The University of Windsor

ELEC4490: Sensors and Vision Systems

Summer 2020

Assignment # 5

Transform Operations and Morphological Image Processing

Wednesday, August 12, 2020

Emmanuel Mati

104418019

Q1.1 Code:

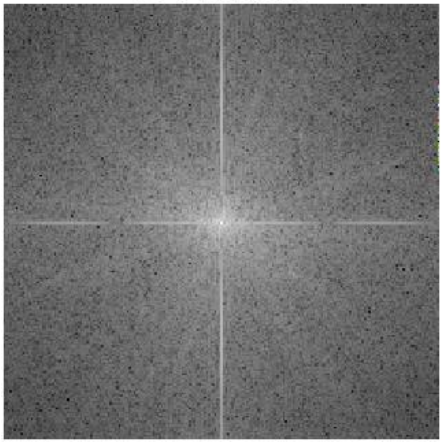
```
%Emmanuel Mati
%Summer 2020
%Sensors and Vision Systems
%Assignment 5-1.1

%Clearing previous results
close all
clear all
clc

%Retreiving our image and displaying it
peppers = rgb2gray(imread('peppers.jpg'));
wicker = rgb2gray(imread('wicker.jpg'));
wood = rgb2gray(imread('wood.jpg'));


%applying FFT2
peppersGraph = abs(fftshift(fft2(peppers)));
wickerGraph = abs(fftshift(fft2(wicker)));
woodGraph = abs(fftshift(fft2(wood)));

figure;imshow(peppers);title('peppers');
figure;imshow(log(peppersGraph), []);title('DFT peppers');

figure;imshow(wicker);title('wicker');
figure;imshow(log(wickerGraph), []);title('DFT wicker');

figure;imshow(wood);title('wood');
figure;imshow(log(woodGraph), []);title('DFT wood');
```

You can solve for frequency response intuitively using Fourier transforms. However, to do it for an image would take a very long time without the help of computers. Thus it is possible without computers but it would take a very long time.

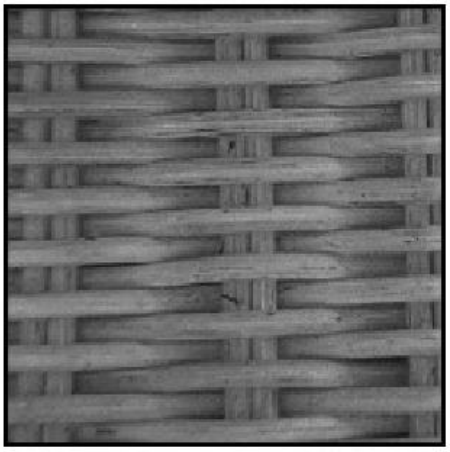
Output:

