CSE585/EE555: Digital Image Processing I

Computer Project # 1:

Mathematical Morphology: Hit-or-Miss Transform

Lindsey Schwartz, Ke Liang, Xilun Liu

Date: 01/25/2019

A. Objectives

The goal of this project was to explore mathematical morphology and use the hit-or-miss transform to detect the smallest and largest circles in an image.

B. Methods

The original image (Figure 1) contains discs of five different radii as well as salt and pepper noise. In order to detect the smallest and largest discs using the hit-or-miss transform, several pre-processing steps need to occur. The first step was to turn the grayscale image into a binary image in order to perform morphological operations. A threshold of 128 was chosen based on the histogram of the image. Each pixel in the original image with a value of less than 128 was set to 0, while each pixel greater than or equal to 128 was set to 255. The next step was to eliminate the salt and pepper noise. Salt and pepper noise was eliminated by filtering the image with a median filter. The open/close morphology method was tested, however, the salt and pepper noise was not single pixel as originally thought (no matter what threshold was picked).

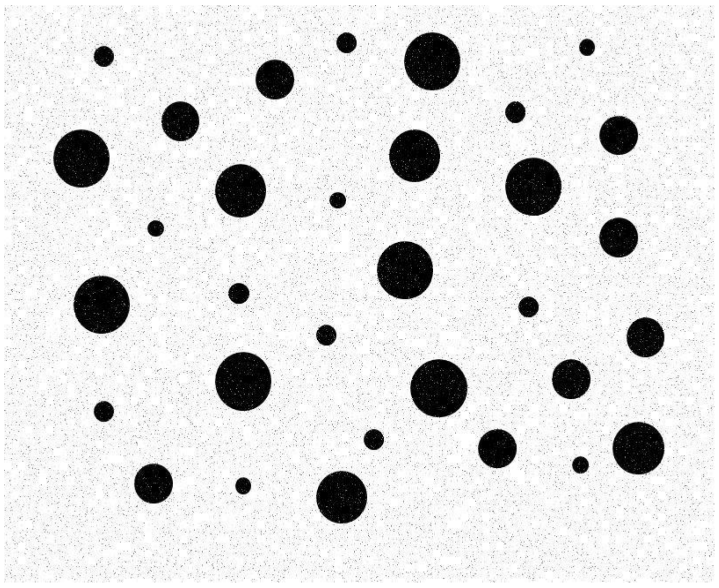
Figure 1. Original image

Once the pre-processing was completed, structuring elements for the hit-or-miss transform could be designed. The hit-and-miss transform is defined by

$$X \odot (A,B) = (X \ominus A^S) - (X \oplus B^S)$$

Where X is the input image and A and B are the structuring elements. Structuring element A is designed to find neighborhoods that match its shape, while structuring element B is designed to find neighborhoods that do not match its shape. To automatically create the structuring elements, MATLAB’s regionprops function was used to determine the diameters of each circle in the image. Then, using a priori knowledge that there were five different sized circles, MATLAB’s kmeans classification algorithm was used to sort the circles into five classes. The diameter from the smallest and largest class are used to create the structuring element. For simplicity, a square structuring element was used. Structuring element A was designed to be large enough to fit inside the circle size being detected (the largest square to fit inside the circle). Structuring element B was designed to be slightly larger to encompass the circle size being detected. Four total structuring elements were used, two for each circle size detected.

Once created, the structuring elements are moved through the image and a response is computed at each pixel. When structuring element A is fully contained, a hit is recorded. And when structuring element B hits the background, a miss is recorded. This results in two detection maps. One for the smallest circles and one for the largest circles. These can be combined into one detection map by performing a union.The following is the flow chart of our algorithm, shown in Figure 2.

