Segmentation of white blood cells

Haneen Elsawy

segmentation of white blood cells

Content:

Project description:

Design a program for the automated segmentation of white blood cells (WBC). The program's input is a WBC image and the output should be a binary image with only the white blood cell (the purplish object).

Code flow:

The code takes the input images from each through looping for each file, then enhancement techniques such as histogram equalization, noise reduction…etc. next we find the accuracy of our model against the ground truth model and the sum of those values and we combine all these sums to find the average accuracy of my segmented images. Finally we extract features that would help in classifying the model at the classifier learning app where we train our model and find the accuracy of the model against test set.

```
sum=0.0;
av=0.0;
stats1;
stats2;
stats3;

%mono
file1 = dir('/MATLAB Drive/Published/mono/*.bmp');
file01 = dir('/MATLAB Drive/Published/monoground/*.bmp');
se = strel('disk',7);
channel2Min = 0.2;
channel2Max = 1.0;
NF1 = length(file1);
images1 = cell(NF1,1);
for k = 1 : NF1
  images1{k} = imread(fullfile('/MATLAB Drive/Published/mono', file1(k).name));
  images01{k} = imread(fullfile('/MATLAB Drive/Published/monoground', file1(k).name));
  figure;
  stretched_Image1{k} =histeq( images1{k});
  noise{k}= medfilt3(stretched_Image1{k});
  hsv{k} = rgb2hsv(im2double(images1{k}));
  mask{k}=(hsv{k}(:,:,2) >= channel2Min ) & (hsv{k}(:,:,2) <= channel2Max);
  opening{k}=imopen(mask{k},se);
  closing{k}=imclose(opening{k},se);
  %BW{k} = imbinarize(I{k})
 similarity{k} = jaccard( closing{k}, images01{k});
 sum=sum+similarity{k};
 stats1=regionprops('table',closing{k},'Area','Perimeter','Eccentricity','Circularity');
%   av=sum/NF1;
```


```
%   av=sum/NF1;
 %stats1 = regionprops('table',closing{k},'Area','Circularity','Perimeter','Eccentricity');
 % temp{k} = regionprops('table',closing{k}, 'area', 'Perimeter','Circularity','Eccentricity');
  %areas{k} = [temp{k}.area];
%   centroids{K} = vertcat(temp.Centroid);
% t = struct2table(closing{k});
 %stats{k} = regionprops('table',closing{k},'Area','Perimeter','Circularity','Eccentricity')
% subplot(2,4,1), imshow(images1{k}), title('Original Image');
% subplot(2,4,2), imshow(stretched_Image1{k}), title('Equalized Image');
% subplot(2,4,3), imshow(noise{k}), title('NoiseFree Image');
 %imshow(closing{k}), title('Binary Image');

end

%lymp
file2 = dir('/MATLAB Drive/Published/lymp/*.bmp');
 file02 = dir('/MATLAB Drive/Published/lympground/*.bmp');
 se2 = strel('disk',8);
 channel2Min2 = 0.2;
 channel2Max2 = 1.0;
 NF2 = length(file2);
 images2 = cell(NF2,1);
 for k = 1 : NF2
   images1{k} = imread(fullfile('/MATLAB Drive/Published/lymp', file2(k).name));
   images2{k} = imread(fullfile('/MATLAB Drive/Published/lympground', file02(k).name));
   figure;
   stretched_Image1{k} =histeq( images1{k});
   noise{k}= medfilt3(stretched_Image1{k});
   hsv{k} = rgb2hsv(im2double(images1{k}));
```

```
    hsv{k} = rgb2hsv(im2double(images1{k}));
    mask{k}=(hsv{k}(:,:,2) >= channel2Min ) & (hsv{k}(:,:,2) <= channel2Max);
    opening{k}=imopen(mask{k},se2);
    closing{k}=imclose(opening{k},se2);
    %BW{k} = imbinarize(I{k})
   similarity{k} = jaccard( closing{k}, images2{k});
   sum=sum+similarity{k};
   % av=sum/NF1;
   stats2=regionprops('table',closing{k},'Area','Perimeter','Eccentricity','Circularity');
   % subplot(2,4,1), imshow(images1{k}), title('Original Image');
% % subplot(2,4,2), imshow(stretched_Image1{k}), title('Equalized Image');
% % subplot(2,4,3), imshow(noise{k}), title('NoiseFree Image');
%  %imshow(closing{k}), title('Binary Image');
 end

%eosi
file3 = dir('/MATLAB Drive/Published/eosi/*.bmp');
 file03 = dir('/MATLAB Drive/Published/eosiground/*.bmp');
 se3 = strel('disk',6);
 channel2Min3 = 0.2;
 channel2Max3 = 1.0;
 NF3 = length(file3);
 images3 = cell(NF3,1);
 for k = 1 : NF3
   images1{k} = imread(fullfile('/MATLAB Drive/Published/eosi', file3(k).name));
   images2{k} = imread(fullfile('/MATLAB Drive/Published/eosiground', file03(k).name));
   figure;
   stretched_Image1{k} =histeq( images1{k});
   noise{k}= medfilt3(stretched_Image1{k});
```



```
    noise{k}= medfilt3(stretched_Image1{k});
    hsv{k} = rgb2hsv(im2double(images1{k}));
    mask{k}=(hsv{k}(:,:,2) >= channel2Min ) & (hsv{k}(:,:,2) <= channel2Max);
    opening{k}=imopen(mask{k},se3);
    closing{k}=imclose(opening{k},se3);
    %BW{k} = imbinarize(I{k})
   similarity{k} = jaccard( closing{k}, images2{k});
   sum=sum+similarity{k};
   % av=sum/NF1;
   stats3=regionprops('table',closing{k},'Area','Perimeter','Eccentricity','Circularity');
%  % subplot(2,4,1), imshow(images1{k}), title('Original Image');
% % subplot(2,4,2), imshow(stretched_Image1{k}), title('Equalized Image');
% % subplot(2,4,3), imshow(noise{k}), title('NoiseFree Image');
%  %imshow(closing{k}), title('Binary Image');
 end

%neut
 file4 = dir('/MATLAB Drive/Published/neut/*.bmp');
 file04 = dir('/MATLAB Drive/Published/neutground/*.bmp');
 se4 = strel('disk',5);
 channel2Min4 = 0.2;
 channel2Max4 = 1.0;
 NF4 = length(file4);
 images4 = cell(NF4,1);
 for k = 1 : NF4
   images1{k} = imread(fullfile('/MATLAB Drive/Published/neut', file4(k).name));
   images2{k} = imread(fullfile('/MATLAB Drive/Published/neutground', file04(k).name));
   figure;
   stretched_Image1{k} =histeq( images1{k});
```

