An Investigation of Image and Video Classification Algorithm for White Blood Cells Detection in Real Time View

by

Nabeela Bt Abu Kasim

16329

Dissertation submitted in partial fulfilment of

the requirements for the

Bachelor of Engineering (Hons)

(Electrical and Electronics)

JANUARY 2015

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

Contents

LIST OF FIGURES

ABSTRACT

Medical industry is struggling in finding the cure for many types of disease especially cancer. It is known that white blood cell is used to protect the body against bacteria and diseases. Nowadays, many ways in separating white blood cells in human body were introduced for example; centrifugation. In this report, the author is using a new approach in separating the white blood cells to help with the immune system of human body. The new approach used in separating it by using image classification algorithm to separate the white blood cells from the blood capillaries and it will be done live from a video. By separating the white blood cells, we can study the behaviour of the immune system since white blood cells is responsible for immune system in human body.

CHAPTER 1: INTRODUCTION

1.1 BACKGROUND OF STUDY

In this project, the author is dealing with microfluidic system where by humans blood capillaries are too small to be seen with naked eyes. Microfluidic systems are one of the new growing technologies which offer miniaturization of analysis systems. It is a multidisciplinary field interconnecting engineering, physics, biochemistry, and nanotechnology with sensible applications to the outline of frameworks in which little volumes of liquids will be taken care of. Microfluidics deals with the flow of liquid inside channels of micrometer size. In general, fluids are moved, mixed, split or otherwise processed into microchannel. Microchannels are channels whose dimensions are less than 1 millimetre and greater than 1 micron. Microfluidic systems have diverse and widespread potential applications for examples blood-cell-separation equipment and biochemical assays.

The human body has many types of blood cells, each with different disease-fighting roles. Focusing only on white blood cells or leukocytes, its role is to protect the body against disease and invader to the body. There are 4 types of white blood cells in human body including neutrophils, eosinophils, basophils and monocytes. White blood cells numbers as of now helped specialists diagnosed a few disease and screen treatment of others. White blood cell-separation can be done in microfluidic system since the size of human's blood capillaries is very small. This project is a new approach in sorting out white blood cells in microfluidic by creating software versions of the algorithms. The input will be either images or videos and the goal of this project is to analyze the captured image to be processed thus separating the white blood cells.

This report was divided into 5 chapters starting from Chapter 2, where the literature review is written and elaborated. While in chapter 3, the methodology in producing the result is explained. On the other hand, in chapter 4, results will be discussed, and finally, a conclusion will be made in chapter 5.

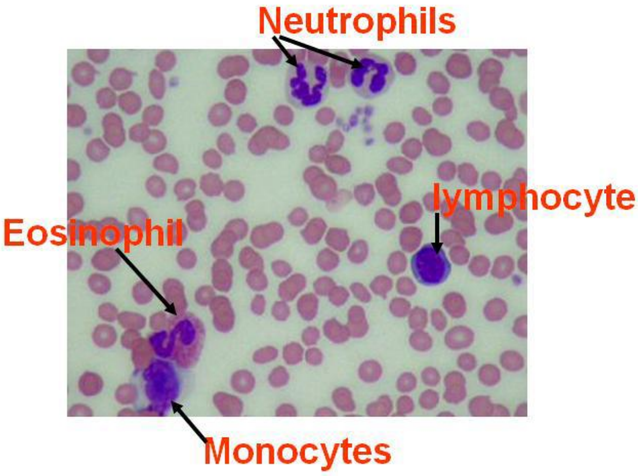


Figure 1 Different types of White Blood Cells

1.2 PROBLEM STATEMENT

Separation of white blood cell in real time view is required to explore the behaviour of immune system in a faster and more efficient way.

1.3 OBJECTIVE

1. To develop an image and video classification algorithm in real time view to detect and count white blood cells

CHAPTER 2: LITERATURE REVIEW

Analyzing the blood cells in human body is very important since the 3 types of blood cells in human body plays different and important role in leading a healthy life. By counting the number of blood cells in human body, different disease can be detected. One of the most observable different in the 3 blood cells is the size of each one of it where the white blood cells is the biggest in size followed by red blood cells and platelets [1]. Below table shows the size of each blood cell, the composition and the function of each blood type:

Type of blood	Functions	Composition	Size (um)
White blood cells	Protecting the body against disease	1%	10-20
Red blood cells	Delivering oxygen to the body tissue	40-50%	6-10
Platelets	Help blood clotting process	1%	2-4
Plasma	Transport blood, circulates dissolved nutrients and removes waste products	55%	-

Lots of researches have been done but there is no general algorithm for segmentation of white blood cells. Ongun et al. [2] proposed a programme in which it segmented the white blood cells using active shape models. In classifying the task, they apply the shape based and texture based features including the twelve classes of white blood cells. Adollah et al. [3] in their study developed an automated system in classifying the blood cells by surveying the segmentations method. Theerapattanakul et al. [4] using a different approach in segmenting the white blood cells by utilizing the active shape. Before scanning the image which is in binary to search for nucleus of each white blood cell, they doubled the thresholding and choose which white blood cell exceeds the intensity of the thresholding value. They then assign a circular shaped to be placed on the nucleus locations found. Finally, active shape model is used with gradient flow vector force as a force to drive the circular shaped contour fitting the white blood cell to be extracted.

In this project, the author decided to use image processing as the new approach in separating the white blood cells. Image processing is a signal processing that is used as an input to a process. The input can be either, pictures or images or video. The image captured will be processed by the software for example MATLAB / Simulink. There are three aspects in image processing which include image enhancement, image restoration and image segmentation. Sharpening the image, removing noise and de-blurring the image are the example of image enhancement in which it will process the image to be better for application. On the other hand, image restoration is the reversing damage to the image done by a known cause for example removal of optical distortion. Apart from that, image segmentation is dividing or isolating an image into few parts such as identifying certain shapes in the image [5].

MATLAB is the easiest software since it comes together with the image processing toolbox and library. MATLAB can be used for image processing application although it was primarily designed to manipulate matrices. In MATLAB, image is being considered as matrices so that the designer can develop optimized matrix operation by using an algorithm. Nevertheless, it also depends on the goal of the project. If the goal is hardware device, the algorithm written will operate in the same way as the proposed hardware system which will result in a slower algorithm [6]. Since almost all Images Processing Toolbox in MATLAB is represented by M-file, this will ease the user in understanding the sophisticated and image processing implementation [7].

CHAPTER 3: METHODOLOGY

In this paper, the author tried to apply 2 approaches and choose one of those two to be implemented into the real time view. The first approach is just by using the radius and size of the white blood cell while the second one is by using image segmentation that will segment the white blood cells from the nucleus inside the white blood cell.

The first approach is when the author tries changing the radius of the white blood cells and by increasing the sensitivity of the detection. Apart from that, the author also uses image processing toolbox in MATLAB that will detect the shape of the cells. As known, white blood cells are the biggest cells in the blood stream compared to other elements in the capillaries and the shape is almost same as round shape. By applying this approach, the author can directly detect the white blood cells.

Move on to the second approach, as for now, the available techniques for image segmentation can be divided into two main groups which are [8] based on gray level (e.g.: thresholding, Edge based segmentation), based on image texture. Based on algorithm done by Madhloom et al. [9], it is modified so the noise can be reduced hence increase the accuracy. The drawback of the algorithm proposed by [8] is the accuracy is 100% depends on the initial contrast of the image. Let say the contrast of an image is being edited, it will capture all the grey levels that are close to the white blood cells and it will also be selected as one of the white blood cells using the test image.

Thus, the author proposed to use the nucleus in the white blood cells while eliminating the non nucleus objects. It is chosen by 50% of the average red blood cells size. The steps taken in segmenting the white blood cells is explained below:

1. Convert the image into grey scale image and named it as B.
2. Adjust Image B’s intensity value with a linear contrast to get image C.
3. Increase the contrast of image C by applying the histogram equalization to get image D.
4. Produce image X which is equal to C + D
5. Produce image Y which is equal to C – D
6. Produce image Z by adding both image and Y
7. Filter 3 times image Z
8. Use Otsu’s method, calculate the global threshold value
9. Convert image Z into binary image

10. Use morphological opening in eliminating all the false objects.
11. Select the objects that are more than 50% of the average red blood cells by applying the size test.

