Name: Brihat Ratna Bajracharya

Roll No.: 19/075

Solution to Exercise 2 (Image Processing and Pattern Recognition)

Q1. Load the images `pollen.tif` and `bone-scan-GE.tif`. Display them with `imtool`. Look at the pixel values in some area. Convert them into `double`. Display and look at the values again.

Solution

Matlab Script

```
pollen = imread('C:\Users\HP\Desktop\IPPLAssignment2\pollen.tif');
imtool(pollen)
pollen_double = double(pollen);
imtool(pollen_double)
bone = imread('C:\Users\HP\Desktop\IPPLAssignment2\bone-scan-GE.tif');
imtool(bone)
bone_double = double(bone);
imtool(bone_double)
```

Matlab Output

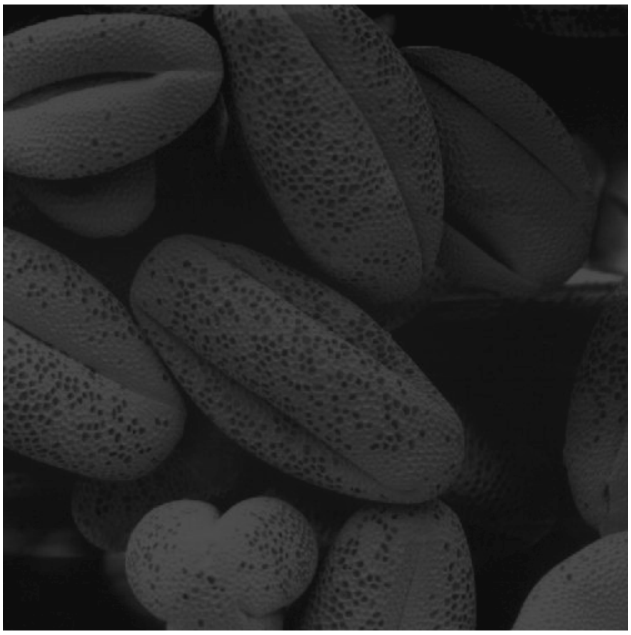

Figure 1: pollen.tif (original)

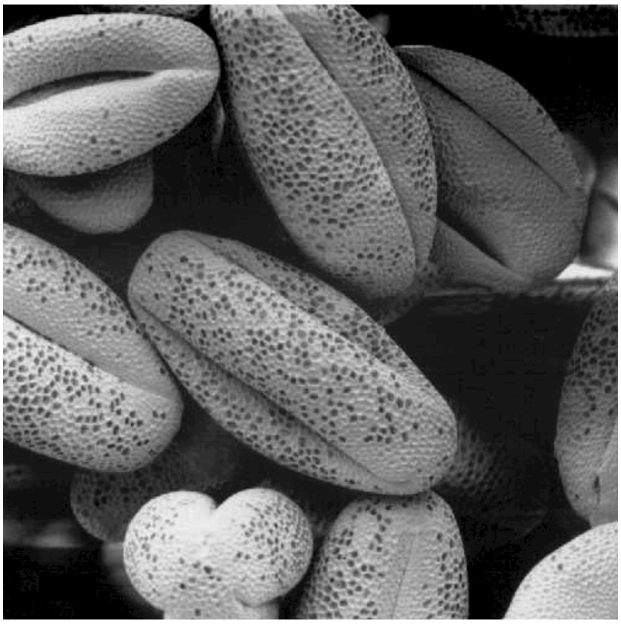

Figure 2: pollen.tif (double)

Figure 2 is contrast adjusted image (from [0,1] to [13,83]) using Adjust Constrast Tool of imtool. The pixel values in both original and double image is found out to be same for both images. Also contrast is adjusted from [0,1] to [0.92] in Figure 4 as well. It is clear that left bone scan is front side scan and right bone scan is the back side scan of a person