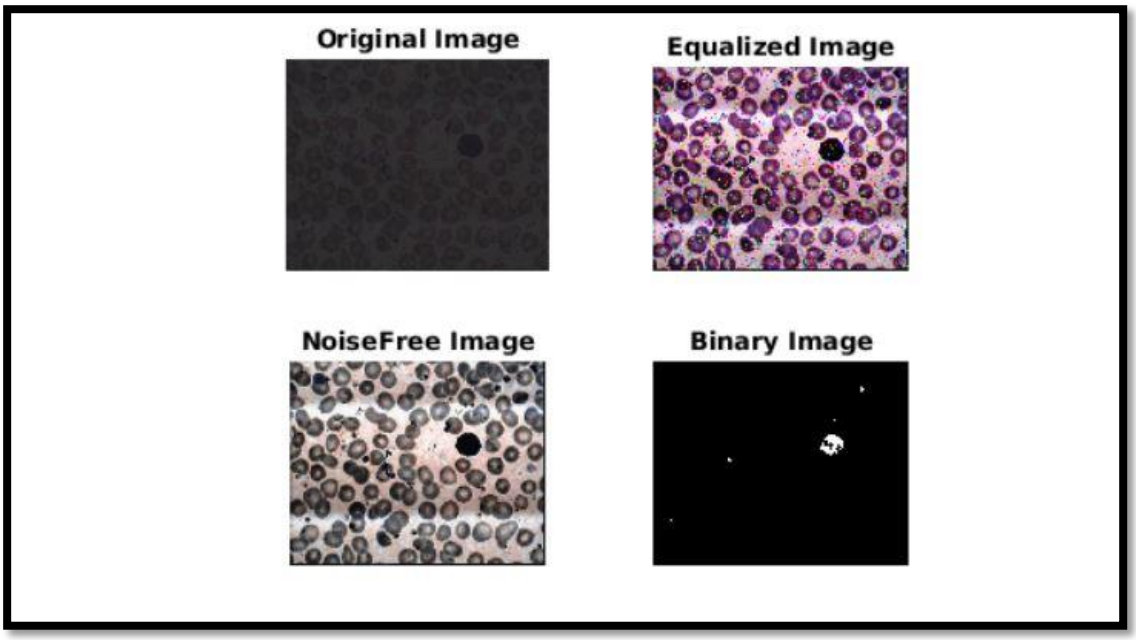
```
    stretched_Image1{k} =histeq( images1{k});
    noise{k}= medfilt3(stretched_Image1{k});
    hsv{k} = rgb2hsv(im2double(images1{k}));
    mask{k}=(hsv{k}(:,:,2) >= channel2Min ) & (hsv{k}(:,:,2) <= channel2Max);
    opening{k}=imopen(mask{k},se4);
    closing{k}=imclose(opening{k},se4);
%   %BW{k} = imbinarize(I{k})
  similarity{k} = jaccard( closing{k}, images2{k});
  sum=sum+similarity{k};
%  % av=sum/NF1;
stats4=regionprops('table',closing{k},'Area','Perimeter','Eccentricity','Circularity');
%  % subplot(2,4,1), imshow(images1{k}), title('Original Image');
% % subplot(2,4,2), imshow(stretched_Image1{k}), title('Equalized Image');
% % subplot(2,4,3), imshow(noise{k}), title('NoiseFree Image');
%  %imshow(closing{k}), title('Binary Image');
 end

%baso
 file5 = dir('/MATLAB Drive/Published/*.bmp');
 file05 = dir('/MATLAB Drive/Published/basoground/*.bmp');
 se5 = strel('disk',1);
 channel2Min5 = 0.2;
 channel2Max5 = 1.0;
 NF5 = length(file5);
 images5 = cell(NF5,1);
 for k = 1 : NF5
   images1{k} = imread(fullfile('/MATLAB Drive/Published', file5(k).name));
   images2{k} = imread(fullfile('/MATLAB Drive/Published/basoground', file05(k).name));
   figure;
```

```
 NF5 = length(file5);
 images5 = cell(NF5,1);
 for k = 1 : NF5
   images1{k} = imread(fullfile('/MATLAB Drive/Published', file5(k).name));
   images2{k} = imread(fullfile('/MATLAB Drive/Published/basoground', file05(k).name));
   figure;
   stretched_Image1{k} =histeq( images1{k});
   noise{k}= medfilt3(stretched_Image1{k});
   hsv{k} = rgb2hsv(im2double(images1{k}));
   mask{k}=(hsv{k}(:,:,2) >= channel2Min ) & (hsv{k}(:,:,2) <= channel2Max);
   opening{k}=imopen(mask{k},se5);
   closing{k}=imclose(opening{k},se5);
%   %BW{k} = imbinarize(I{k})
  similarity{k} = jaccard( closing{k}, images2{k});
  sum=sum+similarity{k};
%  % av=sum/NF1;
stats5=regionprops('table',closing{k},'Area','Perimeter','Eccentricity','Circularity');
%  % subplot(2,4,1), imshow(images1{k}), title('Original Image');
% % subplot(2,4,2), imshow(stretched_Image1{k}), title('Equalized Image');
% % subplot(2,4,3), imshow(noise{k}), title('NoiseFree Image');
%  %imshow(closing{k}), title('Binary Image');
 end

%accuracy
av=(sum/(NF1+NF2+NF3+NF4+NF5))*100;

%features_extraction
A1 = {stats1,stats2,stats3,stats4,stats5};
C = vertcat(A1{:})
```

