseases carried by vectors both within the region and when traveling abroad. While treatment of malaria itself is a challenging problem its quick detection is also a problem with no less significance. There are mainly four species of malaria parasites infecting human beings namely, Plasmodium falciparum, Plasmodium vivax, Plasmodium ovale and Plasmodium malaria. Plasmodium vivax, is found mainly in tropical and subtropical areas and has a severe clinical manifestation. Rapid detection of presence of the parasite in human blood and early institution of antimalarial drugs are the mainstay of management of the disease. WHO recommends that all cases of suspected malaria be confirmed using parasite-based diagnostic testing (either microscopy or rapid diagnostic test) before administering treatment . In the malaria detection test, microscopy based diagnosis has the central importance for species differentiation, parasite quantification, management of severe disease. Additionally, the method may be amenable to larger section of society because of its scalability and low running cost.

Two types of blood smears, thick and thin, are prepared from the blood of patients, who are clinically suspected to be suffering from malaria. The thick smear is more useful for parasite detection whereas the thin smear is particularly used for identification of malaria species. When the parasite load is low, malaria may be detected about 20 times more rapidly in thick smear than in thin smear. The methods based on which image analysis of blood smears have been made, broadly fall into two classes namely, analysis based on morphology and that based on colour. Some of the reported methods use supervised training sets and lack of availability of appropriate training sets may delimit the scalability of such methods.

1.1 Scope and Objective

We present a system where image analysis studies aiming at automated diagnosis or screening of malaria infection in microscope images of thin blood film smears. In this work, we propose a combined algorithm consisting of morphological operations and shaped based discrimination technique to identify malaria parasites from thick smear images. Evaluation of percentage of detection and False Positive Rate (FPR) shows that our proposed algorithm has significantly higher predictive rate, and lower FPR as compared to any existing methodology when tested on same input slides.

1.2 Existing system and their disadvantages

Malaria parasites can be identified by examining under the microscope a drop of the patient's blood, spread out as a "blood smear" on a microscope slide. Prior to examination, the specimen is stained (most often with the Giemsa stain) to give the parasites a distinctive appearance.

- The detection of Malaria parasites is done by pathologists manually using Microscopes. So, the chances of false detection due to human error are high, which in turn can result into fatal condition.
- It takes more time to process many samples of slide.
- In absence of pathologist, results are difficult to obtain.

1.3 Problem statement

Thick blood smear examination is a necessary part for rapid screening of malaria parasite. Primary diagnosis of malaria by thick smear examination is cheap and highly sensitive, advocated by the World Health Organization (WHO).For the examination of thick blood smear, manpower and time can be reduced by using automated computational techniques.

To provide a solution to the above problem we design a Image processing system using Matlab which processes the image to obtain a result whether a person having malaria or not.

1.4 Proposed system

The objective of the project is to develop a fully automated image classification system to positively identify malaria parasites present in thin blood smears, and differentiate the species. In this project, we for the first time, describe an unsupervised approach in which color and morphology based algorithms are put together to formulate an algorithm for Plasmodium vivax detection from thick smear slide. Our approach has a comparative higher predictability and lower false positive rate.

1.5 Advantages of proposed system

- Faster execution of multiple images.
- Results can be obtained even without the pathologists.
- Results can be obtained any time in a day.

Chapter 2

LITERATURE SURVEY

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then next steps are to determine which operating system and programming language can be used for developing the tool. Once the developer starts building the tools the programmer need lot of external support. This support can be obtained from senior programmer, from books and from website. Before building the system the above consideration are taken into account for developing the proposed system.

Title: Detection of Malarial Parasite in Blood Using Image Processing

Authors: Pallavi T. Suradkar

Abstract:

The objective of this paper is to develop a fully automated image classification system to positively identify malaria parasites present in thin blood smears, and differentiate the species. The algorithm generated will be helpful in the area where the expert in microscopic analysis may not be available. The effort of the algorithm is to detect presence of parasite at any stage. One of the parasites grows in body for 7 to 8 days without any Symptoms. So if this algorithm is incorporated in routine tests, the presence of malarial parasite can be detected Automatic parasite detection has based on colour histograms. In a diagnosis scenario In this study we have proposed a solution for the parasite detection problem with two consecutive classifications.

Title: Comparative Study of Malaria Parasite Detection using Euclidean Distance Classifier & SVM

Authors: Ms. Snehal Suryawanshi, Prof. V. V. Dixit

Abstract:

This paper presents enhanced technique for Malaria Parasite Detection, where cell segmentation process consists of various steps such as image binarization using Poisson's distribution based Minimum Error Thresholding, followed by Morphological Opening for

the purpose of refinement. Seed point localization is done by multiscale LoG filter. Since frequency and orientation representations of Gabor filter are same as that of human visual system, it is used for feature extraction. Two algorithms are compared in this paper in order to get superior classification. Results show that SVM (Support Vector machines) gives better accuracy of 93.33% than that of Euclidean Distance Classifier which is 80%.

Title: Computer Vision for Malaria Parasite Classification in Erythrocytes

Authors: Somasekar J, Reddy B, Reddy E, Lai C

Abstract:

In this is paper, a new approach to represent a mathematical modelling technique by means of linear programming as an efficient tool to solve problems related to medical imaging problems especially Malaria Diagnosis through Microscopy Imaging problems .Two applications are approached: formulation of a linear programming based on the given data and solving the given problem using graphical method approach for detecting parasite. Also we applied some image processing techniques namely image segmentation, morphological operations. In the first application, just we have to develop mathematical model from the collected information and in second approach we have to solve problem by Graphical approach. We mark region infected with malaria from the original image leads to identifying parasite and also we classified different species of malaria by using graphical approach. By observation of graph we can predict whether the blood is infected by parasite or not. We can also classify the number of species of parasite infected the erythrocytes and parasite identification by labelling the infected area.

Title: Microscopic determination of malaria parasite load: role of image analysis

Authors: Frean J

Abstract:

There are two acceptable methods of expressing the parasite load: either as the percentage of infected erythrocytes as counted on a stained thin blood film or the number of parasites per unit volume of blood. The latter is usually assessed on a stained thick film by counting parasites against leukocytes, then multiplying by either the patient's own leukocyte count if available, or a standard count. When the parasite count is either very high or very low, thick film and thin film counts, respectively, are inaccurate. Ideally, therefore, both methods should be in a good malaria microscopist's repertoire. Semi

quantitation, although often used in laboratories in developing countries, is inherently imprecise and often incorrectly applied, and is less satisfactory than the other methods.

Title: Digital analysis of changes by Plasmodium vivax malaria in erythrocytes

Authors: Edison, Maombi, Jeeva, J.B. Singh, Megha

Abstract:

Blood samples of malaria patients selected based on the severity of parasitemia, were divided into low (LP), medium (MP) and high (HP) parasitemia, which represent increasing levels of the disease severity. Healthy subjects without any history of disease were selected as a control group. By processing of erythrocytes images their contours were obtained and from these the shape parameters area, perimeter and form factor were obtained. The gray level intensity was determined by scanning of erythrocyte along its largest diameter. A comparison of these with that of normal cells showed a significant change in shape parameters. The gray level intensity decreases with the increase of severity of the disease. The changes in shape parameters directly and gray level intensity variation inversely are correlated with the increase in parasite density due to the disease.

Chapter 3

SOFTWARE REQUIREMENTS SPECIFICATIONS

A **software requirements specification** (**SRS**) is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide. Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers (in market-driven projects, these roles may be played by the marketing and development divisions) on what the software product is to do as well as what it is not expected to do. Software requirements specification permits a rigorous assessment of requirements before design can begin and reduces later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules. Used appropriately, software requirements specifications can help prevent software project failure.

The software requirements specification document enlists enough and necessary requirements that are required for the project development. To derive the requirements we need to have clear and thorough understanding of the products to be developed or being developed. This is achieved and refined with detailed and continuous communications with the project team and customer till the completion of the software.

3.1 System requirements

3.1.1 Hardware requirements

- Processor: Pentium 4 +
- RAM: 2GB
- Hard Disk: 10GB
- Speed: 1.2 GHz+

3.1.2 Software requirements

- Operating System : Windows XP or Higher
- Software : Matlab 2013a.

3.2 Non-functional requirements

Non-functional requirements are constraints that must be adhered to during development. They limit what resources can be used and set bounds on aspects of the software's quality. One of the most important things about non-functional requirements is to make them verifiable. The verification is normally done by measuring various aspects of the system and seeing if the measurements confirm to the requirements.

Non-functional requirements are divided into several groups:

- Availability
- Reliability
- Security
- Usability
- Performance

- Availability

  Availability, in the context of a computer system, refers to the ability of a user to access information or resources in a specified location and in the correct format.

  Our application is available to the customer anytime he wishes to use the system.

- Reliability

  The ability of an apparatus, machine or system to consistently perform its intended or required function or mission on demand and without degradation or failure is called reliability.

  Our software is reliable because it secures all the information.

- Security

  The ability of an apparatus, machine or system to consistently perform its intended or required function or mission on demand and without degradation or failure is called security.

  Information such as images for each user is maintained securely and hence our application is secured.

- Usability

  Usability is making products and systems easier to use, and matching them more closely to user needs and requirements.

  The proposed software can be used by any user without any detailed knowledge above the project. Thus we can say that the proposed software is useable.

- Performance

  The accomplishment of a given task measured against present known standards of accuracy, completeness, cost, and speed. In a contract, performance is deemed to be the fulfilment of an obligation, in a manner that releases the performer from all liabilities under the contract.

  Our software captures the input images and gives the output in a matter of few seconds, thus our software performs efficiently.

3.3 Technology Specification

Matlab

MATLAB can be used for a range of applications, including signal processing and communications, image and video processing, control and computational biology. More than a million engineers and scientists in industry and academia use MATLAB, the language of technical computing where it has functions for integrating algorithms with external applications and languages such as C, Java and .NET.

Chapter 4

SYSTEM DESIGN

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a **system** to satisfy specified requirements. System design consists of use case diagrams, data flow diagrams and sequence diagrams.

4.1 Proposed Architecture

Architecture Diagram is a graphical representation of the concepts, their principles, elements and components that are part of an architecture. For system developers, they need system architecture diagrams to understand, clarify, and communicate ideas about the system structure and the user requirements that the system must support. It's a basic framework can be used at the system planning phase helping partners understand the architecture, discuss changes, and communicate intentions clearly.

The general diagram of system has been shown in Fig. 4.1.