peppers

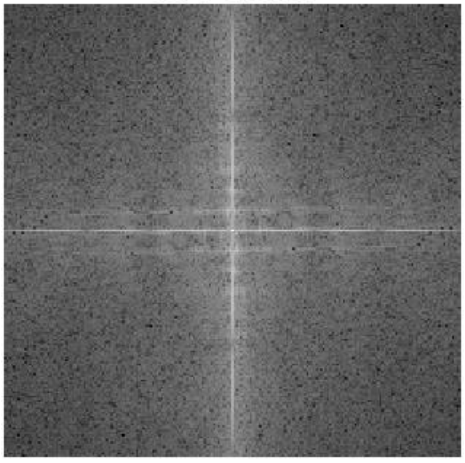
DFT peppers



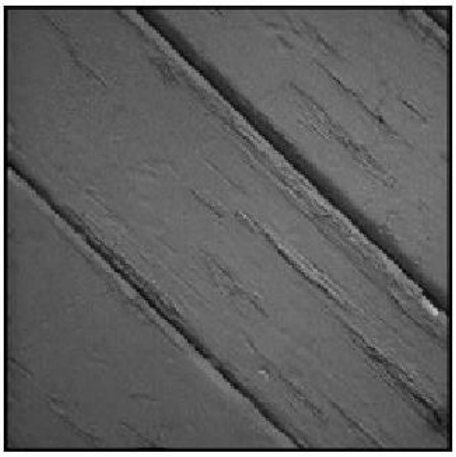
wicker



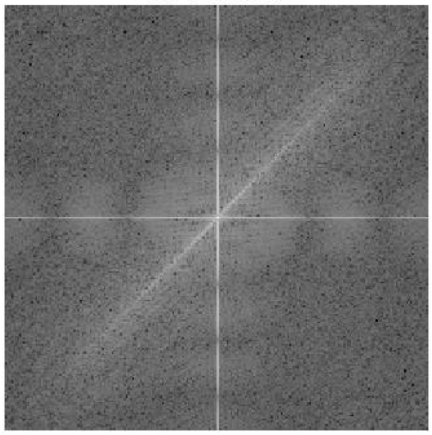
DFT wicker



wood



DFT wood

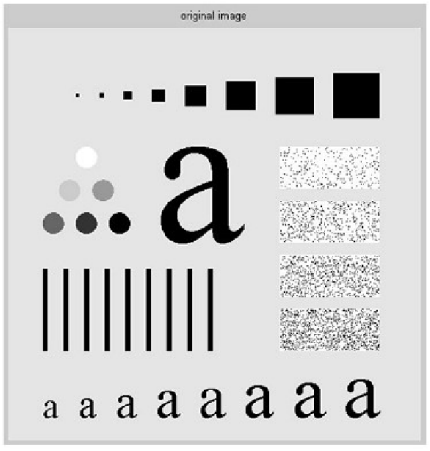


Q1.2 Code:

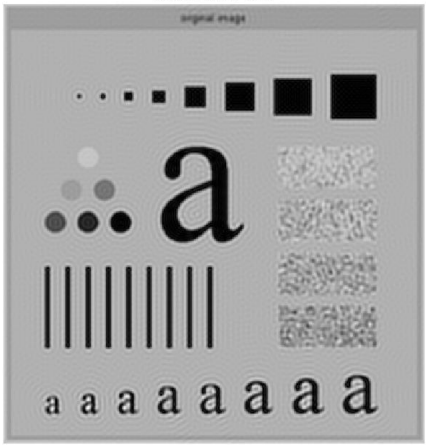
```
%Emmanuel Mati
%Summer 2020
%Sensors and Vision Systems
%Assignment 5-1.2

%Clearing previous results
close all;
clear all;
clc;

%%part a
I = rgb2gray(imread('a.jpg'));
[M,N] = size(I);
radii = 80;

[H] = lpfilter('ideal',M,N,radii);
%Retreiving our image and displaying it
F = fft2(I);
Z = F.*H;
i = ifft2(Z);
m = max(max(i));
figure;imshow(I);title('Original');
figure;imshow(abs(i)/m);title('Ideal Low-Pass');

%%part b
I = rgb2gray(imread('a.jpg'));
[M,N] = size(I);
radii = 80;

[H] = lpfilter('btw',M,N,radii);
%Retreiving our image and displaying it
F = fft2(I);
Z = F.*H;
i = ifft2(Z);
m = max(max(i));
figure;imshow(abs(i)/m);title('btw Low-Pass');

%%part c
I = rgb2gray(imread('a.jpg'));
[M,N] = size(I);
radii = 80;

[H] = lpfilter('gaussian',M,N,radii);
%Retreiving our image and displaying it
F = fft2(I);
Z = F.*H;
i = ifft2(Z);
m = max(max(i));
figure;imshow(abs(i)/m);title('Gaussian Low-Pass');
```

Output:

Original



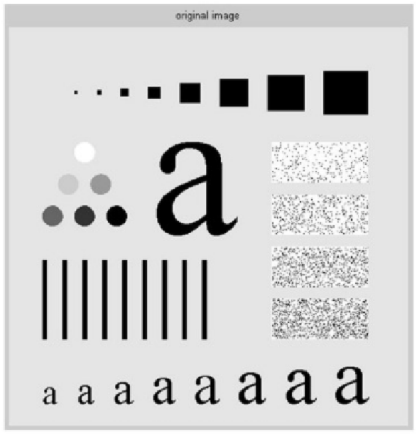
Ideal Low-Pass



a) **dev= 80**

Image edges have become blurred and darkened.