Req.1: Preprocessing of the image for segmentation

- Adjusting the contrast of the image.
- Removal of the salt and pepper noise.

Baso

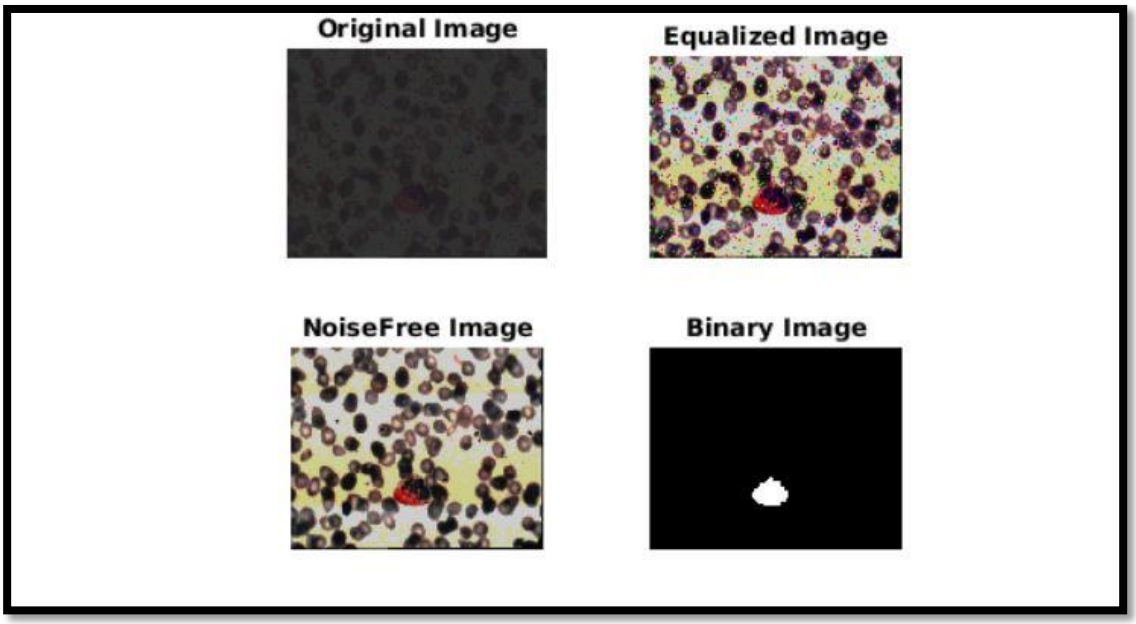



```
%baso
 file5 = dir('/MATLAB Drive/Published/*.bmp');
 file05 = dir('/MATLAB Drive/Published/basoground/*.bmp');
 se5 = strel('disk',1);
 channel2Min5 = 0.2;
 channel2Max5 = 1.0;
 NF5 = length(file5);
 images5 = cell(NF5,1);
 for k = 1 : NF5
   images1{k} = imread(fullfile('/MATLAB Drive/Published', file5(k).name));
   images2{k} = imread(fullfile('/MATLAB Drive/Published/basoground', file05(k).name));
   figure;
   stretched_Image1{k} =histeq( images1{k});
   noise{k}= medfilt3(stretched_Image1{k});
   hsv{k} = rgb2hsv(im2double(images1{k}));
   mask{k}=(hsv{k}(:,:,2) >= channel2Min ) & (hsv{k}(:,:,2) <= channel2Max);
   opening{k}=imopen(mask{k},se5);
   closing{k}=imclose(opening{k},se5);
%   %BW{k} = imbinarize(I{k})
  similarity{k} = jaccard( closing{k}, images2{k});
  sum=sum+similarity{k};
%  % av=sum/NF1;
stats5=regionprops('table',closing{k},'Area','Perimeter','Eccentricity','Circularity');
```

Here we applied histogram equalization using histeq() to adjust the contrast, then for noise removal we applied median filter medfilt3() to the output image, then we extracted the saturation

according to the range of the highest and the lowest level defined for better results when producing binary images, Finally we performed morphological operations with a disc structuring element with value 1, then we opening imopen() (erosion followed by a dilation) for removing small objects (noise), then followed by closing imclose() for filling the gaps. After enhancement we find the accuracy of our segmented pictures through comparing it to the ground truth images. Then we find the sum of these value and save it at a global variable sum which will add the rest of the accuracy values to find the average total accuracy. We use regionprops() to extract the given features to help us train our model and high.