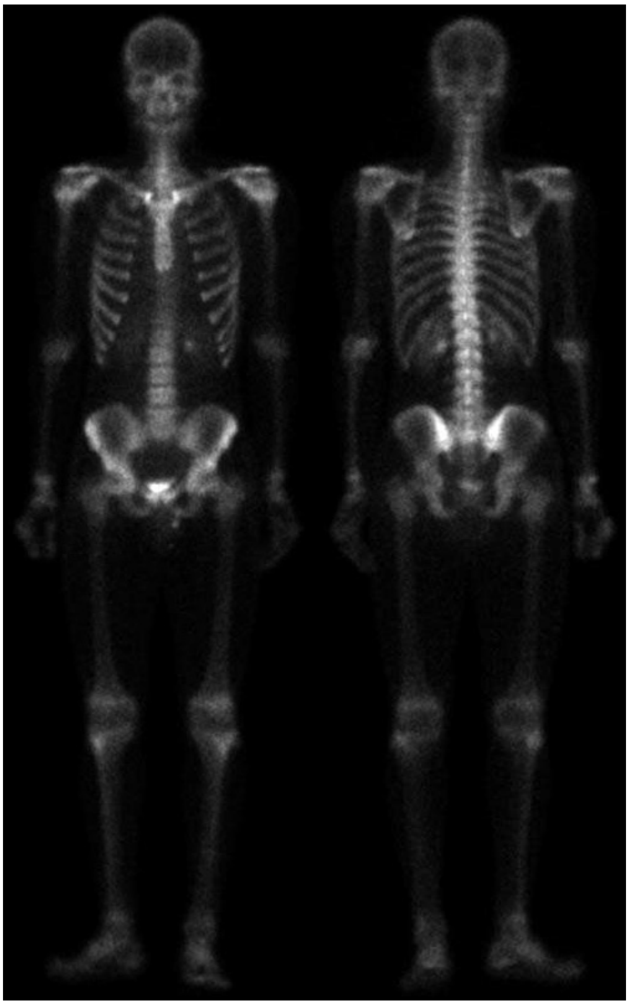

Figure 4: bone-scan-GE.tif (original)

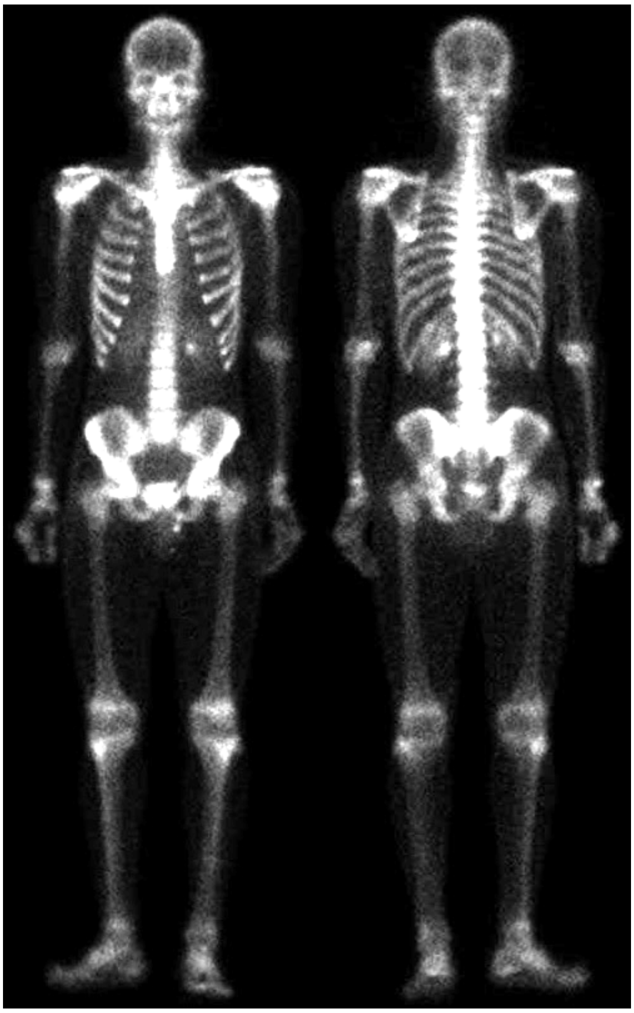

Figure 3: bone-scan-GE.tif (double, adjusted)

Q2. Use `thresholding` and `regionprops` to count the nuclei in the following images:

`nuclei1.tif` **`nuclei2.tif`**

How many nuclei are there in the picture?

