ADIP ASSIGNMENT 2

Automatic RBC and WBC counting.

Team members:

Name	SRN
Prarthana P Bhat	PES1201802829
B.N. Vismaya	PES1201802831

Introduction:

We have developed a code that counts the number of white blood cells, red blood cells in an image.It also gives the overlapping cell regions in the image. WBCs are much larger, scattered than RBCs in a microscopic image of a blood smear. We have used basic image processing techniques such as: rgb to grayscale conversion, adjusting the contrast, binarizing the image, segmentation and other morphological operations such as opening, connected components,region props etc. Thus we will be able to separate WBCs and RBCs and count them.

Theory:

1) **Rgb to grayscale conversion** : This process removes all color information, leaving only the luminance of each pixel. Since digital images are displayed using a combination of red, green, and blue colors, each pixel has three separate luminance values. Therefore, these three values must be combined into a single value when removing color from an image.

2) **Image binarization** : The process is taking a grayscale image and converting it to black-and-white, essentially reducing the information contained within the image from 256 shades of gray to 2- black and white.

3) **Opening**:This process is the dilation of the erosion of a set A by a structuring element B.Opening removes small objects from the foreground .It can be used to find things into which a specific structuring element can fit (edges, corners, etc).

4) **Connected components**: In a 2D image, are clusters of pixels with the same value, which are connected to each other through either 4-pixel, or 8-pixel connectivity. 4-pixel connectivity would group all pixels that contact each other on either of their four faces, while 8-pixel would group pixels that are connected along any face or corner. In 3D, connectivity options, at least for rectangular pixels, are 6, 18, and 26 (faces, faces+edges, faces+edges+corners).

5) **Clearing border**: Suppresses structures in image that are lighter than their surroundings and that are connected to the image border. This is used to clear the image border. For grayscale images, imclearborder (function in matlab) tends to reduce the overall intensity level in addition to suppressing border structures.

Algorithm:

1. Read the original image.
2. As a part of image pre-processing, we convert the colour image into grayscale.
3. Adjust the contrast between foreground and background.
4. Convert it to a binary image.
5. Fill the holes in the binary image.
6. To separate overlapping cells:
   a. Perform opening with a structuring element as a disk.
   b. Clean and clearly segment the image with foreground as white pixels.
7. Finding the WBCs
   a. Find connected components in the binary image.
   b. Find the area and perimeter of these components.
   c. Area and perimeter is used as the criteria to separate WBC and RBC (area should be greater than 800 sq units and the perimeter should be greater than 175 units for this image.It can be arbitrarily set depending on image resolution).
   d. The resulting binary image is cleaned to remove RBCs overlapping on WBC.
   e. Clean the border cells to get only the regions for WBCs.
8. WBC regions are then removed from the original binary image through subtraction to get RBC regions.
9. Count number of WBCs by counting the number of connected components
10. For RBCs - find the number of overlapping RBCs and the total count of RBCs in the image.
11. Find the edge map of the RBC image and then filter elliptical regions to get the regions of overlapping cells.
12. Total number of Overlapped cells is given by counting the number of connected components of the overlapped cells.
13. Total number of RBCs by counting the number of connected components of the circular cells
14. Function to find elliptical regions

a. Clean the image and then specify the dimensions of an elliptical region to filter out overlapping cell regions