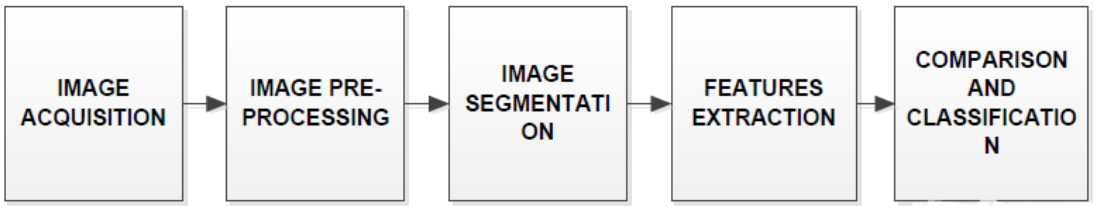


Fig 4.1 System Architecture

As it can be seen, the image received from user is processed followed by segmentation and feature extraction and finally the results are compared and classified.

4.2 Use case diagrams

A use case diagram at its simplest is a graphical representation of a user's interaction with the system and depicting the specifications of a use case and the relationship between the user and the different use cases in which the user is involved. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction

with the textual use case and will often be accompanied by other types of diagrams as well.

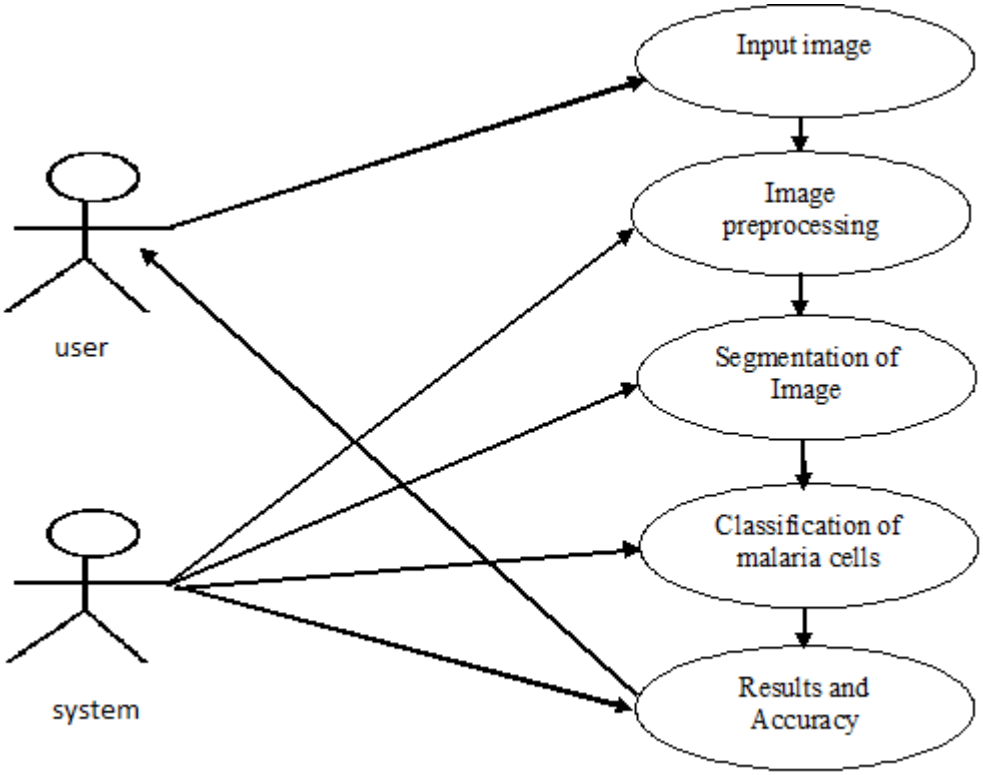


Fig 4.2 Use Case Diagram for Proposed System

In our proposed system sequence diagram there are two actors i.e. the user and system were the image is accepted by the system and it is further preprocessed, segmented and classified to give the results.

4.3 Sequence diagram

A sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams, event scenarios. A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal

arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

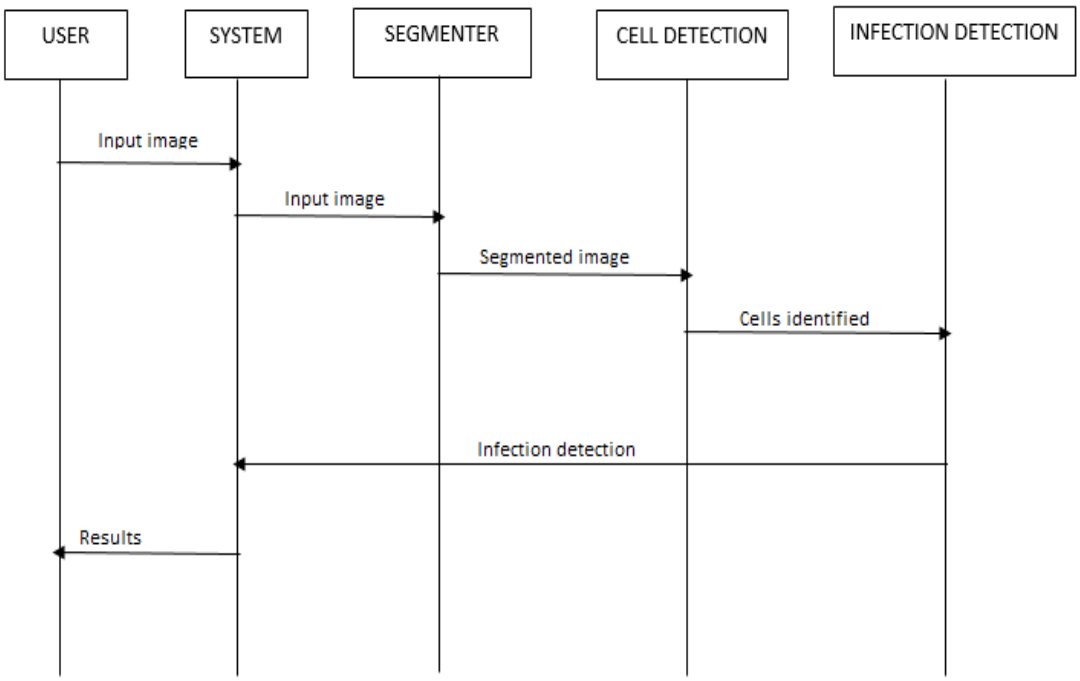


Fig 4.3 Sequence Diagram of Proposed System

Chapter 5

IMPLEMENTATION

Here the proposed system includes two algorithms

1. Morphological Operation.
2. Colour based discrimination.

5.1 Algorithm1: morphological operation

Morphology is a broad set of image processing operations that process images based on shapes. Morphological operations apply a structuring element to an input image, creating an output image of the same size. In a morphological operation, the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbours. By choosing the size and shape of the neighbourhood, you can construct a morphological operation that is sensitive to specific shapes in the input image.

The most basic morphological operations are dilation and erosion. Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries. The number of pixels added or removed from the objects in an image depends on the size and shape of the structuring element used to process the image. In the morphological dilation and erosion operations, the state of any given pixel in the output image is determined by applying a rule to the corresponding pixel and its neighbours in the input image. The rule used to process the pixels defines the operation as a dilation or an erosion.

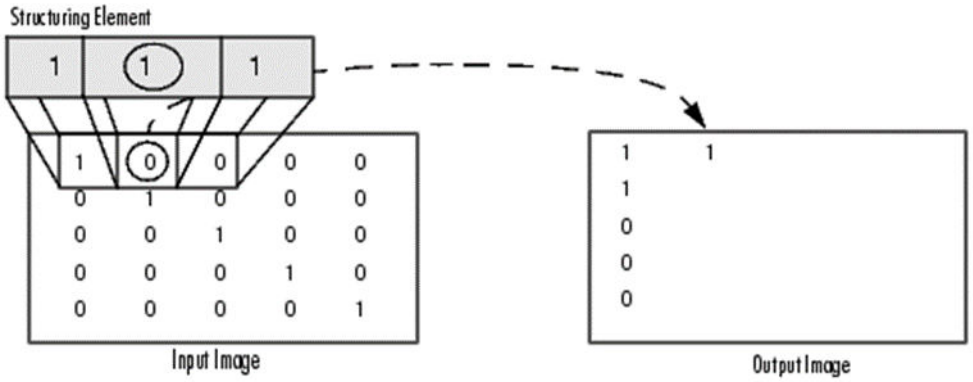


Fig 5.1 Morphological operation

The above figure illustrates this processing for a grayscale image. The figure shows the processing of a particular pixel in the input image. Note how the function applies the rule to the input pixel's neighbourhood and uses the highest value of all the pixels in the neighbourhood as the value of the corresponding pixel in the output image.

5.1.1 Pre-processing of image

Pre-processing of image consists of converting the original image into grayscale image that can be done by using an inbuilt function **rgb2gray()**. RGB to gray conversion is done by averaging all the three components i.e. R, G and B which results in gray scale

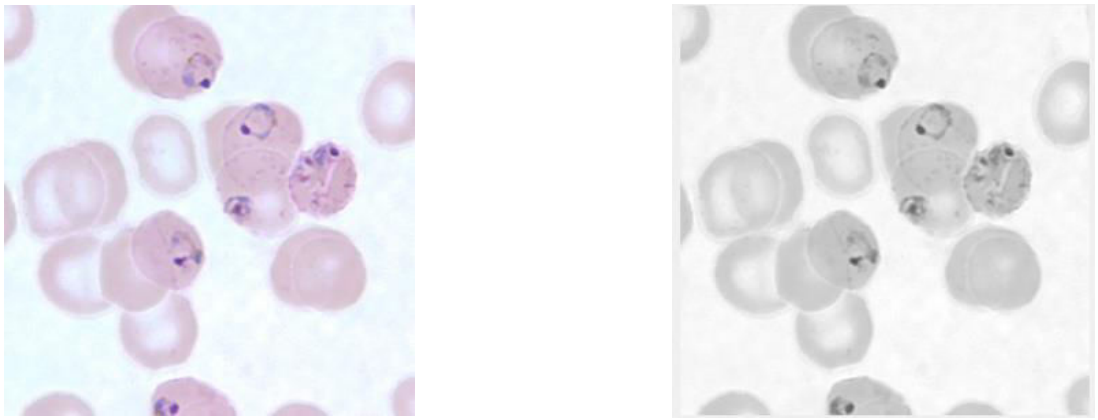


Fig 5.2 RGB to grayscale conversion

After the conversion of grayscale image we apply inbuilt function **edge()** that detect edges using the Canny method. The Canny method finds edges by looking for local maxima of the gradient of I. The edge function calculates the gradient using the derivative of a Gaussian filter. This method uses two thresholds to detect strong and weak edges, including weak edges in the output if they are connected to strong edges. By using two thresholds, the Canny method is less likely than the other methods to be fooled by noise, and more likely to detect true weak edges.

5.1.2 Segmentation

Segmentation process starts with the strel() function which takes the pre-processed image as an input. A strel object represents a flat morphological structuring element, which is an essential part of morphological dilation and erosion operations.

SE = strel('disk',R,N) creates a disk-shaped structuring element, where R specifies the radius. N specifies the number of line structuring elements used to approximate the disk

shape. Morphological operations using disk approximations run much faster when the structuring element uses approximations.