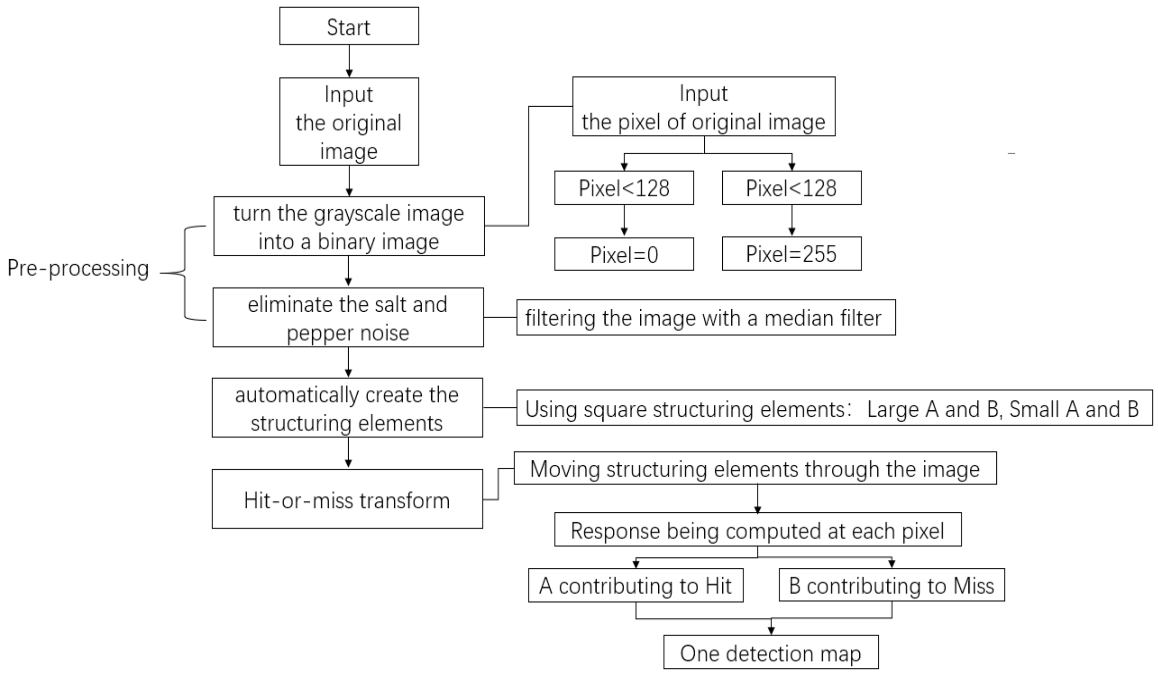
Start
Input the original image
turn the grayscale image into a binary image
Pre-processing
eliminate the salt and pepper noise
automatically create the structuring elements
Hit-or-miss transform
Input the pixel of original image
Pixel<128
Pixel<128
Pixel=0
Pixel=255
filtering the image with a median filter
Using square structuring elements: Large A and B, Small A and B
Moving structuring elements through the image
Response being computed at each pixel
A contributing to Hit
B contributing to Miss
One detection map

Figure 2 The flow chart of the algorithm

To run the code in MATLAB, ensure that the file path is correct and that it points to the location of the ‘RandomDiscs-P10.jpg’. Then hit the run button. The algorithm is divided into the main script with two supporting functions. The supporting functions remove the salt and pepper noise and create the structuring elements. The main script loads the image, performs the transform, and displays the intermediate and final results. The algorithm will display the original image, the binary image, the filtered image, the structuring elements, and two final results images. One binary image with just the smallest and largest circles remaining and the other is the original image with the smallest and largest circles outlined.

C. Results

The purpose of this section is to illustrate the results of the steps outlined in section B. The first intermediate result was creating a binary image by thresholding, which can be seen in Figure 3. As you can see, there is still some residual noise that needs to be removed.