What is the average area of a nucleus?

Which image was easier? Why? How can the segmentation be improved?

Try to write your programs in a generic fashion.

Comment and maybe even include functions.

Solution

Matlab Script

```
function e2q2()
    clc;
    clear;
    imtool close all;
    % Exercise 2 Question 2
    function count_nuclei(file_path, file_name)
        if exist([file_path, file_name])
            disp(['For image: ', file_name]);
            nuclei1 = imread(strcat(file_path,file_name));
            imtool(nuclei1)
            % imhist(nuclei1)

            nuclei1_gray = mat2gray(nuclei1);
            T_nuclei1_gray = graythresh(nuclei1_gray);
            nuclei1_gray_binary = im2bw(nuclei1_gray, T_nuclei1_gray);

            imtool(nuclei1_gray_binary)

            [L_nuclei1, num_nuclei1] = bwlabel(nuclei1_gray_binary);
            disp([' Number of nuclei: ', num2str(num_nuclei1)]);

            D_nuclei1 = regionprops(L_nuclei1, 'area');
            area_list_nuclei1 = [D_nuclei1.Area];
            avg_area_nuclei1 = mean(area_list_nuclei1);

            disp([' Avg Area of nuclei: ', num2str(avg_area_nuclei1)]);
            disp(' ');
            % hist(area_list_nuclei1)
        else
            disp(['For image: ', file_name]);
            disp(' This file path is invalid');
            disp(' ');
        end
    end

    count_nuclei('C:\Users\HP\Desktop\IPPLAssignment2\', 'nuclei1.tif')
    count_nuclei('C:\Users\HP\Desktop\IPPLAssignment2\', 'nuclei2.tif')
end
```

Matlab Output

```
For image: nuclei1.tif
 Number of nuclei: 80
 Avg Area of nuclei: 1113.125
```

```
For image: nuclei2.tif
 Number of nuclei: 18359
 Avg Area of nuclei: 4.8994
```

