CSE585/EE555: Digital Image Processing II

Computer Project # 1:

Shape Detection using Morphological Operations

Names of group members: Saptarashmi Bandyopadhyay ; Bharadwaj Ravichandran

Date: 01/25/2019

A. **Objectives**

The objective of the project is to use mathematical morphology operations like erosion, dilation, opening, closing and hit-or-miss transform to detect the largest and the smallest disks from an image consisting of disks of 5 different sizes.

B. **Methods**

Control Flow of the Project

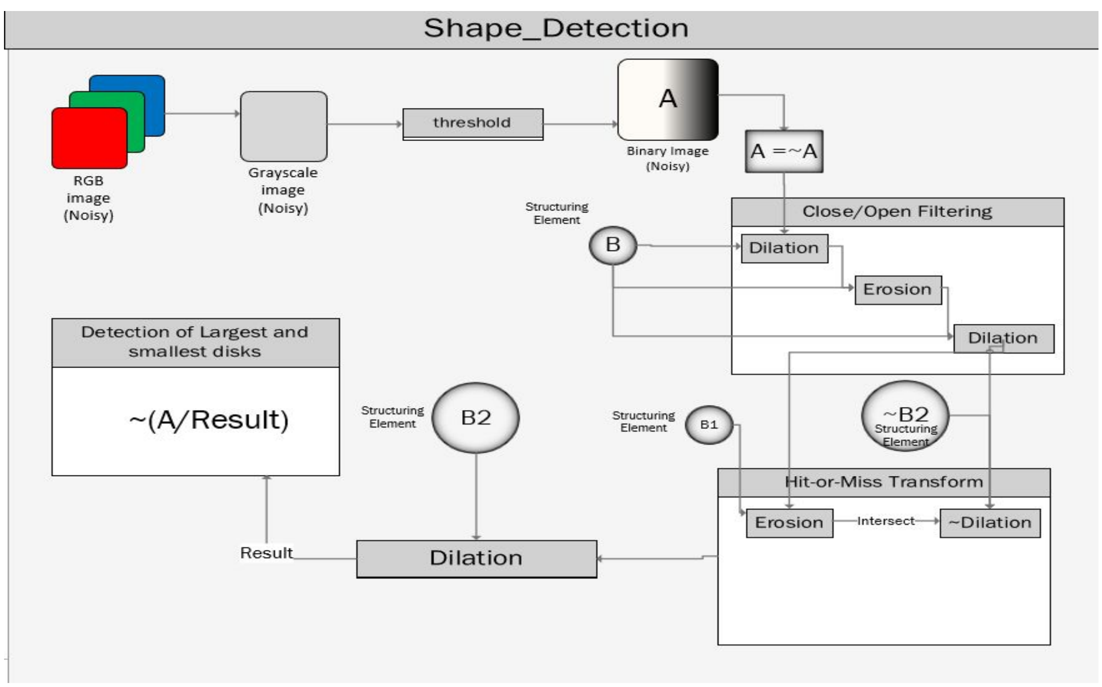


Figure 1. Algorithm to detect the largest and the smallest disks among 5 differently sized disks in the input image

Organization of the Codes:

(1) ***Project_ main.m***: It is the main function which oversees the control flow of the program as in Figure 1. The input image is loaded and the size of the matrix storing the image is obtained.

The input image is then converted to gray scale by using the weights of 0.2989 multiplied to pixel values in the Red dimension, 0.5870 multiplied to pixel values in the Green dimension and 0.1140 multiplied to pixel values in the Blue dimension of the matrix storing the image[1].

The histogram of the grayscale image is obtained to investigate the threshold value. A threshold value of 95 is obtained by a mechanism specified in the Sub-section iii in Section C (Results).

The input image is then converted to binary image (A) by calling the threshold function in the threshold.m code with the grayscale image and the threshold value as parameters.

The image is then complemented (~A) by the not_operator function in the code not_operator.m. This is because the erosion_2 and the dilation_2 functions in the erosion_2.m and the dilation_2.m codes are implemented assuming the background pixels to be 0 and the pixels denoting the image to be 1. But the input image has background pixels to be 1 and the pixels denoting the image to be 0. The erosion_2 and the dilation_2 functions are called by the openclose function.

The image obtained after not operation is sent to the openclose function in the openclose.m code to execute the close/open filter with the structuring element being a 2x2 matrix of ones (B).

The structuring elements are already selected by a mechanism specified in the Sub-section i in Section C (Results).

The hit or miss transform is carried out on the image obtained after close/open filter with the first structuring element (B1) being a disk which is a bit bigger than the smallest disk and and the second structuring element (B2) being the complement of the disk which is a bit smaller than the largest disk from a window W. This gives the three middle sized disks in the image (Result).

The largest and smallest disks are obtained by a set difference operation of A with Result. That can be represented as

$$A/Result = A \cap (\sim Result)$$

Since A was initially complemented to be used in the implementation of the erosion_2 and dilation_2, the final result is complemented again. It is represented as $\sim(A/Result) = \sim(A \cap (\sim Result))$

(2) *threshold.m*: It is used to define the function threshold(A,val) which takes two inputs: noisy grayscale input image(A)and threshold value(val). It sets the pixel values to zero if less than the input threshold and the rest of the pixels to 1 and returns the thresholded (binary) image.

(3) *not_operator.m*: It is used to define the function not_operator(A) which takes one input: a binary image (A) and returns the complement of the image.

(4) *openclose.m*: It is used to define the function openclose(A,B) which takes 2 inputs: the thresholded noisy input image (A) and a structuring element (B) and returns the denoised image, i.e, does salt and pepper noise removal.

(5) *hitormiss.m*: It is used to define the function hitormiss(X,A,B) which takes 3 inputs: Denoised Binary Image (X), structuring element (A) which is the smallest disk and structuring element (B) which is the complement of the largest disk. It returns the result after applying hit-or-miss transform.

(6) *dilation_2.m*: It is used to define the function dilation_2(A,B) which does Minkowski addition on the input image (A) with the structuring element (B) and returns the result.

(7) *erosion_2.m*: It is used to define the function erosion_2(A,B) which does Minkowski subtraction on the input image (A) with the structuring element (B) and returns the result.

Execution of the codes

The 7 MATLAB files along with the input image and the images of the 2 structuring elements are present in the folder Codes_in_Project_1 in the submitted zip file. Project_main.m should be run in MATLAB to obtain the result.

Only 9 primary images (Figures 3,4,8,9,10,14,16,17 and 19 in Section C of Results) among the 34 images presented in the report are displayed on running the code to avoid cluttering of all images after the execution. These 9 images appear with titles for readability.