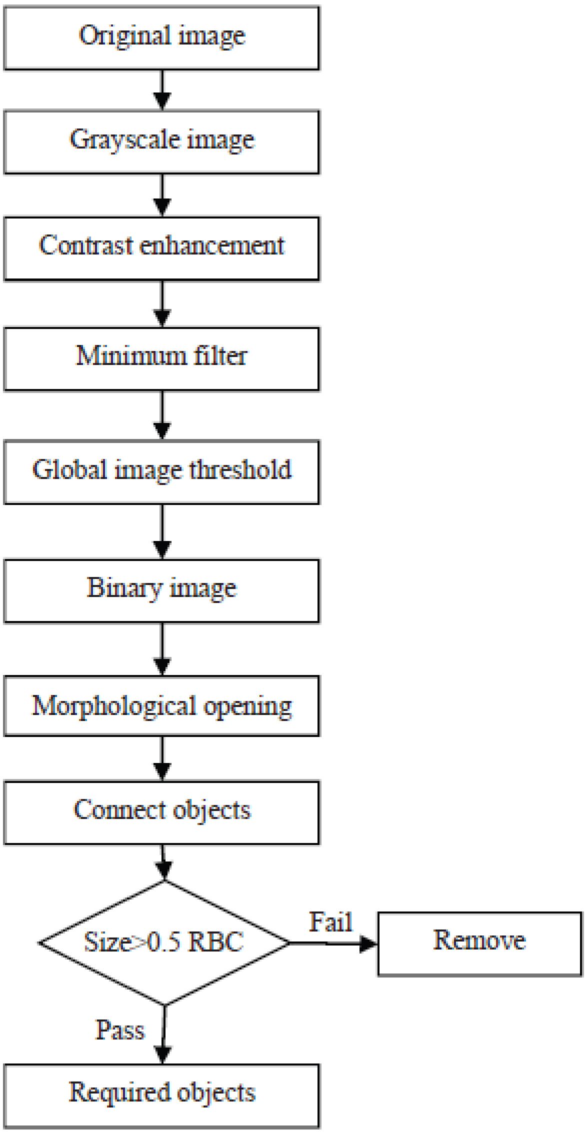


Figure 2 Simplifiedfor segmenting white blood cells

The author used MATLAB software to process the image as the input. Image segmentation is applied to this project since the shape that the author wants to detect is a round shape with certain radius. Once the shape is confirmed, it will be segmented.

From these two approaches, the author concludes that the second approach where the white blood cells being segmented is the best approach in segmenting the white blood cells. This approach is then being implemented into real time view where a video is uploaded and being analyzed to detect the moving white blood cells. Since the author did not have the real video of moving elements in blood stream, the author use a video that contains of several size of balloon to try the algorithm. The video is similar to the real blood stream since the shapes of the balloons are round and the size of the balloons varies in the video.

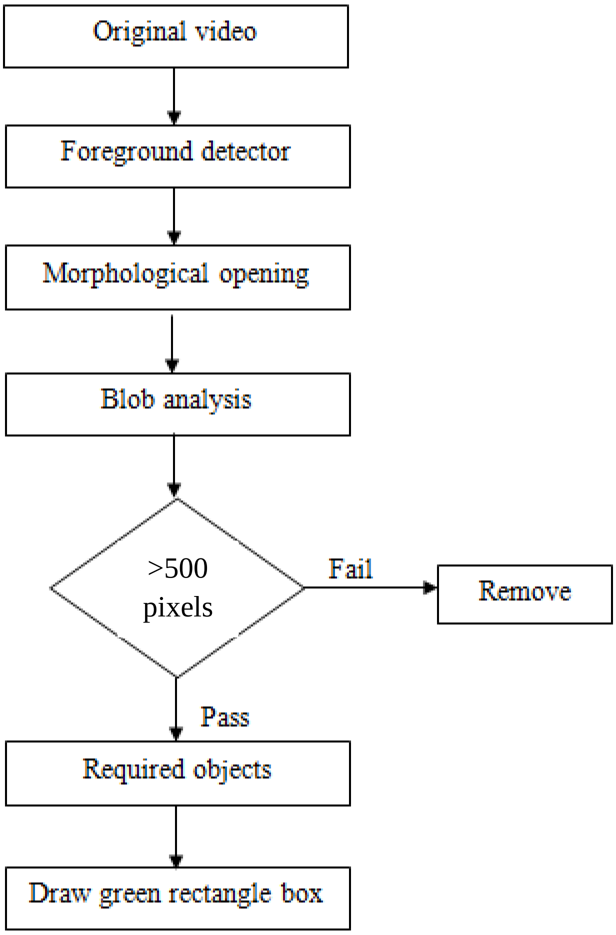


Figure 3 Simplified Real time view algorithm

The first step is to upload the video into the program in MATLAB. Once the video is uploaded, the foreground detector which can be found in image processing toolbox will be activated and start to segment the shape of the required object in the initial frame from the video. The foreground detector uses Gaussian mixture model from numbers of frame it gets from the video. Similar with image, the foreground detector may also detect some noise or unwanted element in the video and in order to remove it, the author applies morphological opening same with what the author use in image segmentation to remove the noise. Next steps are finding the continuity in the object using ‘vision.BlobAnalysis’ found in the image processing toolbox. ‘vision.BlobAnalysis’ computes statistic for connected region in binary image. An extra filter is applied to the blobs and only pixel that is more than 150 is selected others are not included. Around the required object, a green box will be drawn.

GANTT CHART AND MILESTONE

Milestone / Weeks	**1-2**	**3-4**	**5-6**	**7-8**	**9-10**	**11-12**	**13-14**
FYP 1							
Literature review	🟦						
Detection of circular object in image		🟦					
Understand the characteristic of white blood cells			🟦	🟦			
Development of white blood cells algorithm in an image				🟦			
Development of separation of white blood cell algorithm in an image					🟦	🟦	🟦
Documentation	🟦	🟦	🟦	🟦	🟦	🟦	🟦
FYP 2							
Development of segmentation of white blood cell algorithm in an image	🟦	🟦	🟦	🟦			
Development of algorithm for moving object in a video				🟦	🟦		
Development of separation of white blood cell algorithm in a video				🟦	🟦	🟦	
Documentation	🟦	🟦	🟦	🟦	🟦	🟦	🟦

CHAPTER 4: SIMULATION AND RESULTS

4.1 Approach 1: Using Radius to Detect the White Blood Cells

For simulation, MATLAB 2013 is used. There are several approaches the author tried. The first one is by using the radius of the white blood cell which is usually 50% bigger than average red blood cells. In general, the radius of white blood cells is much bigger compared to the radius of other elements in the blood capillaries. So the author uses it to write a program that can detect a circular object with the radius of white blood cells. The input of the image is shown in figure 2 while the output of the image is shown in figure 3.