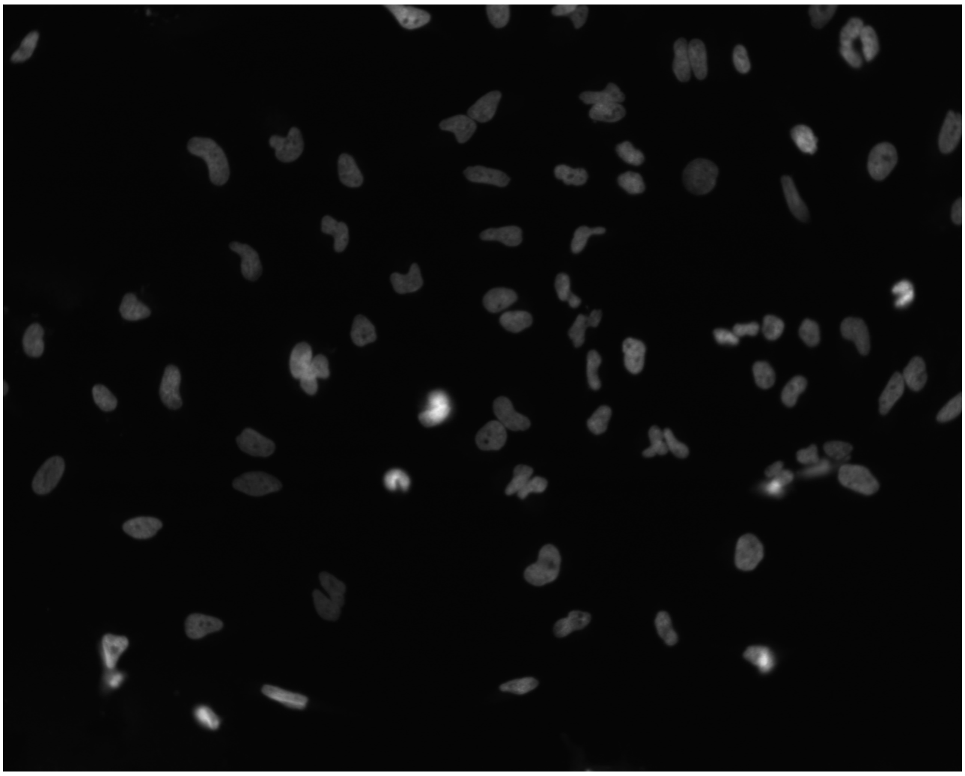

Figure 5: nuclei.tif (original)

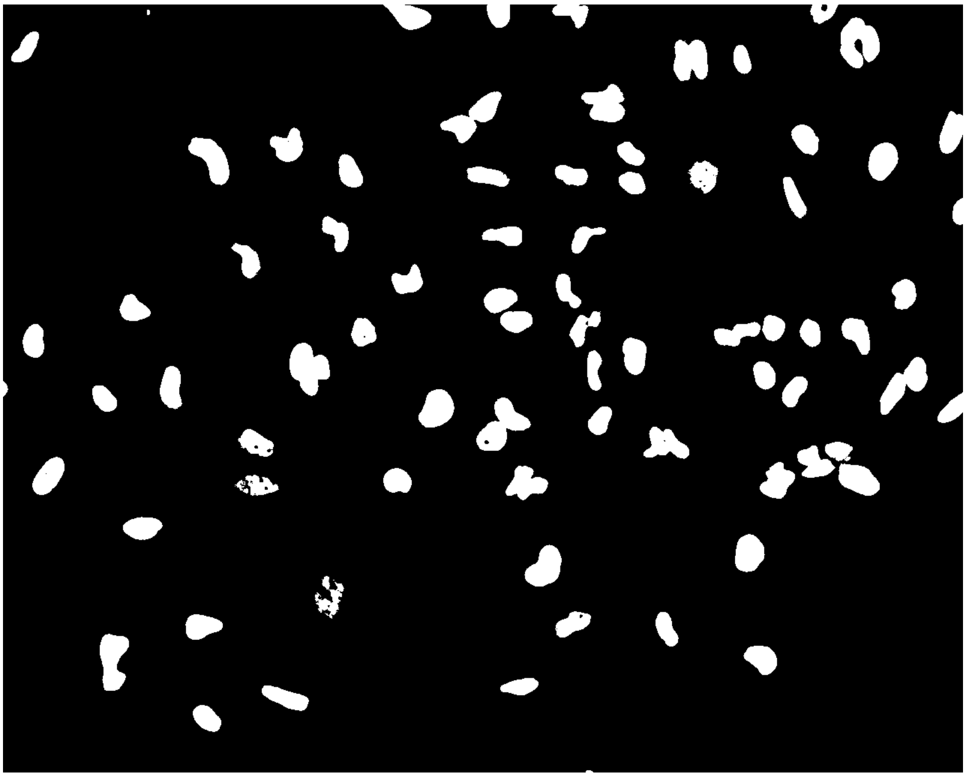

Figure 6: nuclei1.tif (grayscaled, binary)

Figure 7: nuclei2.tif (original)