However the 34 images (Figures 1 to 30 and Graph 1 and Figures in Table 1 in Section C of Results along with Graph1) are stored in the Images_in_Project_1 folder in the submitted zip file. These 34 images are not given any title as they are already defined in the Section C of Results. These 34 images are labeled in the folder with the same Figure number as in the report and these images display every step in the processing of the project **as was asked in Question 3d.**

Morphological Operations and Set Theory Operations used in the Project

i. Complement operation: This operation gives the opposite of the given binary value and is given by,

$A_c = \{x : x \notin A\}$

The complement operator has been executed by the *not_operator.m* code in MATLAB. The pseudocode of not_operator.m is as follows:

Function: not_operator (A)

Objective:

To execute the complement operation on an image

Input Parameters:

A: input binary image

Returned Result:

A_comp: The complement of the image A

Processing Flow:

1. The size of the matrix storing the pixels of the image are obtained to traverse the rows and columns of the matrix.
2. If the pixel value of the image A is 1, it is changed to 0 in the same image.
3. Otherwise if the pixel value of the image A is 0, it is changed to 1 in the same image.
4. The modified image A is assigned to the variable A_comp which is returned by the function.

Restrictions/Notes:

This function requires a binary image as input.

No functions are called.

The following functions are calling:

Project_main.m:

hitormiss.m

ii. Threshold operation: A grayscale image can be converted to the binary image by calling the *threshold.m* function in MATLAB.

The threshold value is selected by observing the histogram of the gray scale of the input image as in Graph 1 in Section C of Results. The pixels nearer to 0 and 255 are concentrated on either end of the histogram plot. After experimenting with a number of threshold values to remove the distortions in the input image as much as possible, the threshold value of 95 was selected. The pseudocode of the *threshold.m* is as follows:

Function: threshold (X, val)

Objective:

To convert a grayscale image to a binary image

Input Parameters:

X: image object

val: threshold value

Returned Result:

bX: binary image obtained from X

Processing Flow:

1. The size of the matrix storing the pixels of the image are obtained to traverse the rows and columns of the matrix.
2. The variable bX is initialized to an array of all zeros with the same number of rows and columns of the matrix storing the final image as in the initial image.
3. If the grayscale pixel value of the image X is less than the threshold value val, the pixel value at the same position (same row and same column) in bX is updated to 0.
4. If the grayscale pixel value of the image X is more than the threshold value val, the pixel value at the same position (same row and same column) in bX is updated to 1.

Restrictions/Notes:

This function requires a grayscale image and a threshold value as inputs.

No functions are called.

The following functions are calling:

Project_main.m

iii. Dilation:

It is a morphological operation which performs Minkowski Set Addition on an input image A, using a structural element B and is given by,

$$A \oplus B = \{z \in E : z = a + b, a \in A, b \in B\} = \bigcup_{b \in B} A_b$$

Function: dilation_2 (A, B)

Objective: To perform Minkowski Set Addition for the given input image and structuring element.

Input Parameters:

A **-** input binary image

B **-** input structuring element

Returned Result:

temp **-** Result of dilation on the input parameters

Processing Flow:

1. The size of the structuring element B is assigned to the variable P (rows), Q (columns) and R (color channels).
2. The index of the center of the structuring element is given by,

   center $\leftarrow$ $(\lceil P/2 \rceil, \lceil Q/2 \rceil)$
3. Initialise x_B and y_B to the indices of B which has the value 1.
4. Initialise n_B to the length of x_B

   n_B $\leftarrow$ length (x_B)

5. If n_B is greater than 0
   a. x_B $\leftarrow$ x_B - $\lceil P/2 \rceil$
   b. y_B $\leftarrow$ y_B - $\lceil Q/2 \rceil$

6. The size of the image A is assigned to the variables r (rows), c (columns) and chann (color channels).
7. [r2, c2] $\leftarrow$ [r + P - 1, c + Q -1]
8. x_limit $\leftarrow$ [$\lceil P/2 \rceil$ to $\lceil P/2 \rceil$+ r - 1]
9. y_limit $\leftarrow$ [$\lceil Q/2 \rceil$ to $\lceil Q/2 \rceil$+ c - 1]
10. Assign image to matrix with new border:

    border_handle [x_limit, y_limit] $\leftarrow$ A
11. Replace image A with border_handle

    A $\leftarrow$ border_handle [x_limit, y_limit]
12. Assign 1 to neighbouring pixels based on condition on A.

    for i in [x_limit]

    a. for j in [y_limit]

        i. if A[i,j] = 1

            1. for k in [1, n_B]

                a. result[i + x_B [k], j + y_B [k]] $\leftarrow$ 1
13. temp $\leftarrow$ result [x_limit, y_limit]
14. Since the problem is setup with a black background and white disks, we take the complement of temp in order to view the result.

Restrictions/Notes:

This input image and structuring element should be in binary.

No functions are called.

The following functions are calling:

Project_main.m

hitormiss.m, openclose.m

iv. Erosion:

It is a morphological operation which performs Minkowski Set Subtraction on an input image A, using a structural element B and is given by,

$$A \ominus B = \textit{intersection of all translates } A_b \textit{ wrt } b \in B = \bigcap_{b \in B} A_b$$

Function: erosion_2 (A, B)

Objective: To perform Minkowski Set Subtraction for the given input image and structuring element.

Input Parameters:

A **-** input binary image

B **-** input structuring element

Returned Result:

temp **-** Result of erosion on the input parameters

Processing Flow:

1. The size of the structuring element B is assigned to the variable P (rows), Q (columns) and R (color channels).
2. The index of the center of the structuring element is given by,

   center ← $(\lceil P/2 \rceil, \lceil Q/2 \rceil)$
3. Initialise x_B and y_B to the indices of B which has the value 1.
4. Initialise n_B to the length of x_B

   n_B ← length (x_B)
5. If n_B is greater than 0
   a. x_B ← x_B - $\lceil P/2 \rceil$
   b. y_B ← y_B - $\lceil Q/2 \rceil$
6. The size of the image A is assigned to the variables r (rows), c (columns) and chann (color channels).
7. [r2, c2] ← [r + P - 1, c + Q -1]
8. x_limit ← [$\lceil P/2 \rceil$ to $\lceil P/2 \rceil$+ r - 1]
9. y_limit ← [$\lceil Q/2 \rceil$ to $\lceil Q/2 \rceil$+ c - 1]

10. Assign image to matrix with new border:

    border_handle [x_limit, y_limit] $\leftarrow$ A

11. Replace image A with border_handle

    A $\leftarrow$ border_handle [x_limit, y_limit]

12. Assign 1 to neighbouring pixels based on condition on A.

    for i in [x_limit]

    a. for j in [y_limit]

        i. if A[i,j] = 1

            Initialise t $\leftarrow$ 1

            1. for k in [1, n_B]

                a. t $\leftarrow$ A [i + x_B [k], j + y_B [k]] * t

            2. result [i,j] $\leftarrow$ t

13. temp $\leftarrow$ result [x_limit, y_limit]

14. Since the problem is setup with a black background and white disks, we take the complement of temp in order to view the result.

Restrictions/Notes:

This input image and structuring element should be in binary.

No functions are called.

The following functions are calling:

hitormiss.m

openclose.m

v. Opening:

It is a morphological operation which performs erosion of an input image X by a structuring element B followed by dilation by the same structuring element B on the result of the erosion.

$$X_B = X \circ B = (X \ominus B^S) \oplus B \text{ where } B^S = B$$

vi. Closing:

It is a morphological operation which performs dilation of an input image X by a structuring element B followed by erosion by the same structuring element B on the result of the dilation.

