Project 5

Object Counting by Morphological Processing

Md Zobaer Islam
PhD student, Electrical Engineering
Oklahoma State University
CWID: A20270547

Explanation of submission files (only .m files in alphabetical order)

File name	What it is---
part1_a_b.m	For showing individual and clustered discs of the noise free image
part1_c_1_counting.m	Implementation of counting algorithm, showing the detected discs as single pixel
part1_c_2_circles.m	Showing the detected discs as circular rings, using our algorithm and Hough transform
part2_a_b_c.m	Color segmentation, followed by morphological processing, for the noisy image
part2_d_1.m	For showing individual and clustered discs, for the noisy image
part2_d_2.m	Implementation of counting algorithm, showing the detected discs as single pixel, for the noisy image
part2_d_3.m	Showing the detected discs as circular rings, using our algorithm and Hough transform, for the noisy image

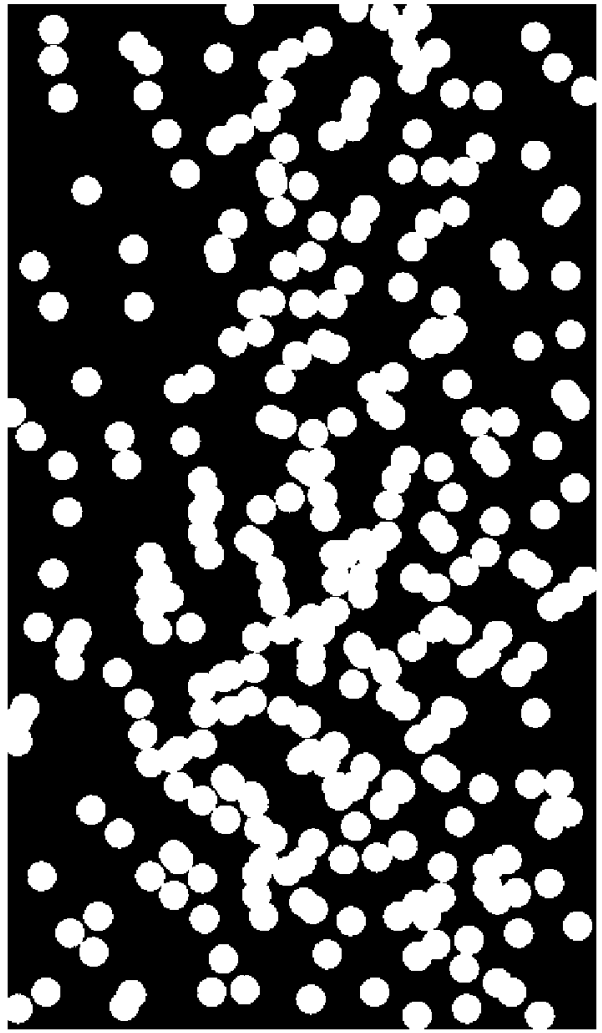
Part I: Object Counting from a Binary Image

(a) and (b): Isolated and Clustered Discs – MATLAB code

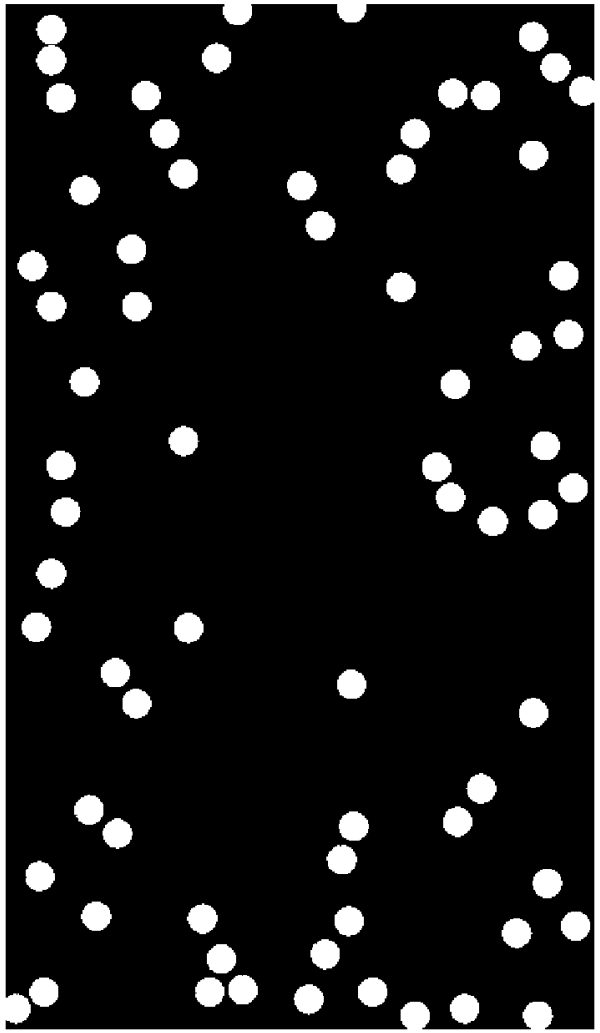
```
clc; clear; close all; f=20;
I = im2double(imread('cells.bmp'));s = size(I);
figure(1); imshow(I); title("Original image",FontSize=f);
B = im2double(imread('disc.bmp'));
Area=sum(sum(B));
[X,Y]=size(I);
[L,num]=bwlabel(I,8);
count_id=0; IID=zeros(X,Y);
count_cd=0; ICD=zeros(X,Y);
for i=1:num
    [r c]=find(L==i);
    ni=length(r);
    if ni-Area<20
        count_id=count_id+1;
        IID(find(L==i))=1;
    else
        count_cd=count_cd+1;
        ICD(find(L==i))=1;
    end
end
figure(2);imshow(IID);title(sprintf('Individual discs only (count = %d)',count_id),FontSize=f);
figure(3);imshow(ICD);title(sprintf('Connected discs only (count = %d)',count_cd),FontSize=f);
```

(a) and (b): Isolated and Clustered Discs

Original image

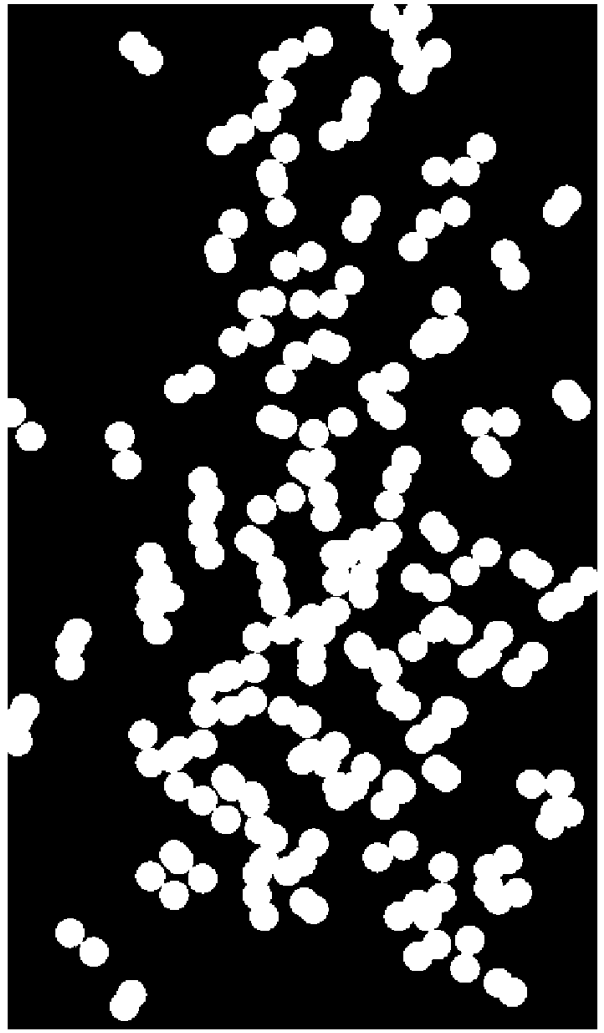


Individual discs only (count = 70)



Fractional disc at the border

Connected discs only (count = 59)



Fractional cluster at the border

Total number of connected components = 70+59 = 129, which is the same as what we got using bwlabel()

Isolated disc count = **70** (it includes fractional single discs at the border)

Clustered discs count = **59** (it includes clustered discs at the border with fractional discs too)

(c) Disc detection, visualization as single pixel and counting – MATLAB code

```
clc; clear; close all;f=14;I = im2double(imread('cells.bmp'));s = size(I);
B = im2double(imread('disc.bmp'));Area=sum(sum(B));
[L,num]=bwlabel(I,8); E=[0 1 0; 1 1 1; 0 1 0]; countdisc = 0; Iglobal = zeros(s);
for iii = 1:num
    A = zeros(s); A(find(L==iii))=1;
    B1 = imerode(B,E); %New structuring element
    D = imerode(A,B1); [r,c] = find(D==1);
    n = length(r); %the number of white pixels in that connected component
    ranks = zeros(1,n);
    for i=1:n
        G = D; G(r(i),c(i))=0; H = imdilate(G,B1);
        ranks(i) = sum(xor(A,H),'all');
    end
    [rval, sortind] = sort(ranks,"descend"); %sort in order of ranks
    tempi = zeros(s); complast = 0;
    for i=1:length(sortind)
        tempi(r(sortind(i)), c(sortind(i))) = 1; %add the next highest ranked pixel
        J = imdilate(tempi,B1); compnew = sum(sum(J));
        if (compnew <= complast+70) %if there's no significant improvement
            %since there's no significant improvmnt aftr addin this pxl, set it to 0
            tempi(r(sortind(i)),c(sortind(i)))=0;
        end
        complast = compnew;
    end
    countdisc = countdisc+sum(sum(tempi)); Iglobal(find(tempi==1))=1;
end
%Uncomment the following line to store the single pixel representation data
% imwrite(Iglobal,'detected_disc.bmp');
figure(2); imshow(Iglobal); title("Detected discs as single pixel",FontSize=f);
countdisc
count = sum(sum(Iglobal))
```