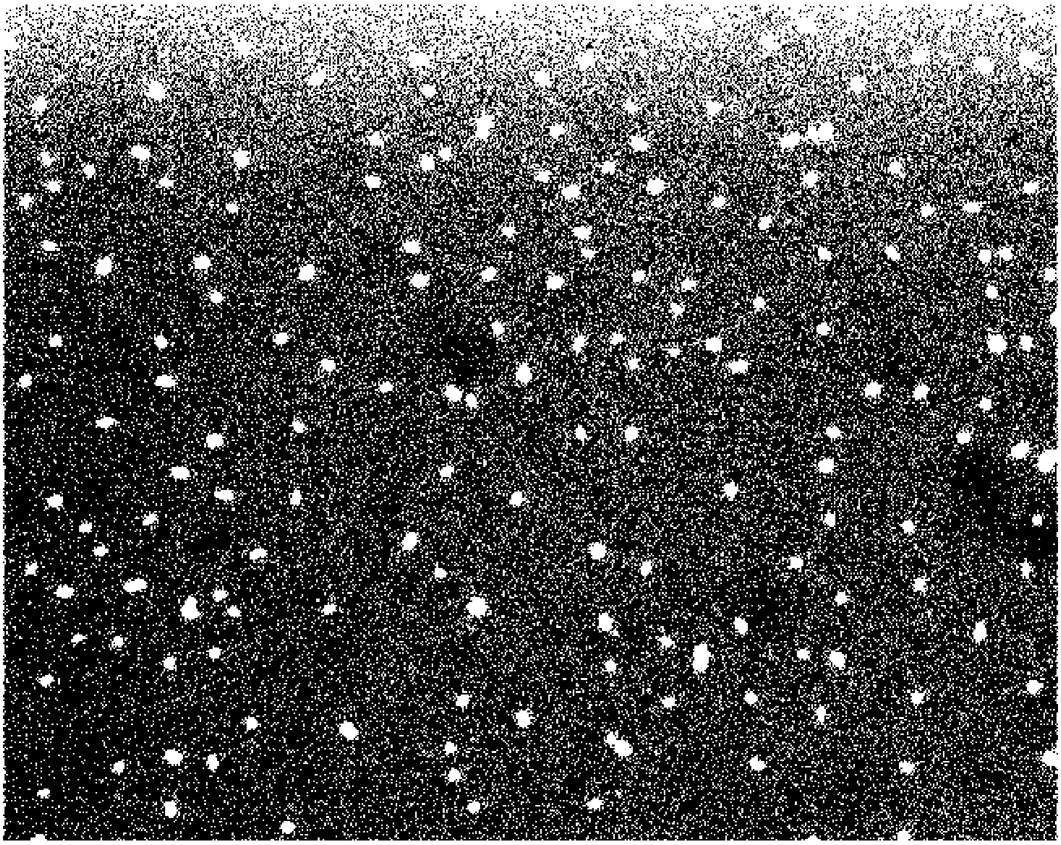

Figure 8: nuclei2.tif (grayscaled, binary)

Analysis

The total number of nuclei in *Figure 5* is **80** with average area of **1113.125** and the total number of nuclei in *Figure 7* is **18359** with average area of **4.8994.**

Figure 5 is easier than Figure 7 to interpret. In Figure 7, the nuclei were hidden due to image contrast and by using *thresholding operation* in Matlab, we can see the hidden nuclei as shown in figure 8. Finally, the number of nuclei were counted using **`regionprops`** command with **`Area`** property

Q3. Eliminate "bad" rice granules from the `rice.png` picture. Use `regionprops` to identify the rice you want to eliminate. Use the `find` function to change eliminate these grains from the labeled image. (You will probably want to write a loop that goes through each object in the labeled file)

Compare the original image to your new image (`imshow` and `hist`). How well did you do? Qualitatively, how many real grain got eliminated/how many bad grains remain?