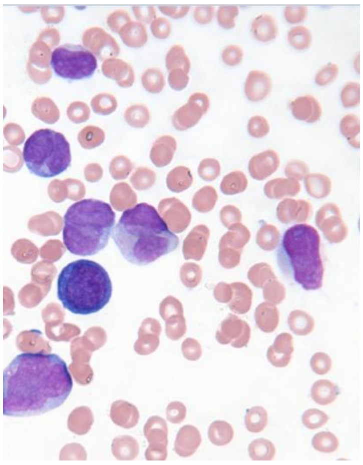
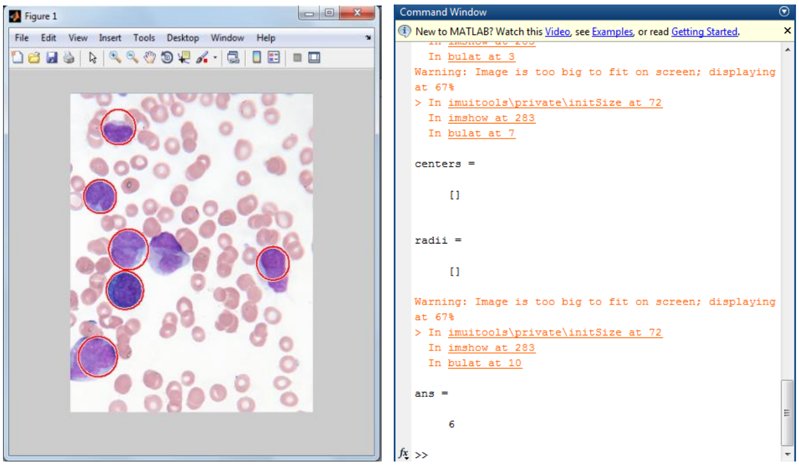


Figure 4 Input image



New to MATLAB? Watch this Video, see Examples, or read Getting Started.

```
  In imshow at 283
  In bulat at 3
Warning: Image is too big to fit on screen; displaying
at 67%
> In imuitools\private\initSize at 72
  In imshow at 283
  In bulat at 7

centers =

     []


radii =

     []

Warning: Image is too big to fit on screen; displaying
at 67%
> In imuitools\private\initSize at 72
  In imshow at 283
  In bulat at 10

ans =

     6

>>
```

Figure 5 Output image and command window

From the output image, a total of 6 out of 7 white blood cells detected. This is considered as not accurate since the other one white blood cell cannot be detected. This is due to the command use in the program where we want to detect a circular object. The reason why the other one white blood cell cannot be detected is because the shape of the white blood cell in not circular enough for the program to recognize it. This is the drawback of using the radius approach program where only the circular image can be detected.

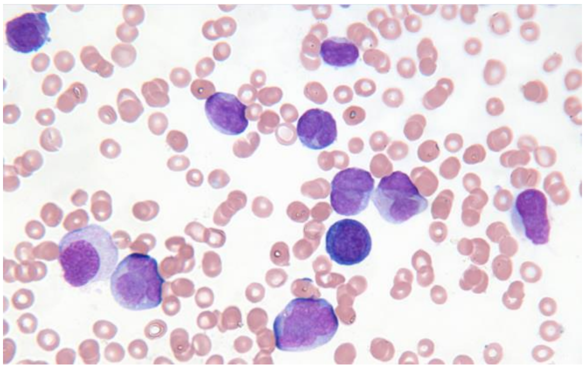


Figure 6 Original Image

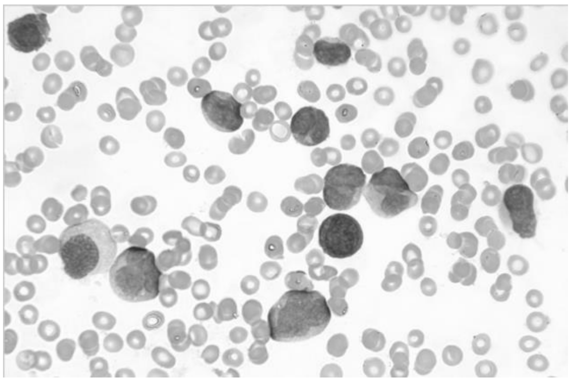


Figure 7 Image after converted to greyscale

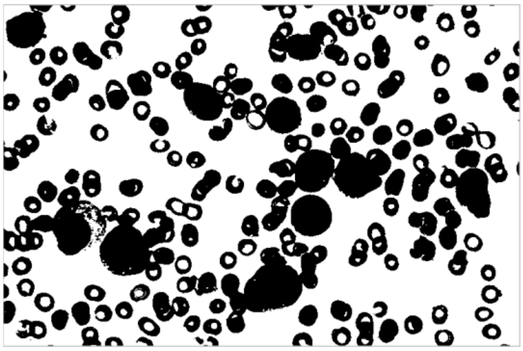


Figure 8 Image after converted to threshold

From figure 5 and 6, the image is converted to greyscale and threshold. The author is planning on detecting the white blood cells by segmentation approach where the greyscale image will turn into binary image. The most important part of this approach is in selecting the threshold value.

Apart from the method mention above, the author also tried to segment the white blood cells.

4.2 Approach 2: Using image segmentation technique to detect White Blood Cells

From the previous chapter, the author already describes the steps taken in segmenting the white blood cells while in this chapter, the result will be shown. In figure 8, the steps by steps picture were taken and displayed to show how the process works starting from the first steps, up until the last step

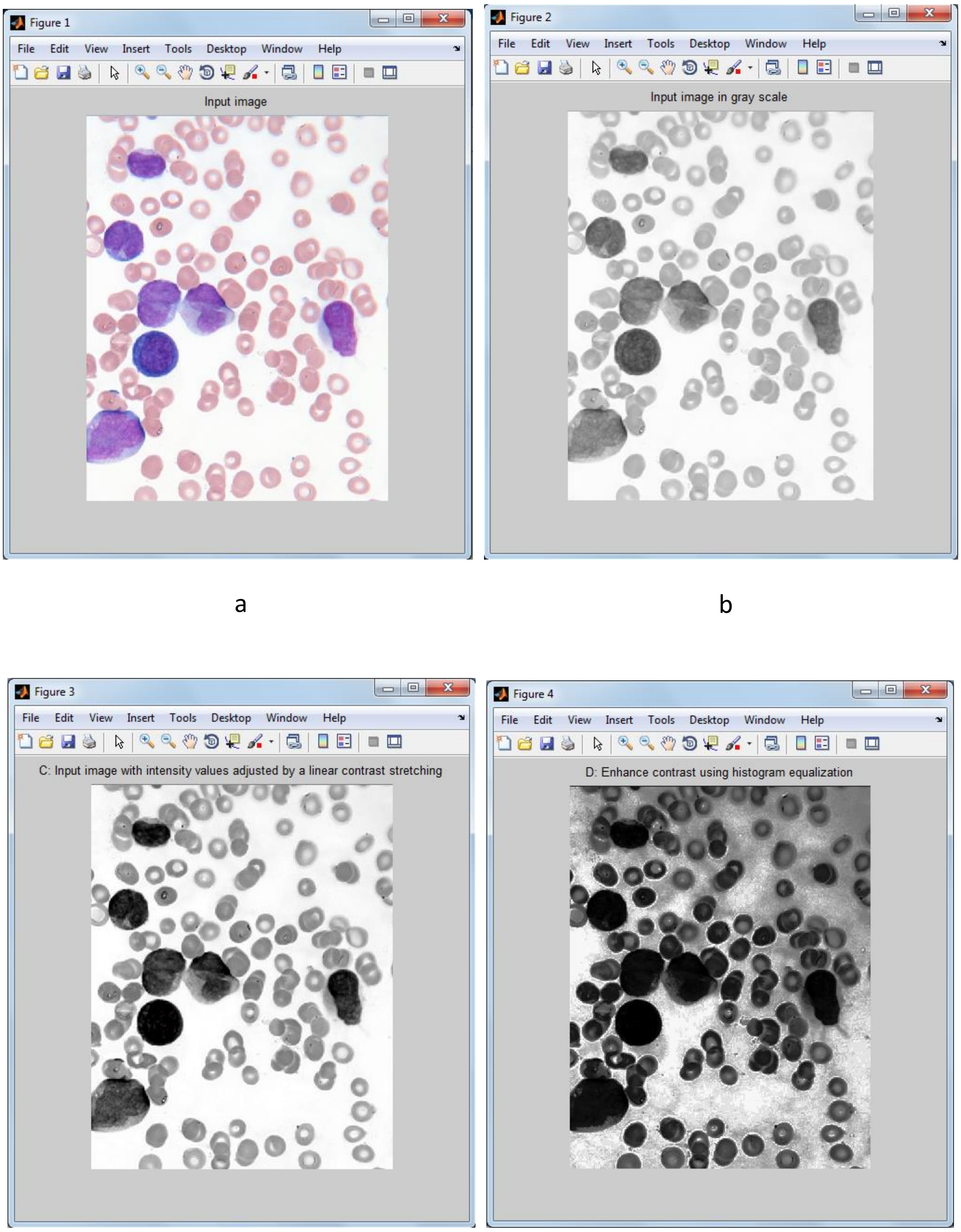


Figure 1
File Edit View Insert Tools Desktop Window Help
Input image
Figure 2
File Edit View Insert Tools Desktop Window Help
Input image in gray scale
Figure 3
File Edit View Insert Tools Desktop Window Help
C: Input image with intensity values adjusted by a linear contrast stretching
Figure 4
File Edit View Insert Tools Desktop Window Help
D: Enhance contrast using histogram equalization