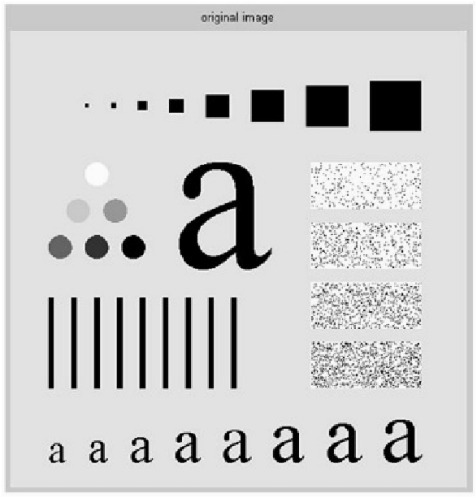
btw Low-Pass



b) **deviation = 160**

Image edges have become sharper.

Gaussian Low-Pass



c) **deviation = 240**

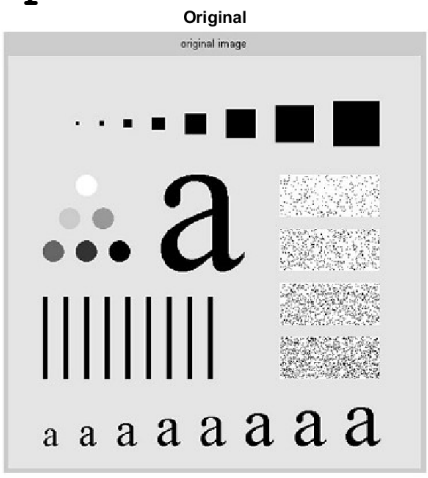
Edges have become just as sharp as original.

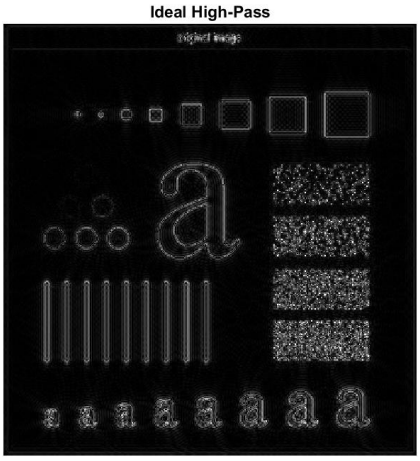
Q1.3 Code:

```
%Emmanuel Mati
%Summer 2020
%Sensors and Vision Systems
%Assignment 5-1.3

%Clearing previous results
close all;
clear all;
clc;

%%part a
I = rgb2gray(imread('a.jpg'));
[M,N] = size(I);
radii = 80;

[H] = hpfilter('ideal',M,N,radii);
%Retreiving our image and displaying it
F = fft2(I);
Z = F.*H;
i = ifft2(Z);
m = max(max(i));
figure;imshow(I);title('Original');
figure;imshow(abs(i)/m);title('Ideal High-Pass');

%%part b
I = rgb2gray(imread('a.jpg'));
[M,N] = size(I);
radii = 160;

[H] = hpfilter('btw',M,N,radii);
%Retreiving our image and displaying it
F = fft2(I);
Z = F.*H;
i = ifft2(Z);
m = max(max(i));
figure;imshow(abs(i)/m);title('btw High-Pass');

%%part C
I = rgb2gray(imread('a.jpg'));
[M,N] = size(I);
radii = 160;

[H] = 0.5+1.5*hpfilter('btw',M,N,radii, 2);
%Retreiving our image and displaying it
F = fft2(I);
Z = F.*H;
i = ifft2(Z);
m = max(max(i));
figure;imshow(abs(i)/m);title('btw High-Pass with High Frequency');
```

```
%%part d
I = rgb2gray(imread('a.jpg'));
[M,N] = size(I);
radii = 240;

[H] = hpfilter('gaussian',M,N,radii);
%Retreiving our image and displaying it
F = fft2(I);
Z = F.*H;
i = ifft2(Z);
m = max(max(i));
figure;imshow(abs(i)/m);title('Gaussian High-Pass');
```

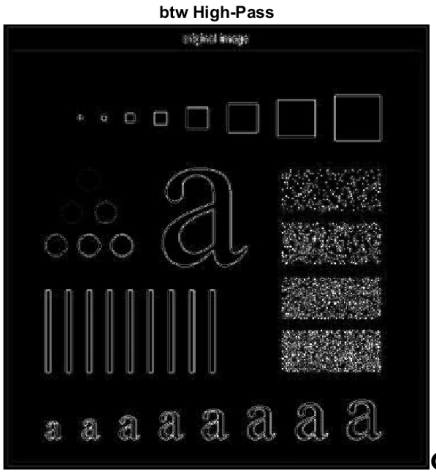
Output:

a) 



