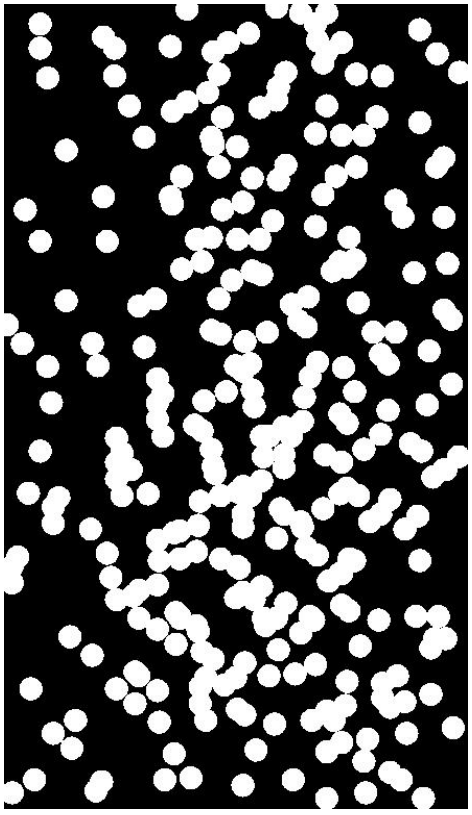
Proj5 - Object Counting by Morphological Processing

This project has two parts (10 points for each part). Part I is about object counting from a given binary image and Part II is doing the same task from a color image.

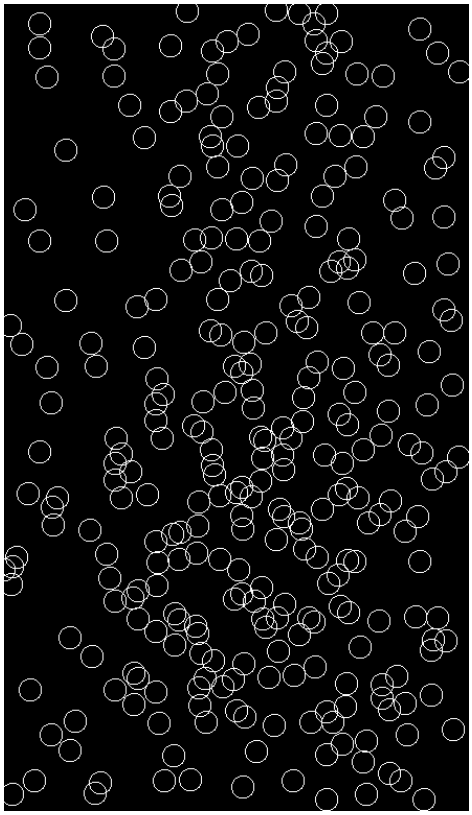
Part I. Given a binary image below ('Cells.bmp') and a template image of a disc ('disc.bmp'), develop appropriate morphological processing algorithms that can achieve the following tasks.

(a) Find, display and count the individual discs that are isolated from others;
(b) Find, display and count the discs that are clustered together;
(c) Count the total number of discs including the isolated and overlapped ones. Visualize each disc by drawing a circle at each location as shown on the right (test2.m);
(d) For comparison, you can use the Matlab function (imfindcircles) to find all discs (test3.m). In addition to the analysis at the whole image level, some region-by-region local analysis should be used for detailed comparison.

Useful Matlab functions:
"imerode" (Erosion), "imdilate" (Dilation), "Error=sum(xor(A,I),'all')" that returns the number of unmatched pixels between A and I, "sort" (sorting in the ascending or descending order ), "bwlabel" (extraction of connected components).



Given input image



Expected detection and counting results.