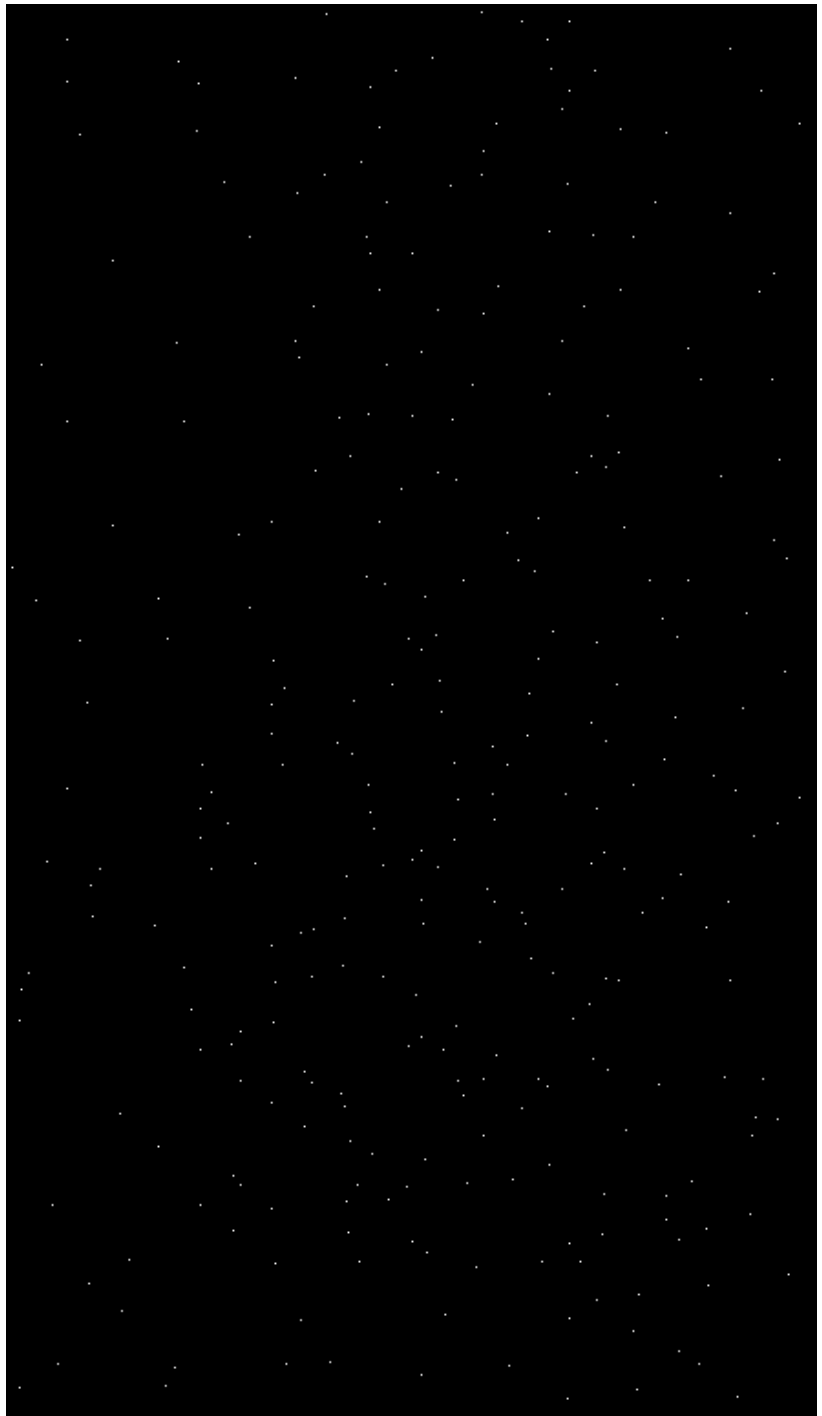
(c) Disc detection and visualization as hollow circles-MATLAB code

1. The image where each disc was represented by a single white pixel was stored as detected_disc.bmp file, for accessing it promptly without recalculation.

2. The image where individual discs were represented as rings by using Hough transformation was stored too from the provided code demo5.m using `imwrite(C,'rings_hough.bmp');`

3. The image created by our own algorithm and the one found by using Hough transform are shown overlayed in a single figure window for comparison. Red color has been added to the circles in the Hough transformed image to show the difference clearly

```
clc; clear; close all;f=16;
DD = im2double(imread('detected_disc.bmp'));
s = size(DD);
T=imread('disc.bmp');
SE=bwmorph(T,'remove');
B=imdilate(DD,T);
C=imdilate(DD,SE);
figure(1); subplot(121);imshow(C);
title("Visulization of discs using circles",Fontsize = f);
H = im2double(imread('rings_hough.bmp'));

%Add red color
HC = zeros(size(H,1), size(H,2), 3, 'uint8');
HC(:,:,1) = 255*H;
figure(1); subplot(122);imshow(HC);
title("Using Hough Transform",Fontsize = f);

%Show overlayed rings
figure(2);imshow(C);hold on; hc = imshow(HC);
set(hc, 'AlphaData', H);
```



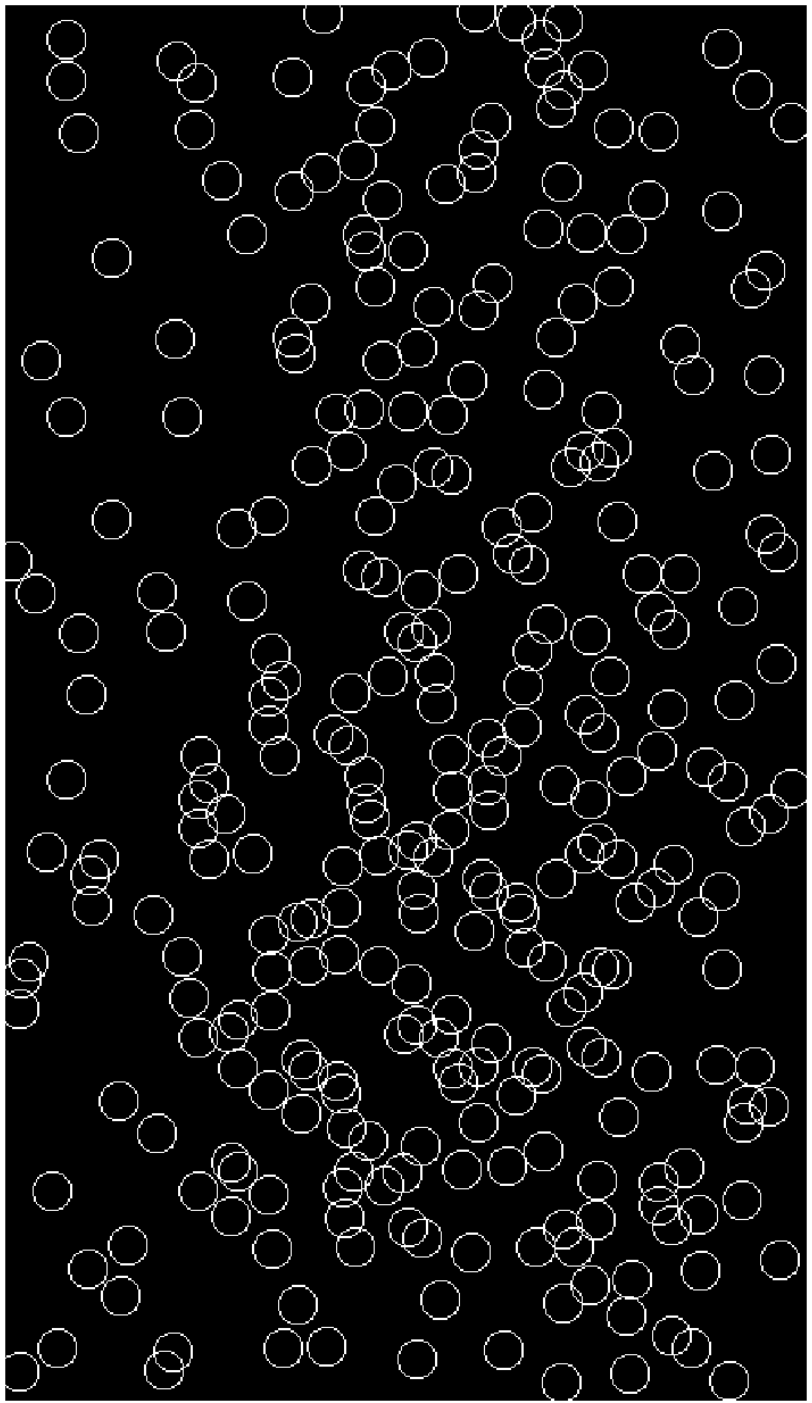
(c) Results - detected discs as single pixels (left) and rings (right)

Both are found using our own algorithm

```
countdisc =

   301


count =

   301

>>
```

Total number of discs 301

Visulization of discs using circles

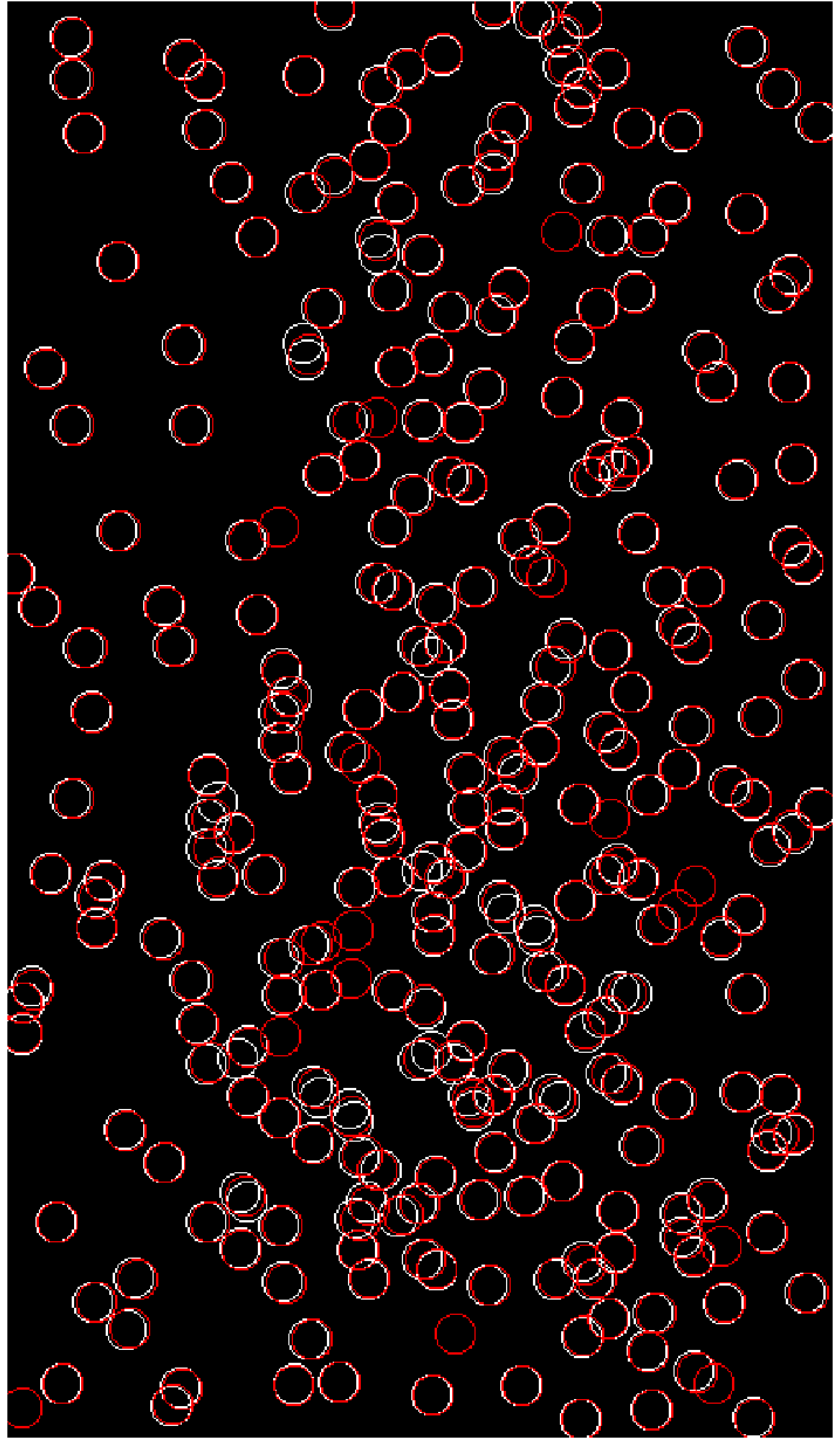


(d) Results - detected discs as rings using our algorithm and Hough transform (overlayed)