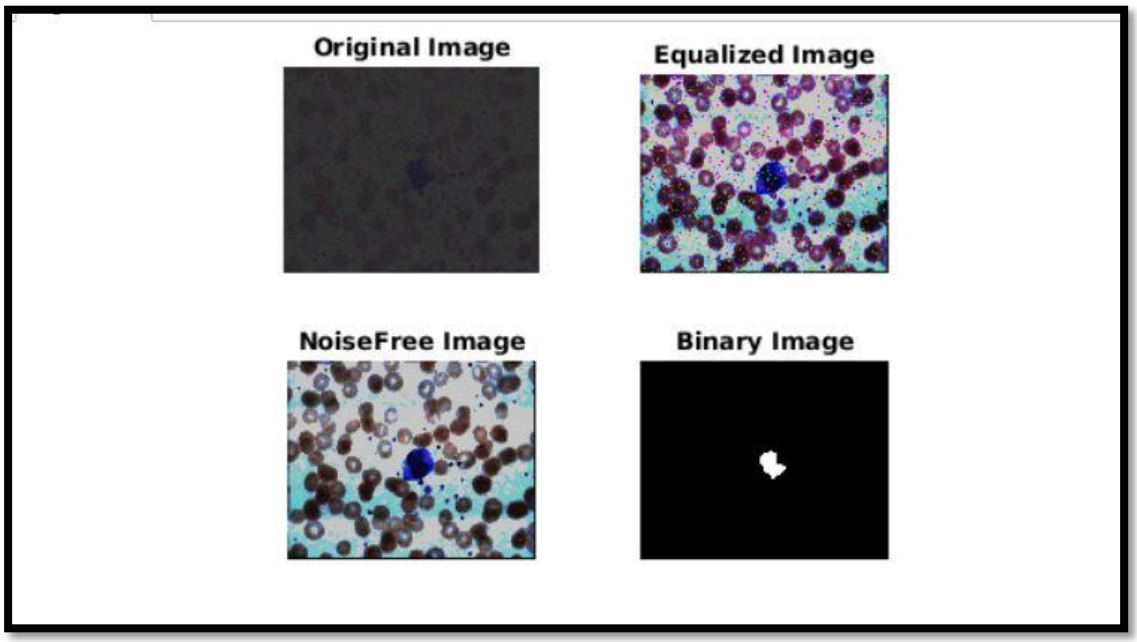
Eosi



```
%eosi
file3 = dir('/MATLAB Drive/Published/eosi/*.bmp');
 file03 = dir('/MATLAB Drive/Published/eosiground/*.bmp');
 se3 = strel('disk',6);
 channel2Min3 = 0.2;
 channel2Max3 = 1.0;
 NF3 = length(file3);
 images3 = cell(NF3,1);
 for k = 1 : NF3
   images1{k} = imread(fullfile('/MATLAB Drive/Published/eosi', file3(k).name));
   images2{k} = imread(fullfile('/MATLAB Drive/Published/eosiground', file03(k).name));
   figure;
   stretched_Image1{k} =histeq( images1{k});
   noise{k}= medfilt3(stretched_Image1{k});
   hsv{k} = rgb2hsv(im2double(images1{k}));
   mask{k}=(hsv{k}(:,:,2) >= channel2Min ) & (hsv{k}(:,:,2) <= channel2Max);
   opening{k}=imopen(mask{k},se3);
   closing{k}=imclose(opening{k},se3);
   %BW{k} = imbinarize(I{k})
  similarity{k} = jaccard( closing{k}, images2{k});
  sum=sum+similarity{k};
  % av=sum/NF1;
  stats3=regionprops('table',closing{k},'Area','Perimeter','Eccentricity','Circularity');