a b

c d

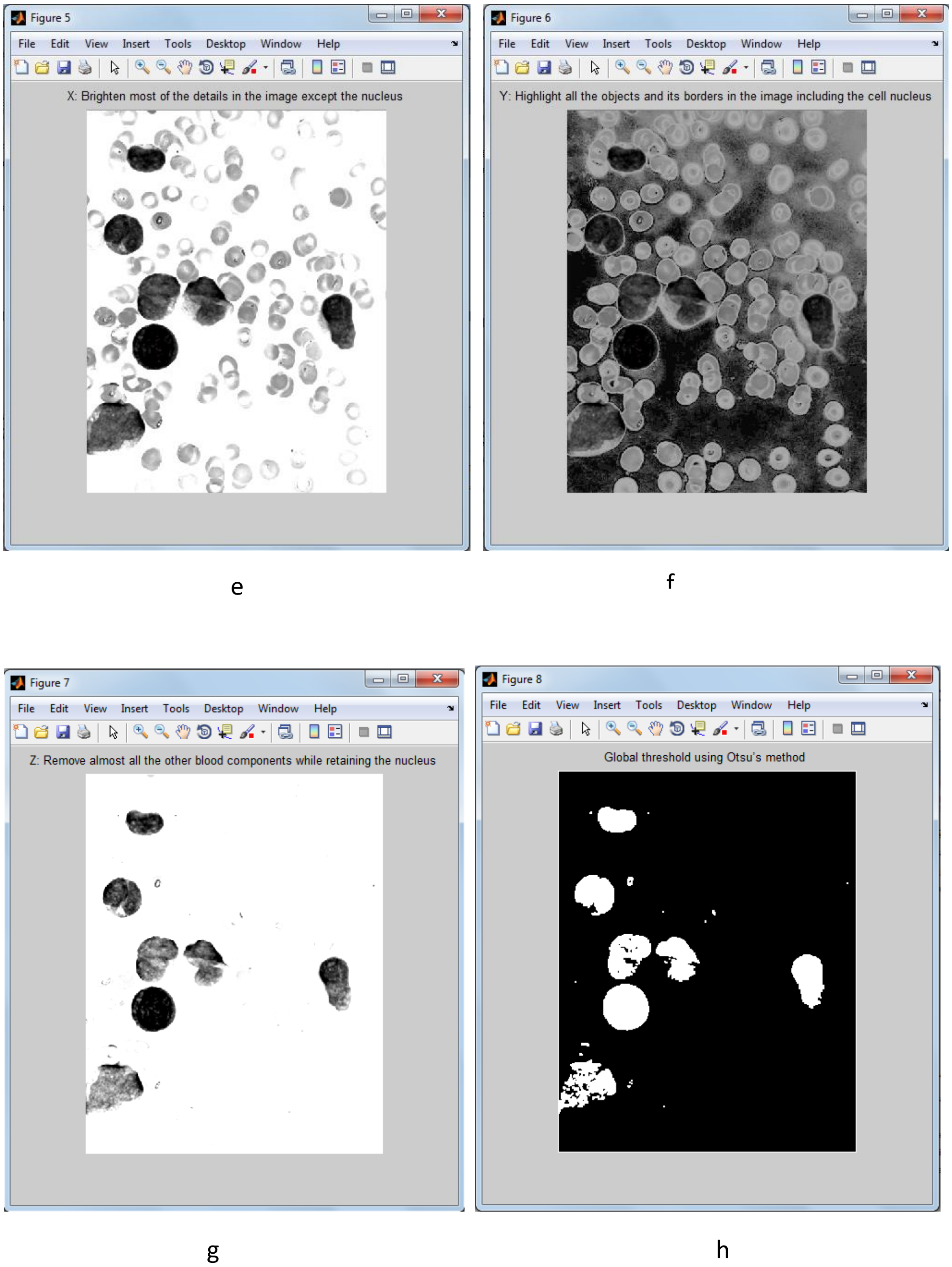


Figure 5
File Edit View Insert Tools Desktop Window Help
Figure 6
File Edit View Insert Tools Desktop Window Help
Figure 7
File Edit View Insert Tools Desktop Window Help
Figure 8
File Edit View Insert Tools Desktop Window Help

X: Brighten most of the details in the image except the nucleus

Y: Highlight all the objects and its borders in the image including the cell nucleus

e

f

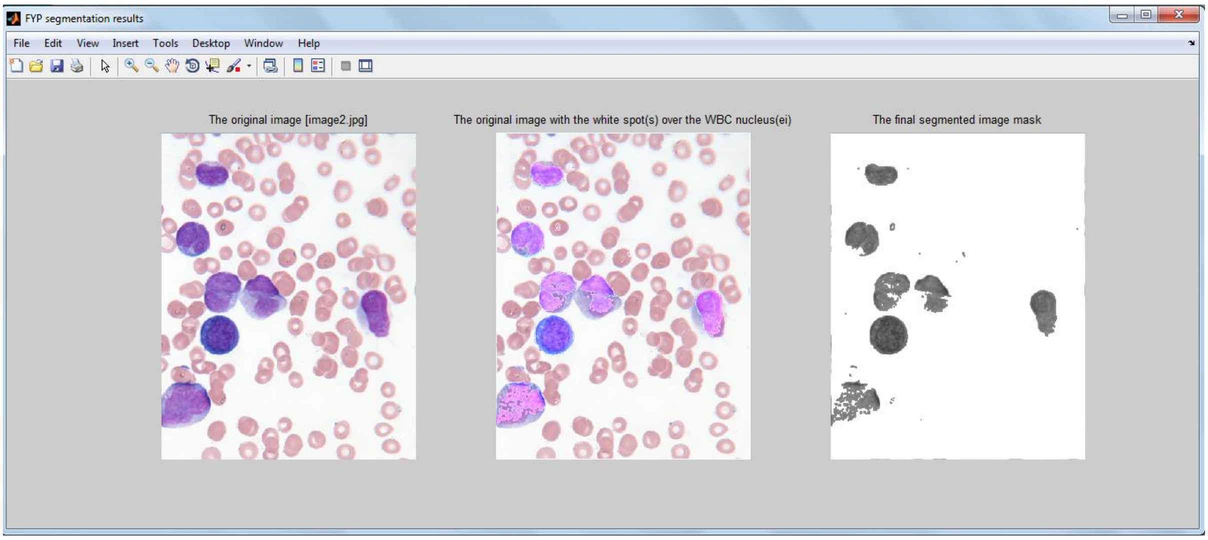
Z: Remove almost all the other blood components while retaining the nucleus

Global threshold using Otsu's method

g

h

Figure 9 a) Input image, b) Grey scale image, c) Image contrasted, d) Image contrasted using Histogram, e) Brighten Image, f) Highlighted image, g) Remove unwanted components, h) Image with Otsu's method



FYP segmentation results
File Edit View Insert Tools Desktop Window Help
The original image [image2.jpg]
The original image with the white spot(s) over the WBC nucleus(ei)
The final segmented image mask

Figure 10 The final segmented image

Figure 9a shows the original image the author use for both approaches. In the second approach, the main focus was to detect the nucleus of the white blood cells since the size is much bigger than other cells. For figure 9b, the author changed the image into greyscale. Then, the image's intensity values were adjusted by a linear contrast stretching and named as C. the following figure which is figure D is a result of enhancement of contrast using histogram equalization. The next step was to add figure C and figure D and were named as X. X is actually the image that was brighten with most of the details except the nucleus. Next figure is when image C minus image D and named as Y. Y is where the objects and its borders in the image including the cell nucleus were highlighted. Figure 9g it when all other component except the nucleus is removed from the image and the final step was using Otsu method where the global threshold is applied and only object with more than radius of red blood cells is counted.