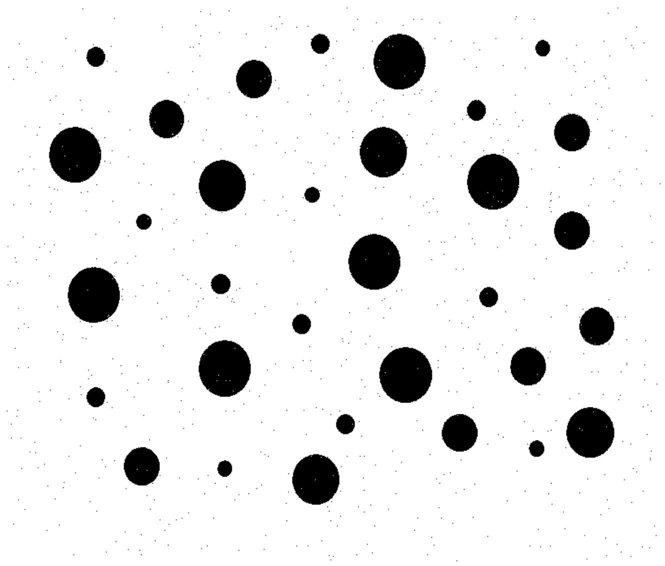

Figure 3. Binary image as a result of thresholding.

The second result is about the noise contained in the original image. In this project the salt and pepper noise is removed by using a median filter. The result is indicated in Figure 4. If the salt and pepper noise is not removed from the image, the hit-or-miss transform will not work properly. This is because the structuring element A cannot be contained within a disk if there is a hole. The same is true for structuring element B.

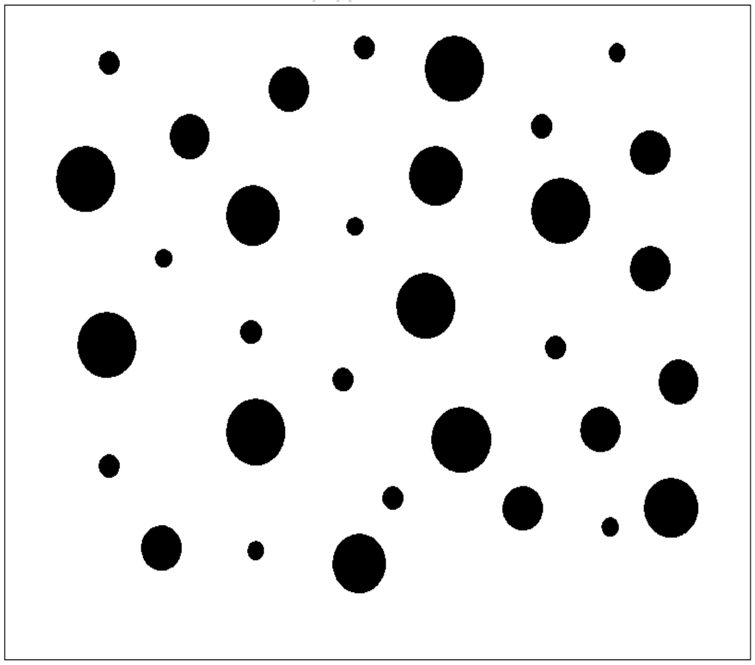
Figure 4. Image noise removed by using a median filter

The third result is about the structuring elements we choose to complete the hit-or-miss transform.There are 4 structuring elements chosen in the project, which can be divided in 2 parts named as the small structuring element and large structuring element, corresponding to the small and large disks. For each part, both of its shapes are square. For each sized structural part, there is a pair of elements named A and B, which are the shape itself and the local background of the shape with respect to window. The small structuring elements are used to find the smallest circles in the original image and the large structuring elements are used to find the largest circles. The four structuring elements are shown in Figure 5.