```

Same steps except se value 6 to obtain a better output when performing morphological operations.

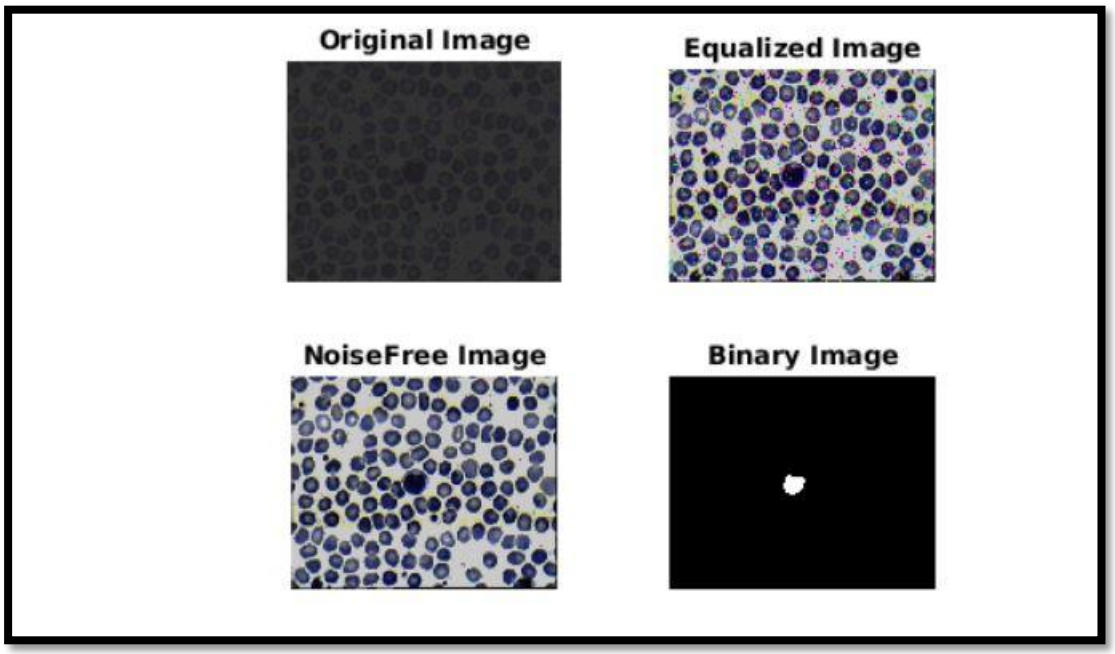
Mono



```
%mono
file1 = dir('/MATLAB Drive/Published/mono/*.bmp');
file01 = dir('/MATLAB Drive/Published/monoground/*.bmp');
se = strel('disk',7);
channel2Min = 0.2;
channel2Max = 1.0;
NF1 = length(file1);
images1 = cell(NF1,1);
for k = 1 : NF1
  images1{k} = imread(fullfile('/MATLAB Drive/Published/mono', file1(k).name));
  images01{k} = imread(fullfile('/MATLAB Drive/Published/monoground', file1(k).name));
  figure;
  stretched_Image1{k} =histeq( images1{k});
  noise{k}= medfilt3(stretched_Image1{k});
  hsv{k} = rgb2hsv(im2double(images1{k}));
  mask{k}=(hsv{k}(:,:,2) >= channel2Min ) & (hsv{k}(:,:,2) <= channel2Max);
  opening{k}=imopen(mask{k},se);
  closing{k}=imclose(opening{k},se);
  %BW{k} = imbinarize(I{k})
 similarity{k} = jaccard( closing{k}, images01{k});
 sum=sum+similarity{k};
```