After applying strel function, the output of this function is taken as input for **imdialte()** function.

IM2 = imdilate(IM,SE) dilates the grayscale, binary, or packed binary image IM, returning the dilated image, IM2. The argument SE is a structuring element object, or array of structuring element objects, returned by the strel or offset strel function.

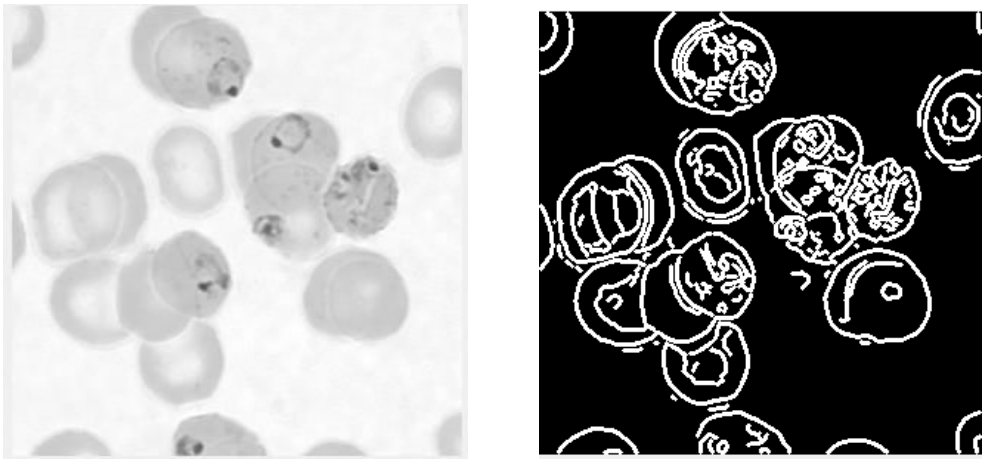


Fig 5.3 Conversion of grayscale to dilated image

Next, BW2= imfill(BW,'holes') fills holes in the input binary image BW using **imfill()**. In this syntax, a hole is a set of background pixels that cannot be reached by filling in the background from the edge of the image.

Fig 5.4 Filling the holes

Next, BW2 = bwareaopen(BW,P) removes all connected components (objects) that have fewer than P pixels from the binary image BW, producing another binary image, BW2. This operation is known as an area opening.

5.1.3 Classification

After applying bwareopen we apply regionprops, graycomatrix and graycoprops to measure properties of image regions and apply red rectangular box for each detected region using bounding box.

glcms = graycomatrix(I,Name,Value,...) returns one or more gray-level co-occurrence matrices, depending on the values of the optional name/value pairs. Parameter names can be abbreviated, and case does not matter. **stats = graycoprops(glcm,properties)** calculates the statistics specified in properties from the gray-level co-occurrence matrix glcm. glcm is an *m*-by-*n*-by-*p* array of valid gray-level co-occurrence matrices. If glcm is an array of GLCMs, stats is an array of statistics for each glcm.

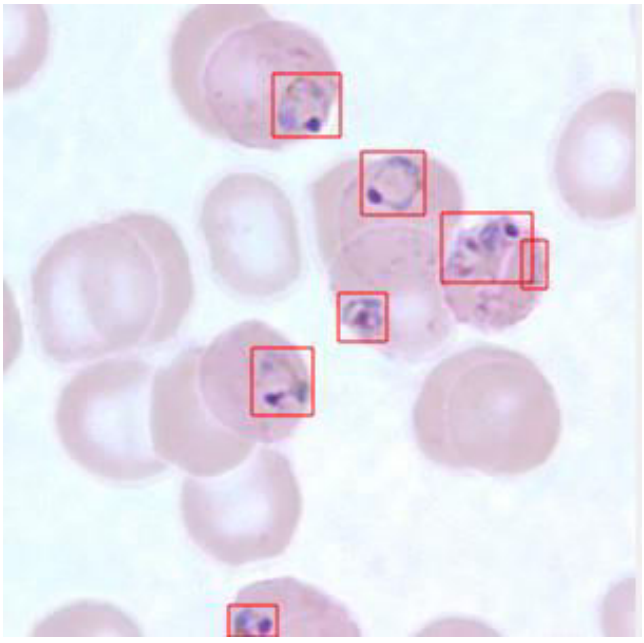


Fig 5.5 Detection of malaria infected cells

CODE:

```
[filename, pathname] =
uigetfile({'*.png';'*.bmp';'*.jpg';'*.*'}, 'pick an image
file');
```

```
s1 = imread([pathname,filename]);
imshow(s1);
handles.s1 = s1;
guidata(hobject, handles);
i= handles.s1;
r=i(:,:,1);
g=i(:,:,2);
b=i(:,:,3);
[m n]=size(r);
for i=1:m
    for j=1:n
           if r(i,j)>200
             r(i,j)=0;
               g(i,j)=0;
            b(i,j)=0;
        end
    end
end
v=cat(3,r,g,b);
bw=edge(rgb2gray(v),'canny');
se = strel('disk',1);
bw2 = imdilate(bw,se);
bw1=imfill(bw2,'holes');
bw1=bwareaopen(bw1,100);
stats=regionprops(bw1,'all');
n1=cat(1,stats.boundingbox);
[rr1 rr2]=size(n1);
togglefig('gray scale image');
imshow(bw1)
togglefig('output image');
```

```
imshow(i)

hold on

for i=1:rr1

rectangle('position',n1(i,:),'edgecolor','red');

end

for i=1:rr1

rr=imcrop(r,n1(i,:));

gg=imcrop(g,n1(i,:));

bb=imcrop(b,n1(i,:));

glcm2 = graycomatrix(rr,'offset',[2 0;0 2]);

stats1 =
graycoprops(glcm2,{'contrast','homogeneity','energy','correl
ation'});

ss1=cat(2,stats1.contrast,stats1.homogeneity,stats1.energy,s
tats1.correlation);

glcm2 = graycomatrix(gg,'offset',[2 0;0 2]);

stats2 =
graycoprops(glcm2,{'contrast','homogeneity','energy','correl
ation'});

ss2=cat(2,stats2.contrast,stats2.homogeneity,stats2.energy,s
tats2.correlation);

glcm2 = graycomatrix(bb,'offset',[2 0;0 2]);

stats3 =
graycoprops(glcm2,{'contrast','homogeneity','energy','correl
ation'});

ss3=cat(2,stats3.contrast,stats3.homogeneity,stats3.energy,s
tats3.correlation);

  rectangle('position',n1(i,:),'edgecolor','red');

end

glcm2 = graycomatrix(bw1,'offset',[2 0;0 2]);

stats3 =
graycoprops(glcm2,{'contrast','homogeneity','energy','correl
ation'});

feat=cat(2,stats3.contrast,stats3.homogeneity,stats3.energy,
stats3.correlation);
```

5.2 Algorithm2: colour based discrimination

Colour based discrimination is finding out malaria cells using difference between each pixels value and finding out the accuracy of the using mathematic classifier like SVM. This process can be explained by the series of process

- Segmentation.
- Water segmentation.
- Finding number of RCBs cells in a image.
- Intensity adjustment and identification of infected malarial cells

5.2.1 Segmentation

When we "segment" an image, we distinguish the regions of interest (ROIs) from the non-ROI portion, generally creating a binary mask of what we want to qualify, quantify, track, etc. This segmentation done by an app called image segmenter which is inbuilt in matlab and able to export code.

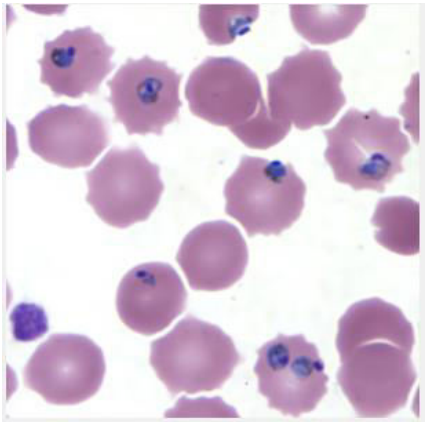
CODE:

```
% Convert to grayscale
if size(im,3) == 3
    im = rgb2gray(im);
end

% Initialize segmentation with Otsu's threshold
level = graythresh(im);
mask = im2bw(im,level);

% Filter components by area
BW = bwareafilt(mask, [920 Inf]);

% Form masked image from input image and segmented image.
maskedImage = im;
maskedImage(~BW) = 0;
BW = ~BW;
```



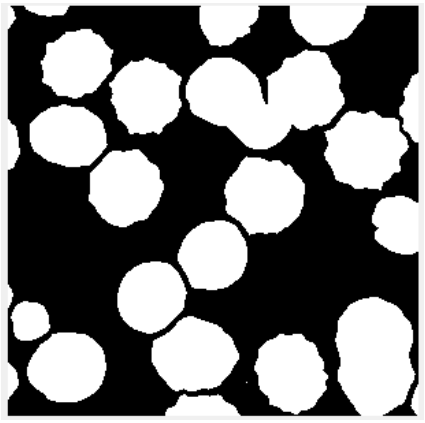


Fig 5.6 segmentation of cells from original image

You can improve the results by using in built function **bwareaopen()** on the segmentation image.

BW2 = bwareaopen(BW,P) removes all connected components (objects) that have fewer than P pixels from the binary image BW, producing another binary image, BW2. This operation is known as an area opening. But this results of segmentation is not enough so we go for water segmentation.

5.2.2 Water segmentation

a watershed is a transformation defined on a grayscale image. The name refers metaphorically to a geological watershed, or drainage divide, which separates adjacent drainage basins. The watershed transformation treats the image it operates upon like a topographic map, with the brightness of each point representing its height, and finds the lines that run along the tops of ridges.

To apply water segmentation on a grayscale image, we use inbuilt function called **watershed() .**

By applying this function we get a results as fig 5.7, but this image has lots of watershed line which is over segmenting the cells. To get a watershed line in between the each cells for proper segmentation we need to further process it using IMHMIN.

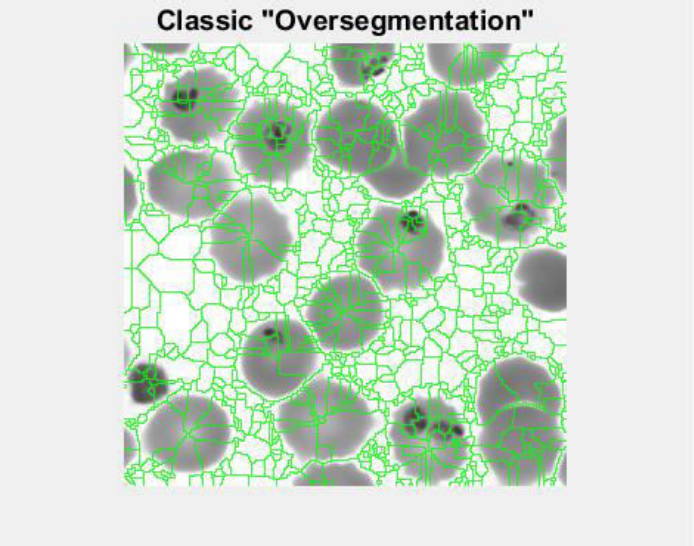


Fig 5.7 Over segmentation of watershed

Improving that result, I2 = imhmin(I,h) suppresses all minima in the intensity image I whose depth is less than h, where h is a scalar. Regional minima are connected components of pixels with a constant intensity value, t, whose external boundary pixels all have a value greater than t. After applying imhmin function we get a suppressed grayscale image.