Code:

```matlab
% The below code counts the number of white blood cells, red blood cells

% and overlapping cells in the image

%White blood cells are much larger and scattered in a microscopic image of

%a blood smear. This code separates out large WBC's and smaller RBC's. RBC count

%is generally very high and the chances of overlapping cells in RBC is

%higher.

clc;

clear all;

inputImage = imread('blood_smear_ASH.jfif'); %read the original image

figure(1);

imshow(inputImage);

title('Orignal Image');

%image preprocessing

grayImage = rgb2gray(inputImage); %convert to grayscale image

IM_ad = imadjust(grayImage); %adjust the contrast between foreground and background

IB=imbinarize(grayImage); %convert to a binary image

IB_filled = imfill(~IB, 'holes'); %fill the holes in the binary image

figure(2);

imshow(IB_filled);

title('All holes filled')

%separate overlapping cells

se = strel('disk',6);
```

```matlab
IM3 = imopen(IB_filled,se); %perform opening with a structuring element as a
disk

figure(3);

imshow(IM3); %clean and clearly segmented image with foreground as white pixels

title("Clearly segmented Image")

%%finding and counting WBCs

 cc = bwconncomp(IM3); %find connected components in the binary image

stats = regionprops(cc,'Area','Perimeter'); %find area and perimeter of these
components

idx = find([stats.Area] > 800 & [stats.Perimeter] > 175);

BW2 = ismember(labelmatrix(cc),idx); %area and perimeter is used as the
criteria to separate out WBC and RBC

%The resulting binary image is cleaned to remove RBC's overlapping on WBC

%and those on border cells to get only the regions for WBC's

se = strel('disk',15);

IM_WBC = imopen(BW2,se);

IM_WBC= imclearborder(IM_WBC);

figure(4);

imshow(IM_WBC);

title("WBC regions")

%WBC regions are then removed from the original binary image

IM_RBC=IM3-IM_WBC;

figure(5);

imshow(IM_RBC);