Same steps except se value 7 to obtain a better output when performing morphological operations.

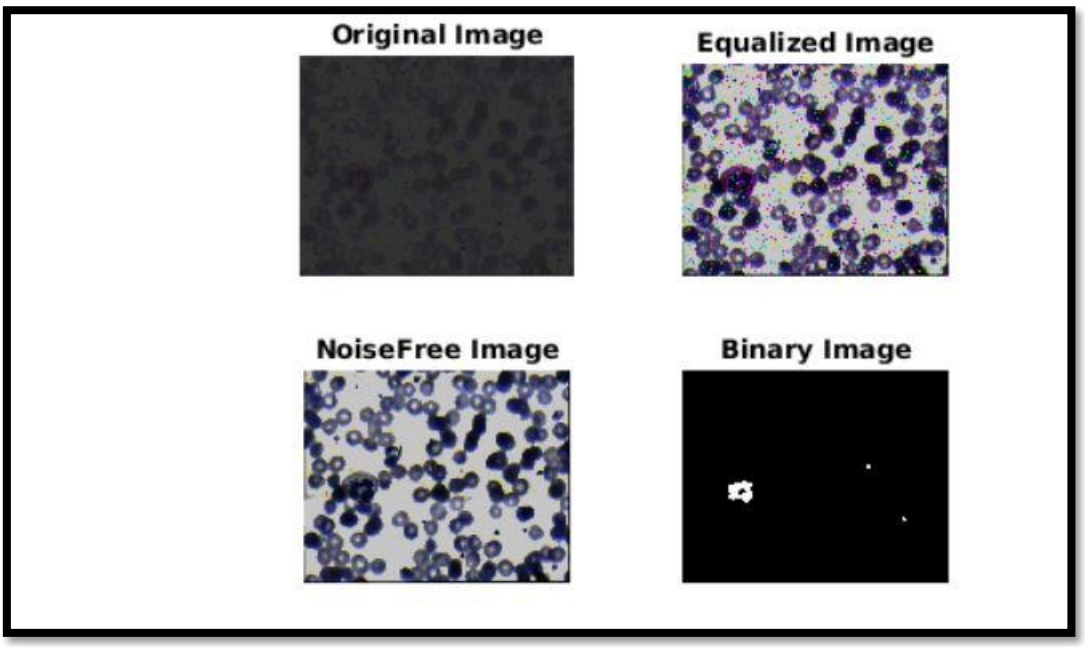
lymp



```
%lymp
file2 = dir('/MATLAB Drive/Published/lymp/*.bmp');
 file02 = dir('/MATLAB Drive/Published/lympground/*.bmp');
 se2 = strel('disk',8);
 channel2Min2 = 0.2;
 channel2Max2 = 1.0;
 NF2 = length(file2);
 images2 = cell(NF2,1);
 for k = 1 : NF2
   images1{k} = imread(fullfile('/MATLAB Drive/Published/lymp', file2(k).name));
   images2{k} = imread(fullfile('/MATLAB Drive/Published/lympground', file02(k).name));
   figure;
   stretched_Image1{k} =histeq( images1{k});
   noise{k}= medfilt3(stretched_Image1{k});
   hsv{k} = rgb2hsv(im2double(images1{k}));
   mask{k}=(hsv{k}(:,:,2) >= channel2Min ) & (hsv{k}(:,:,2) <= channel2Max);
   opening{k}=imopen(mask{k},se2);
   closing{k}=imclose(opening{k},se2);
   %BW{k} = imbinarize(I{k})
  similarity{k} = jaccard( closing{k}, images2{k});
  sum=sum+similarity{k};
  % av=sum/NF1;
  stats2=regionprops('table',closing{k},'Area','Perimeter','Eccentricity','Circularity');
```

Same steps except se value 8 to obtain a better output when performing morphological operations.