$X^B = X \bullet B = (X \oplus B^S) \ominus B$ where $B^S = B$

To remove the remaining noise present in the image, X^B image is again dilated with the structuring element B.

$Z = X^B \oplus B$

Function: openclose(A, B) (involves v and vi)

Objective: To remove salt and pepper noise with the help of the closing and opening operations.

Input Parameters:

A **-** input binary image

B **-** input structuring element

Returned Result:

Z2 **-** Result of close/open filtering

Processing Flow:

1. A is the noisy binary input image and B is a 2x2 structuring element of ones.
2. Z $\leftarrow$ dilation (A,B)
3. close_filter $\leftarrow$ erosion (Z,B)
4. Z2 $\leftarrow$ dilation (close_filter, B)

Restrictions/Notes:

This input image and structuring element should be in binary.

The following functions are called:

erosion_2.m

dilation_2.m

The following functions are calling:

Project_main.m

vii. **Hit or miss transform**:

It is a morphological operation which performs erosion of input image X with a structuring element A, a dilation of the same input image with a structuring element B and finally performs a set difference of the dilation result from the erosion result.

$X \odot (A, B) = (X \ominus A^s) / (X \oplus B^s)$

where X is the image object and A^s=A and B^s=B are the structuring elements.

$\odot$ denotes the hit or miss transform

$\ominus$ denotes the erosion operation

$\oplus$ denotes the dilation operation

Since the indices used to represent the matrix of the images in MATLAB are all positive and the indices represent the points where the pixels in the images are located, it can be assumed that the images are present only in the positive axes. So, A and B are used in the hit or miss transform instead of A^s and B^s and the modified formula is

$$X \odot (A, B) = (X \ominus A) / (X \oplus B)$$
$$= (X \ominus A) \cap (X \oplus B)^c$$

The hit or miss transform has been executed in the code *hitormiss.m* in MATLAB. It is invoked by the main function *Project_main.m*. The pseudocode of *hitormiss.m* is as follows:

Function: **hitormiss(X,A,B)**

Objective:

To execute the hit or miss transform morphological operation

Input Parameters:

X: image object

A: foreground structuring element

B: background structuring element

Returned Result:

ret: The image obtained after executing hit or miss transform

Processing Flow:

1. The image X is eroded by the structuring element A by calling the erosion_2 function and the resulting image is stored in the X1 variable.
2. The image X is dilated by the structuring element A by calling the dilation_2 function and the resulting image is stored in the X2 variable.
3. The complement operation is carried on X2 by calling the not_operator function and the resulting image is stored in the X3 variable.
4. The intersection operation is executed on X1 and X3 by the .* (element -wise multiplication) operator of MATLAB. This is used as X1 and X3 are binary images of type double which increases the precision of the image.
5. The ret variable (output of the hit or miss transform) is returned by the function.

Restrictions/Notes:

This function requires a binary image as input and 2 binary images as the 2 structuring elements.

The following functions are called:

erosion_2.m

dilation_2.m

not_operator.m

The following functions are calling:

Project_main.m

C. **Results**

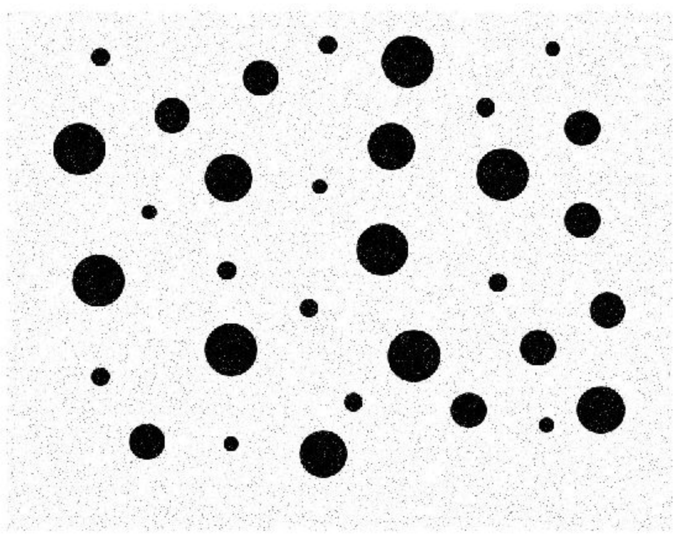


Figure 2. Original input image with 5 types of disks

This is the original RGB input image with salt and pepper noise. Using this image directly will lead to incorrect result as it is of type uint8. So, it is converted to a grayscale image in Figure 3 by the procedure described in the Methods section.

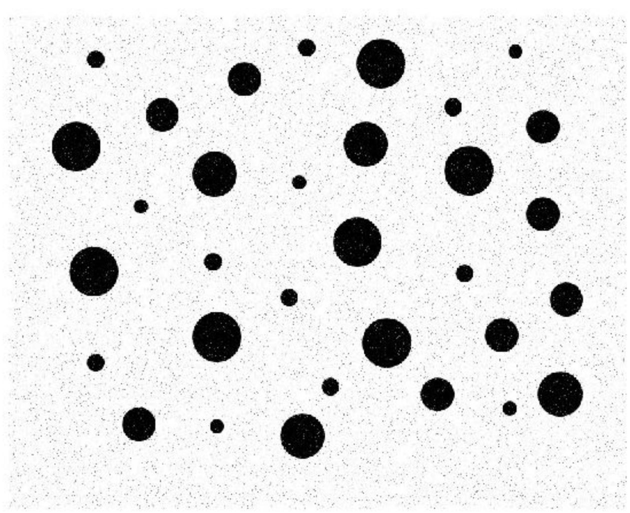
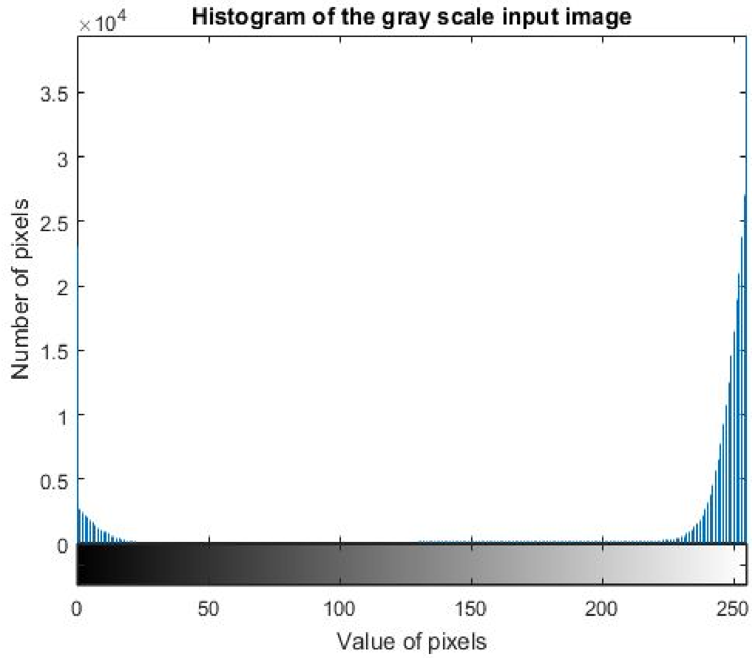


Figure 3. Gray scale image after converting the RGB image in Figure 2 to grayscale

This image is of type double. But the pixel values in the image are between 0 and 255 which cannot be used for processing in the image. So, this image is converted to binary image in Figure 4 as described in the Methods section.