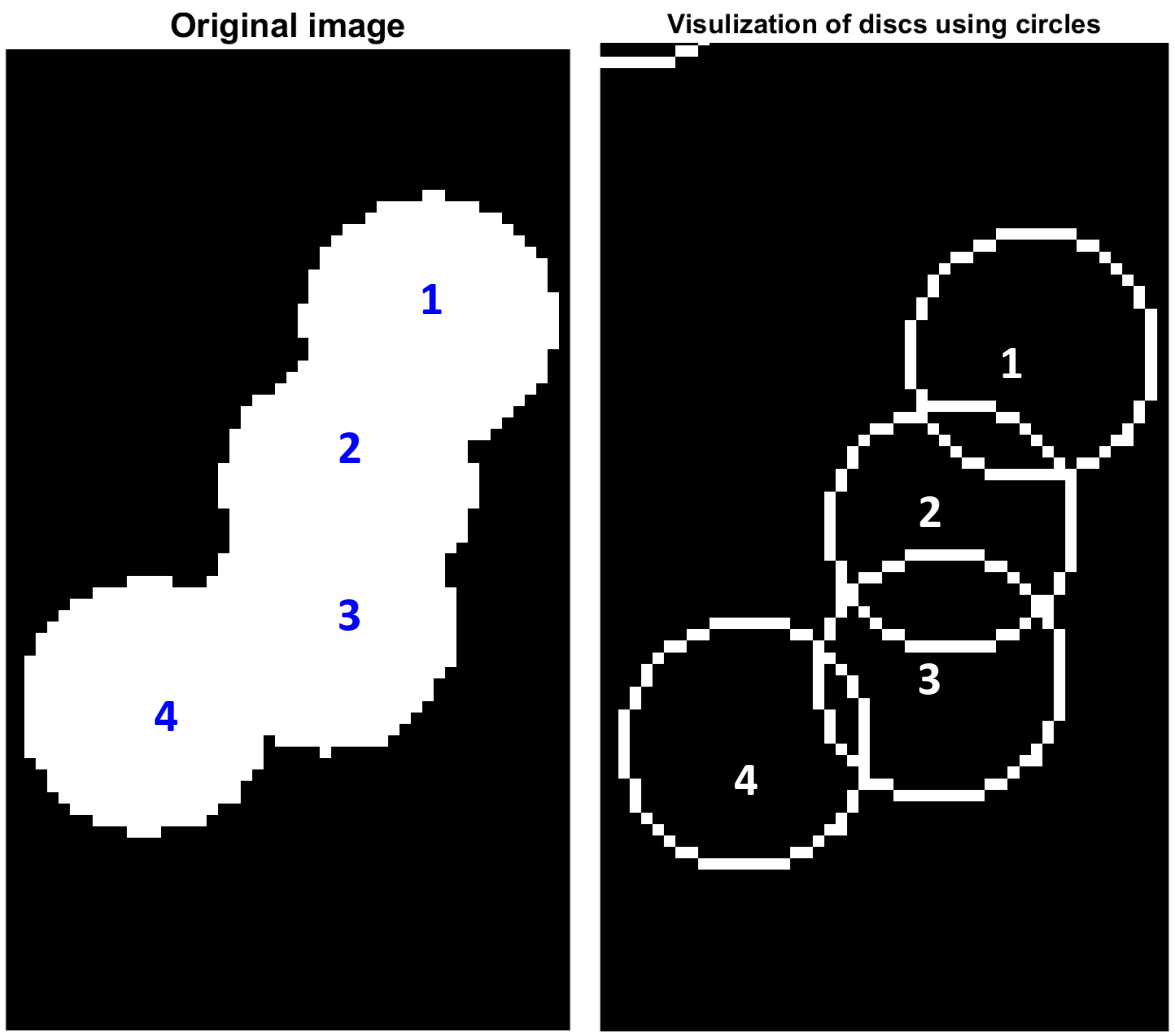
Legend:

White: Our algorithm

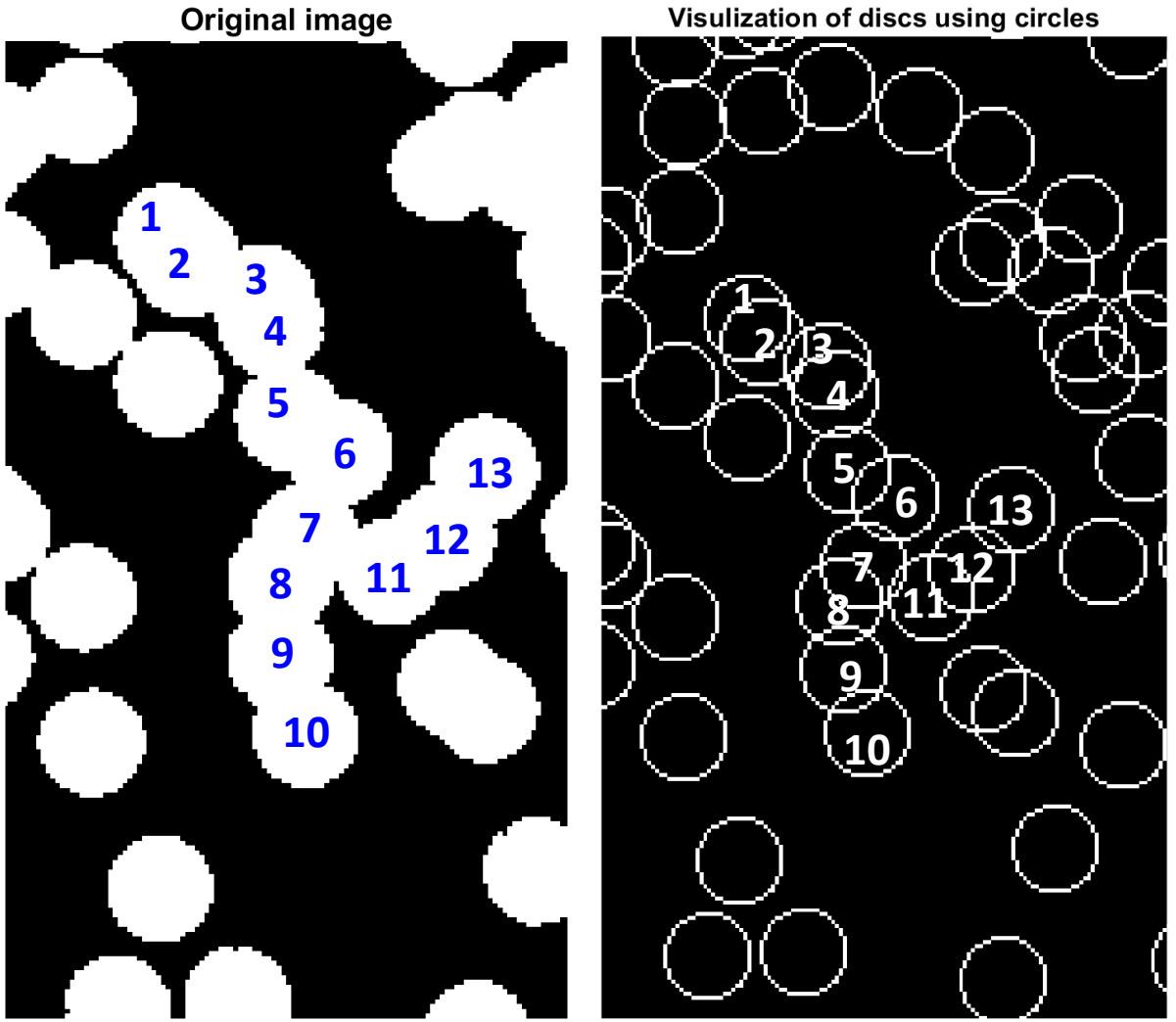
Red: Hough transform (using the given code)