Neut



```
%neut
 file4 = dir('/MATLAB Drive/Published/neut/*.bmp');
 file04 = dir('/MATLAB Drive/Published/neutground/*.bmp');
 se4 = strel('disk',5);
 channel2Min4 = 0.2;
 channel2Max4 = 1.0;
 NF4 = length(file4);
 images4 = cell(NF4,1);
 for k = 1 : NF4
   images1{k} = imread(fullfile('/MATLAB Drive/Published/neut', file4(k).name));
   images2{k} = imread(fullfile('/MATLAB Drive/Published/neutground', file04(k).name));
   figure;
   stretched_Image1{k} =histeq( images1{k});
   noise{k}= medfilt3(stretched_Image1{k});
   hsv{k} = rgb2hsv(im2double(images1{k}));
   mask{k}=(hsv{k}(:,:,2) >= channel2Min ) & (hsv{k}(:,:,2) <= channel2Max);
   opening{k}=imopen(mask{k},se4);
   closing{k}=imclose(opening{k},se4);
%   %BW{k} = imbinarize(I{k})
  similarity{k} = jaccard( closing{k}, images2{k});
  sum=sum+similarity{k};
%  % av=sum/NF1;
stats4=regionprops('table',closing{k},'Area','Perimeter','Eccentricity','Circularity');
```

Same steps except se value 5 to obtain a better output when performing morphological operations.

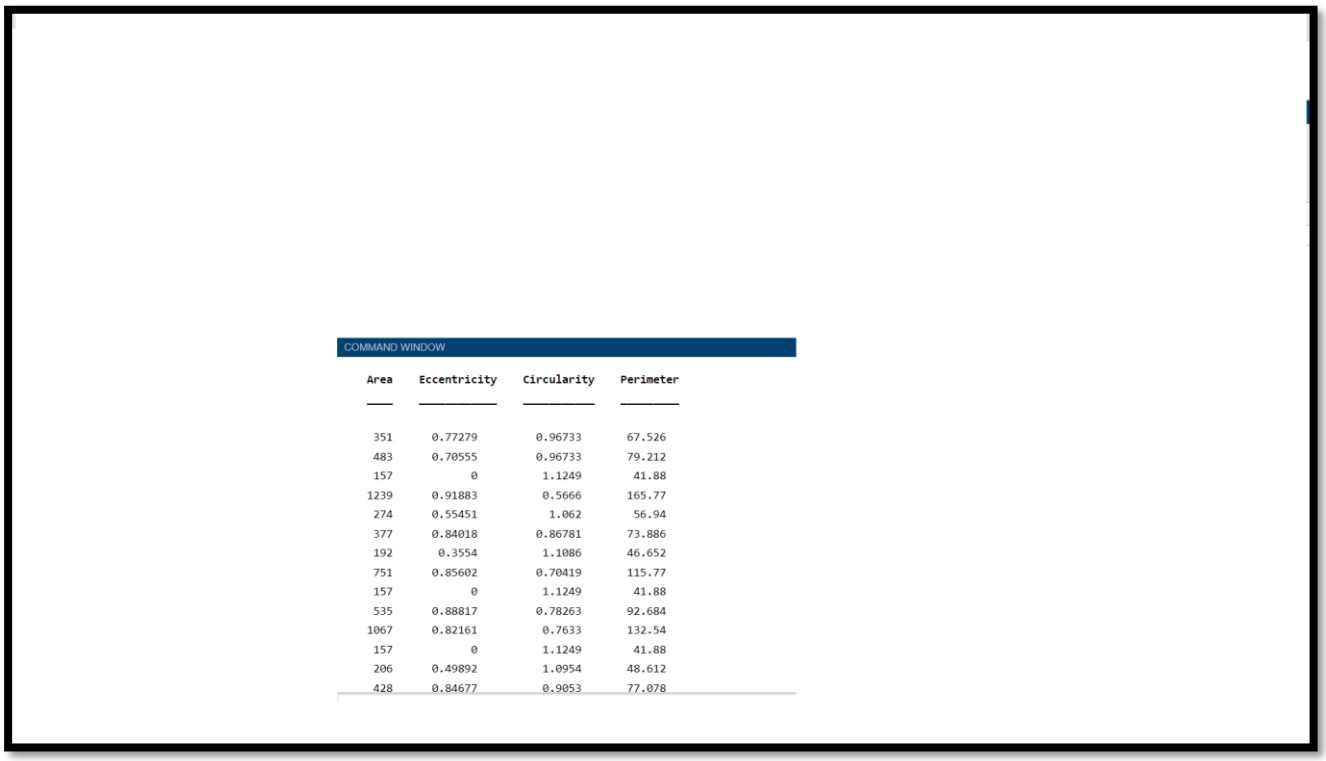
- Req.2: White blood cell segmentation (black and white image with the extracted WBC)