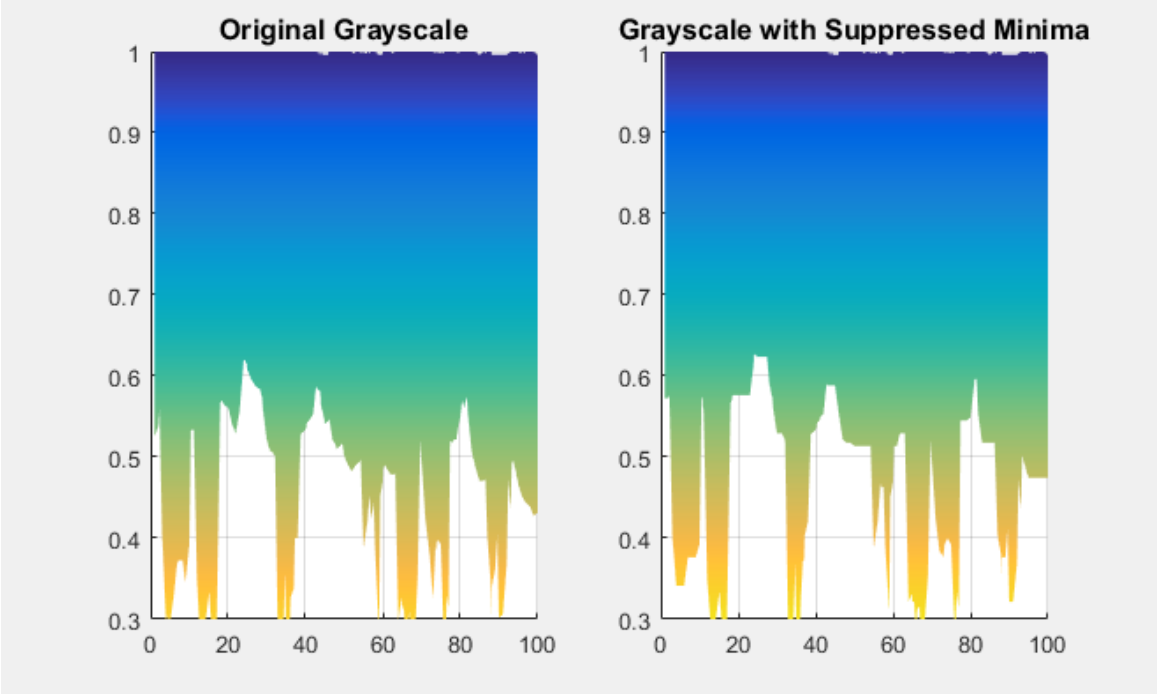


Fig 5.8 Suppressed minima in the intensity of grayscale image.

After applying suppressed minima on a grayscale image if we apply watershed we get a better segmentation of cells using water segmentation.

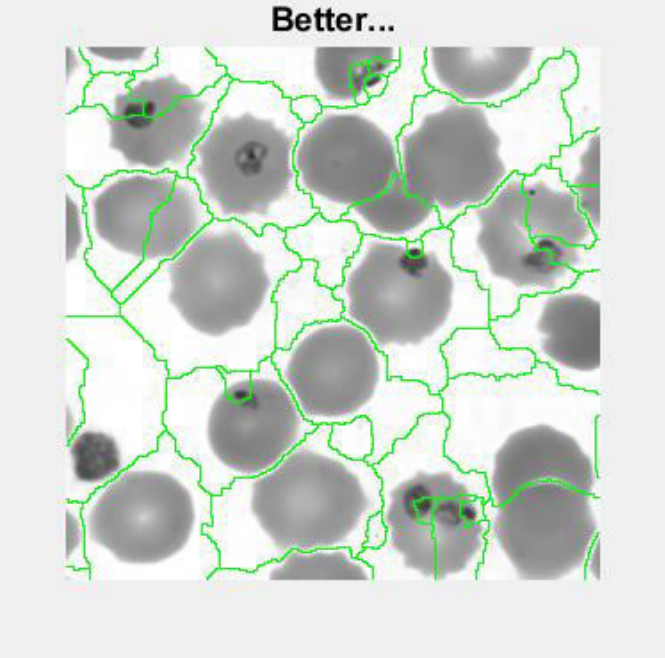


Fig 5.9 Better segmentation of wartershed.

This segmented watershed image is masked with the grayscale image to nice segmented view of cells.

CODE:

```
mask = segmentImageFcn(tmpImg);
mask = imfill(bwareaopen(mask,30),'holes');
tmpImg = imhmin(rgb2gray(tmpImg),13);
tmpImg = watershed(tmpImg);
tmpImg = tmpImg == 0;
tmpImg = bwareaopen(tmpImg,200,8);
mask(tmpImg) = 0;
```

5.2.3 Finding number of RBCs cells in a image

To find the RBC cells in a image we need to start with how questioning how many circles are there in a cells. So by finding out how many circles are there in a image we can determine the number of cells on a image. There are a lot of technique to find a circle like imfindcircles-Find circles using circular Hough transform inbuilt function. But in our project we are using an app called circleFinder.

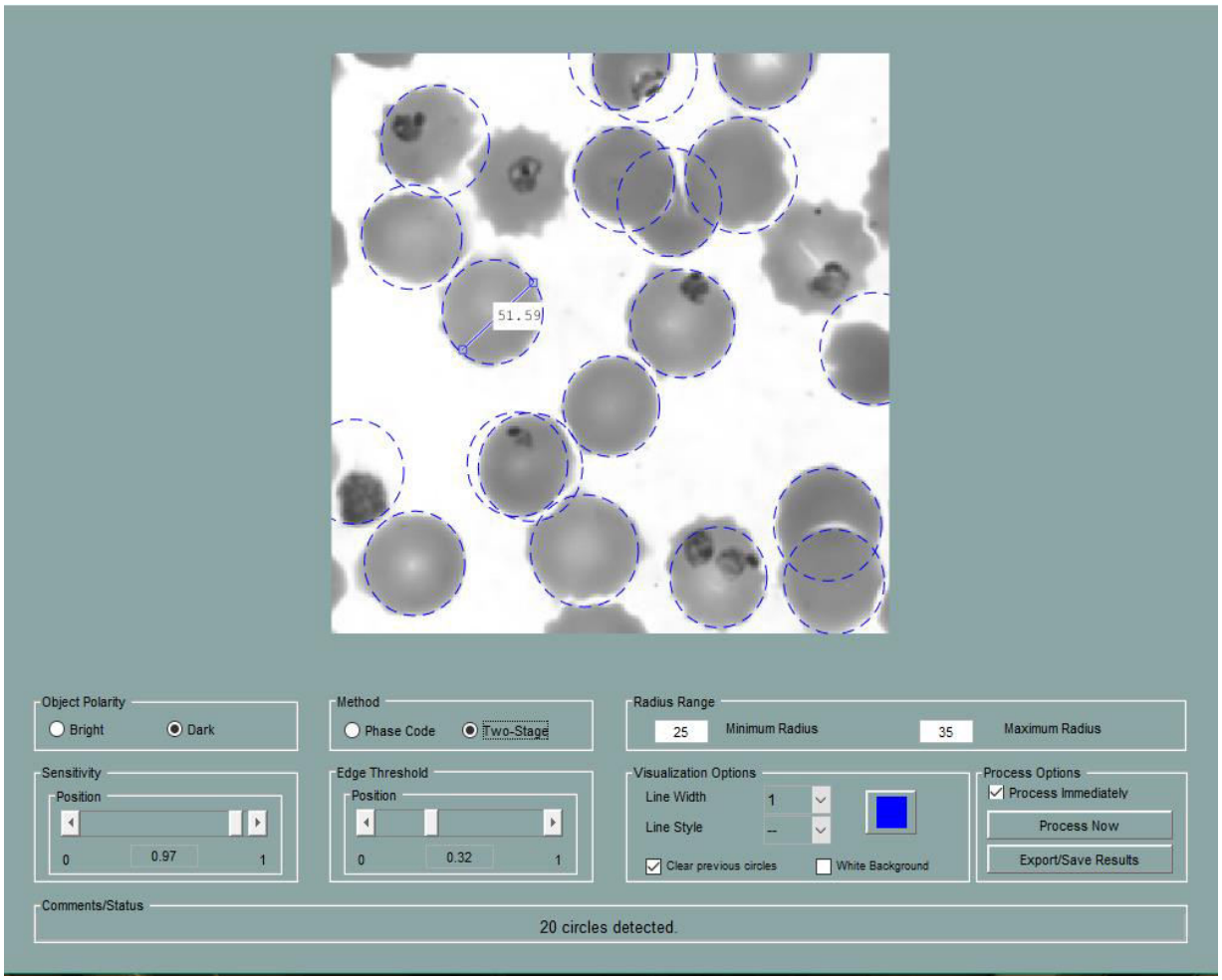


Fig 5.10 CircleFinder App in matlab.

Which can be setup as

- Object polarity which in this case is a dark i.e. dark cells on bright background.
- Radius range by measuring
- Sensitivity and method to detect a cells on a image as shown above.

After exporting a results from a circleFinder app from single image you use the same results to find the cells on all the images.