Graph 1. Histogram of the gray scale image in Figure 3 to select a threshold value

The threshold value is determined by observing the histogram of the gray scale image in Figure 3. The pixels which are whiter are concentrated on one end of the histogram while the pixels which are blacker are concentrated on the other end of the histogram. From the histogram, there are more white pixels than black as the background of the image is white which is a major section of the image. A threshold value of 95 is selected as described in Section iii of Results.

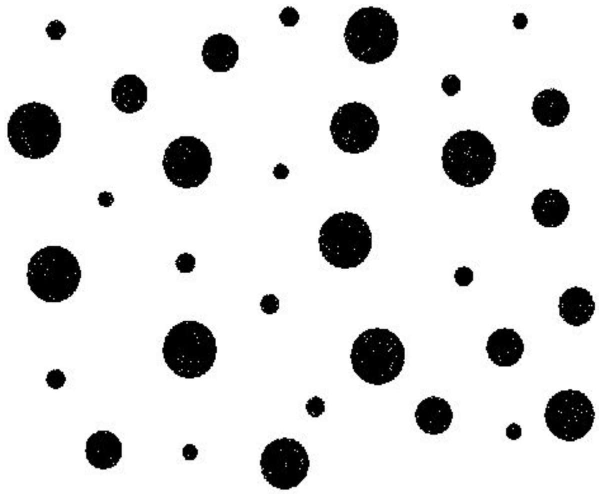


Figure 4. Binary image after converting the gray scale image in Figure 3 to binary with threshold value 95 based on observing graph 1

The binary image obtained still has noise which can be removed by passing the image through a close/open filter.

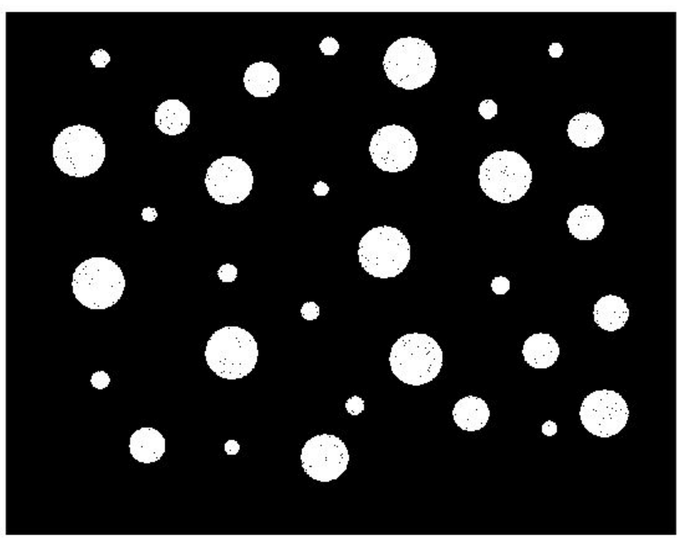


Figure 5. Complement of the binary image in Figure 4 to be used as input to the openclose and hitormiss functions

The binary image is complemented before processing by the openclose function which uses the erosion_2 and dilation_2 functions. This is because the implementation of erosion_2 and dilation_2 considers the background pixels to be 0 and the foreground pixels to be 1 while the image in Figure 4 is represented in just the opposite manner.

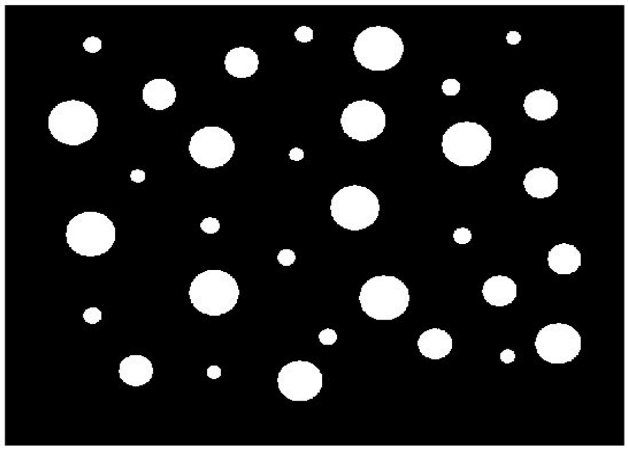


Figure 6. Dilation of the image in Figure 5 with an image stored in 2x2 matrix of ones as a part of the closing operation

Figure 6 obtained after dilation is the first step of the closing operation.

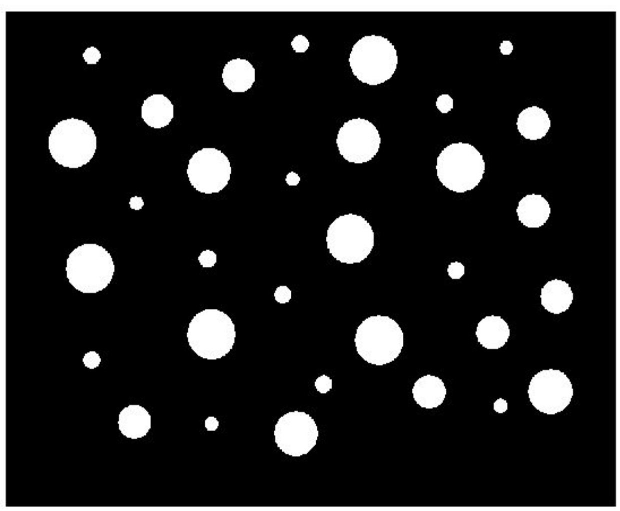


Figure 7. Erosion of Figure 6 with a structuring element of 2x2 ones (Closing)

Figure 7 obtained after erosion is the image obtained after the closing operation.

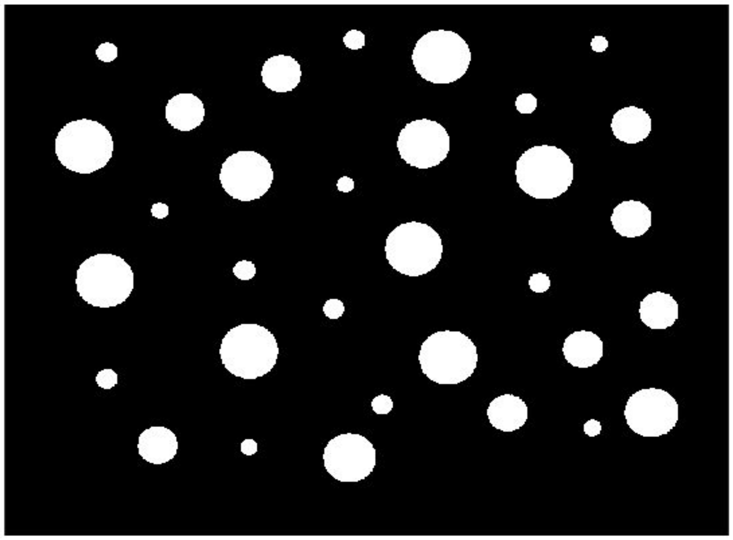


Figure 8. Dilation of Figure 7 to obtain the final close/open filter result

The closing of the image is dilated once to remove the remaining noise in Figure 7.