- and accuracy measured using the average IoU metric (jaccard index) over all images in the
- dataset against the provided ground truth segmentations.
  - Req.3: The extracted features table.

Accuracy and feature extraction

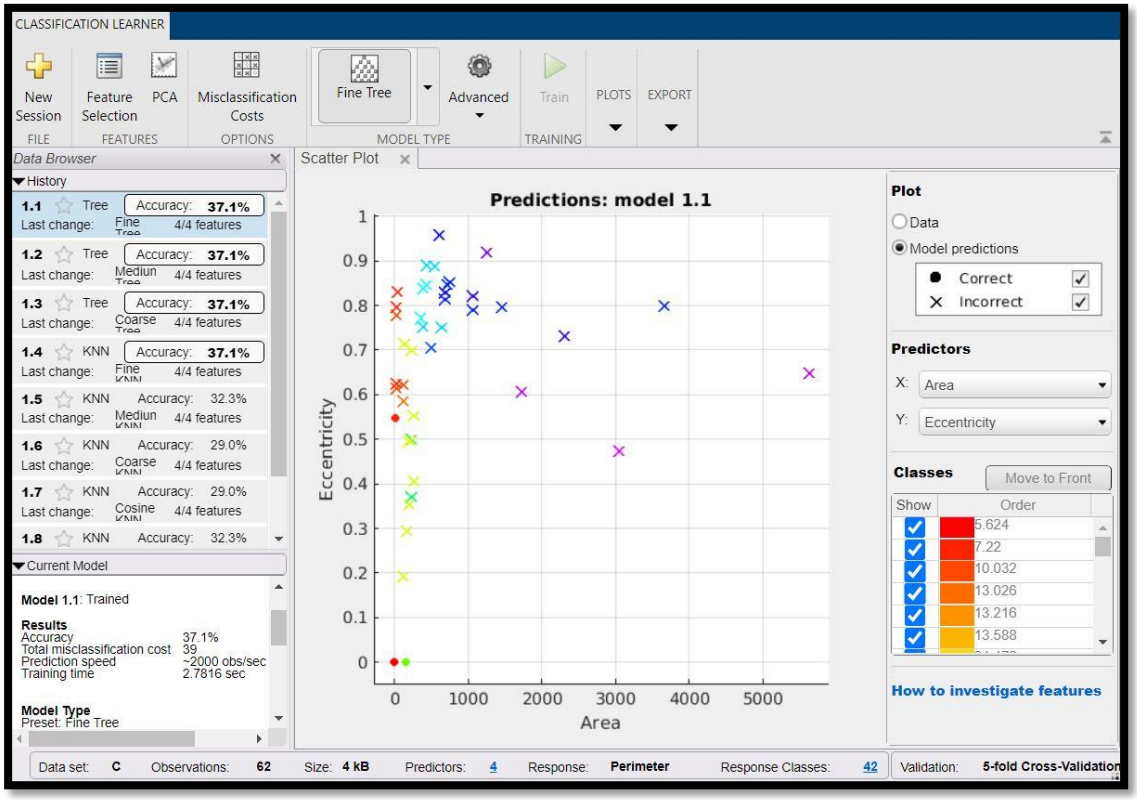
```
%accuracy
av=(sum/(NF1+NF2+NF3+NF4+NF5))*100;

%features_extraction
A1 = {stats1,stats2,stats3,stats4,stats5};
C = vertcat(A1{:})
```

Here we find the average and percentage of our segmented value accuracy. Concatenation of all the extracted features in one table for later classification.



- Req.4: Classification output and measuring the accuracy of the classifier on the test set.

Classification using classification learner



Here we trained our data and we obtained an accuracy of 37.1% against the test set.