title("RBC regions")

 %count number of WBCs

connectedComponent = bwconncomp(IM_WBC, 4);

circularCellCount = connectedComponent.NumObjects;

fprintf('%s %d\n','Total number of white blood cells = ',circularCellCount);
```

```matlab
%For RBC- we find the number of overlapping RBCs and the total count of
%RBC's in the image
%find the edge map of the RBC image and then filter elliptical regions to
%get the regions of overlapping cells
[Gmag,Gdir] = imgradient(IM_RBC);
[extractOverlappedCells,properties]=filterRegionsElliptical(Gmag);
figure(6);
imshow(extractOverlappedCells);
title("overlapping RBCs regions")
% Total number of Overlapped cells
connectedComponent = bwconncomp(extractOverlappedCells, 4);
overlappedCellsCount = connectedComponent.NumObjects;
fprintf('%s %d\n','Total number of overlapped cells = ',overlappedCellsCount);
%Total number of RBCs
connectedComponent = bwconncomp(IM_RBC, 4);
circularCellCount = connectedComponent.NumObjects;
fprintf('%s %d\n','Total number of red blood cells = ',circularCellCount);
%function to find elliptical regions
function [BW_out,properties] = filterRegionsElliptical(BW_in)
%clean the image and then specify the dimensions of an elliptical region to
%filter out overlapping cell regions
BW_out = imbinarize(BW_in);
BW_out = imclearborder(BW_out);
BW_out = imfill(BW_out, 'holes');
BW_out = bwpropfilt(BW_out, 'MajorAxisLength', [30, 50]);
BW_out = bwpropfilt(BW_out, 'MinorAxisLength', [0, 30]);
BW_out = bwpropfilt(BW_out, 'Eccentricity', [0.795, 0.907]);
BW_out = bwpropfilt(BW_out, 'Area', [300, 2700]);
```
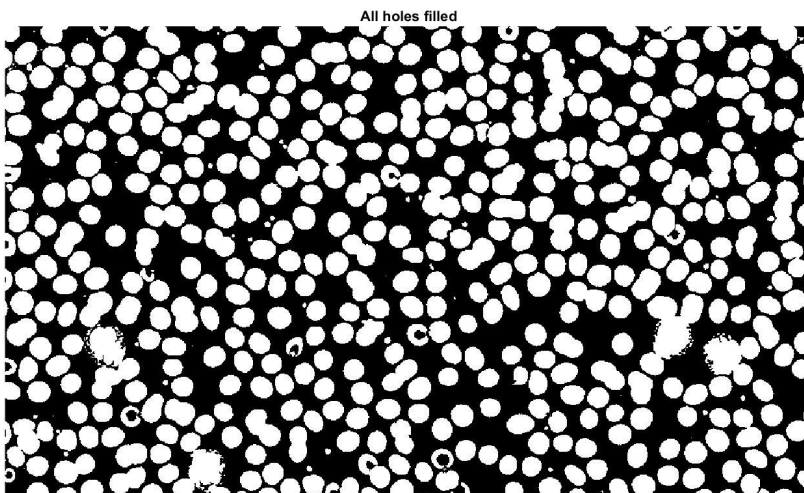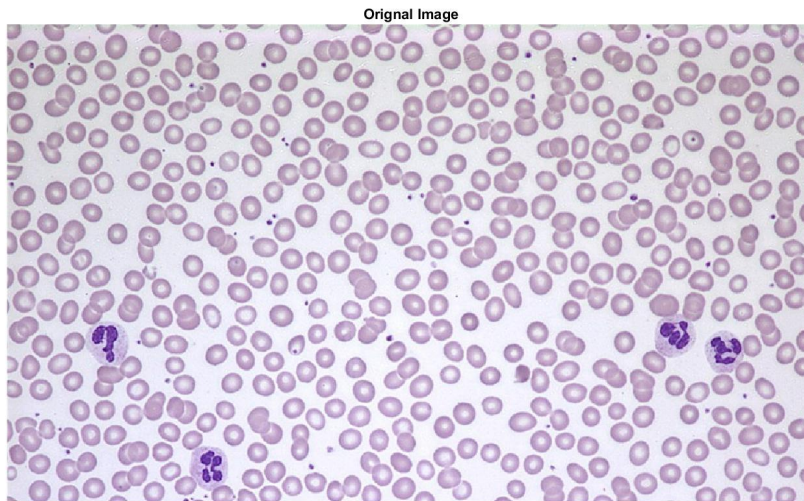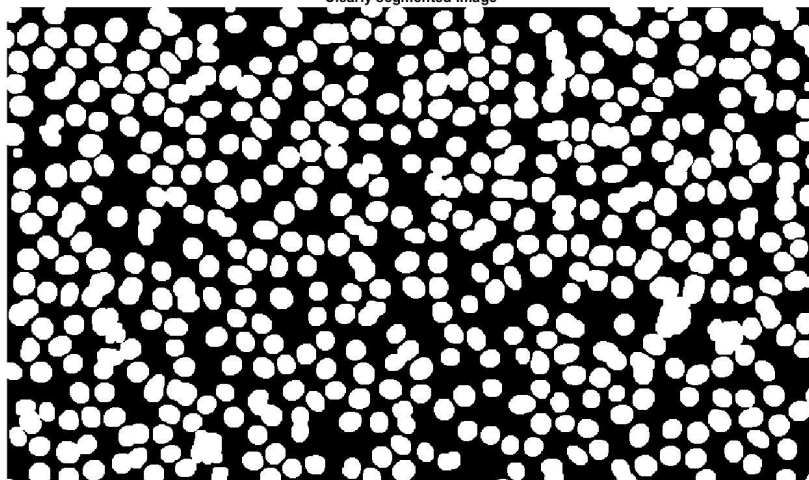
```
properties = regionprops(BW_out, {'Eccentricity','MajorAxisLength',
'MinorAxisLength','Perimeter'});

end
```
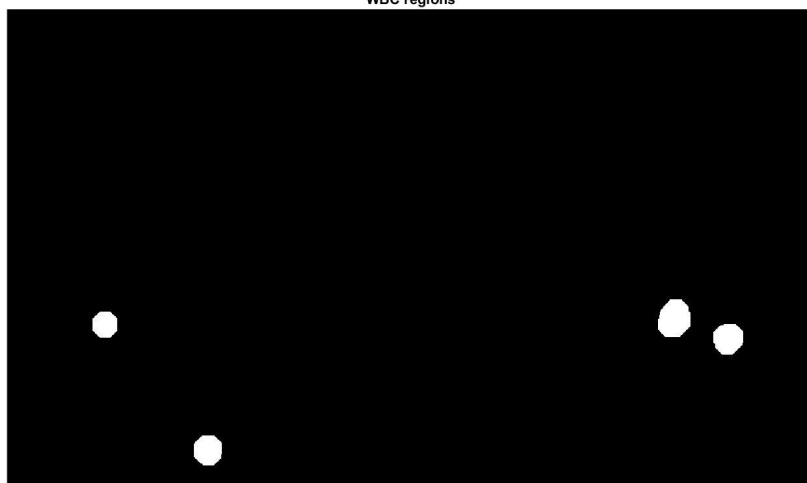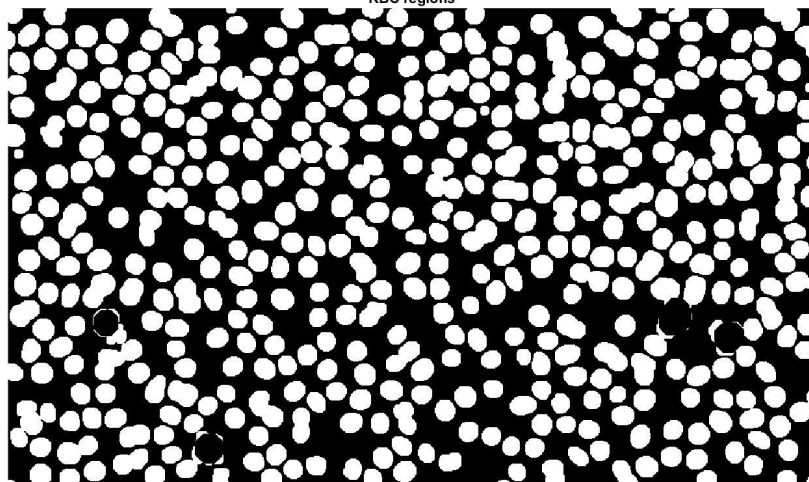
Results:

Orignal Image

All holes filled

Clearly segmented Image
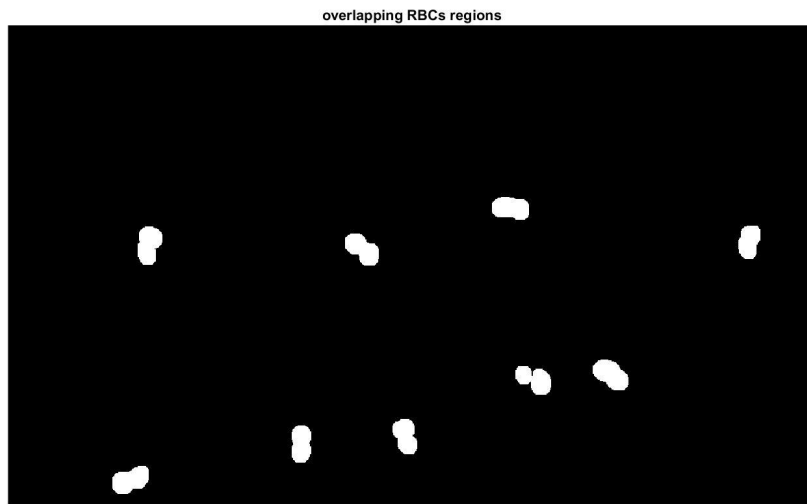

WBC regions

RBC regions

overlapping RBCs regions

Command Window

```
Total number of white blood cells =   4
Total number of overlapped cells =   9
Total number of red blood cells =   448
fx >> |
```

Observation:

The input is a microscopic image of a blood smear, when we run the code, it fills the holes , clearly segments the image and gives the different regions of interest . The algorithm as a result gives the separate WBC region, RBC region and the overlapped RBC region and the total count of WBCs, overlapped cells and RBCs are also given.