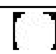
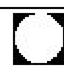
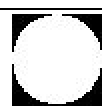
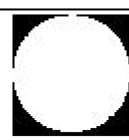
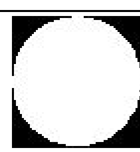
Serial no.	Description of the disk present in Figure 8	Image	Diameter (in units)	Number of disks
1	5th largest disk		19	5
2	4th largest disk		27	8
3	3rd largest disk		47	8
4	2nd largest disk		63	4
5	Largest disk		68	7

Table 1. Description of the 5 differently sized disks present in Figure 8

i. Selection of the structuring elements A and B (Answer to question 3b)

The 5 different sized disks in the image were cropped out of the image by using the imtool function of MATLAB. After removal of salt and pepper noise as mentioned in Figure 8 under Section C (Results), the imtool operation displayed the image. The diameter of the 5 types of disks were measured by manually selecting them and using the Measure Distance operation from the Tools MenuBar. The 5 disks and their diameters are documented in Table 1 under Section C (Results).

The selected structuring element A image cropped had dimensions 19x23 which is a bit bigger than the 5th largest disk with a diameter of 19 units. This was done, so that the 5th largest disk is removed by erosion from the image X and only the eroded images of the largest, 2nd largest, 3rd largest and 4th largest disks are present in the image once ($X \ominus A$) is executed.

The structuring element B was the complement of a disk from a 66*66 window. The disk is slightly smaller than the 5th largest largest disk with diameter 68 units. This was done as the complement of the dilation represent the smallest, 4th largest, 3rd largest and 2nd largest disks.

An intersection of the operations described in the above 2 paragraphs will give eroded images of the 2nd, 3rd and 4th largest disks.

These 3 types of disks can be removed from the initial image by a set difference operation on the input image obtained after removal of salt and pepper noise.

Thus, the largest and the smallest disks can be detected in the image by the above selection of structuring elements.

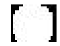


Figure 9. Structuring element A in the hit or miss transform

The structuring element A is selected by the procedure described in section i above with dimension 19x23. It is a bit larger than the smallest disk with diameter 19.

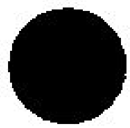


Figure 10. Structuring element B in the hit or miss transform

The structuring element B is selected by the procedure described in section i above with dimension 66x66. The disk which is complemented from the 66x66 window is a bit smaller than the largest disk with diameter 68.

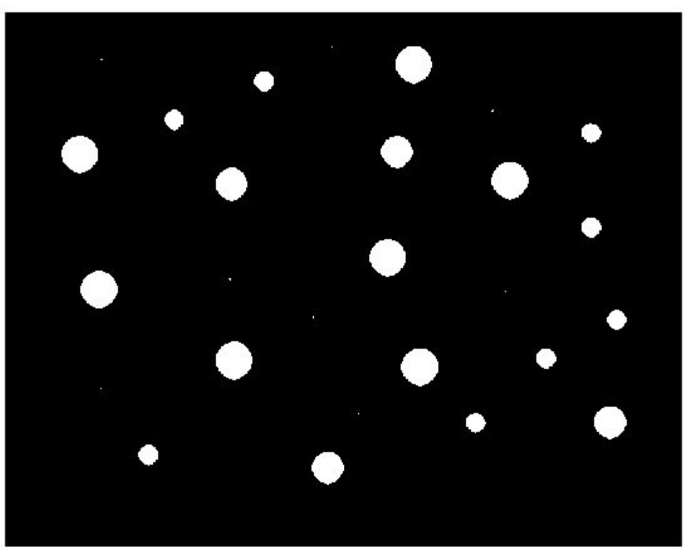


Figure 11. Erosion of the image in Figure 8 by the structuring element in Figure 9 as the 1st step of the hit or miss transform

This is the first step of the hit or miss transform. Eroded disks which are the largest, 2nd largest, 3rd largest and the fourth largest are obtained while the fifth disk is removed totally.

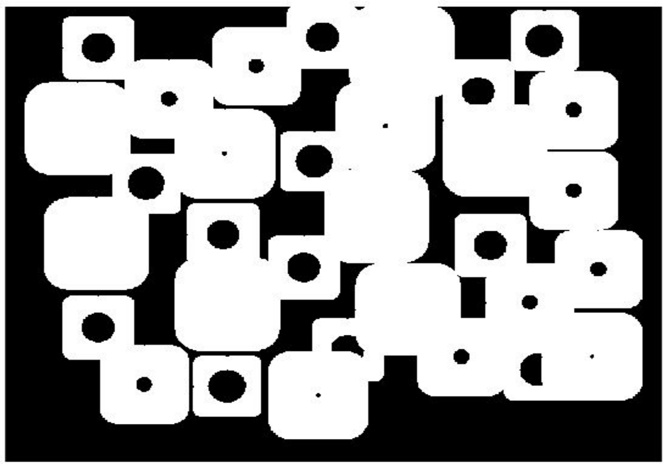


Figure 12. Dilation of the image in Figure 8 by the structuring element in Figure 10 as the 2nd step of the hit or miss transform

This is the second step of the hit or miss transform.

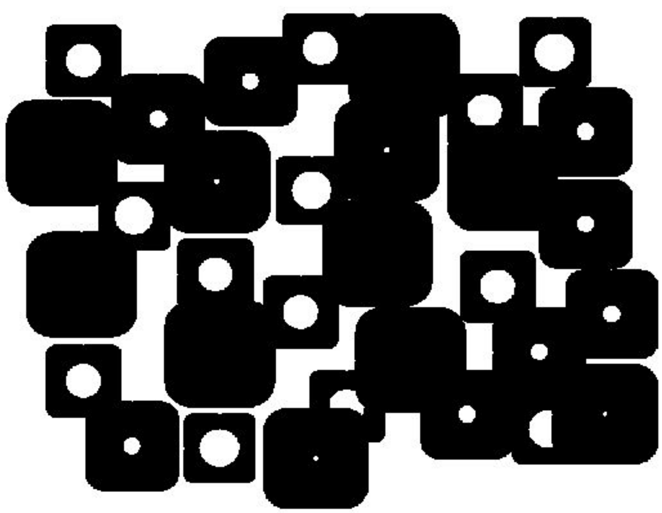


Figure 13. Complement of the image in Figure 12

Complement of the dilated image in Figure 12 leads to the smallest, 2nd largest, 3rd largest and 4th largest disks to stay in the image while the disks which are the largest are removed totally.