From figure 10, the totals of segmented white blood cells are 7 which are the same of the input image. This shows that this method, segmentation of white blood cells is better compared to the first method which uses the radius of white blood cells as the approach.

4.3 Detecting White Blood cells in real time view

In detecting the white blood cells fir real time view, the steps taken is almost same with the steps taken in segmenting the white blood cells. The author try to upload an image with varies size of balloons and almost same shape with white blood cells. Figures below shows the result:

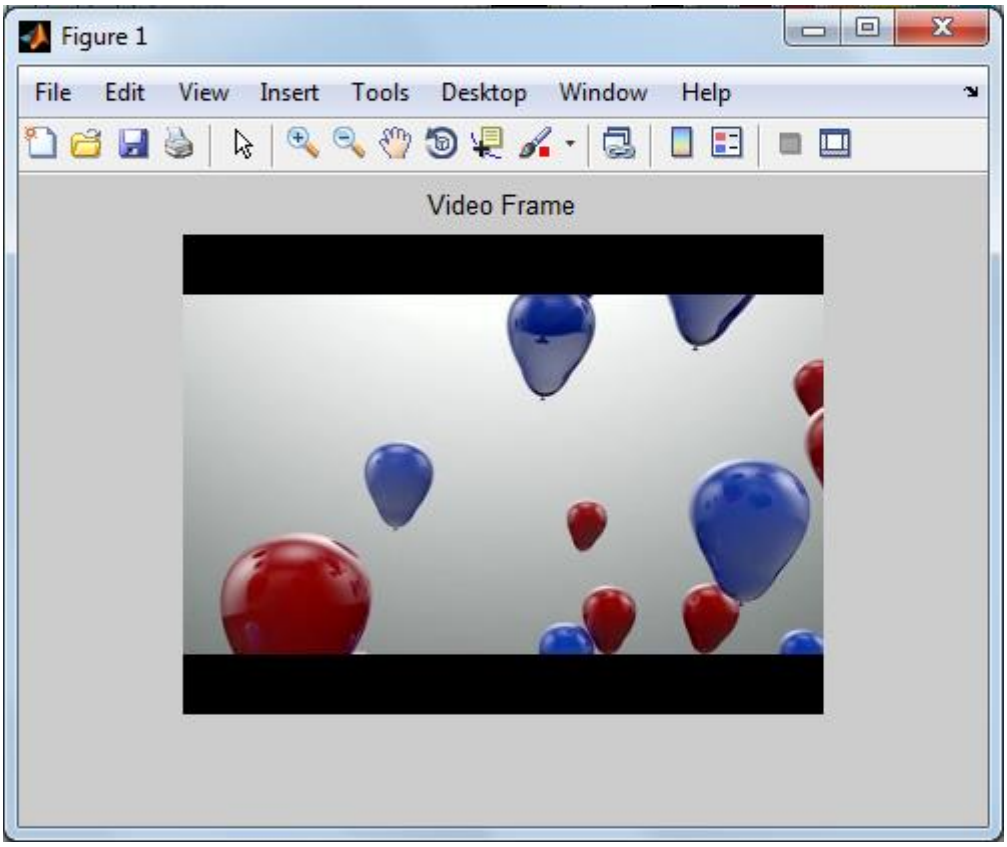


Figure 11 Video frame

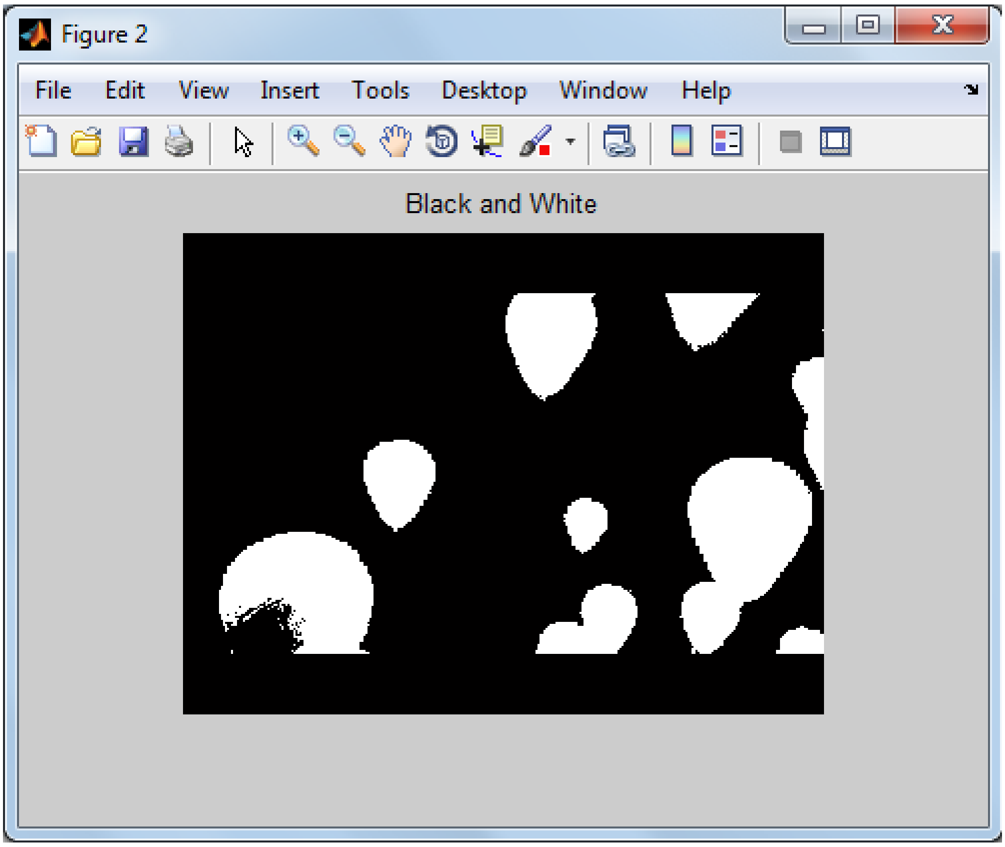


Figure 12 Black and white image

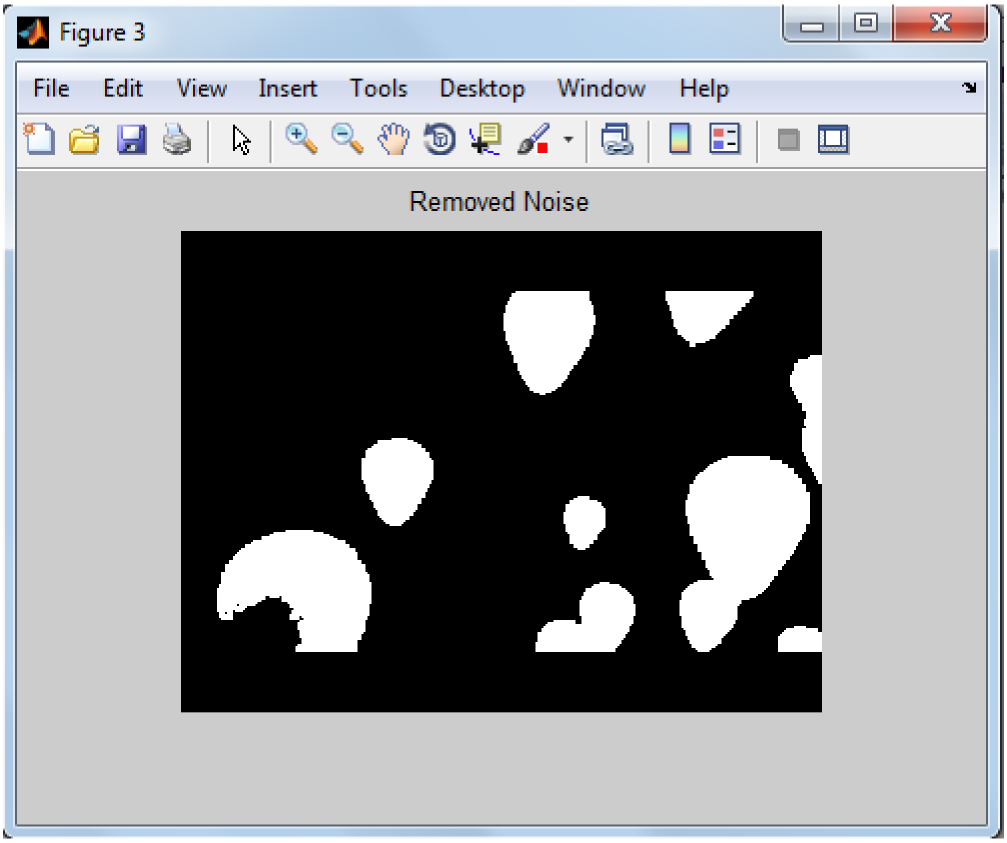


Figure 13 Removed noise

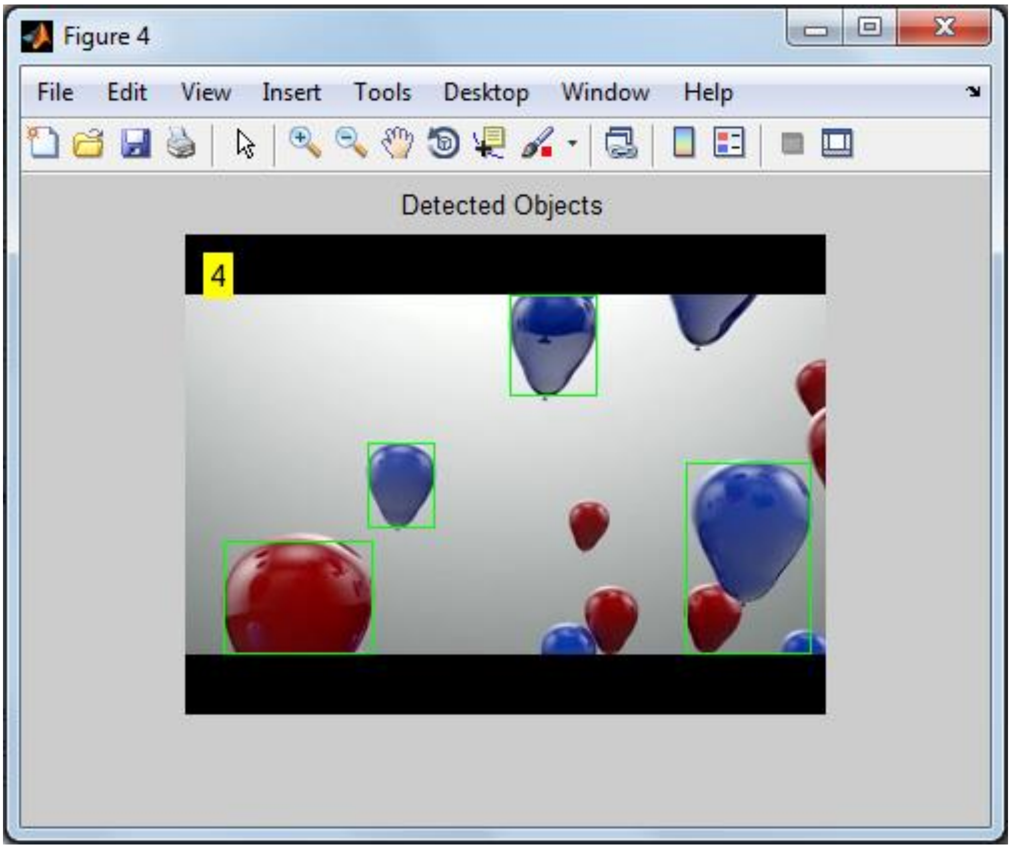


Figure 14 Detected object being screenshot

In figure 14, around the balloons, a rectangle green box being drawn. This shows the detected object the algorithm manage to detect while at the upper side of the image is the number of object it managed to detect which are 9.

CHAPTER 5: CONCLUSION AND RECOMMENDATION

In this paper, the author has developed a software based solution to count and separate the number of white blood cells. It is a new approach where only by using image and video input, the white blood cells can be separated. The author came out with the idea of white blood cell segmentation algorithm which is actually a modification of the one proposed by Madhloom et al. [9]. The minimum size constraints of 50% RBC size to reject small false objects were introduced.

When applying the algorithm to real time view, some of the steps were used while some does not needed. In real time view, the pixel of image in a frame is taken into account to detect the required object.

