Image Processing

CMP3108M

19701707

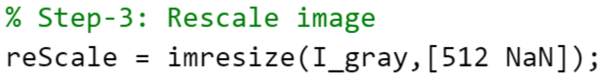
19701707@students.lincoln.ac.uk

Douglas Carrie

Task 1:

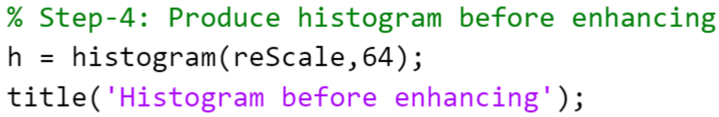
In task 1, the first step is to load the image under the variable name I.
Next, the image is converted to greyscale, this makes it easier to examine the image of blood cells as there is only two colours in the image now.

Then the image needed to be resized to a smaller size of height 512 whilst keeping the same aspect ratio for the image.

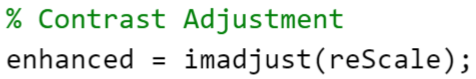


```
% Step-3: Rescale image
reScale = imresize(I_gray,[512 NaN]);
```

Originally the vector of ‘IMG_01’ was 1396 x 844 . After rescale ‘IMG_01’ was 847 x 512. The sample images do not all share the same resolution, so by rescaling them all it allows consistency and is essential for the histogram.

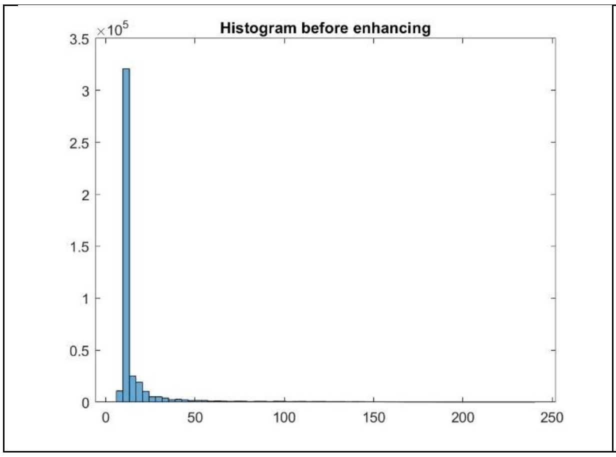


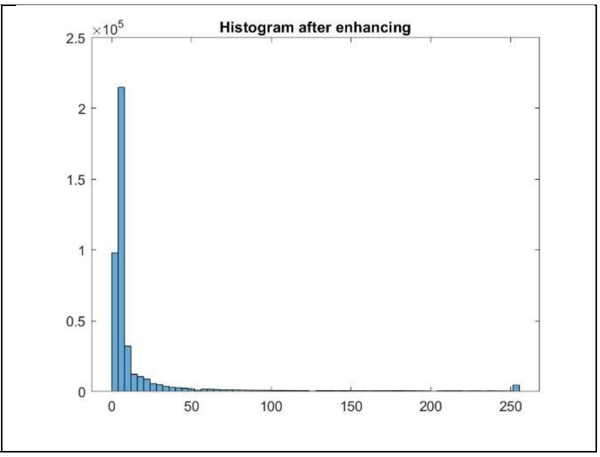
```
% Step-4: Produce histogram before enhancing
h = histogram(reScale,64);
title('Histogram before enhancing');
```



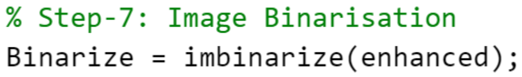
```
% Contrast Adjustment
enhanced = imadjust(reScale);
```

Normal Image vs. Enhanced Image

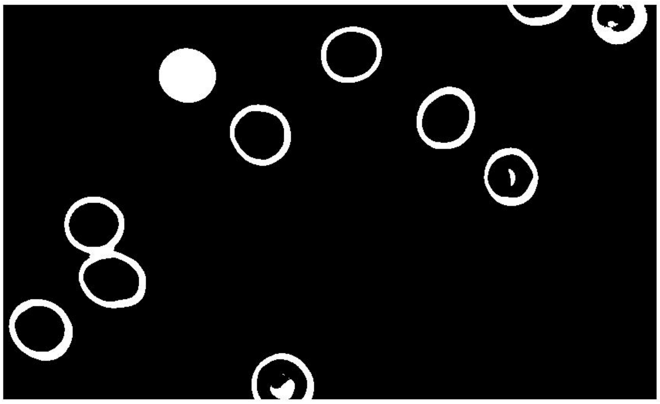




Converting the image to binary



```
% Step-7: Image Binarisation
Binarize = imbinarize(enhanced);
```