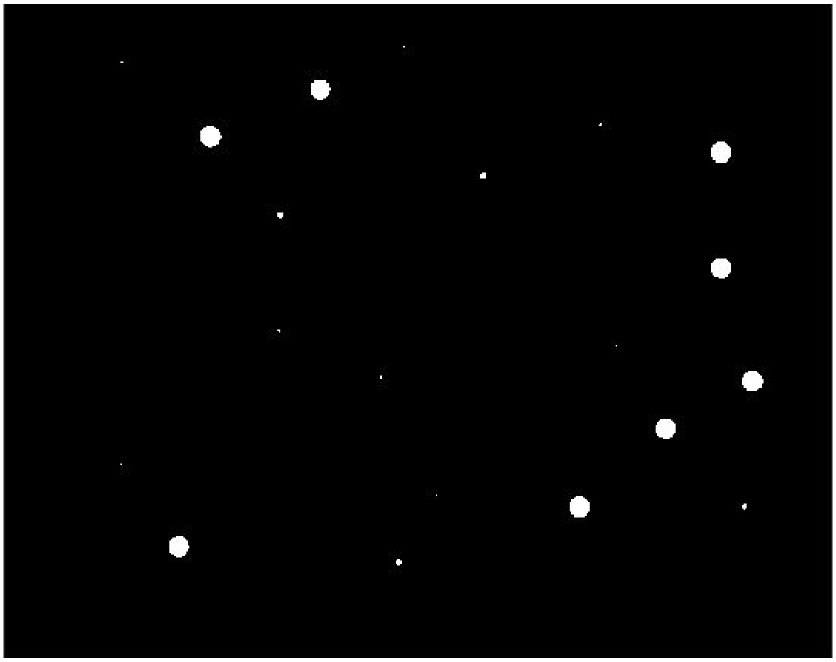


Figure 14. Hit or miss transform of the image in Figure 8 by the structuring elements (A,B) in Figures 9 and 10 respectively.

The hit or miss transform leads to 20 eroded disks which represent the sizes of 2nd largest, 3rd largest and 4th largest. This matches with the observation in Table 1 where there are four 2nd largest disks, eight 3rd largest disks and eight 4th largest disks which add up to a total of 20 disks. But since, they are eroded, they need to be dilated by the disk which was complemented in Figure 10.

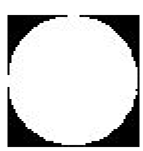


Figure 15. Structuring element to dilate the output of the hit or miss transform and get the three middle-sized disks

This is a complement of the image in Figure 10 which is the disk which is a bit smaller than the largest disk but larger than all other types of disks.

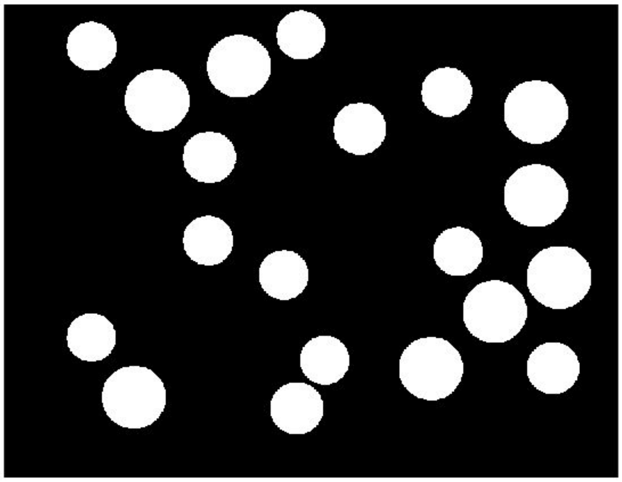


Figure 16. Dilation of image in Figure 14 by image in Figure 15 to recognize the 3 middle-sized disks from the image in Figure 8

20 disks which are the 2nd, 3rd and 4th largest are obtained after dilation.

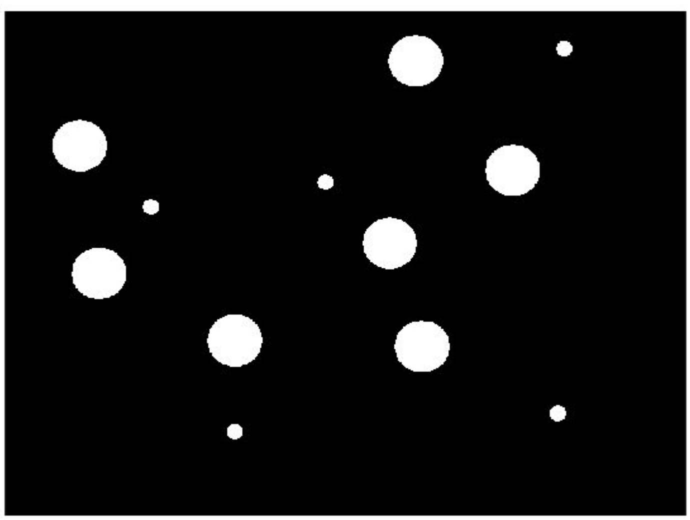


Figure 17. Set difference of image in Figure 8 by image in Figure 16 to recognize the largest disks and the smallest disks in the image in Figure 8

A set difference is done on the image in Figure 8 by the image in Figure 16, which recognises the largest and the smallest disks. But the background is black and the foreground is white due to the assumption as observed in Figure 5.

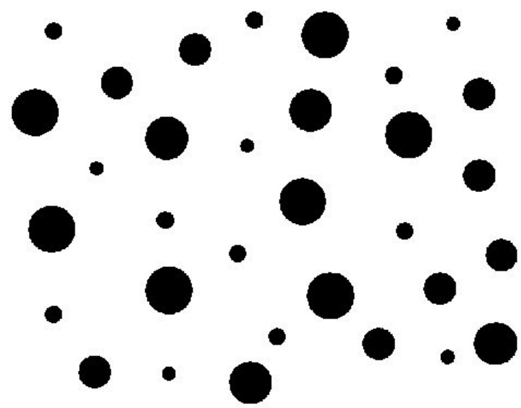


Figure 18. Complement of the image in Figure 8

This is a representation of the actual input image as asked in the question which is a complement of the image in Figure 8.

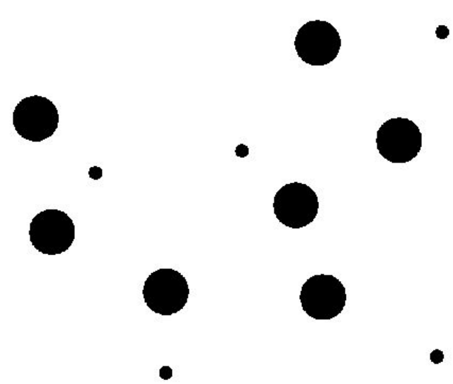


Figure 19. FINAL RESULT of the detected largest and the smallest disks in Figure 18 obtained by a complement operation on the image in Figure 17

The final result is obtained by complementing the image in Figure 17. Thus only the largest and smallest disks are present in Figure 19 which can be realized by comparing with Figure 18 which was the original image.

ii. Detection of the disks without close/open filter (Answer to question 3c)

The hit or miss transform will run on the noisy image but it will not give the expected output of the largest and the smallest disks as noisy pixels will be eroded and dilated by the structuring elements. This can be proved in the results obtained from Figure 20 to Figure 27.

Figure 20. Erosion of the image in Figure 5 by the structuring element in Figure 9 as the 1st step of the hit or miss transform

The erosion operation leads to many close eroded images due to the noise.