Image classification method in separating the white blood cells is cost effective and almost accurate where about 90 – 100% of the blood cells can be separated and save a lot of time.

For future study, the author would like to recommend developing another algorithm that can count the entire element in the blood capillaries such as platelets and red blood cells. This may help in detecting a new disease and not only helping in curing cancer.

REFERENCES

[1] S. Khan, A. Khan, F. S. Khattak, “An Accurate and Cost Effective Approach to Blood Cell Count”, Comsats Institute of Information Technology Wah Campus, WahCantt. Pakistan, 2012.

[2] G. Ongun, U. Halici, K. Leblebicioglu and V. Atala, "An automated differential blood count system," in Int. Conf. of the IEEE Engineering in Medicine and Biology Society, 2001, Istanbul.

[3] R. Adollah, M. Y. Mashor, N. F. Mohd Nasir and H. Rosli, "Blood Cell Image Segmentation: A Review," in IFMBE, Kuala Lumpur, 2008.

[4] J. Theerapattanakul, J. Plodpai and C. Pintavirooj, "An efficient method for segmentation step of automated white blood cell classifications," in TENCON, Bangkok, 2004.

[5] H. Papasaika-Hanusch, “Digital Image Processing Using Matlab”, Institute of Geodesy and Photogrammetry, ETH Zurich.

[6] A. E. Nelson, Implementation of image processing algorithm on FPGA hardware”, Ms.C. Thesis, Vanderbilt University, 2000.

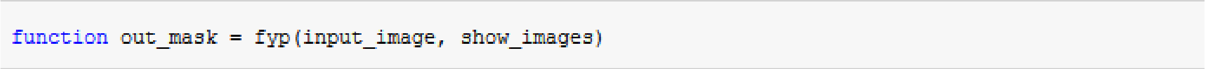
[7] S. L. Eddins, M. T. Orchard., “Using MAT Lab and C in image processing lab course”, The MathWorks, Inc., University of Illinois, 1994

[8] N. Sharma and L. M. Aggarwal, "Automated medical image segmentation techniques," *Journal of Medical Physics*, vol. 35, no. 1, pp. 3-14, 2010.