The results generated by cirlceFinder app **:**

```
detectCircles = @(x) imfindcircles(x,[20 35], ...
      'Sensitivity',0.89, ...
      'EdgeThreshold',0.04, ...
      'Method','TwoStage', ...
      'ObjectPolarity','Dark');
[centers, radii, metric] = detectCircles(grayscale);
```

When we use the exported value from cirlceFinder app in our algorithm to find the cells in a image. We get

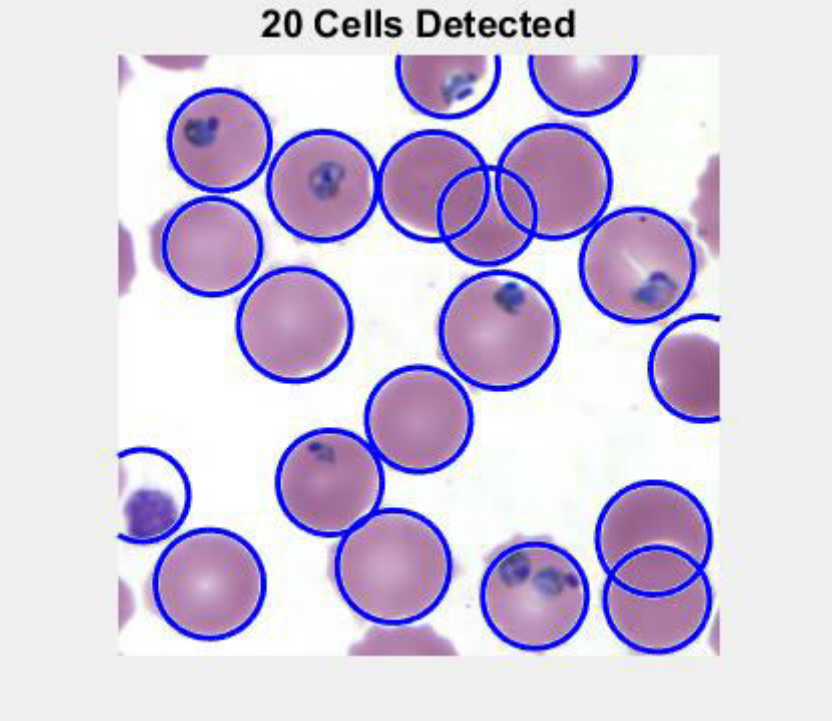


Fig 5.11 Cells detection.

5.2.4 Intensity adjustment and Identification of infected malarial cells

Before identification of infected cells on a image, we need to adjust the intensity of a image because there will be heterogeneous data that can have varying in intensity. Adjusting a image intensity helps in identifying a cells very fast and effective because lots of infected images will have darker regions which has high threshold value.

Adjusting a intensity is done by inbuilt function called imhistmatch-Adjust image to match its histogram to that of another image.

b = imhistmatch(A, REF) transforms the input grayscale or TrueColor image A so that the histogram of the output image B approximately matches the histogram of the reference image REF, when the same number of bins are used for both histograms. For TrueColor images, each color channel of A is matched independently to the corresponding color channel of REF.

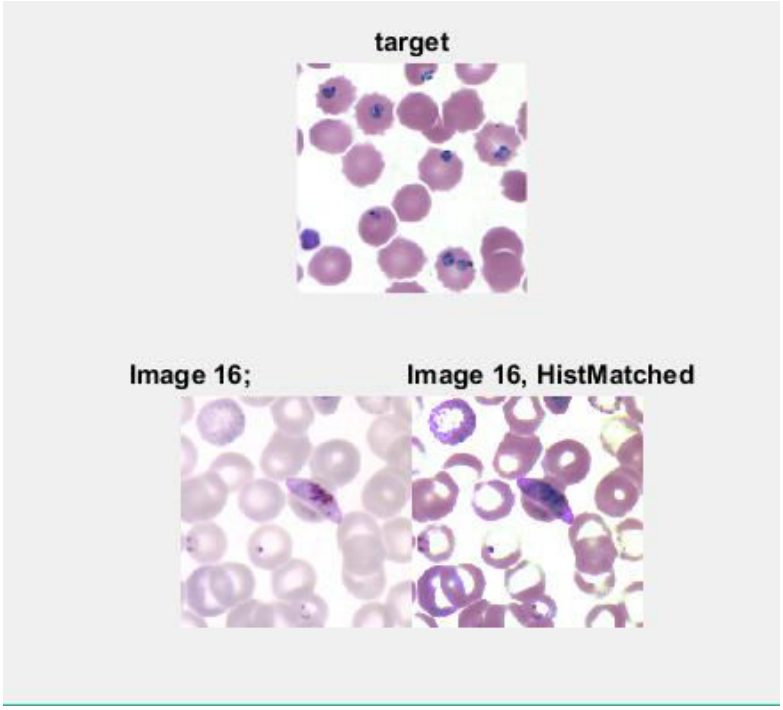


Fig 5.12 Intensity adjustment

Image 16 doesn't meet the intensity of the target image so imhistmatch is applied as shown in the fig 5.12

By adjusting intensity we can able to find the threshold value of infected cells by finding out the pixel value of the infected pixel cells. This is done by matlab app called imtool which helps in identifying each pixel value in the image.

Looking at the fig 5.13 u can tell that the infected cells have higher pixel value and other region in which we have no interest has lower pixel value from that we decide the threshold for identification of infected cells in a image.

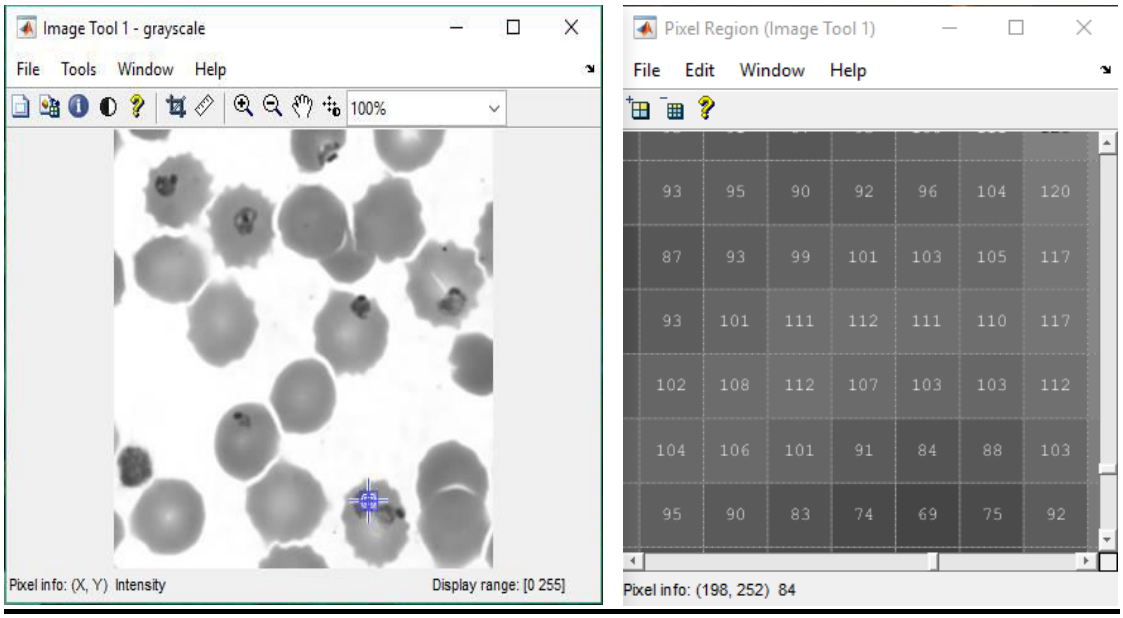


Fig 5.13 Threshold for infection using imtool

After applying every step explained as above we can able to identify a infected cell in a image using this algorithm.

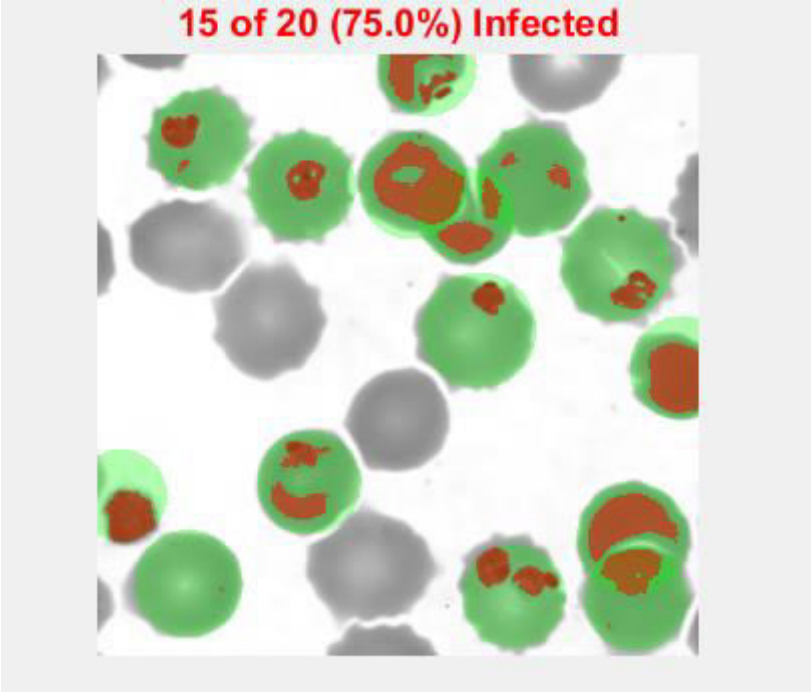


Fig 5.14 Identification of infected cells

CODE:

```
cellMask = segmentImageFcn(targetImage);
cellMask = bwareaopen(cellMask,100);
CM=cellMask;
togglefig('Cell Mask',true)
imshow(CM);
title('Cell mask','fontsize',14);


%% Watershed Segmentation
grayscale = rgb2gray(targetImage);
imshow(grayscale)
wsImg = watershed(grayscale);
showMaskAsOverlay(0.8, wsImg==0, 'g')
%% Improving that result:
% Consider the image as a "topography"
ds = 3;
surf(im2double(grayscale(1:ds:end,1:ds:end)));
shading interp;
rotate3d on
set(gca,'view',[0 90],...
    'xlim',[0 100],'ylim',[0 100],'zlim',[0.3 1])
title('Original Grayscale','fontsize',12)
% IMHMIN:
newGrayscale = imhmin(grayscale,13);
% ax3(2) = subplot(1,2,2);
surf(im2double(newGrayscale(1:ds:end,1:ds:end)));
shading interp
rotate3d on
set(gca,'view',[0 90],...
   'xlim',[0 100],'ylim',[0 100],'zlim',[0.3 1])
title('Grayscale with Suppressed Minima','fontsize',12)
linkprop(ax3,'view');
colormap(flipud(parula));

%%  Better segementation
togglefig('Exploration')
newGrayscale = imhmin(grayscale,13);
wsImg = watershed(newGrayscale);
showMaskAsOverlay(1,wsImg == 0, 'g');
title('Better...','fontsize',14);

%% Impose watershed lines
% regions along watershed lines
 togglefig('Cell Mask',true)
wsEdges = wsImg == 0;
wsEdges = bwareaopen(wsEdges,200,8);
cellMask(wsEdges) = 0;
 imshow(cellMask);

% circleFinder(grayscale)
```

```
%%
detectCircles=@(x)imfindcircles(x,[20,35],'Sensitivity',0.89
,'EdgeThreshold',0.04,'Method','TwoStage','ObjectPolarity','
Dark');

detectCircles = @(x) imfindcircles(x,[20 35], ...
    'Sensitivity',0.89, ...
    'EdgeThreshold',0.04, ...
    'Method','TwoStage', ...
    'ObjectPolarity','Dark');
[centers, radii, metric] = detectCircles(grayscale);

togglefig('Target Image',true)
imshow(targetImage)
viscircles(centers,radii,'edgecolor','b')
title(sprintf('%i Cells
Detected',numel(radii)),'fontsize',14);
 infectionThreshold=135;

 [centers,radii] = detectCircles(grayscale);
isInfected = false(numel(radii),1);
nCells = numel(isInfected);

%
x = 1:size(grayscale,2);
y = 1:size(grayscale,1);
[xx,yy] = meshgrid(x,y);
togglefig('Result mask',true);
imshow(grayscale)
infectionMask = false(size(grayscale));
for ii = 1:numel(radii)
    mask = hypot(xx - centers(ii,1), yy - centers(ii,2)) <=
radii(ii);
    currentCellImage = grayscale;
    currentCellImage(~mask) = 0;
    infection = ...
        currentCellImage > 0 & currentCellImage <
infectionThreshold;
    infectionMask = infectionMask | infection;
    isInfected(ii) = any(infection(:));
    if isInfected(ii)
        showMaskAsOverlay(0.3,mask,'g',[],false);
    end
end
showMaskAsOverlay(0.5,infectionMask,'r',[],false)
title(sprintf('%i of %i (%0.1f%%) Infected',...
    sum(isInfected),numel(isInfected),...
    100*sum(isInfected)/numel(isInfected)),...
    'fontsize',14,'color','r');
```