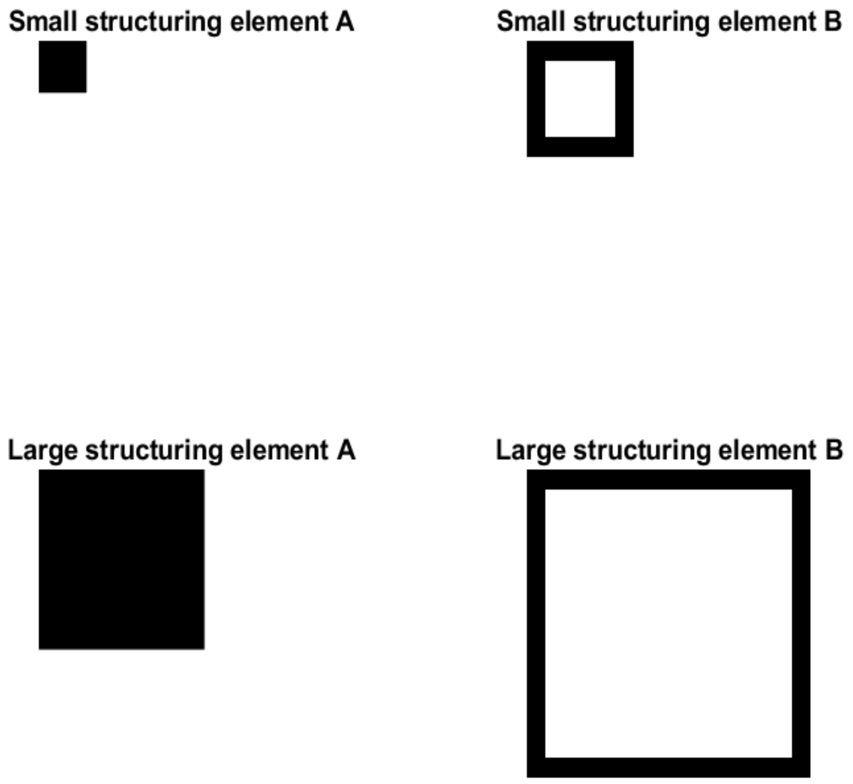
Small structuring element A
Small structuring element B
Large structuring element A
Large structuring element B

Figure 5. Four structuring elements to dealing with hit-or-miss transform

The fourth result are the images of the transformation by using structuring elements. The result of sliding the large structuring element A through the image with the noise removed is shown in Figure 6, and that is also the result of image hit by the large structuring element. The result of using large structuring element B to deal with the image removed the noise is shown in Figure 7, and that is also the result of image missed by the large structuring element.

Figure 6. Image with noise removed hit by large structuring element A

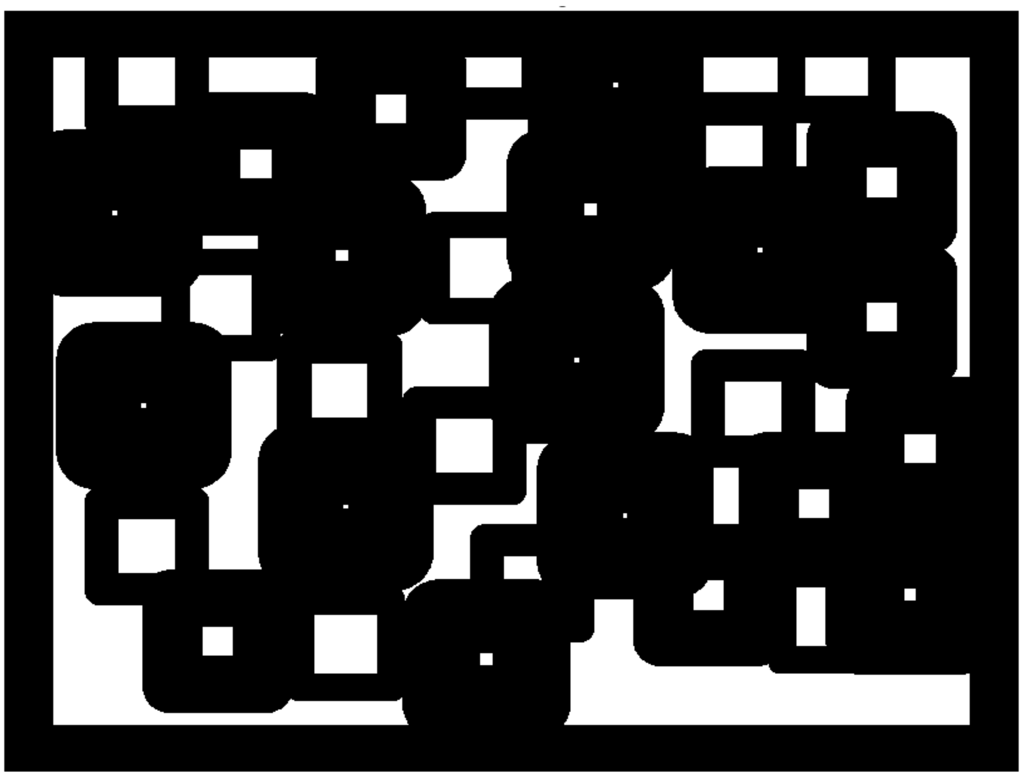
Figure 7. Image with noise removed missed by large structuring element B