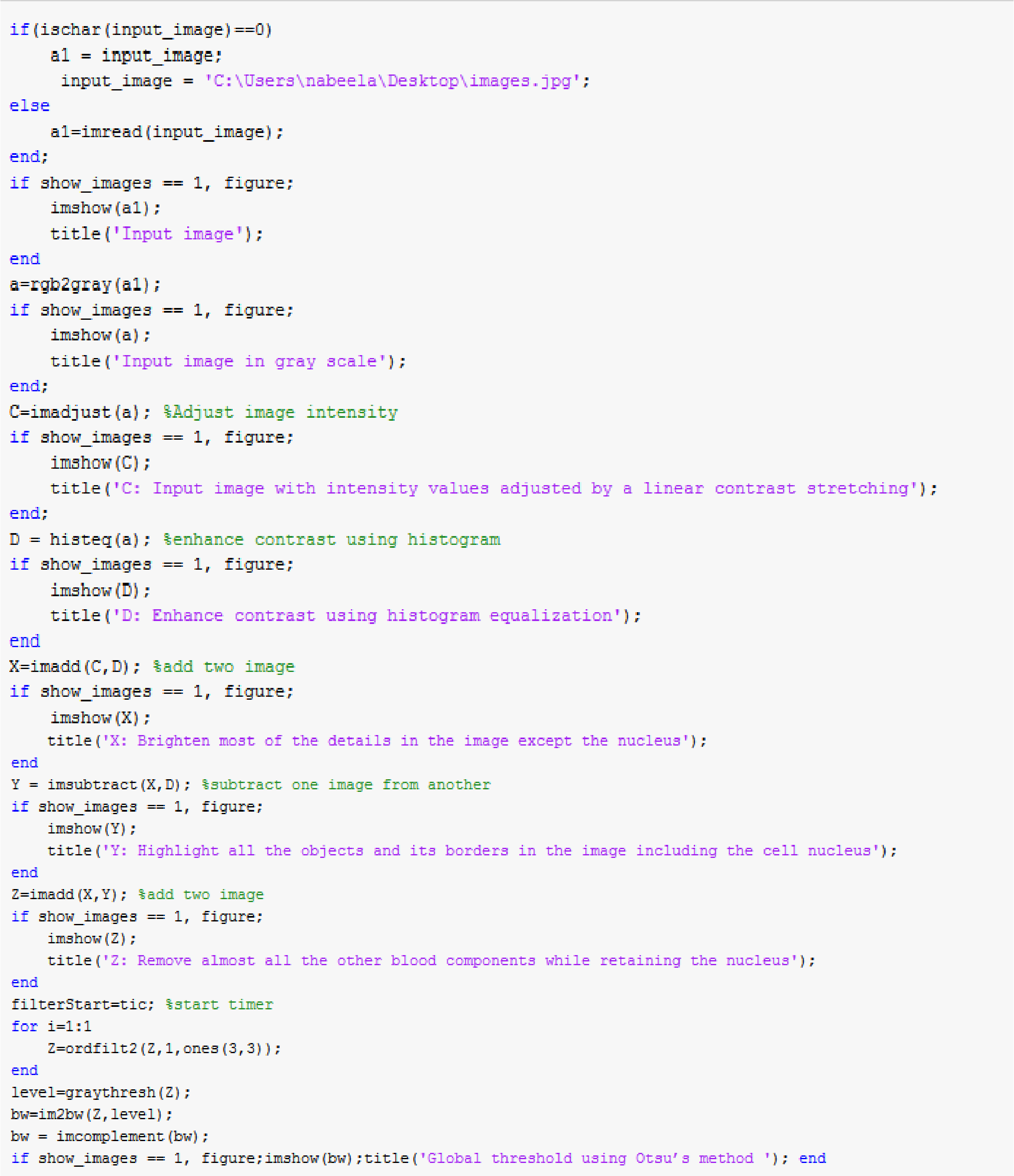
[9] H. T. Madhloom, S. A. Kareem, H. Ariffin, A. A. Zaidan, H. O. Alanazi and B. B. Zaidan, "An Automated White Blood Cell Nucleus Localization and Segmentation using Image Arithmetic and Automatic Threshold," *Journal of Applied Sciences*, vol. 10, no. 11, pp. 959-966, 2010.

APPENDICES

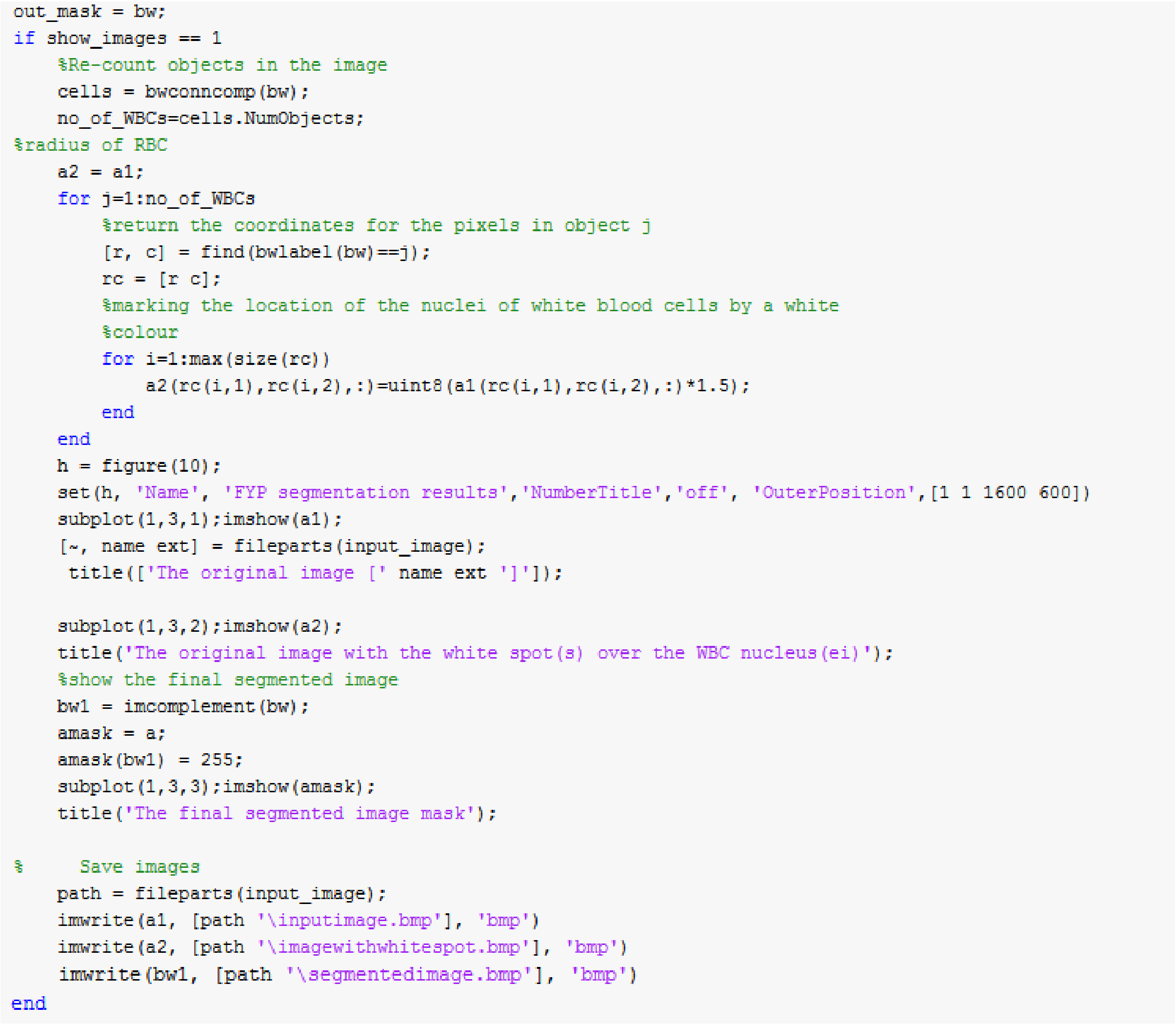
Appendix 1: Algorithm for image segmentation



```
function out_mask = fyp(input_image, show_images)
```



```
if(ischar(input_image)==0)
    a1 = input_image;
     input_image = 'C:\Users\nabeela\Desktop\images.jpg';
else
    a1=imread(input_image);
end;
if show_images == 1, figure;
    imshow(a1);
    title('Input image');
end
a=rgb2gray(a1);
if show_images == 1, figure;
    imshow(a);
    title('Input image in gray scale');
end;
C=imadjust(a); %Adjust image intensity
if show_images == 1, figure;
    imshow(C);
    title('C: Input image with intensity values adjusted by a linear contrast stretching');
end;
D = histeq(a); %enhance contrast using histogram
if show_images == 1, figure;
    imshow(D);
    title('D: Enhance contrast using histogram equalization');
end
X=imadd(C,D); %add two image
if show_images == 1, figure;
    imshow(X);
    title('X: Brighten most of the details in the image except the nucleus');
end
Y = imsubtract(X,D); %subtract one image from another
if show_images == 1, figure;
    imshow(Y);
    title('Y: Highlight all the objects and its borders in the image including the cell nucleus');
end
Z=imadd(X,Y); %add two image
if show_images == 1, figure;
    imshow(Z);
    title('Z: Remove almost all the other blood components while retaining the nucleus');
end
filterStart=tic; %start timer
for i=1:1
    Z=ordfilt2(Z,1,ones(3,3));
end
level=graythresh(Z);
bw=im2bw(Z,level);
bw = imcomplement(bw);
if show_images == 1, figure;imshow(bw);title('Global threshold using Otsu’s method '); end
```