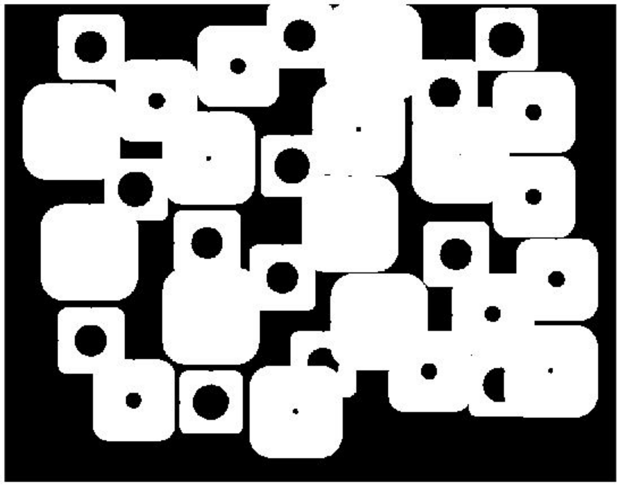


Figure 21. Dilation of the image in Figure 5 by the structuring element in Figure 10 as the 2nd step of the hit or miss transform

The dilation of the image is not much affected when comparatively observed with Figure 12.

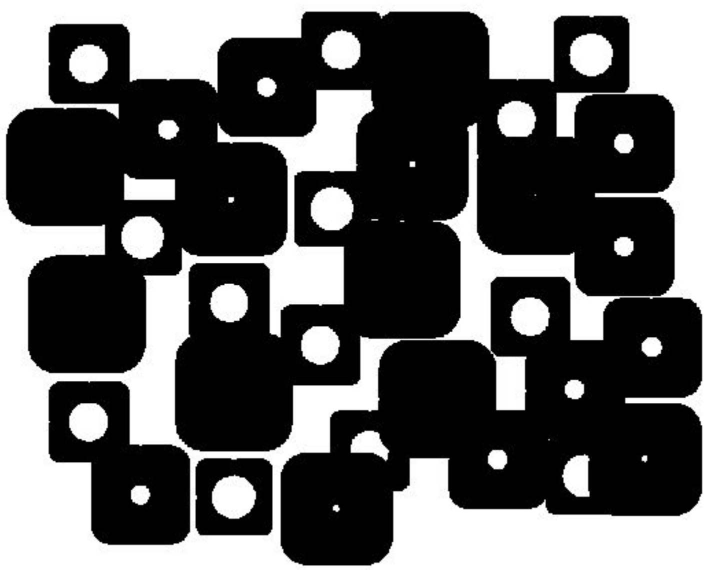


Figure 22. Complement of the image in Figure 21

The negation of the image in Figure 21 does not entirely remove the largest disks as observed Figure 22.

Figure 23. Hit or miss transform of the image in Figure 5 by the structuring elements (A,B) in Figures 9 and 10 respectively

The hit or miss transform has only 3 eroded disks in comparison to 20 eroded disks in Figure 14.

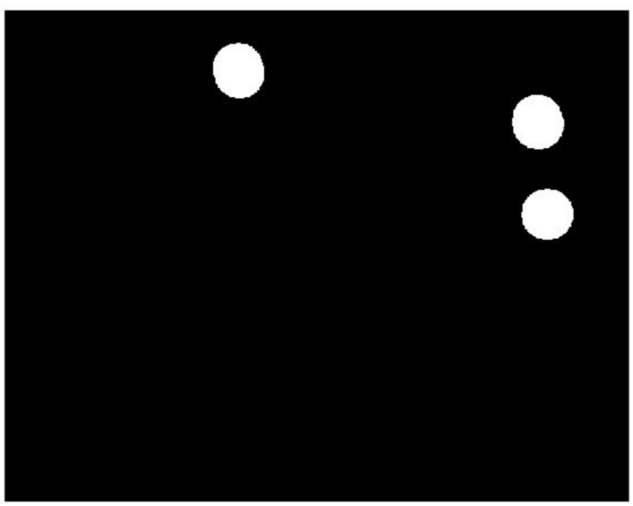


Figure 24. Dilation of image in Figure 23 by image in Figure 15 in an attempt to recognize the 3 middle-sized disks in the image in Figure 4

The 3 eroded disks are dilated in the same way as described in Figure 16.

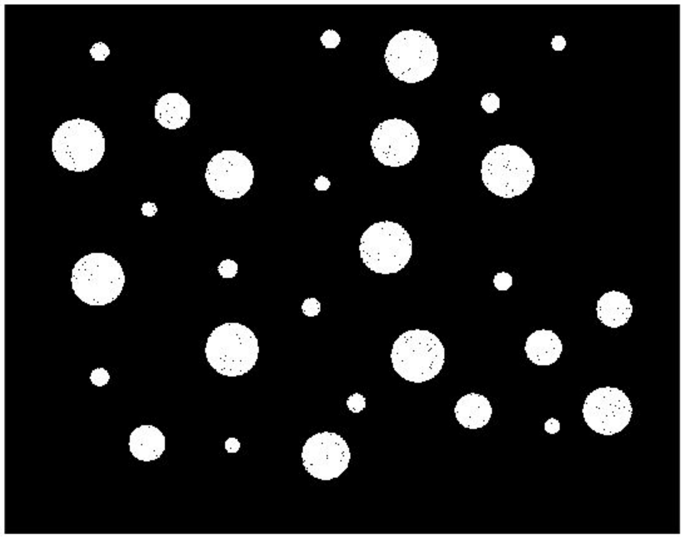


Figure 25. Set difference of image in Figure 5 by image in Figure 24 in an attempt to recognize the largest and the smallest disks in Figure 5

A set difference of the initial input image as in Figure 5 by the image in Figure 24 gives 29 disks with noise instead of the 12 disks which are the largest and the smallest as observed in Figure 17.

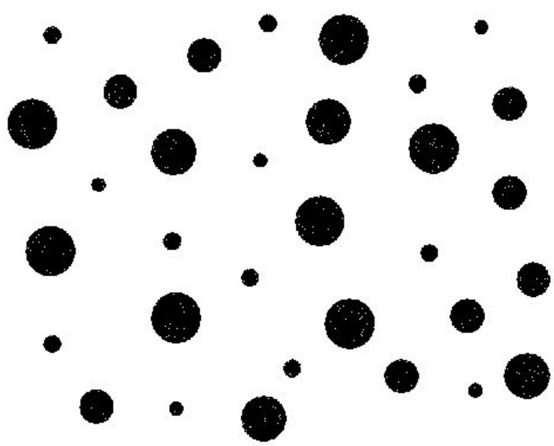


Figure 26. Complement of the image in Figure 5

This is a representation of the initial input image whose background is white and the foreground is black but noise is present.