Then, the location of large circles in original image can be found by performing a union of the images in Figure 6 and Figure 7. This is displayed in Figure 8.

Figure 8. The location of large circles in the image

The result of using small structuring element A to transform the image with the noise removed is shown in Figure 9, and that also is the result of image hit by the small structuring element. The result of using small structuring element B to deal with the image removed the noise is shown in Figure 10, and that also is the result of image missed by the small structuring element.

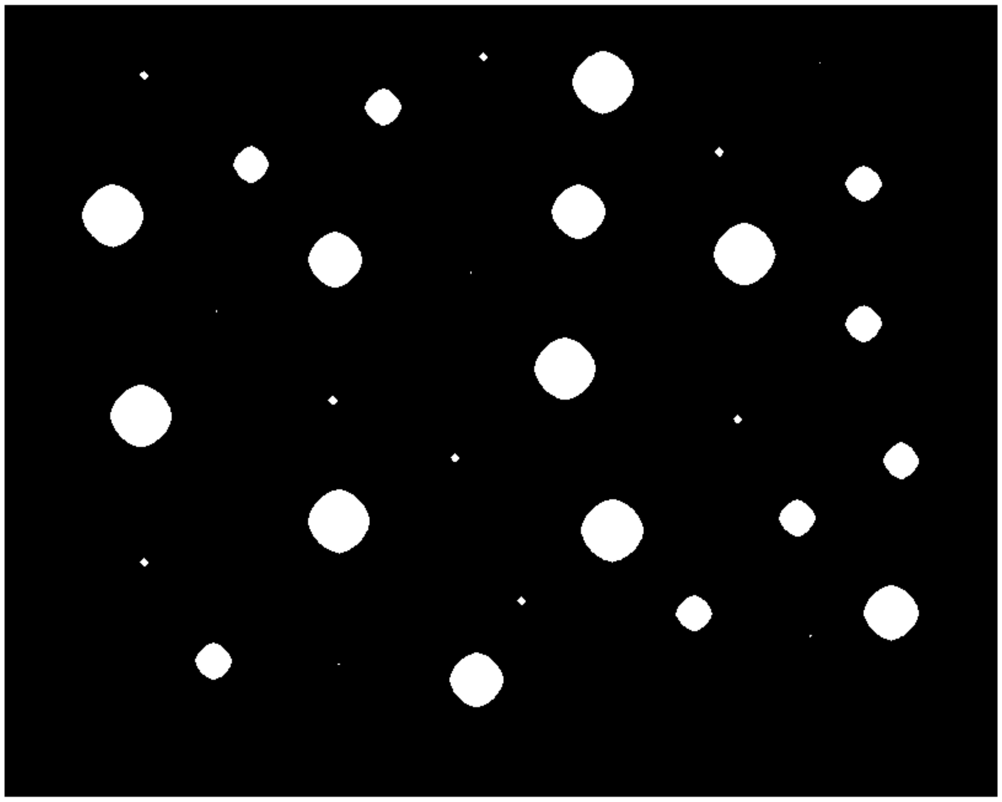

Figure 9. Image with the noise removed missed by small structuring element A