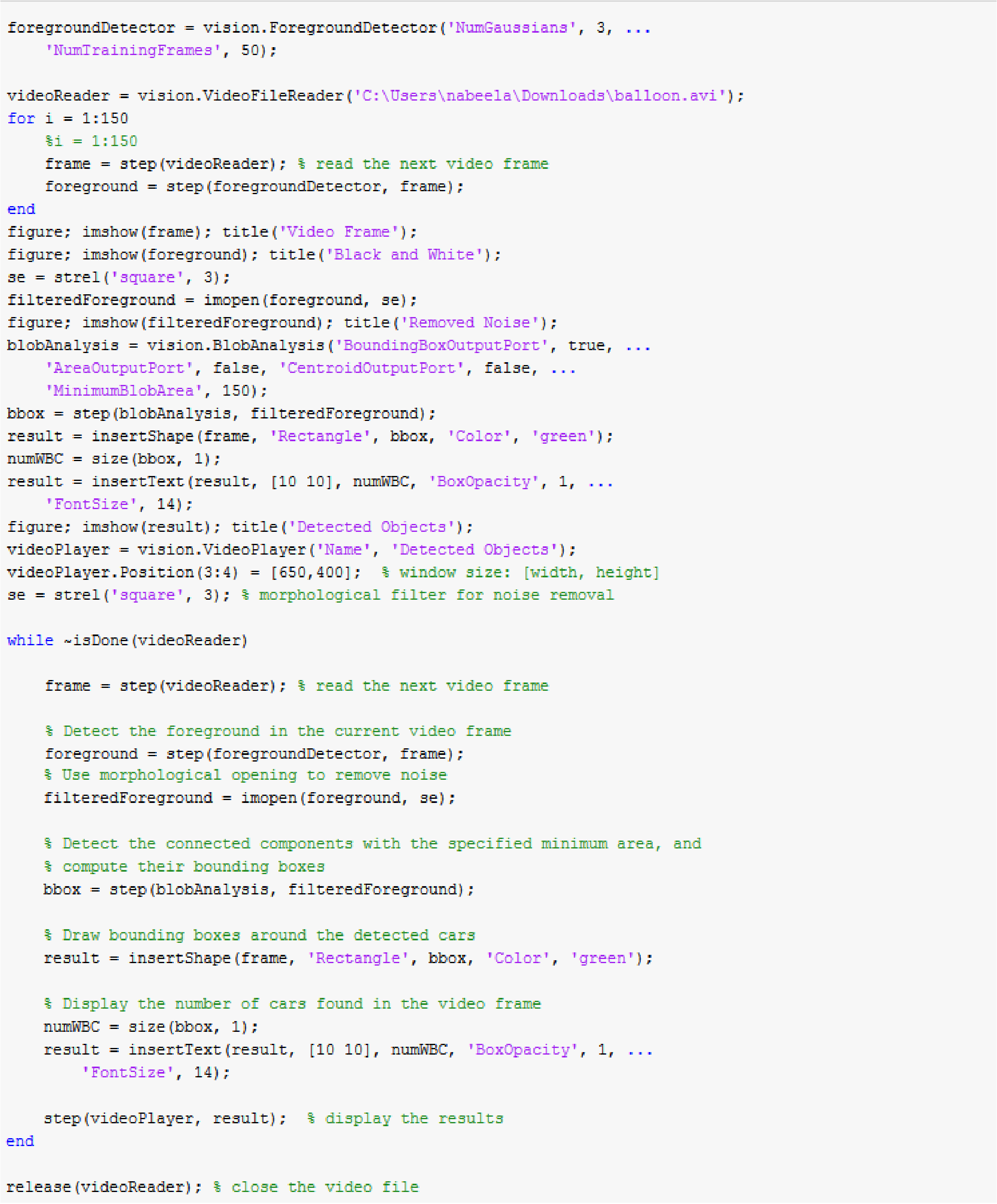


```
out_mask = bw;
if show_images == 1
    %Re-count objects in the image
    cells = bwconncomp(bw);
    no_of_WBCs=cells.NumObjects;
%radius of RBC
    a2 = a1;
    for j=1:no_of_WBCs
        %return the coordinates for the pixels in object j
        [r, c] = find(bwlabel(bw)==j);
        rc = [r c];
        %marking the location of the nuclei of white blood cells by a white
        %colour
        for i=1:max(size(rc))
            a2(rc(i,1),rc(i,2),:)=uint8(a1(rc(i,1),rc(i,2),:)*1.5);
        end
    end
    h = figure(10);
    set(h, 'Name', 'FYP segmentation results','NumberTitle','off', 'OuterPosition',[1 1 1600 600])
    subplot(1,3,1);imshow(a1);
    [~, name ext] = fileparts(input_image);
     title(['The original image [' name ext ']']);

    subplot(1,3,2);imshow(a2);
    title('The original image with the white spot(s) over the WBC nucleus(ei)');
    %show the final segmented image
    bw1 = imcomplement(bw);
    amask = a;
    amask(bw1) = 255;
    subplot(1,3,3);imshow(amask);
    title('The final segmented image mask');

%     Save images
    path = fileparts(input_image);
    imwrite(a1, [path '\inputimage.bmp'], 'bmp')
    imwrite(a2, [path '\imagewithwhitespot.bmp'], 'bmp')
    imwrite(bw1, [path '\segmentedimage.bmp'], 'bmp')
end
```

Appendix 2: Algorithm for video



```
foregroundDetector = vision.ForegroundDetector('NumGaussians', 3, ...
    'NumTrainingFrames', 50);

videoReader = vision.VideoFileReader('C:\Users\nabeela\Downloads\balloon.avi');
for i = 1:150
    %i = 1:150
    frame = step(videoReader); % read the next video frame
    foreground = step(foregroundDetector, frame);
end
figure; imshow(frame); title('Video Frame');
figure; imshow(foreground); title('Black and White');
se = strel('square', 3);
filteredForeground = imopen(foreground, se);
figure; imshow(filteredForeground); title('Removed Noise');
blobAnalysis = vision.BlobAnalysis('BoundingBoxOutputPort', true, ...
    'AreaOutputPort', false, 'CentroidOutputPort', false, ...
    'MinimumBlobArea', 150);
bbox = step(blobAnalysis, filteredForeground);
result = insertShape(frame, 'Rectangle', bbox, 'Color', 'green');
numWBC = size(bbox, 1);
result = insertText(result, [10 10], numWBC, 'BoxOpacity', 1, ...
    'FontSize', 14);
figure; imshow(result); title('Detected Objects');
videoPlayer = vision.VideoPlayer('Name', 'Detected Objects');
videoPlayer.Position(3:4) = [650,400];  % window size: [width, height]
se = strel('square', 3); % morphological filter for noise removal

while ~isDone(videoReader)

    frame = step(videoReader); % read the next video frame

    % Detect the foreground in the current video frame
    foreground = step(foregroundDetector, frame);
    % Use morphological opening to remove noise
    filteredForeground = imopen(foreground, se);

    % Detect the connected components with the specified minimum area, and
    % compute their bounding boxes
    bbox = step(blobAnalysis, filteredForeground);

    % Draw bounding boxes around the detected cars
    result = insertShape(frame, 'Rectangle', bbox, 'Color', 'green');

    % Display the number of cars found in the video frame
    numWBC = size(bbox, 1);
    result = insertText(result, [10 10], numWBC, 'BoxOpacity', 1, ...
        'FontSize', 14);

    step(videoPlayer, result);  % display the results
end

release(videoReader); % close the video file
```