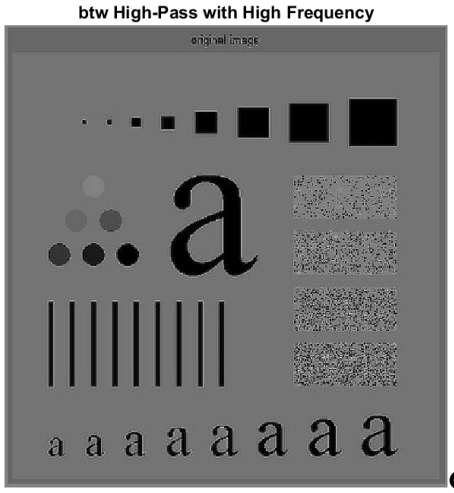
dev= 80

Image edges have become very blurred.

b) 

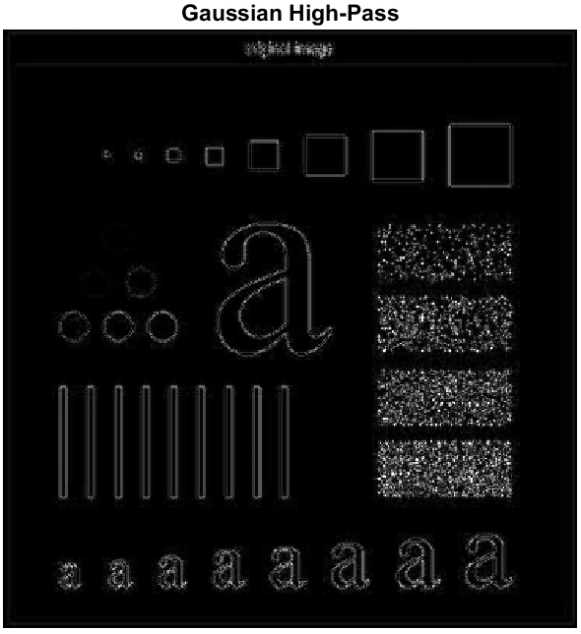
deviation = 160

Image edges are slightly lighter and sharper.



c) deviation = 160

Image edges have become sharper and turned gray.



d) deviation = 240

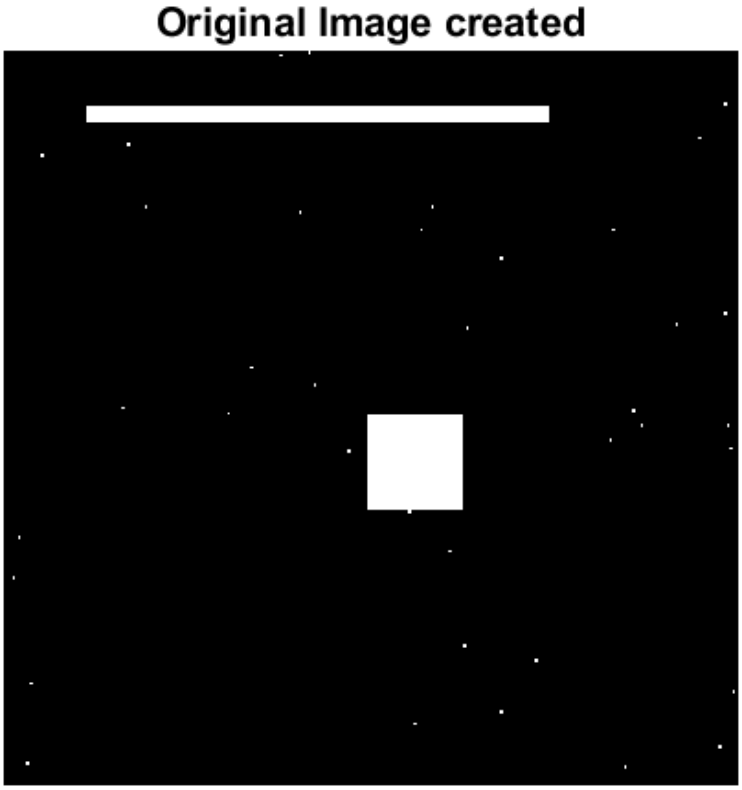
Edges have become blurred once more.