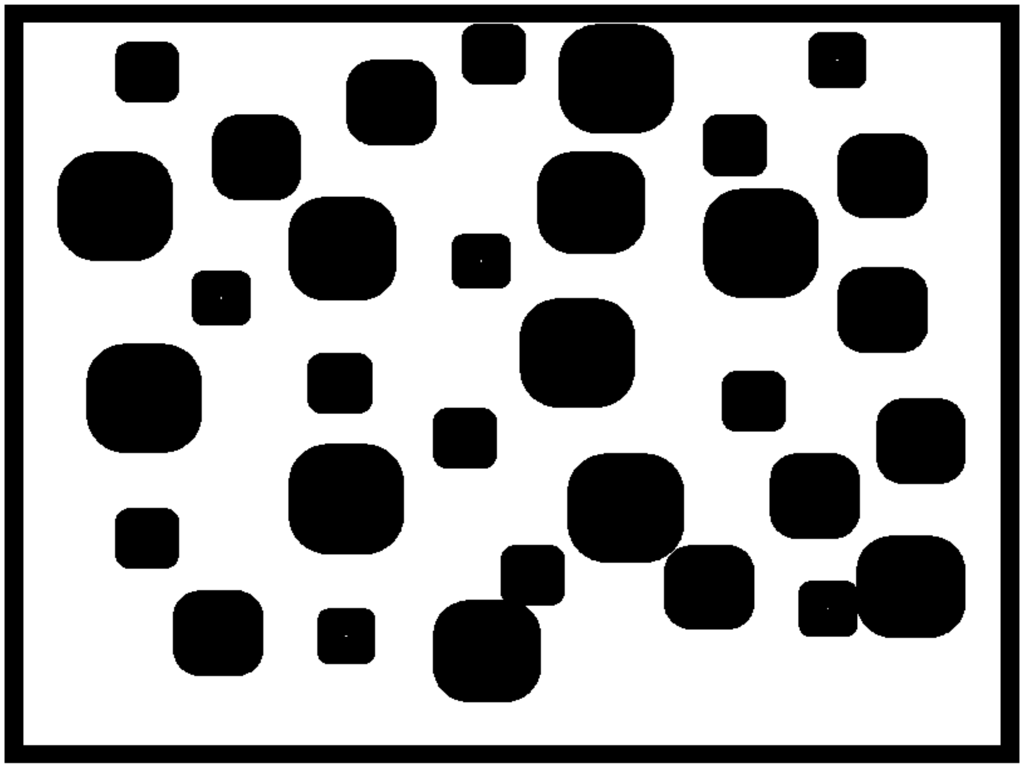

Figure 10. Image with the noise removed missed by small structuring element B

Then, the location of small circles in original image can be found by performing a union of the images in Figures 9 and 10, which is shown in Figure 11.

Figure 11. The location of small circles in the image

The fifth result is showing the smallest circles and biggest circles. Figure 12 shows only the circles detected while Figure 13 shows the original image with the smallest and largest circles highlighted in red. Overall, there were five of the smallest disks and seven of the largest disks.

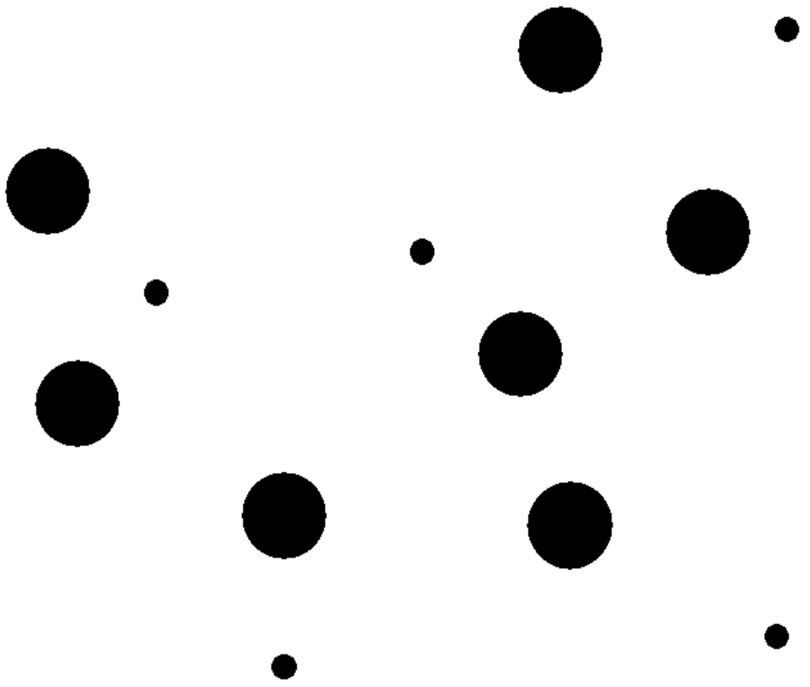
Figure 12. The smallest and largest disks in the image