(d) Region by region analysis

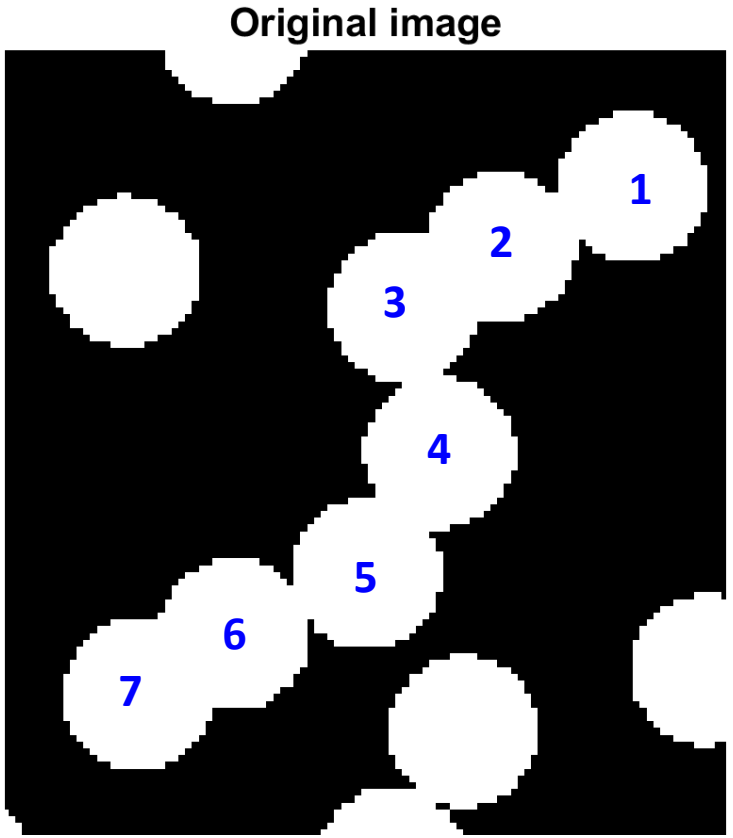


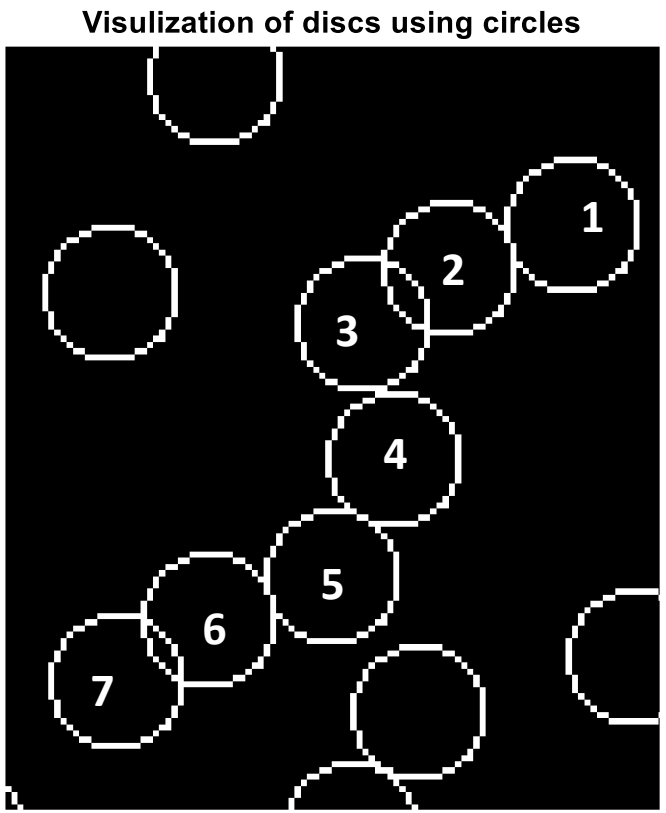
Accurately detected by using our algorithm.



Accurately detected by using our algorithm.

(d) Region by region analysis





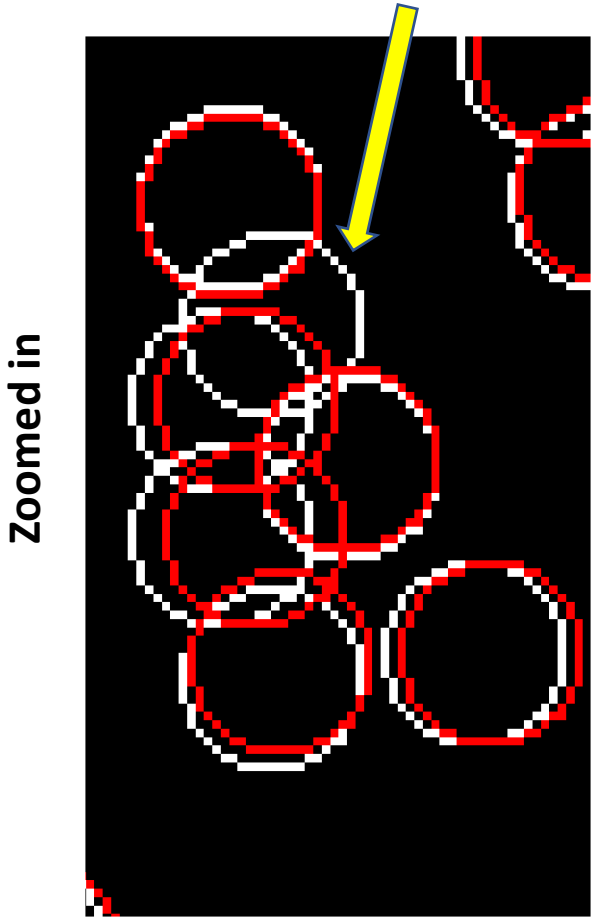
Accurately detected by using our algorithm.

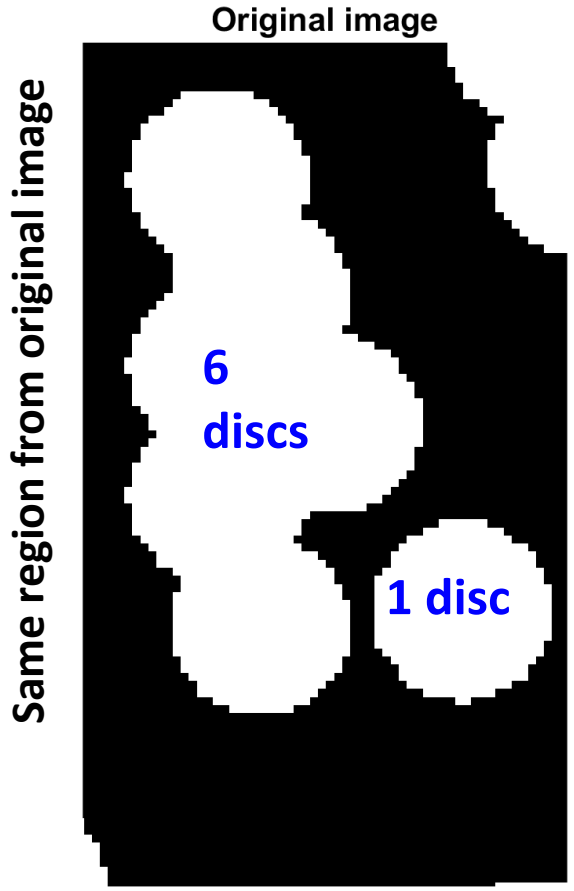
(d) Region by region analysis

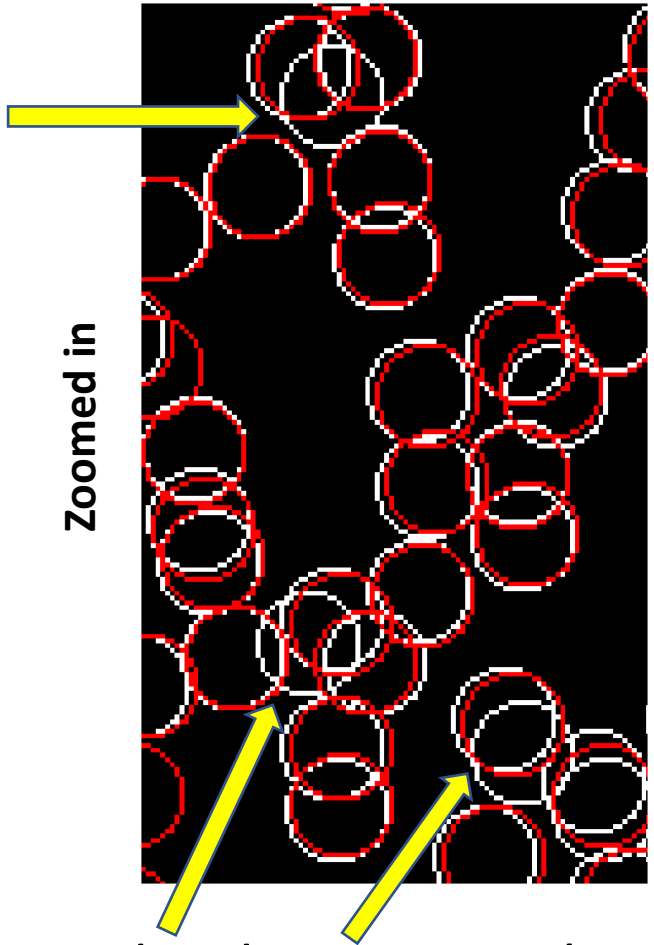
Legend:

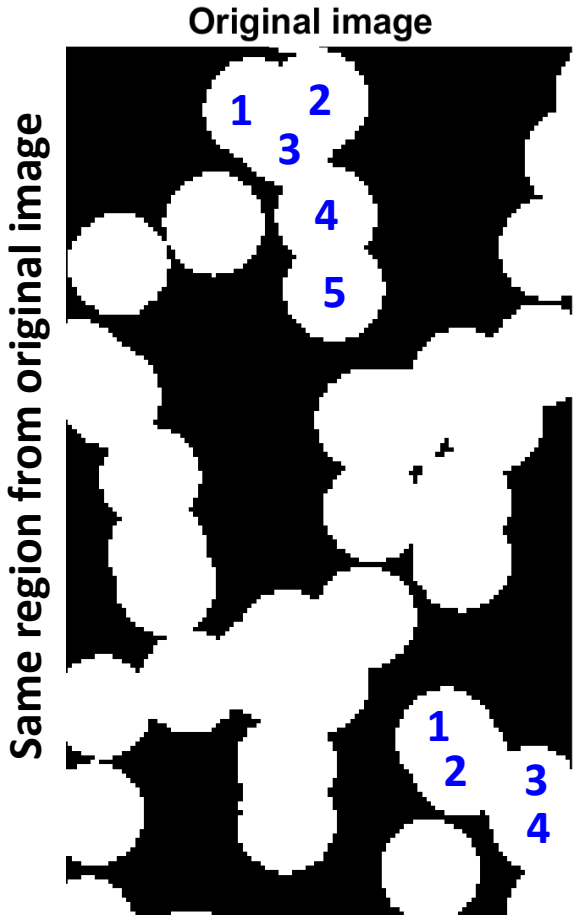
White: Our algorithm

Red: Hough transform (using the given code)









Hough transformation is doing under-detection in these cases

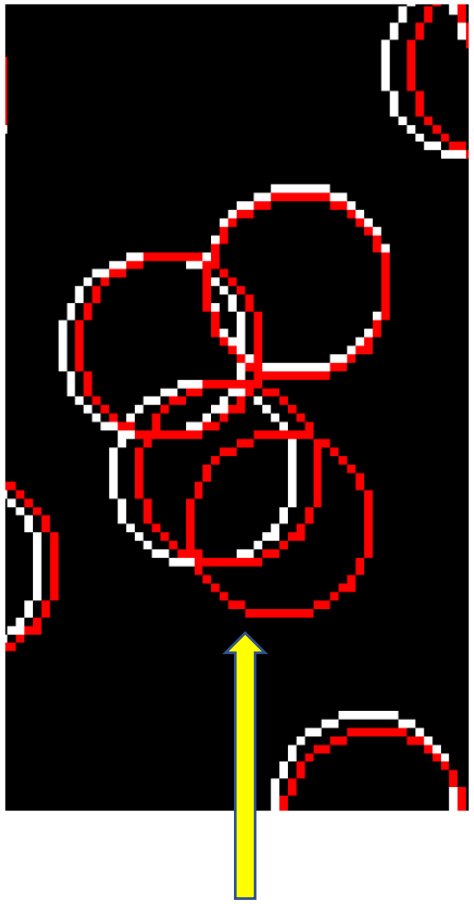
(d) Region by region analysis

Legend:

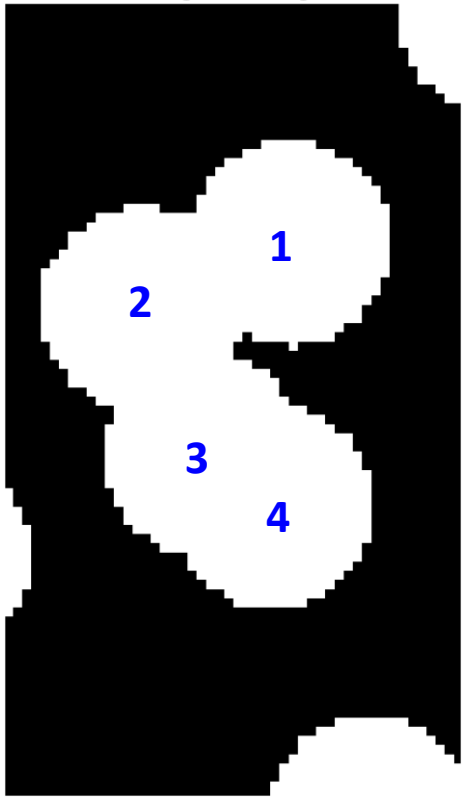
White: Our algorithm

Red: Hough transform (using the given code)

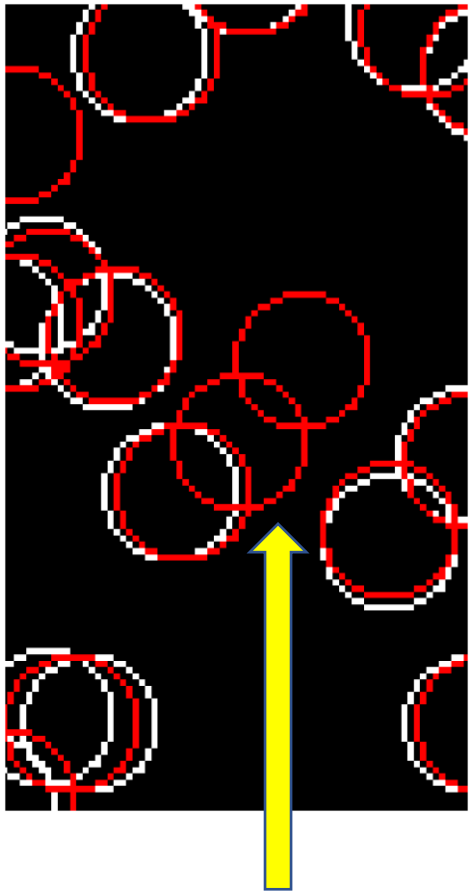
Zoomed in



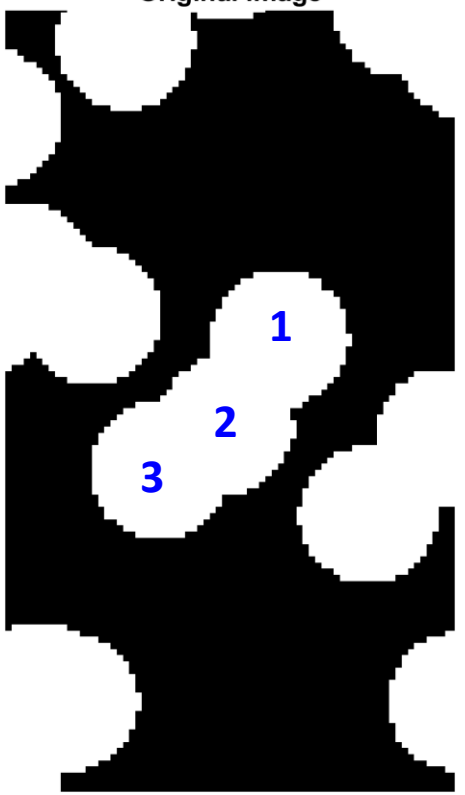
Same region from original image



Zoomed in



Same region from original image



Our algorithm is doing under-detection in these cases

(c)+(d) Discussion:

- It has been observed that the circles corresponding to the same discs are almost overlapped. There are some horizontal and vertical shifts noticed between red and white circles, but the shifts are insignificant, and more importantly, these shifts will not impact the counting result or detection result.

- More interesting observation is that the Hough transform detected **287 discs** (found from the size of centers array in the given code) while our algorithm detected **301 discs**. Both of Hough transform and our algorithm is doing some under-detection, but Hough transform is doing it more, resulting in overall lower number of discs. Hence our algorithm is giving a better estimate of the number of discs.

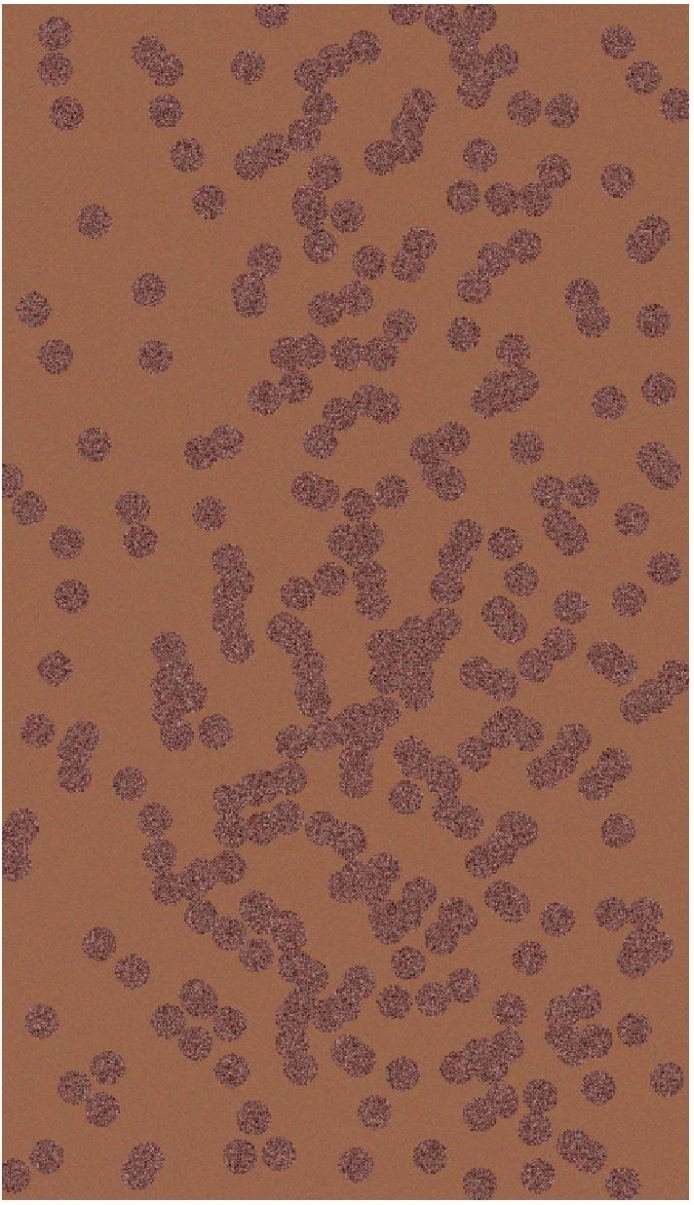
Part II: Object Counting from a Color Image

(a) + (b) + (c) Color segmentation and Morphological Processing-MATLAB code

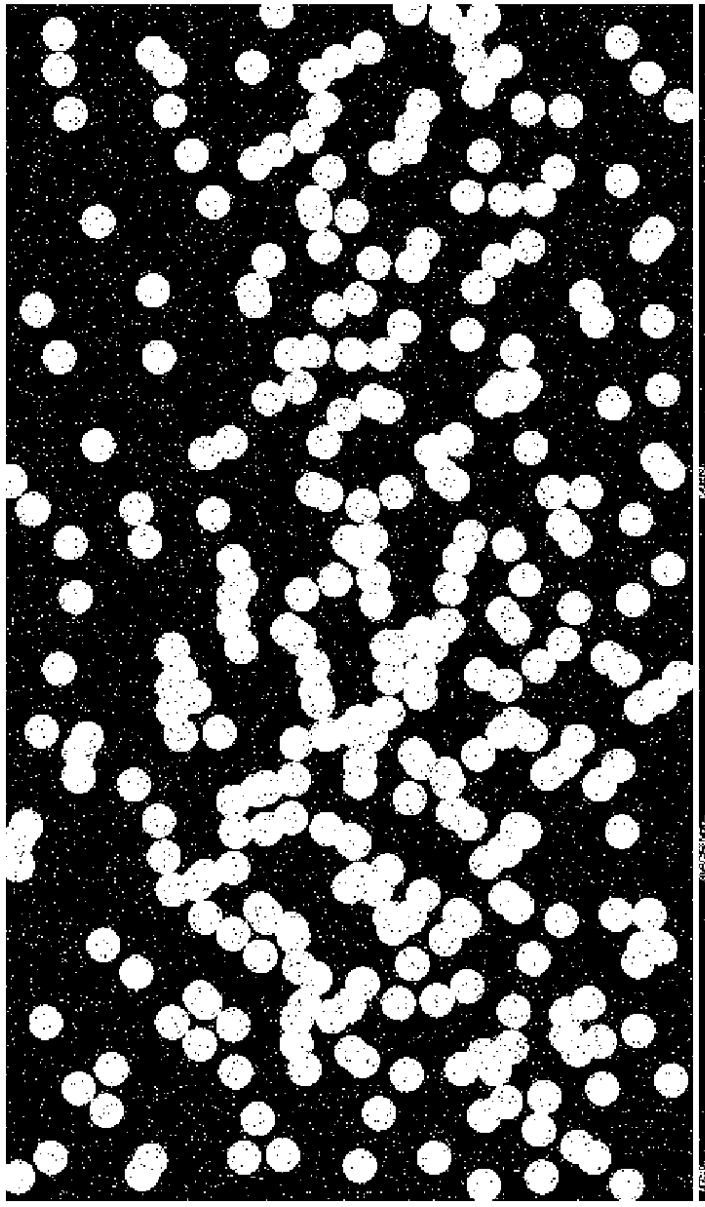
```
clc; clear; close all;f =16;
I = imread('color_cell.bmp');I = im2double(I);IBW = imread('cells.bmp');IBW = im2double(IBW);
R = I(:,:,1);G = I(:,:,2);B = I(:,:,3);s = size(I);
%Two windows are taken manually by using data tip option of MATLAB
y_min1 = 224; y_max1 = 244; x_min1 =457; x_max1 = 471;
y_min2 = 304; y_max2 = 323; x_min2 =673; x_max2 = 690;
samples=[reshape(I(x_min1:x_max1,y_min1:y_max1,1:3),[],3);reshape(I(x_min2:x_max2,y_min2:y_max2,1:3),[],3)];
mean_val = mean(samples); covariance = cov(samples);
IM = zeros(s(1),s(2)); thresh_mahal = 3.1; thresh_eucl = .09;
for i = 1:s(1)
    for j = 1:s(2)
        x = [R(i,j), G(i,j), B(i,j)]; d = x-mean_val;
        D_mahal = sqrt(d*inv(covariance)*d'); D_eucl = sqrt(d*d');
        if min(D_mahal) < thresh_mahal
            IM(i,j) = 1;
        end
        if min(D_eucl) < thresh_eucl
            IE(i,j) = 1;
        end
    end
end
figure(1);imshow(I); title("Original color image",FontSize=f);
figure(2);imshow(IM); title("Color segmentatn using Mahalanobis dist.",FontSize=f-2);
figure(3);imshow(IE); title("Color segmentatn using Euclidean dist.",FontSize=f);
IQA_init_mahal = sum(xor(IM,IBW),'all')
IQA_init_eucl = sum(xor(IE,IBW),'all')

%Perform erosion to get rid of background noise
se1=strel('disk',1);Ierode = imerode(IM,se1);
figure(4);imshow(Ierode); title("Eroded Image",FontSize=f-2);

%Perform closing (dilation followed by erosion) to fill the gaps on discs
se2 = strel('disk',2); Iclose1 = imdilate(Ierode,se2); Iclose2 = imerode(Iclose1,se1);
figure(5);imshow(Iclose1); title("Image after erosion and dilation",FontSize=f-2);
figure(6);imshow(Iclose2); title("Final image (after erosion and closing)",FontSize=f-2);
IQA_mahal_morph1 = sum(xor(Iclose1,IBW),'all')
IQA_mahal_morph2 = sum(xor(Iclose2,IBW),'all')
imwrite(Iclose2,"seg_morph.bmp");
```