Solution

Matlab Script

```
clc;
clear;
close all;

figure('Name', 'Question 3', 'NumberTitle', 'off');
font_size = 10;

% read image: rice.png
rice = imread('C:\Users\HP\Desktop\IPPLAssignment2\rice.png');

% show actual image: rice.png
subplot(2, 3, 1);
imshow(rice);
title('Rice (Actual Image)', 'FontSize', font_size, 'Interpreter', 'None');

% change png image to grayscale and calculate the threshold value,
% then create a new binary image using this calculated threshold value
rice_gray = mat2gray(rice);
T_rice_gray = graythresh(rice_gray);
```

```
rice_gray_binary = im2bw(rice_gray, T_rice_gray);

% plot gray image and binary image
subplot(2, 3, 2);
imshow(rice_gray);
title('Rice (Grayscale)', 'FontSize', font_size, 'Interpreter', 'None');
% imtool(rice_gray)

subplot(2, 3, 3);
imshow(rice_gray_binary);
title('Rice (Binary)', 'FontSize', font_size, 'Interpreter', 'None');
% imtool(rice_gray_binary)


% enumerate all the objects in the figure
[L_rice, num_rice] = bwlabel(rice_gray_binary);

% then display no of object (this gives number of rice grain
disp(['Number of rice grain: ', num2str(num_rice)]);
% num_rice

% regionprops command is used to obtain quantitative information about the objects
% gray_image is needed for 'MeanIntensity' for 'table'
D_rice = regionprops('table', L_rice, rice_gray, 'Area', 'MeanIntensity',
'SubarrayIdx');
% D_rice = regionprops(L_rice, 'area');

% making copy of binary image for two different analysis
rice_gray_binary_good_area = rice_gray_binary;
rice_gray_binary_good_mi = rice_gray_binary;


% sorting Area prop for analysis
area_list_rice = [D_rice.Area];
sort(area_list_rice, 'descend');

% specify threshold value from sorted list
% then remove all those element that do not match threshold value
thres = 120;
D_rice.isGreater = D_rice.Area > thres;
objRemoveIdxA = find(~D_rice.isGreater);
```

```
for i = find(~D_rice.isGreater).'
      rice_gray_binary_good_area(D_rice.SubarrayIdx{i,1},D_rice.SubarrayIdx{i,2}) =
false;
end
% imtool(rice_gray_binary);

% plot this binary image (gives only good rice grain using Area prop)
subplot(2, 3, 4);
imshow(rice_gray_binary_good_area);
title('Rice (Binary, Good Area)', 'FontSize', font_size, 'Interpreter', 'None');

% display number of eliminated rice grains
eliminated_rice_grains = length(objRemoveIdxA);
disp(['Number of eliminated rice grains (prop: Area): ',
num2str(eliminated_rice_grains)]);



% sorting MeanIntensity prop for analysis
mi_list_rice = [D_rice.MeanIntensity];
sort(mi_list_rice, 'descend');

% specify threshold value from sorted list
% then remove all those element that do not match threshold value
thres = 0.80;
D_rice.isGreater = D_rice.MeanIntensity > thres;
objRemoveIdxMI = find(~D_rice.isGreater);
for i = find(~D_rice.isGreater).'
        rice_gray_binary_good_mi(D_rice.SubarrayIdx{i,1},D_rice.SubarrayIdx{i,2}) =
false;
end

% plot this binary image (gives only good rice grain using MeanIntensity prop)
subplot(2, 3, 5);
imshow(rice_gray_binary_good_area);
title('Rice (Binary, Good MI)', 'FontSize', font_size, 'Interpreter', 'None');

% display number of eliminated rice grains
eliminated_rice_grains = length(objRemoveIdxMI);
disp(['Number of eliminated rice grains (prop: MI): ',
num2str(eliminated_rice_grains)]);
```

Matlab Output

```
Number of rice grain: 151
Number of eliminated rice grains (prop: Area): 78
Number of eliminated rice grains (prop: MI): 93
```