Binary image of IMG_01.png

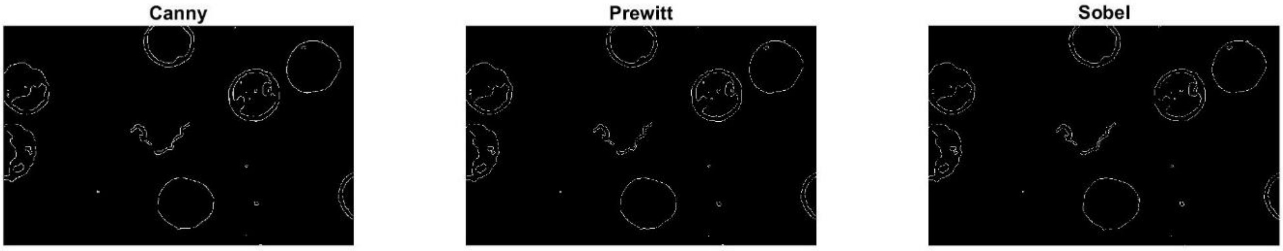
Some of the images did not respond well when converting to binary. Image ‘IMG_03.png’ only contained a string of bacteria near the centre of image and no blood cells. When binarized large white objects were created due to small light intensities fluctuations around the bacteria.

By changing the contrast level, it could help reduce large filled incorrect areas in the image, but in doing this it seemed to ruin the other images.

Task 2:

Edge detection

Using the MATLAB edge() function, three edge filters were applied to the binary image.
Below this the results of each filter.



Canny

Prewitt

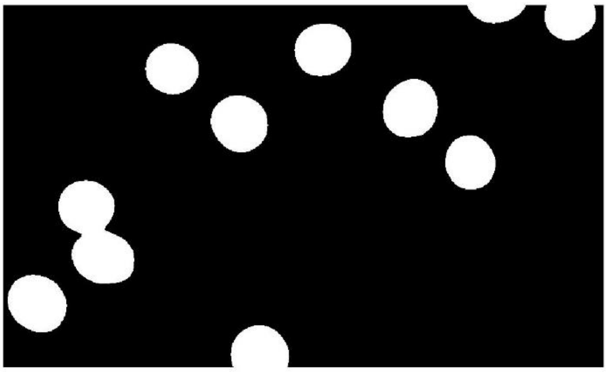
Sobel

Canny edge detection was the algorithm chosen. All three methods did yield very similar results, however Canny was chosen as it’s the most popular edge detection algorithm and its better at detecting circular edges and edges at the corner. It also provides a high quality edge map of the image (more pronounced lines) compared with Prewitt and Sobel.

Task 3:

To fill the red blood cells the images, first the image must be cropped on top,bottom,left and right. Then a 1 px white border is added to the top and right side of the image. The image then uses imfill() to fill the holes in the image. Next the white border top and right is remove from the image. The same process then happens again but padding the left and bottom side and then filling in the circles and removing the padding. The reason for not adding a white border on all sides at once is because when using imfill() after applying a border on every side, the algorithm would fill in the whole image white as its affectivly filling an outline of a rectangle. The reason the white border is added in the first place is because circles (red blood cells) near the edge of the image are not complete circles and the white thin border helps imfill() to regnise that they are circles that need to be filled. Without this the image would only fill complete circles, making the red blood cells at the edge ignored.