(a) and (b): Color segmentation - results

Original color image

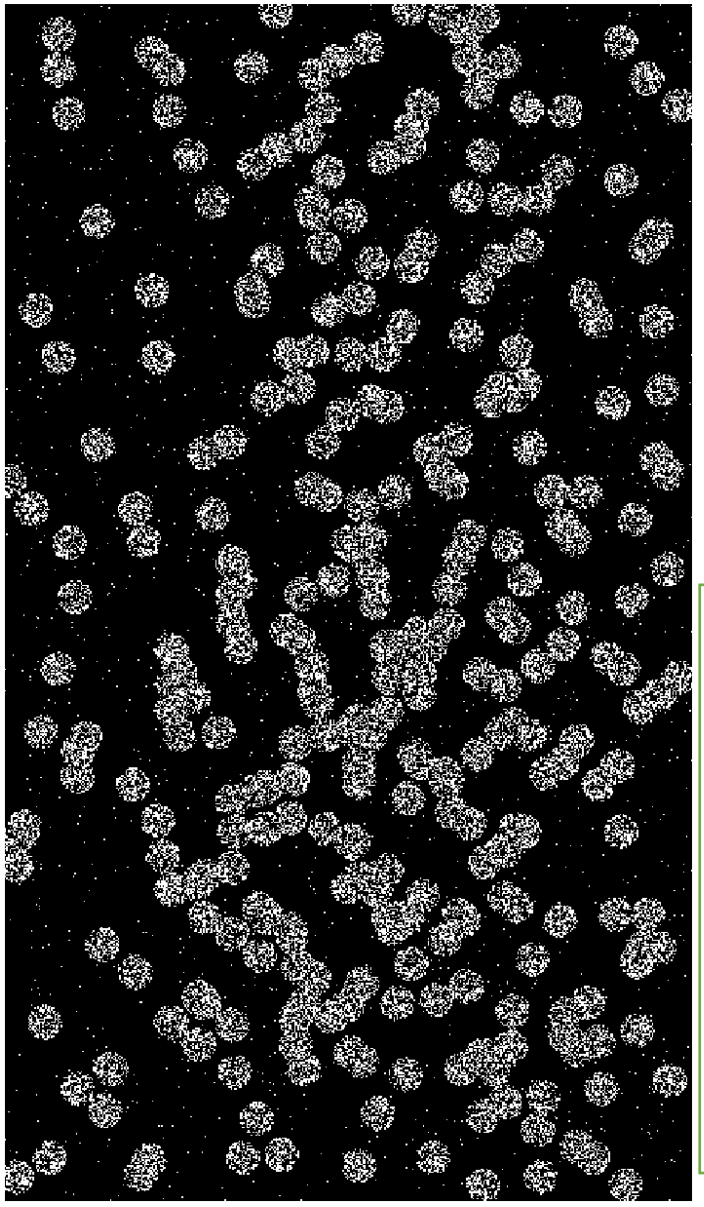


Color segmentatn using Mahalanobis dist.



Initial error = 7424

Color segmentatn using Euclidean dist.

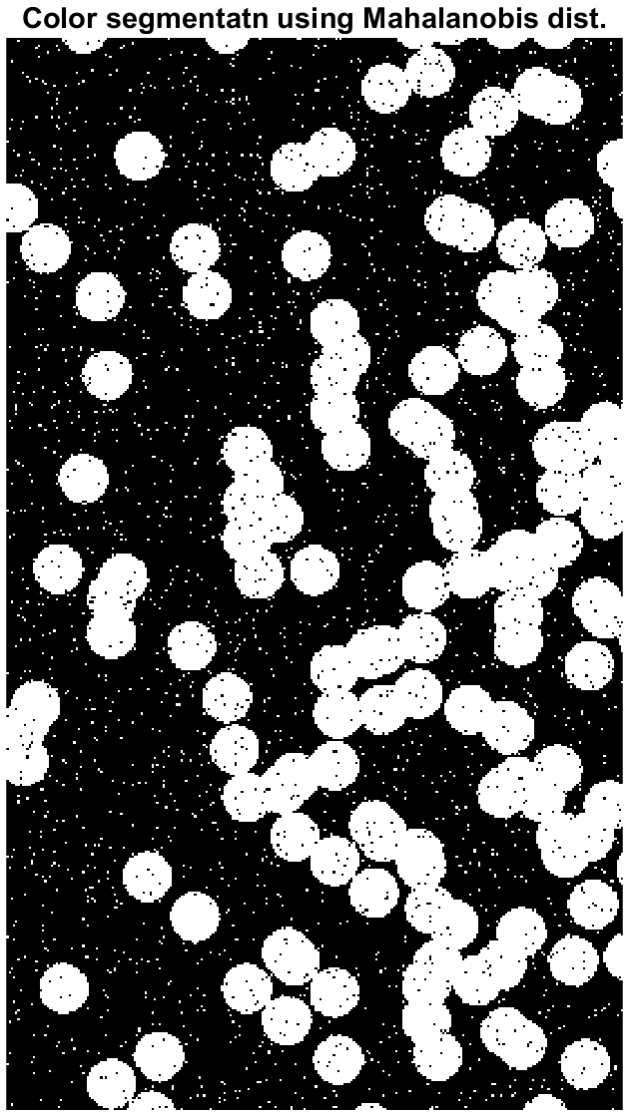


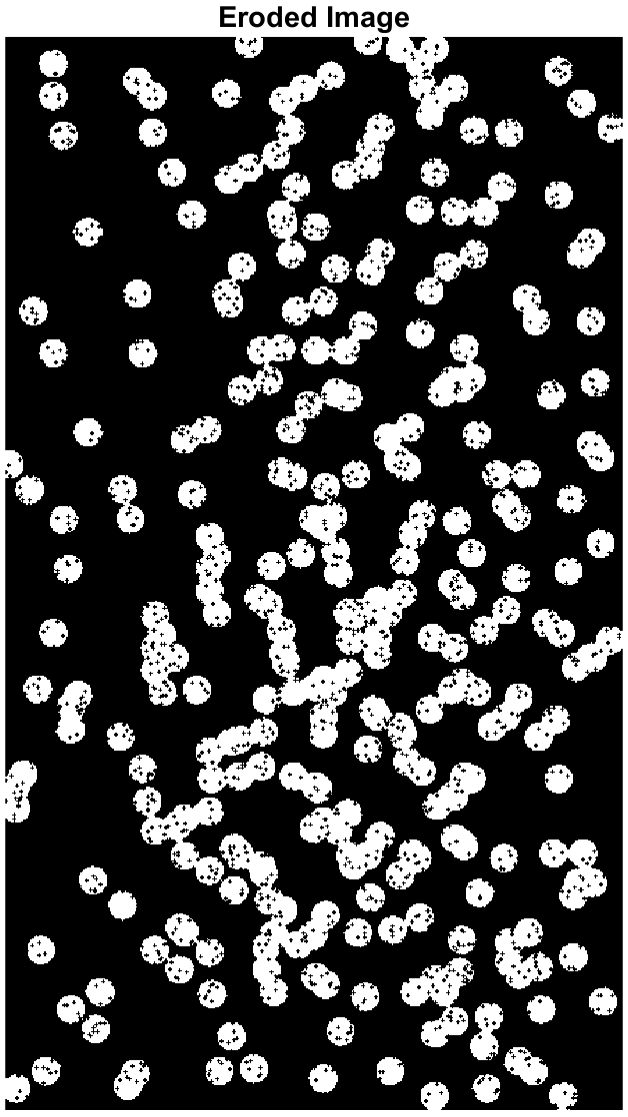
Error = 59766 (not good)

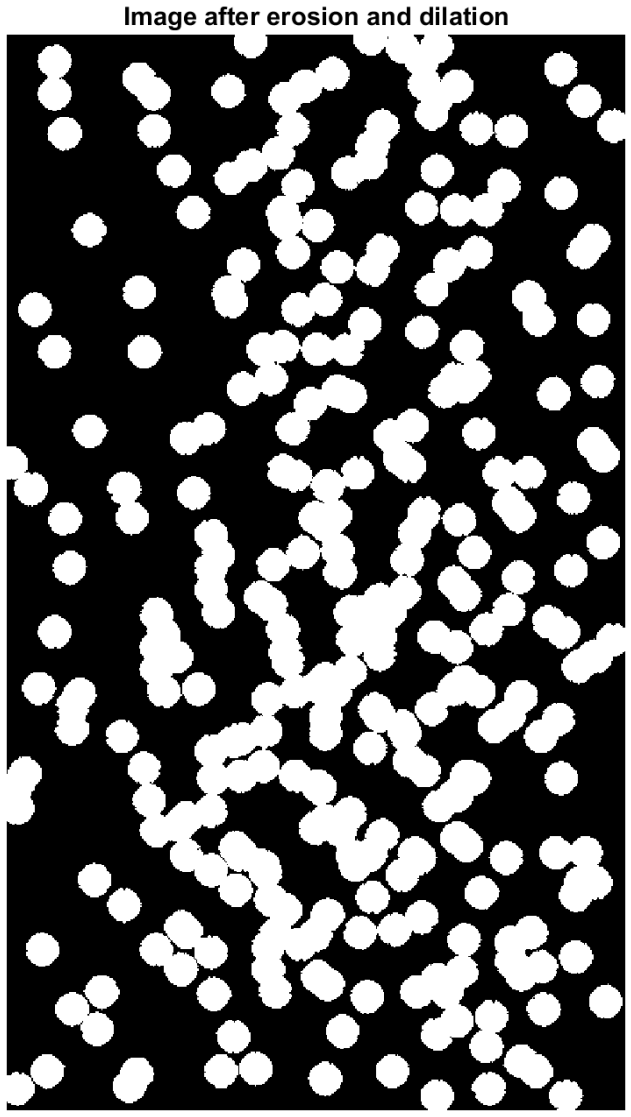
```
IQA_init_mahal =

        7424


IQA_init_eucl =

        59766
```