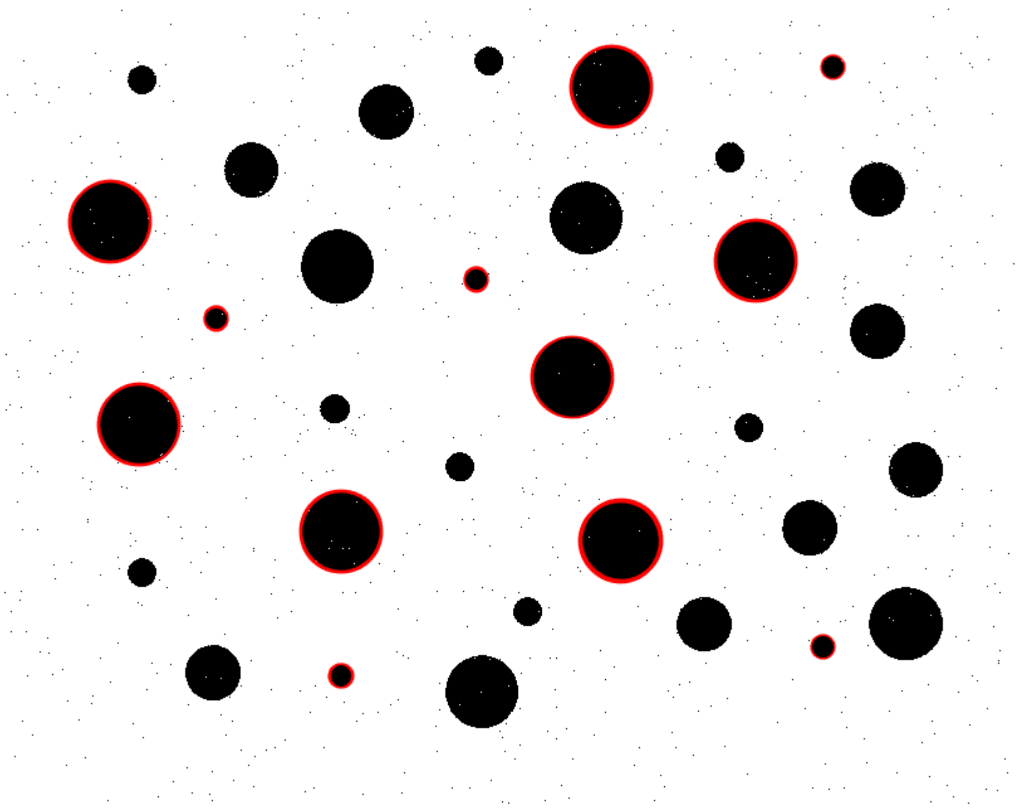
Figure 13. Largest circles and smallest circles displayed on the original image

D. Conclusions

This project demonstrates that the hit-or-miss transform can be used for shape detection by creating two structuring elements. One that is contained within the shape being detected and the other that detects regions that do not match. As demonstrated in this particular project, the structuring element does not have to be identical to the detected shape. We demonstrated that square structuring elements can be used to detect disks. However, the hit-and-miss transform only works if the image has been preprocessed to eliminate noise sources in the image that are not present in the shape being detected.