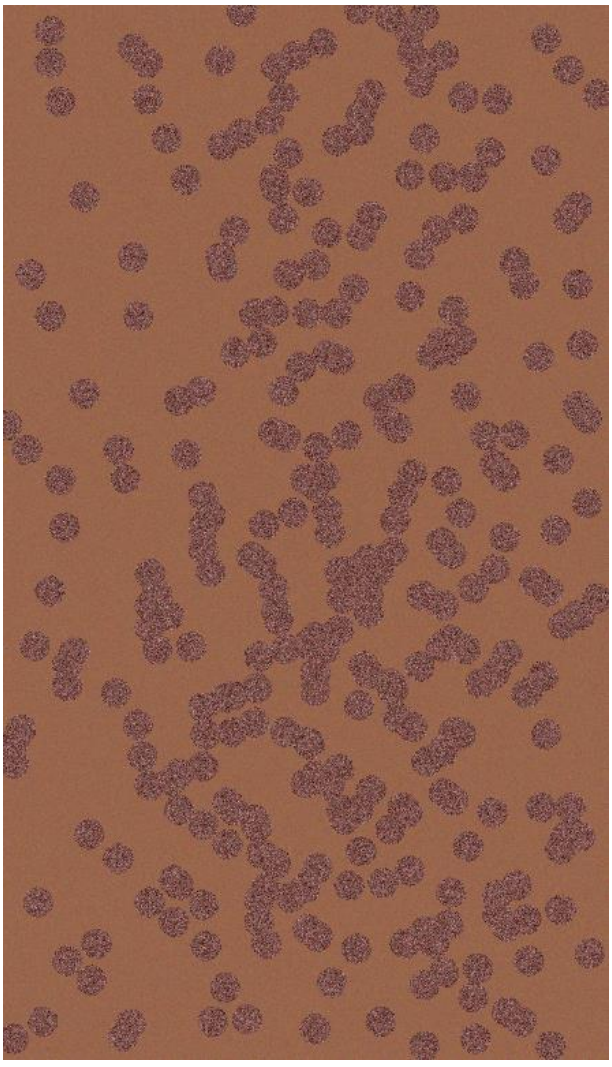
Part II. Given a color image (color_cell.bmp) shown below, first do color segmentation to detect all disc regions and repeat Part I by counting the number of discs.

(a) Manually choose a set of pixels from the disc regions and compute the statistics of color distribution in terms of the mean and the covariance matrix (Slide 10 of Lecture 26);

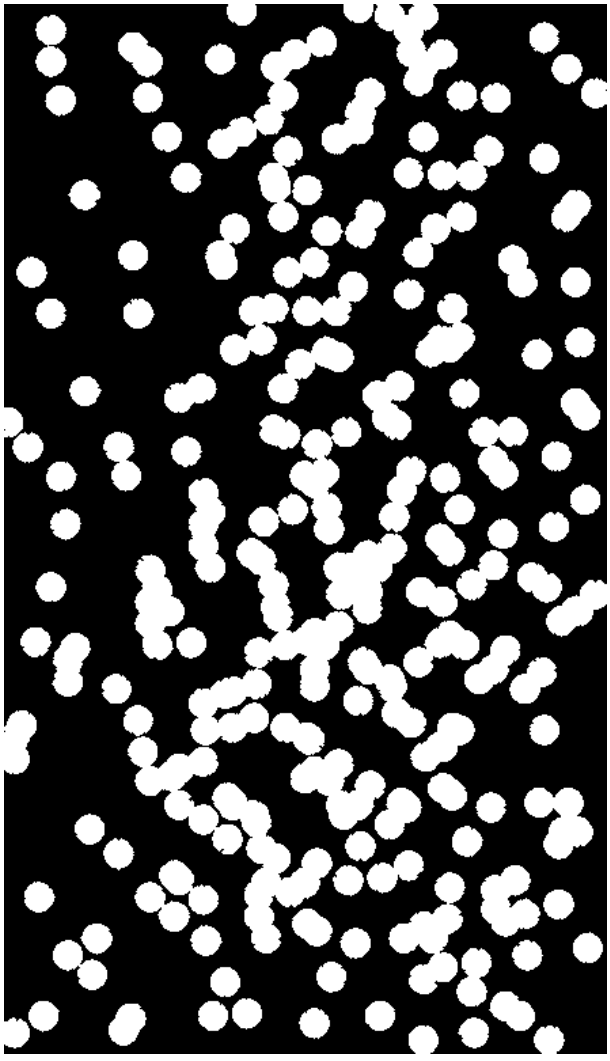
(b) Then using the Mahalanobis distance (Slide 8 of Lecture 26) to segment all disc regions from the background. Carefully choose a distance threshold to create a reasonable initial segmentation by comparing the result (J) with the ground-truth segmentation (O) (Cells.bmp) with a relatively small error (sum(xor(J,O),'all');

(c) Then apply appropriate morphological operations to further refine the segmentation results with the minimum error (sum(xor(J,O),'all');

(d) Repeat Part I based on the final segmentation result and compare the new counting results with the previous ones. Some thresholds or parameters may have to be adjusted to achieve the optimal results.



Given color image



Expected final segmentation result.