Chapter 6

ACCURACY OF ALGORITHMS

In many areas of information science, finding predictive relationships from data is a very important task. Initial discovery of relationships is usually done with a training set while a test set and validation set are used for evaluating whether the discovered relationships hold. More formally, a training set is a set of data used to discover potentially predictive relationships. A test set is a set of data used to assess the strength and utility of a predictive relationship. Test and training sets are used in intelligent systems, machine learning, genetic programming and statistics."ground truth" refers to the accuracy of the training set's classification. By classifying this data sets we can able determine the Accuracy of classification applied on a algorithm.

6.1 Accuracy in morphological operation

Accuracy in morphological operation is obtained by applying k-nearest neighbour classification on train set and test set of sample.

Class = knnclassify(Sample, Training, Group) classifies the rows of the data matrix Sample into groups, based on the grouping of the rows of Training. Sample and Training must be matrices with the same number of columns. Group is a vector whose distinct values define the grouping of the rows in Training. Each row of Training belongs to the group whose value is the corresponding entry of Group. knnclassify assigns each row of Sample to the group for the closest row of Training. Group can be a numeric vector, a character vector, or a cell array of character vectors. Training and Group must have the same number of rows. knnclassify treats NaNs or empty character vectors in Group as missing values, and ignores the corresponding rows of Training. Class indicates which group each row of Sample has been assigned to, and is of the same type as Group.

Code:

```
ObtainedLables=knnclassify(TestFM,TrainFM,GT_Train);

temp1=length(find((ObtainedLables-GT_Test)==0));

Accuracy=(temp1/length(GT_Test))*100;
```

The obtained results of the morphological operation u sing knnclassifer is

```
Accuracy = 57.142857

Time taken to classify 28 samples is = 6.691605e-02
```

6.2 Accuracy in colour based discrimination

To find out accuracy in colour based discrimination, we use of Machine learning. Machine learning is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of computer programs that can change when exposed to new data.

In machine learning and statistics, classification is the problem of identifying to which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known.

In machine learning, support vector machines (SVMs, also support vector networks[1]) are supervised learning models with associated learning algorithms that analyse data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

First we take trained data set and extract a features from which we can able to find a infection i.e. extraction of infected regions to check against test data set. This feature of extraction is called bag of feature.

After specifying train set of images to our algorithm which contains both clean cells and infected cells, we can call classificationLearner app from which we can apply a classification techniques like SVM and train the data set to obtain the accuracy of that classification.

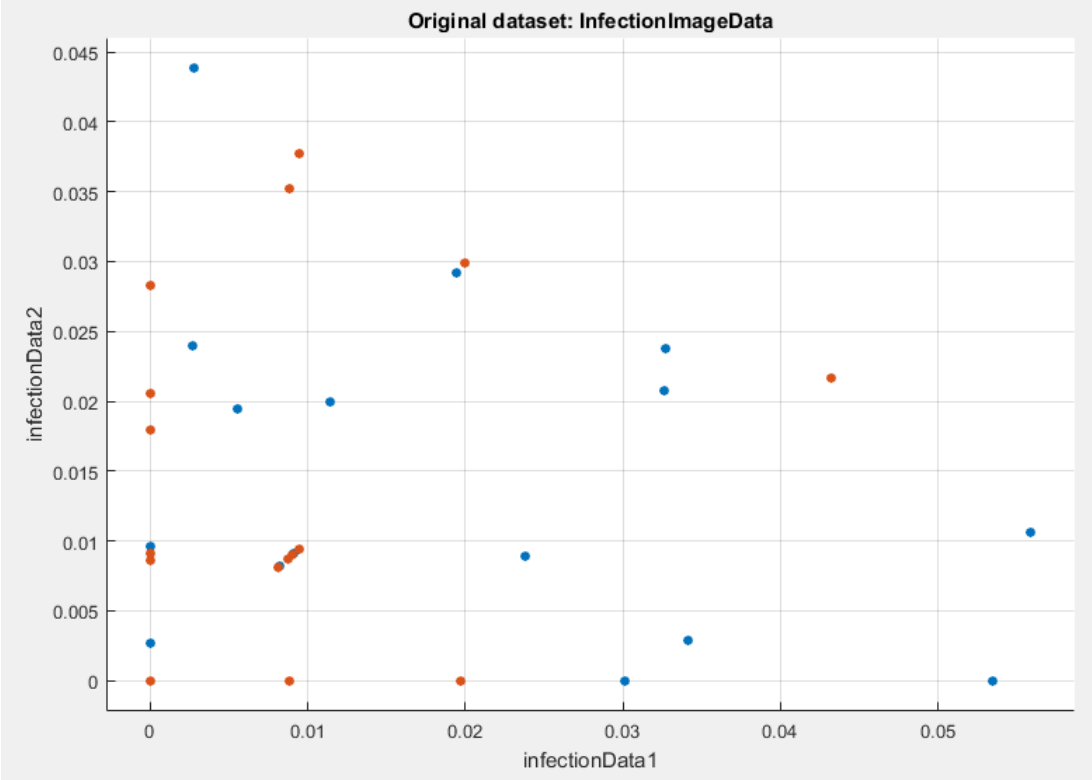


Fig 6.1 Distribution of clean and infected images in x-y plane.

Blue dots in the plane tells about the infected cells and orange dots tells about the cleacells from original data set.

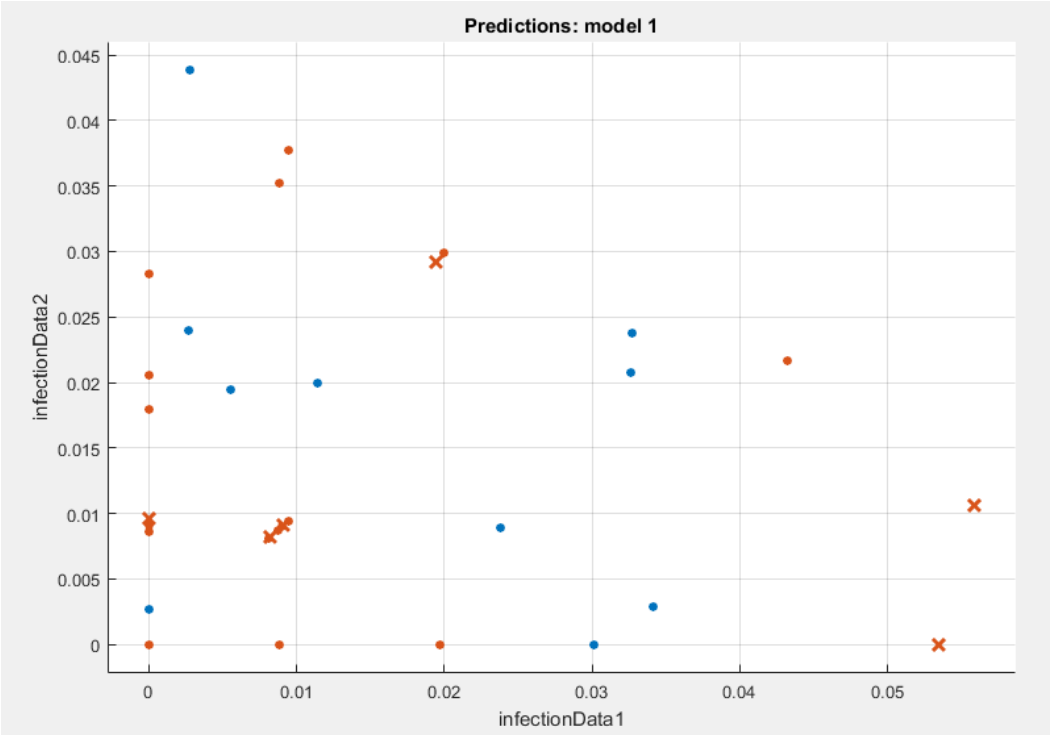


Fig 6.2 Distribuithion after applying SVM

The cross mark on the fig 6.2 tells about the wrong deterministic of reults i.e. a infected cell can be determined as cells and vice versa.

When we apply a SVM we get a results of accuracy and speed as follows:

Model number 1

```
Status: Trained
Accuracy: 82.9%
Prediction speed: ~170 obs/sec
Training Time: 0.94227 secs
Classifier Present: Medium Gaussian SVM
Kernel function: Gaussian
```

Comparative Results of our image data

Sl. No.	Method	Accuracy(in percentage)
1	Morphological operation	**57.14**
2	Colour based discrimination	**82.9**

As we can see the accuracy found in morphological operation i.e., 57.14 is significantly lower than the accuracy of colour based discrimination 82.9

CHAPTER 7

TESTING

Testing is the process of evaluating a system or its component(s) with the intent to find that whether it satisfies the specified requirements or not. This activity results in the actual, expected and difference between their results. In simple words testing is executing a system in order to identify any gaps, errors or missing requirements in contrary to the actual desire or requirements.

Testing is the practice of making objective judgments regarding the extent to which the system (device) meets, exceeds or fails to meet stated objectives.

A good testing program is a tool for the agency and the integrator/supplier; it typically identifies the end of the "Development" phase of the project, establishes the criteria for project acceptance, and establishes the start of the warranty period.

7.1 Purpose of testing

There are two fundamental purposes of testing:

- Verifying Procurement Specifications and
- Managing Risk

First, testing is about verifying that what was specified is what was delivered: it verifies that the product (system) meets the functional, performance, design, and implementation requirements identified in the procurement specifications.

Second, testing is about managing risk for both the acquiring agency and the system's vendor/developer/integrator. The testing program is used to identify when the work has been "completed" so that the contract can be closed, the vendor paid, and the system shifted by the agency into the warranty and maintenance phase of the project.

Following are some of important factors for which Testing for an application is required:

- Reduce the number of bugs in the code.
- To provide a quality product.
- To verify whether all the requirements are met.
- To satisfy the customer's needs.