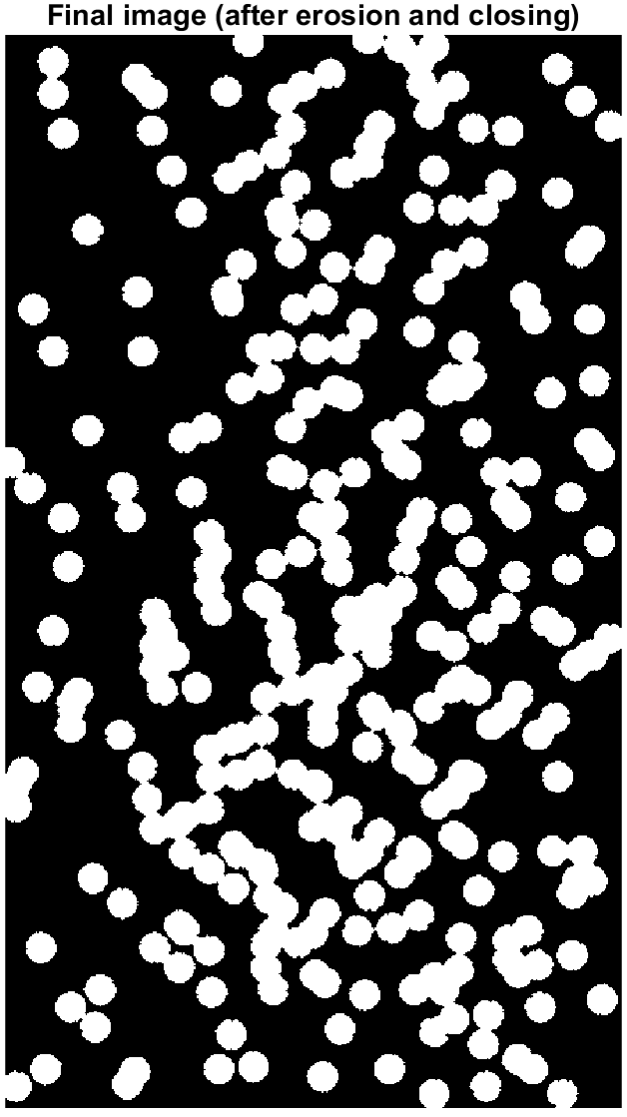
- Two windows were chosen manually from foreground to calculate statistics.
- Some morphological processing is necessary on the segmented image using Mahalanobis distance.

(C): Morphological Processing - Results









Initial error = 7424

Error = 12805 (worse)

Final error = 1802 (a lot better)

(C): Discussion

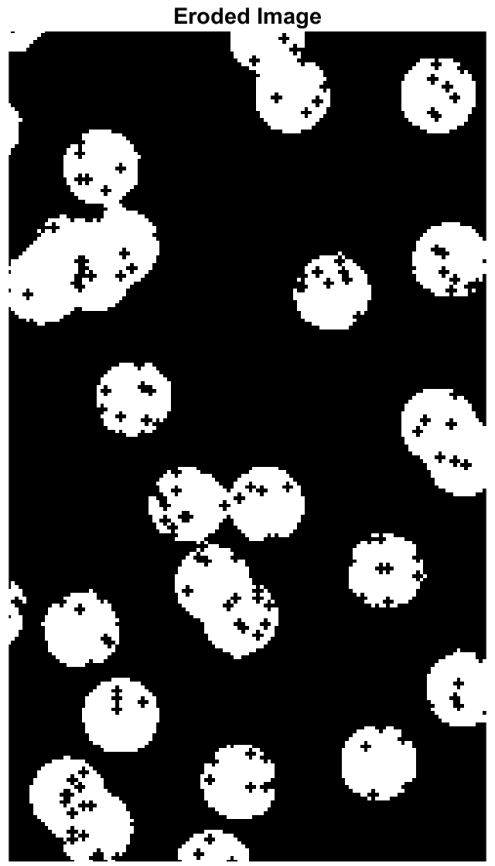
- Erosion was performed to get rid of background noise. Background noise was mostly single pixel or two pixels. Hence, a small disk with radius 1 worked good as the structuring element.

- After erosion, all background noises were seen to be removed. But the foreground noise on the discs increased; noise pixels (black pixels on the discs) have formed small disk size shape (similar to the structuring element) and in a few cases, a bit thicker.

- To remove foreground noise, which is equivalent to closing the gaps on the discs, closing operation was performed. To accomplish that, dilation was performed with a bit larger (radius=2) structuring element disk, so that it covers the holes. It did a good job, but all the discs became slightly larger due to dilation which caused higher error (12805). Another erosion (to complete the closing operation) with a structuring element disk of radius 1 solved this and the error became minimum (1802)

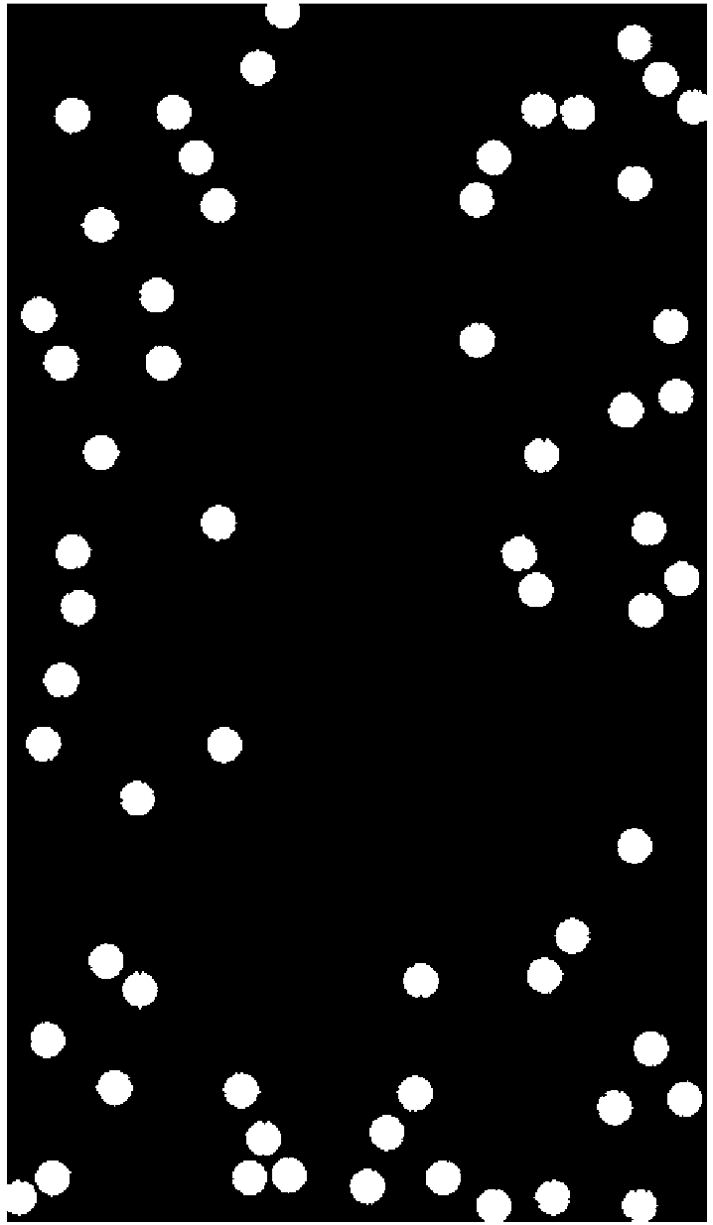
```
IQA_init_mahal =

        7424

IQA_init_eucl =

       59766

IQA_mahal_morph1 =

       12805

IQA_mahal_morph2 =

        1802

>>
```

Close-up look at foreground noises after initial erosion

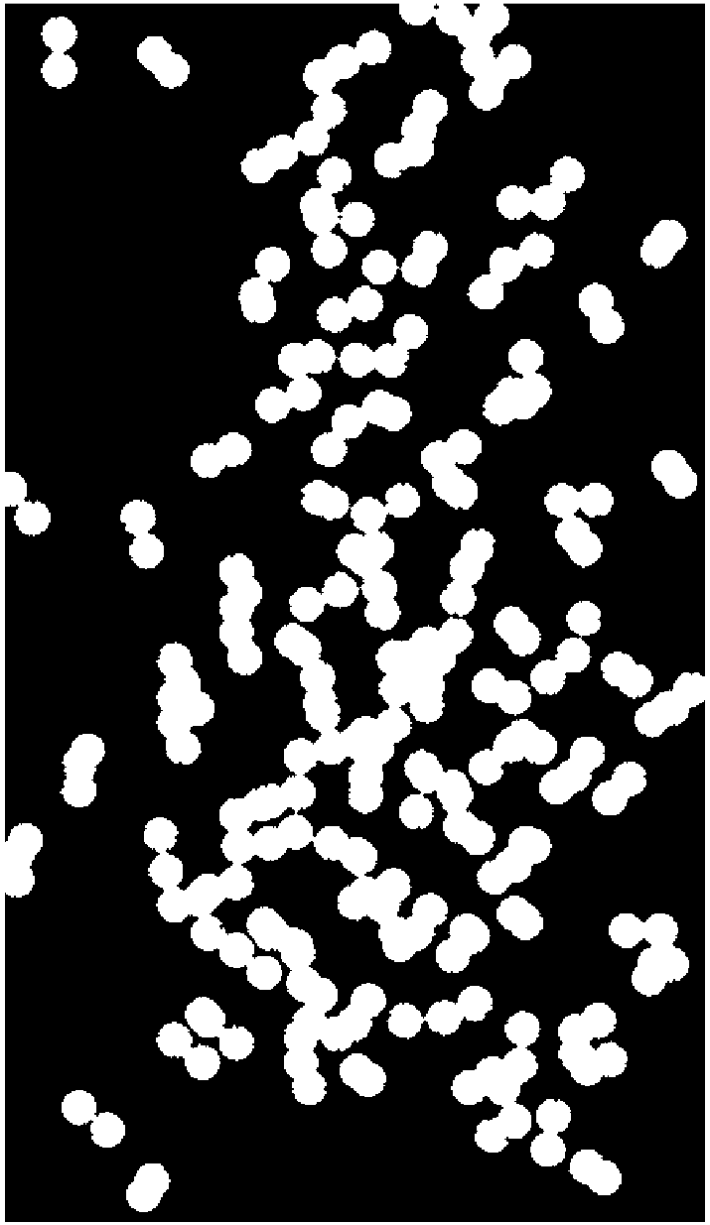


(d) Counting like part I

Individual discs only (count = 61)



Connected discs only (count = 53)



Similar code that was used for part 1 have been used here with slightly different parameters, in some cases.