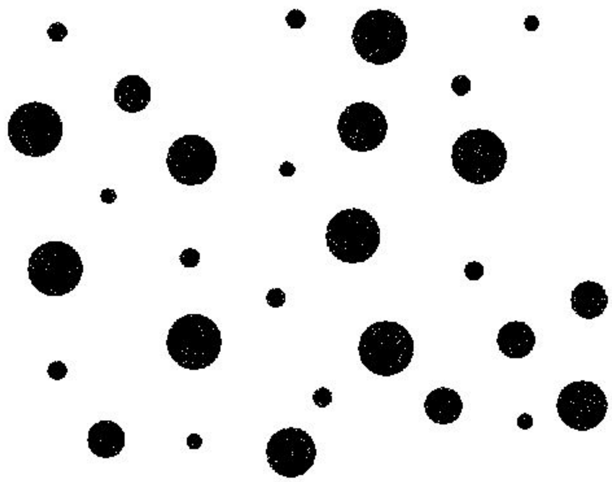


Figure 27. ATTEMPTED FINAL RESULT to detect the largest and the smallest disks in Figure 26 obtained by a complement operation on the image in Figure 25

Instead of the correctly recognized 12 disks which are the largest and the smallest in Figure 19, 29 disks are recognized when noise is present in the input image which indicates that the output is incorrect when noise is present in the input image to the hit or miss transform.

iii. Choice of thresholding for close/open filter (Question 3a)

The following figure is a small block diagram of the process flow in the process of removing salt and pepper noise. B is the structuring element input to all the three processes within the Close/Open filtering block. The details regarding this are in the Methods Section after definitions of Opening and Closing. (After Section B (vi)).

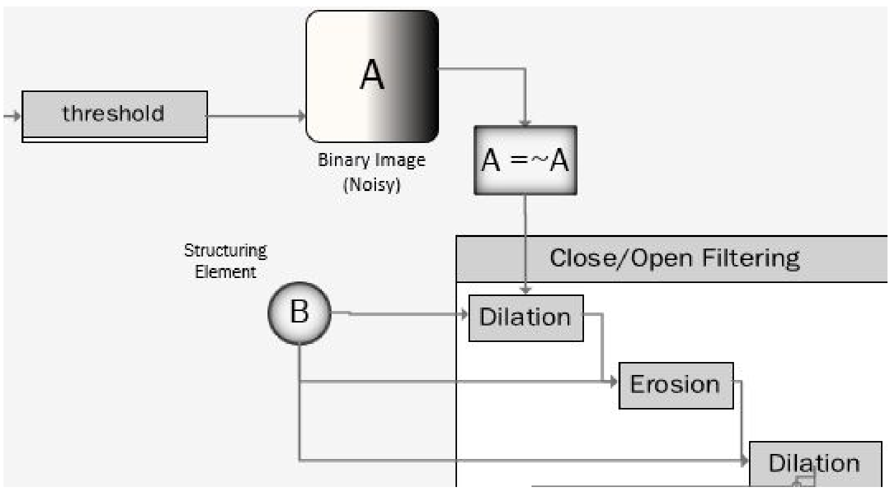


Figure 28: This is the process flow of the close and open filtering.

The first step to ensuring a good result for the Close/Open filtering is to choose an appropriate threshold value. The following (figure 26 and 27) contains results of Close/Open filtering on two different threshold values. These are the failure cases and figure 7 shows the successful result with a threshold value of 95.

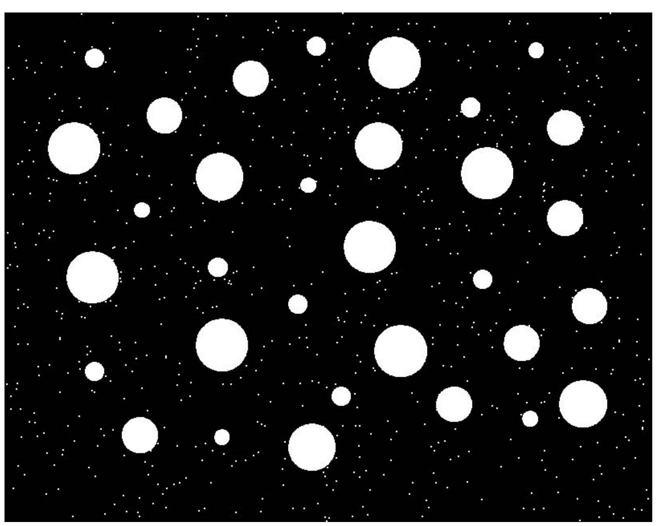


Figure 29: Close/Open Filter on a binary image which was thresholded with a value = 128. So, the pixels with values less than 128 are set to 0 and the rest are set to 1. There is still noise in this picture.

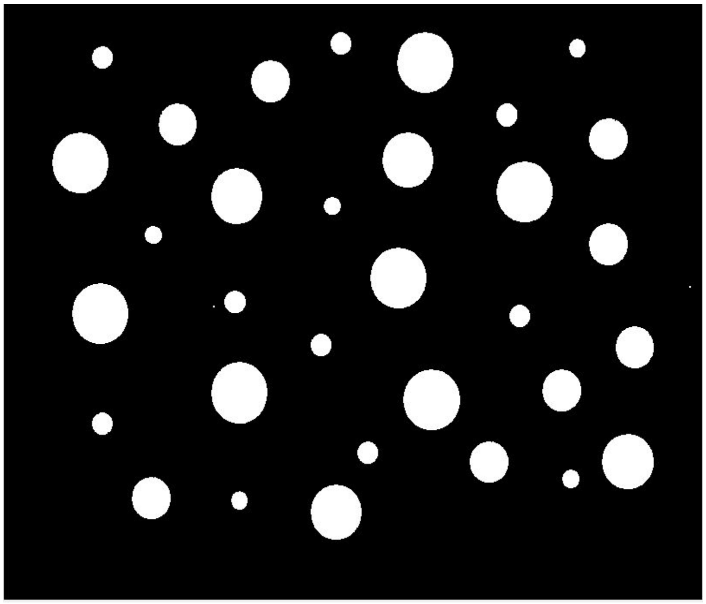


Figure 30: Close/Open Filter on a binary image which was thresholded with a value = 100. So, the pixels with values less than 100 are set to 0 and the rest are set to 1. There are a couple dots visible on the far middle right corner and one slightly left to the middle of the image.

Then, when we chose a threshold value of 95, the Close/Open filter result came out without any salt and pepper noise. (Figure 7). Basically, after the thresholding, the close/open filter is implemented by performing the following operation:

$$X = ((A \oplus B) \ominus B) \oplus B$$

A is the noisy binary input image
B is the 2x2 structuring element of ones.

This is how the close/open filter was exactly done.

D. Conclusions

It can be concluded that the largest and the second largest disks are correctly recognized by selecting the structuring element A as in Figure 9 which is a bit bigger than the smallest disk but smaller than the 2nd smallest disks and selecting the second structuring element B as in Figure 10 which is the complement of the disk from a window W=66x66 and the disk is slightly smaller than the largest disk but is larger than the 4th largest disk. There are 12 disks in the final result in Figure 19 which matches with the observation of 7 largest and 5 smallest disk as noted in Table 1.

The threshold value selection for the input image was very important. The reason for choosing a particular threshold value has been explained in figure 29 and 30. If the noise is not removed by the close/open filter and the noisy input image is used for hit or miss transform, the hit or miss transform will work but the disks will be recognized incorrectly as demonstrated from Figure 20 to Figure 27. The Salt and pepper noise removal was crucial in getting the correct results.

E. References

1. http://qu.edu.iq/el/mod/resource/view.php?id=51999