- To provide a Bug free software.
- To earn the reliability of the Software.
- To avoid the user from detecting problems.
- Verify that it behaves “as specified”.
- Validate that what has been specified is what the user actually wanted.

There are different methods which can be used for Software testing. The following briefly describes some of those methods:

- **Black box testing**

The technique of testing without having any knowledge of the interior workings of the application is Black Box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, when performing a black box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

- **White box testing**

White box testing is the detailed investigation of internal logic and structure of the code. White box testing is also called glass testing or open box testing. In order to perform white box testing on an application, the tester needs to possess knowledge of the internal working of the code. The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inappropriately.

7.2 Test cases

Test Case No.	Description	Expected Result	Actual Result	Status of Execution (Pass/Fail)
1.	Uploading of image	Image accepted by system	Image uploaded successfully	Pass
2.	Applying morphological operation	Segmentation of image and detection of infected cells	Algorithm applied successfully	Pass
3.	Finding accuracy of morphological operation	Accuracy in percentage and time elapsed	Accuracy calculated successfully	Pass
4.	Applying colour based algorithm	Cell mask of a image and finding infected cells	Algorithm applied successfully	Pass
5.	Group execution of colour based algorithm	All images should be executed at once	Group execution of images executed successfully	Pass
6.	Finding accuracy of colour based operation	Accuracy in percentage using classifier	Accuracy calculated successfully	Pass

CONCLUSION AND FUTURE ENHANCEMENTS

The detection of malaria parasites is done by pathologists manually using microscopes. so, the chances of false detection due to human error are high, which in turn can result into fatal condition. this seminar curbs the human error while detecting the presence of malaria parasites in the blood sample by using image processing and automation. we achieved this goal using image segmentation smoothing processing techniques to detect malaria parasites in images acquired from giemsa stained peripheral blood samples. the system in a robust manner so that it is unaffected by the exceptional conditions and achieved high percentages of sensitivity, specificity, positive prediction and negative prediction values. and the extraction of red blood cells achieves a reliable performance and the actual classification of infected cells.

Our project focuses on detection of malaria detected cells, future work can be carried out to predict different stages of infected cells, different kinds of diseases like H1N1,Dengue etc. using machine learning techniques.

BIBLIOGRAPHY

[1] Frean J (2010) Microscopic determination of malaria parasite load: role of image analysis. Micrsocopy: Science, Technology, Applications, and Eductaion 862-866.

[2] Somasekar J, Reddy B, Reddy E, Lai C (2011) Computer vision for malaria parasite classification in erythrocytes, International Journal on Computer Science and Engineering 3: 2251-2256.

[3] Anggraini D, Nugroho AS, Pratama C, Rozi IE, Iskandar A A, et al. (2011) Automated status identification of microscopic images obtained from malaria thin blood smears. In: Electrical Engineering and Informatics (ICEEI), 2011 International Conference on, IEEE.

[4] Makkapati VV, Rao RM (2011) Ontology-based malaria parasite stage and species identification from peripheral blood smear images. Conf. Proc. IEEE Eng Med Biol Soc 2011 2011:6138-41

[5] F. B. Tek, A. G. Dempster, and I. Kale, "Computer vision for microscopy diagnosis of malaria," Malaria Journal 2009.

[6] Hirimutugoda Y, Wijayarathna G (2009) Artificial intelligence-based approach for determination of haematologic diseases. in: Image and Signal Processing, 2009. CISP'09. 2nd International Congress on, IEEE, 2009. 1-5.

[7] Yu YW, Wang JH (1999) Image segmentation based on region growing and edge detection. in: Systems, Man, and Cybernetics, 1999 IEEE International Conference on. 6: 798-803.

[8] Zack GW, Rogers WE, Latt SA (1977) Automatic measurement of sister chromatid exchange frequency. J Histochem Cytochem 25: 741-753.

[9] Acharya T, Ray AK (2005) Image processing: principles and applications, John Wiley & Sons 2005.

[10] Anyona SB, Schrier SL, Gichuki CW, Waitumbi JN (2006) Pitting of malaria parasites and spherocyte formation. Malar J. 5: 64.