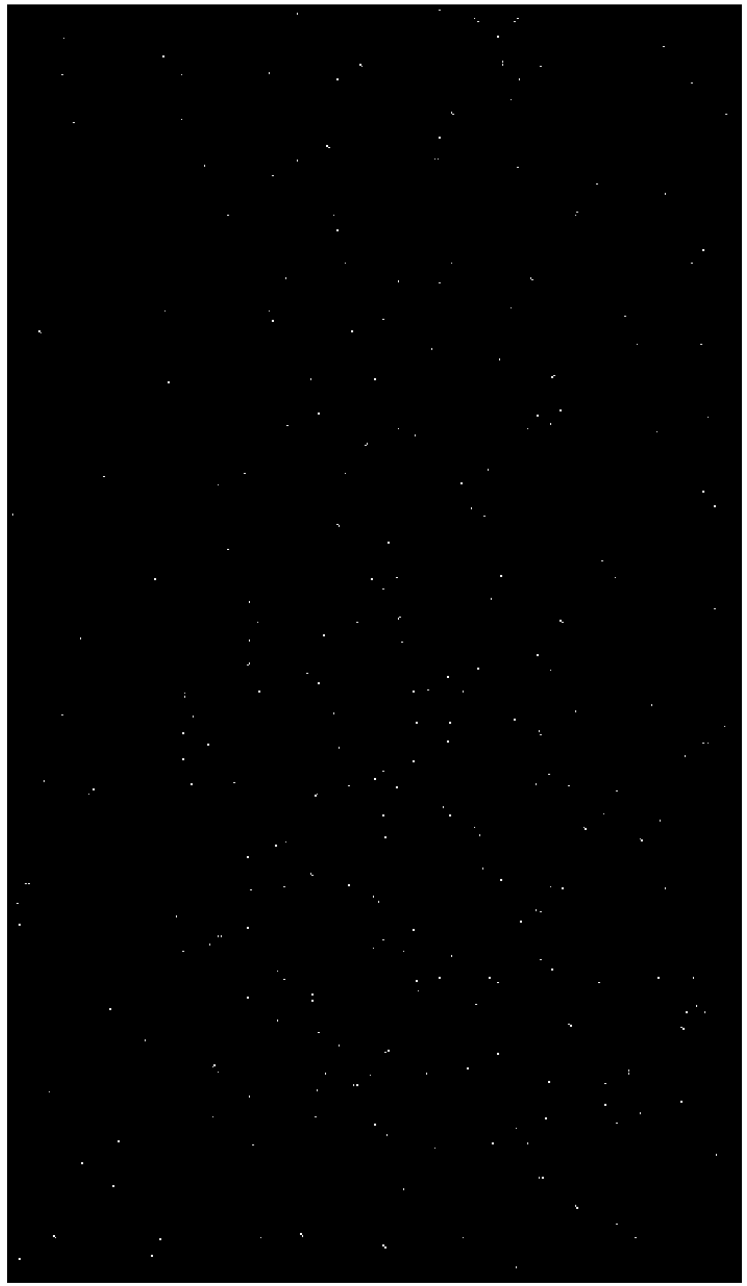
Discussion:

Because of closing operation performed, some of the individual discs or clusters, which were previously in close proximity, became connected and treated as one connected component. Thus, the total number of connected components became 114, while it was previously higher (129) with noise free original image.

	Number of individual discs	Number of clusters	Total number of connected components
With noise free binary image	70	59	129
With noisy color image	61	53	114

(d) Counting like part I - continued

Detected discs as single pixel



Discussion:

Similar code that was used for part 1 have been used here with slightly different parameters. The threshold for significant change in the dilated image after adding a pixel was previously 73. But this threshold was brought down to 23 for colored noisy image, because the overlap among the discs increased because of morphological processing which made the improvement smaller, even after adding a whole new disc through dilation.

	Total Disc count
With noise free binary image	301
With noisy color image	295

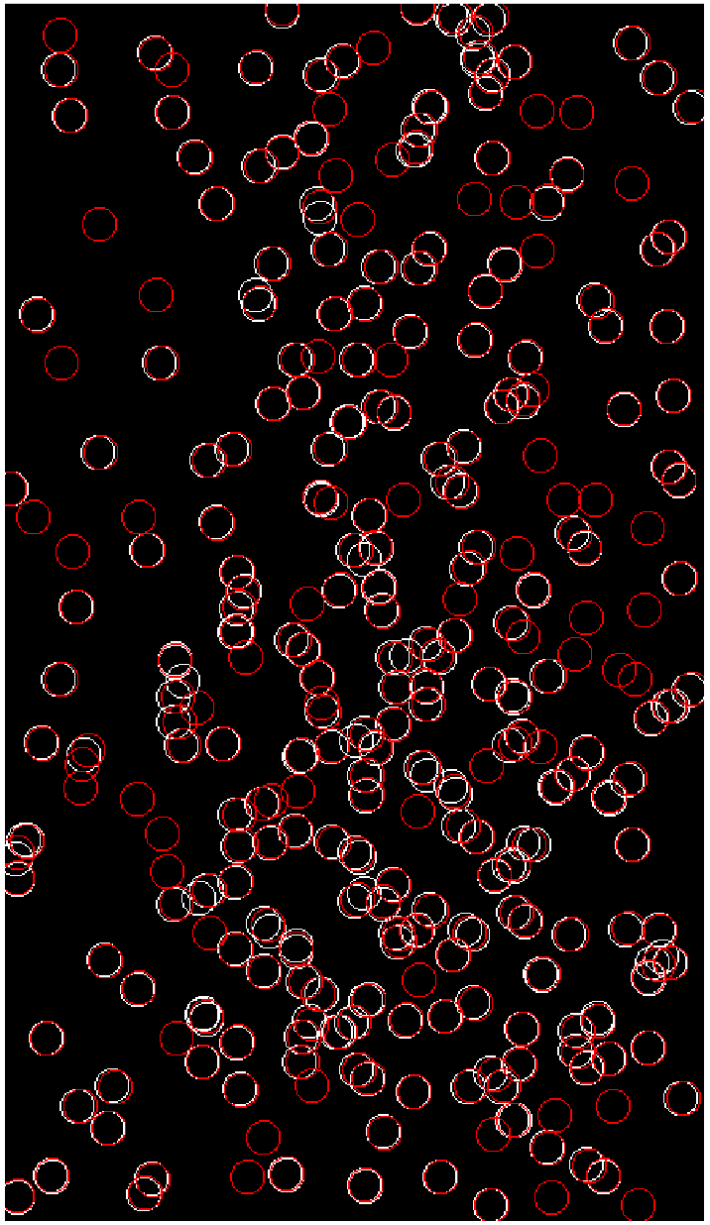
(d) Counting like part I - continued

Image	Disc count	
	Using Hough transform	Using our algorithm
With noise free binary image	287	301
With noisy color image	282	295

Discussion:

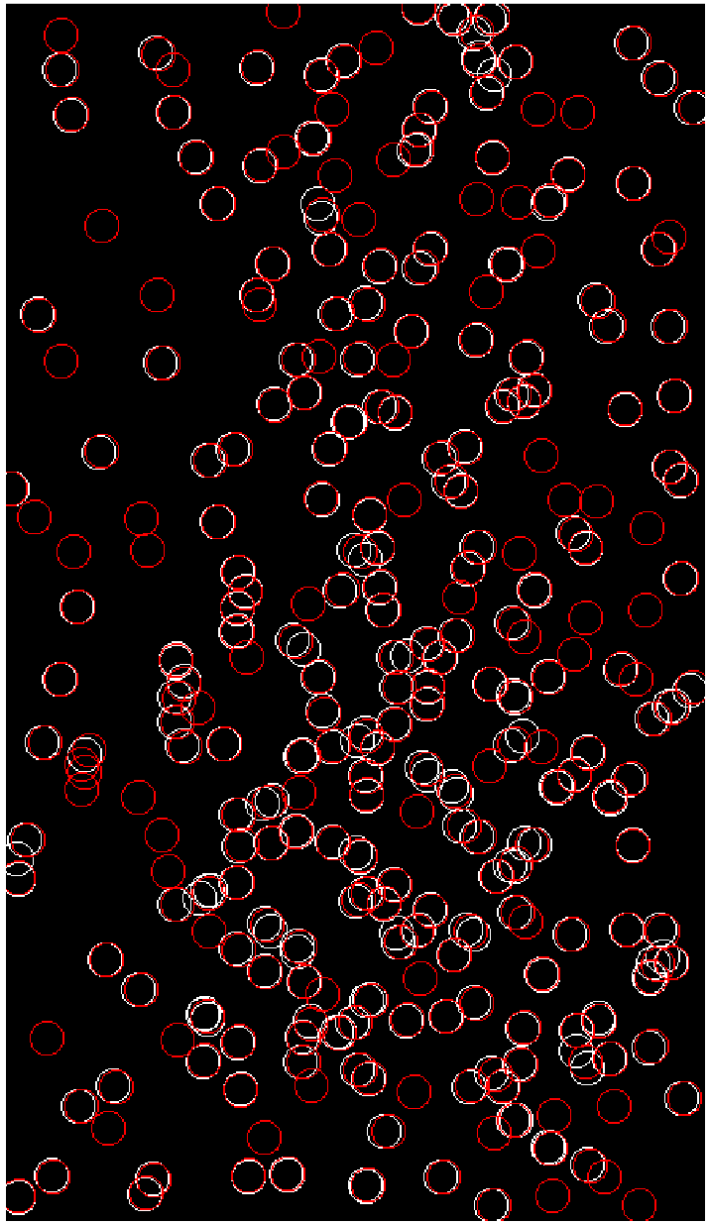
Although from the images at the right, it seems that a lot of red circles are undetected by our algorithm, the overall count is not that bad. The reason is: in the single pixel representation of the discs, the noise and morphological processing shifted some pixels thus changed the location of the hollow circle afterwards. So, there is more mismatch now, but still the count is only slightly underestimated.

Circular representatn: White-our algorithm, red-Hough Transform



(Here, Hough transform was applied on the original noise free binary image, like part 1)

Circular representatn: White-our algorithm, red-Hough Transform



(Here, Hough transform was applied on the color image – after binarizing and morphological processing)

Thanks