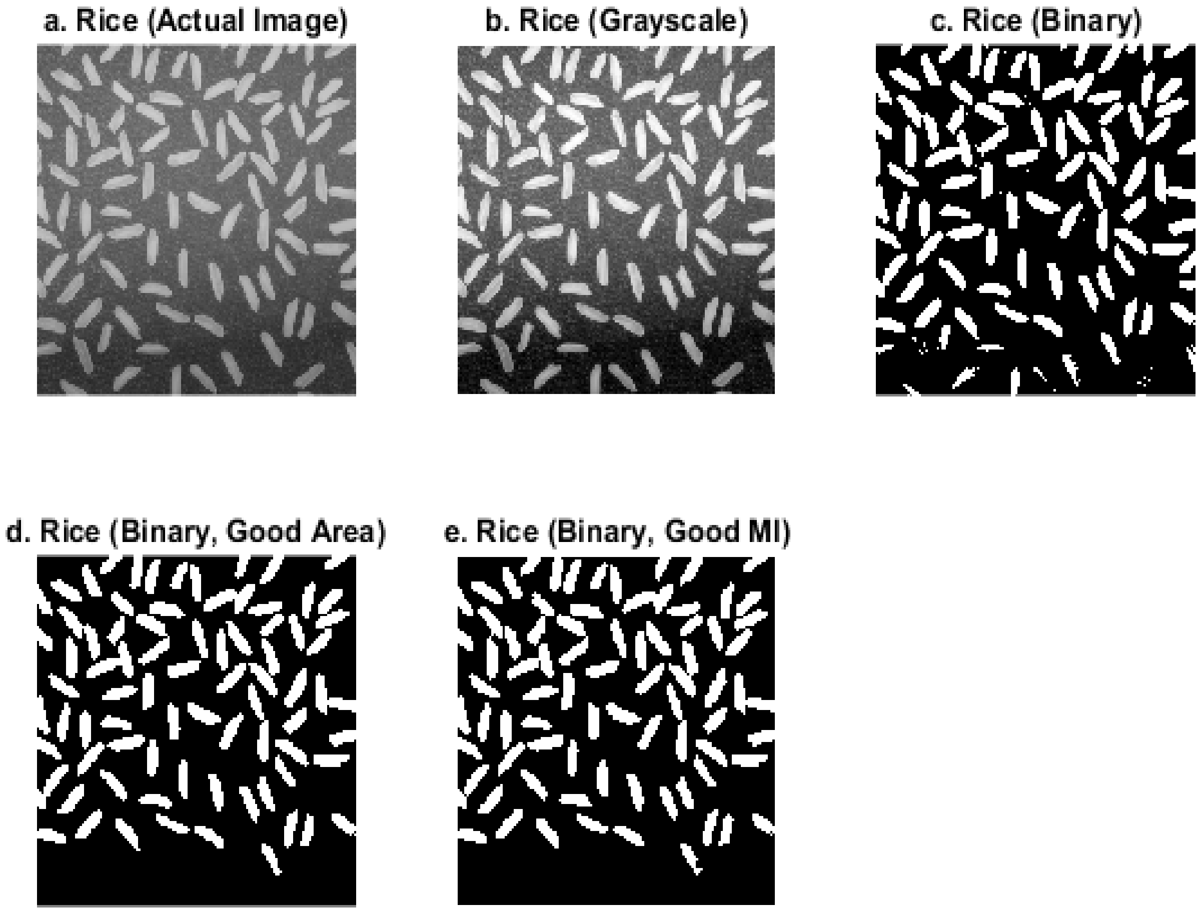

Figure 9: Separating bad rice grains from rice.png

Analysis

The original image (Figure 9a.) has darker side at the bottom of the image. So, when applying grayscaling function (Figure 9b.), some of the good rice grains were not detected properly in binary image (Figure 9c.) and thus around *eight (8)* good grains were eliminated in our analysis. Also the rice grains around the edges of the image is connected so that the area of those half and quarter rice grains are detected big. So we can see some half and quarter sized rice grains around the edges (Figure 9d and Figure 9e.). As such around *seven (7)* bad grains remained after the analysis. But looking at the overall detection, we can say that almost all of the bad grains are completely removed from the original image.