APPENDIX A

SNAPSHOTS

- Home page

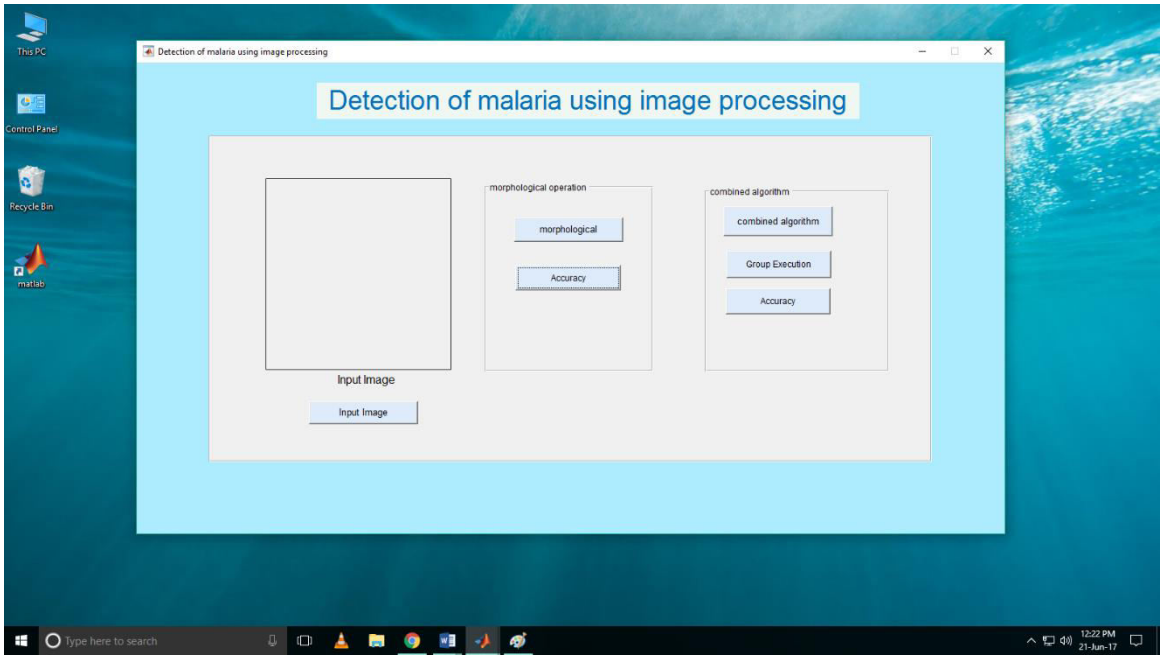


Fig A.1 Home page

- Input image selection

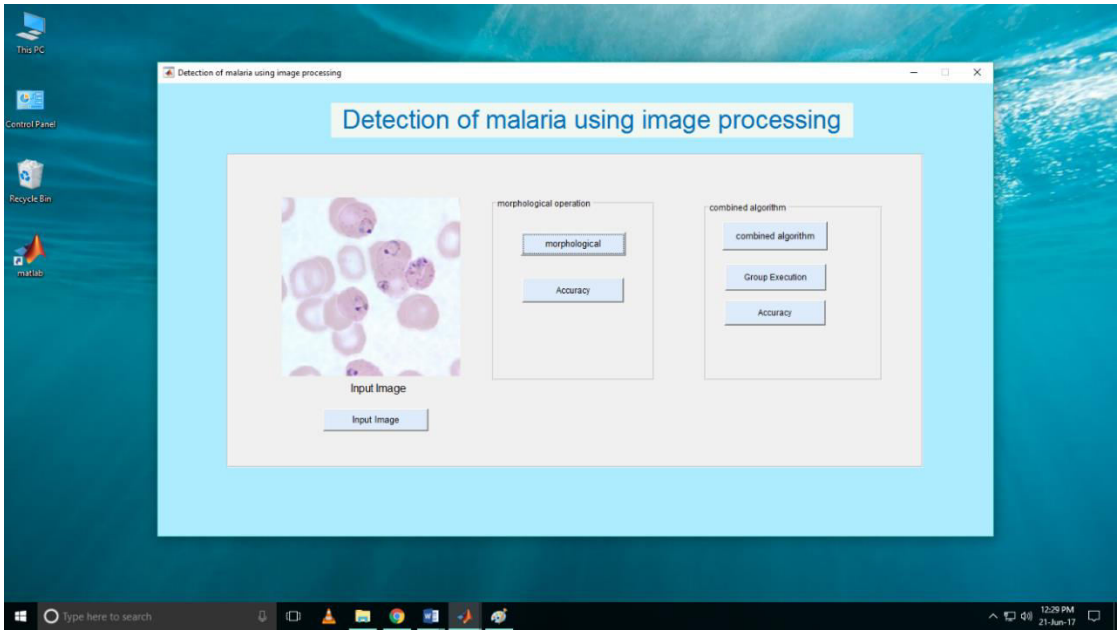


Fig A.2 Input image selection

- Morphological operation

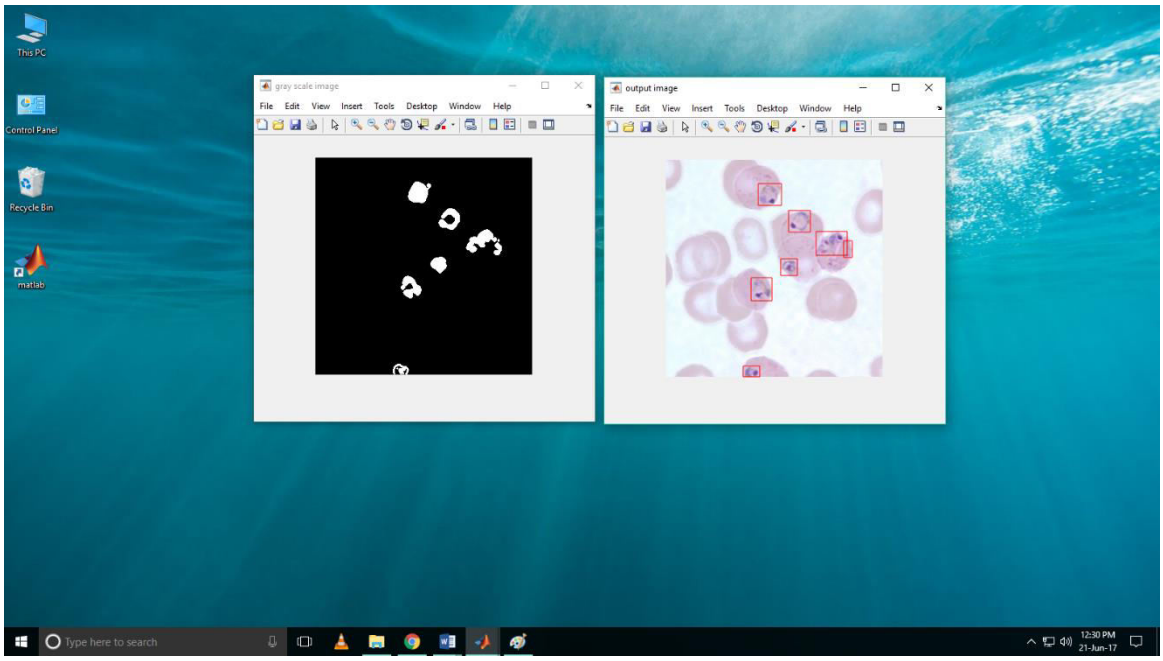


Fig A.3 Morphological operation

- Accuracy of morphological operation

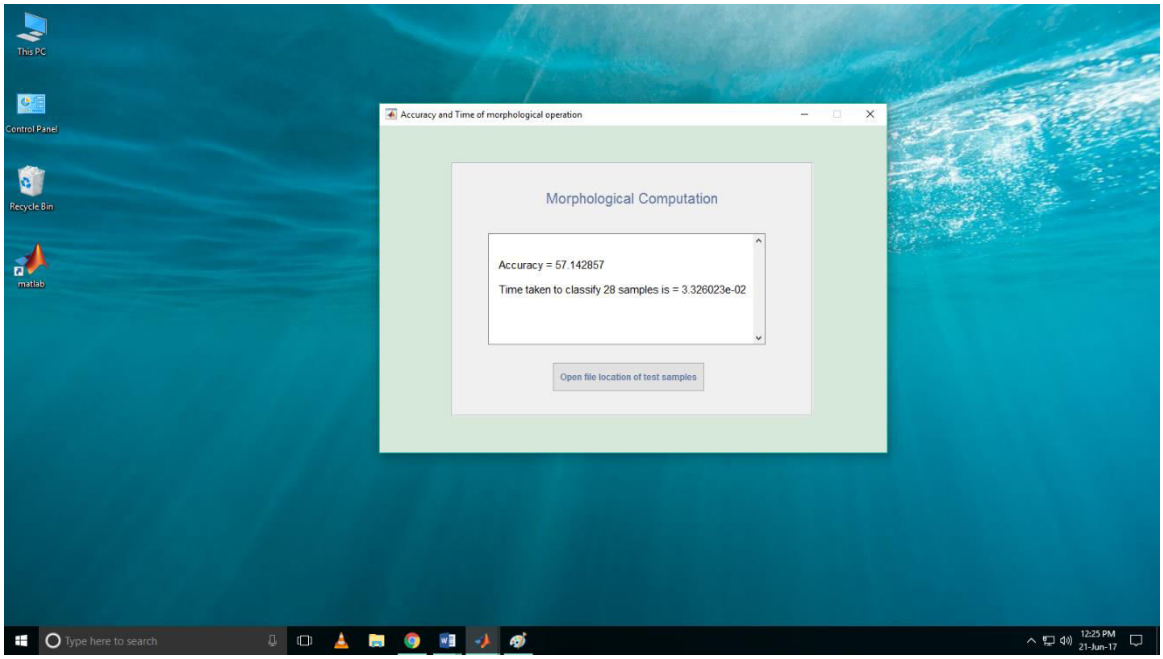


Fig A.4 Accuracy of morphological operation.

- Colour based discrimination algorithm

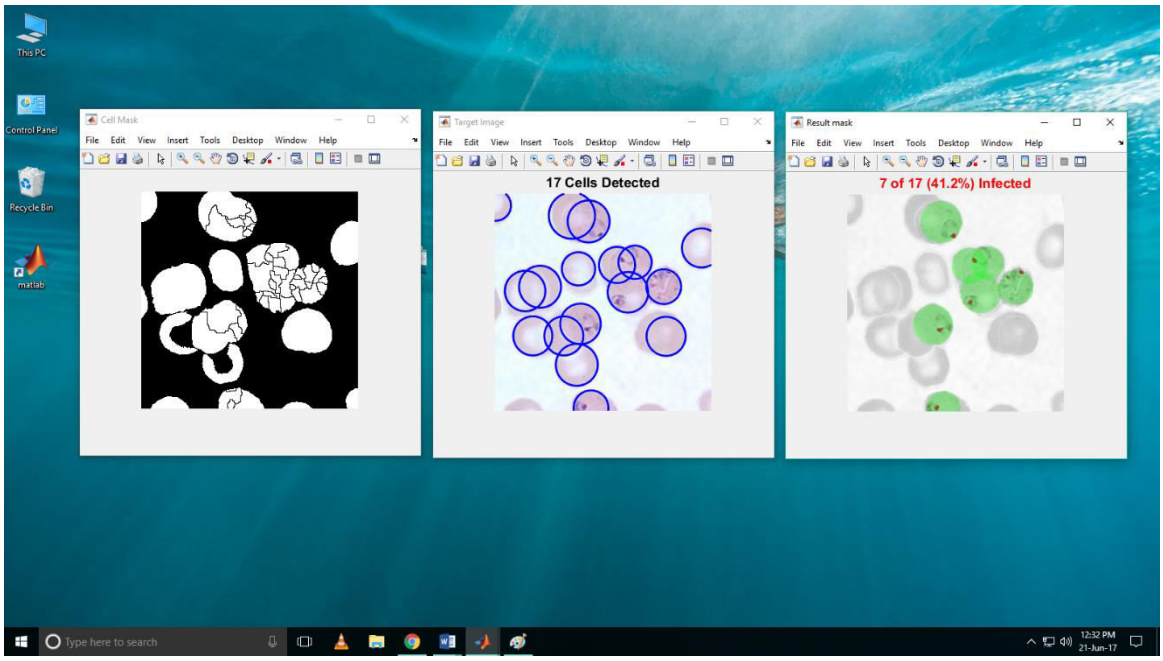


Fig A.5 colour based algorithm

- Group execution

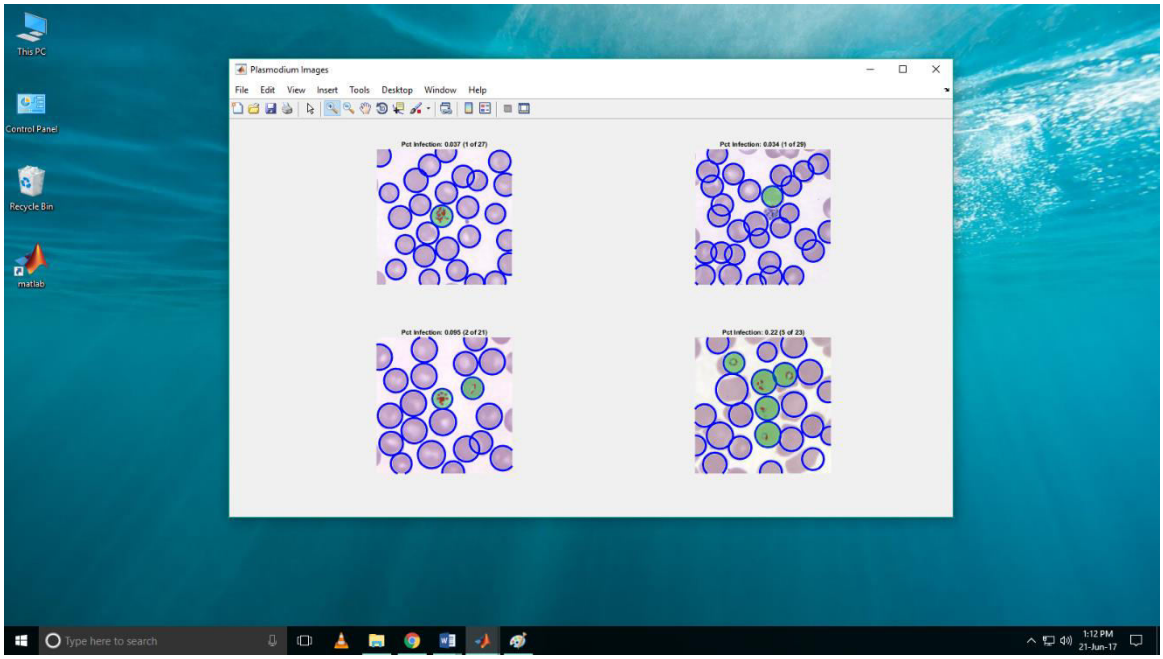


Fig A.6 multiple image execution.

- Accuracy of colour based algorithm

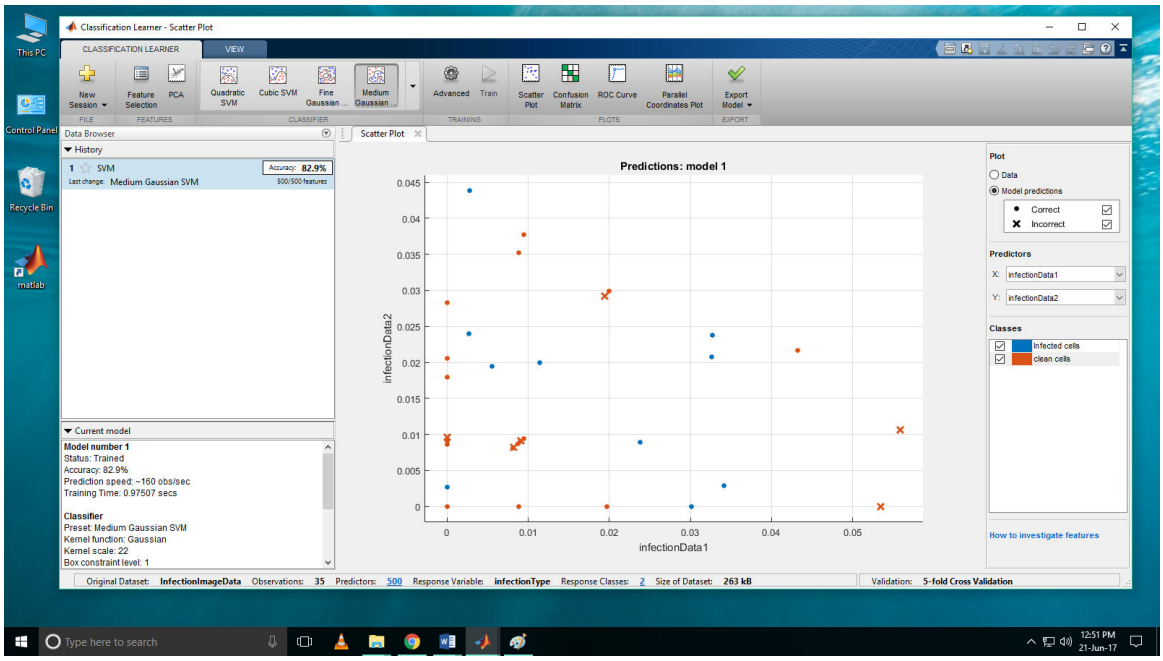


Fig A.7 Accuracy of colour based algorithm.