Below is the output image (IMG_01.png):



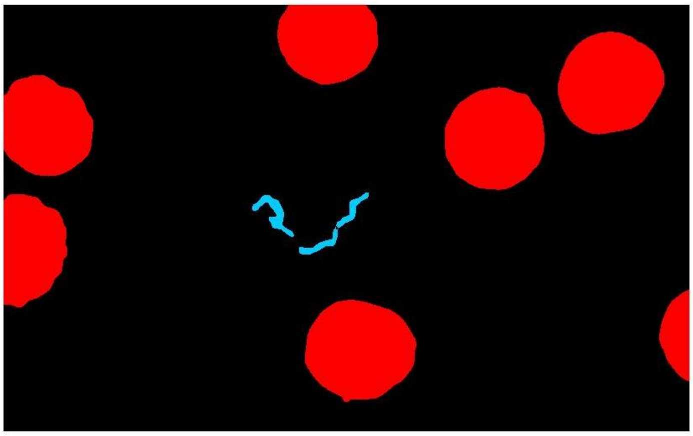
Task 4:

For this task, the blood cells needed to be sperated from the baceria by colour.

The first step was to remove any noise from the image. This was important small noise objects will make it harder to process and differentiate the cells and bacteria. Then to highlight the bacteria in blue, the function bwareafilt() was used to filter the three smallest white pixel areas in the image. The three smallest objects are the three strands of bacteria near the centre of the image. This is why the image had any noise removed first as the bwareafilt() would have recognized them as small areas. After this, a RGB colour map is created for the colour blue. Then the ind2RGB function is used to convert the image to colour and fill the white pixels with the RGB colour map blue.

As there are seven circles (red blood cells) in image 'IMG_01.png', the same function bwareafilt() can be used to find the 7 largest areas in the image. This is stored under the vairable 'blood' which can then follow a similar process to the bacteria. By creating a new RGB colour map of red. Then applying it using the ind2rgb() function.

The bacteria and blood are both now filled in with the correct colour but are being stored as two seperate images. With the function imadd() both these images can be combined together.



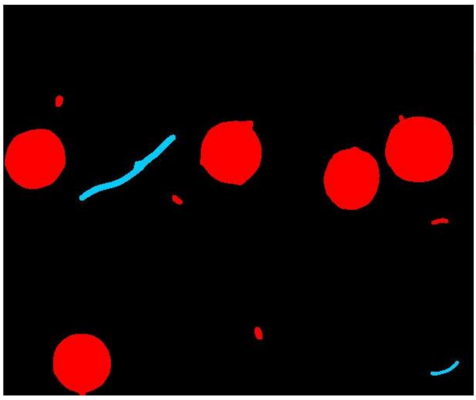
Output for task 4

Task 5:

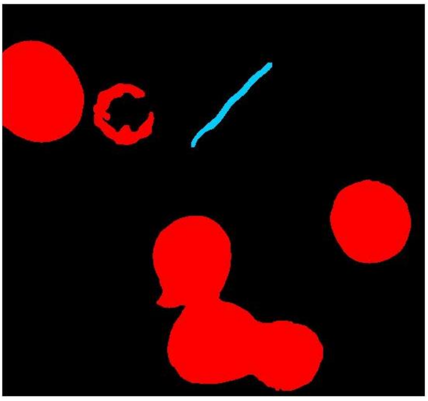
The script for task 4 is mostly specific for the image ‘IMG_11.png’. This task is about developing an automatic solution for every image.

The first step was using bwareafilt() function to remove small objects from the images, this is noise. Next bwlabel() function was used to define the connected components within the image. With this, regionprops() function can be used to find min and max lengths for areas of objects in the image. Then a variable aspectRatios can be created by taking the props variable of major axis length and performing an element wise right division with the props variable of minor axis length. This variable gives the calculated aspect ration for each object, next a threshold can be set for when to class an object as bacteria.
The threshold for the bacteria is set at 5 which detects the bacteria and not blood cells.
With this, a variable for only the blood cells can be created using the ismember() function to check the labeledImage is smaller the threshold. The colour of the cells can now be changed using the Cmap2 from task 4 and applied using ind2rgb() function.
A similar process happens for the bacteria mask where its checked labeledImage to find aspect ratios greater than or equal to the threshold. This is then converted to the colour blue using Cmap from task 4. Finally both images are combined together using imadd() to show blood cells and bacteria separated by colour.

IMG_07.png



IMG_13.png



All images are provided in folder ‘output’

The algorithm was able to distinguish between blood cells and bacteria to a good degree, however certain images like IMG_03.png struggled due to the method picking up incorrect details. It is able to detect bacteria in images but did struggle when the bacteria is connected to a blood cell.