Q2.1 Code:

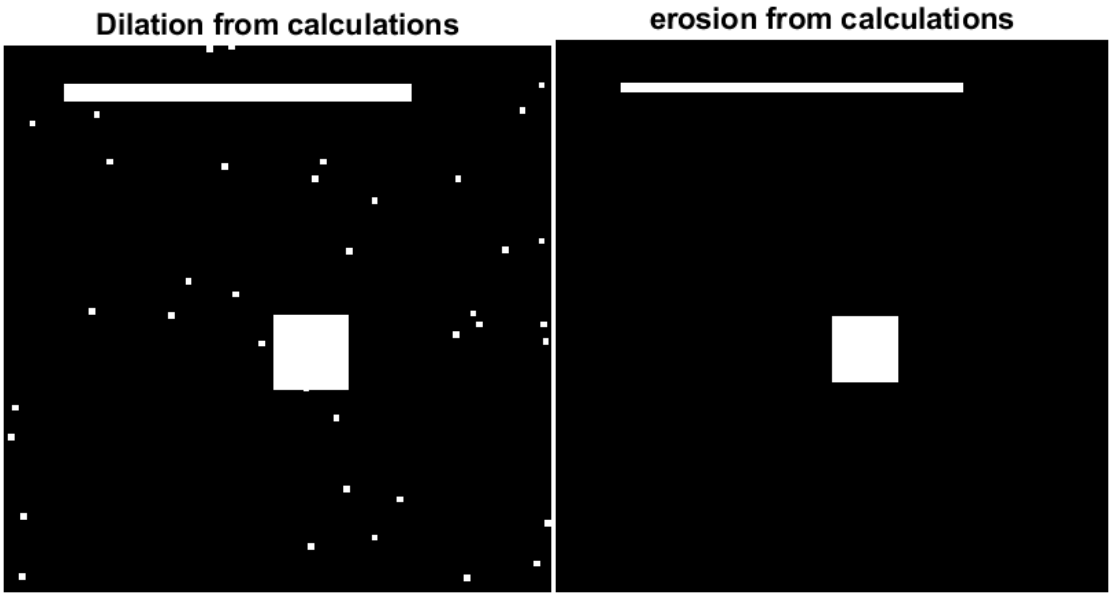
```
%Emmanuel Mati
%Summer 2020
%Sensors and Vision Systems
%Assignment 5-2.1

%Clearing previous results
close all;
clear all;
clc;

%%part a

%User input for SE
SE = input('Enter your structuring element. Press enter for deault [1 1 1;1 1
1;1 1 1]: ');

if isempty(SE)
    SE = [1 1 1;1 1 1;1 1 1]
end
% Code from lecture
% Creating 256x256 image

B=zeros(256,256);
for i=128:160
    for j=128:160
        B(i,j)=1;
    end
end

for i=20:25
    for j=30:190
        B(i,j)=1;
    end
end

i=1;
while i<40 % generate 40 random pixels
    x=uint8(rand*254)+1;
    y=uint8(rand*254)+1;
    B(x,y)=1;i=i+1;
end

figure;imshow(B);title('Original Image created'); %Original Image


%dialation
padB = padarray(B,[1,1]);
newImg = zeros(size(B));
for x = 1: size(B,1)
    for y = 1: size(B,2)
        newImg(x, y) = sum(SE & padB(x:x+2, y:y+2), 'all'); %summing matrix
of values that are to be dialated
    end
```

```
end

figure;imshow(newImg);title('Dilation from calculations');
figure;imshow(imdilate(B, SE));title('imdilate results'); %part b

%erosion
padBe = padarray(B,[1,1],1);
newImge = zeros(size(B));
for x = 1:size(padBe, 1)-2
    for y = 1:size(padBe, 2)-2
       xx = padBe(x:x+2,y:y+2);
       yy = find(SE == 1);

        if(xx(yy)==1)
            newImge(x,y)=1; %erroding the values in the image outside of mask
        end
    end
end

figure;imshow(newImge);title('erosion from calculations');
figure;imshow(imerode(B, SE));title('imerode results'); %part b
```

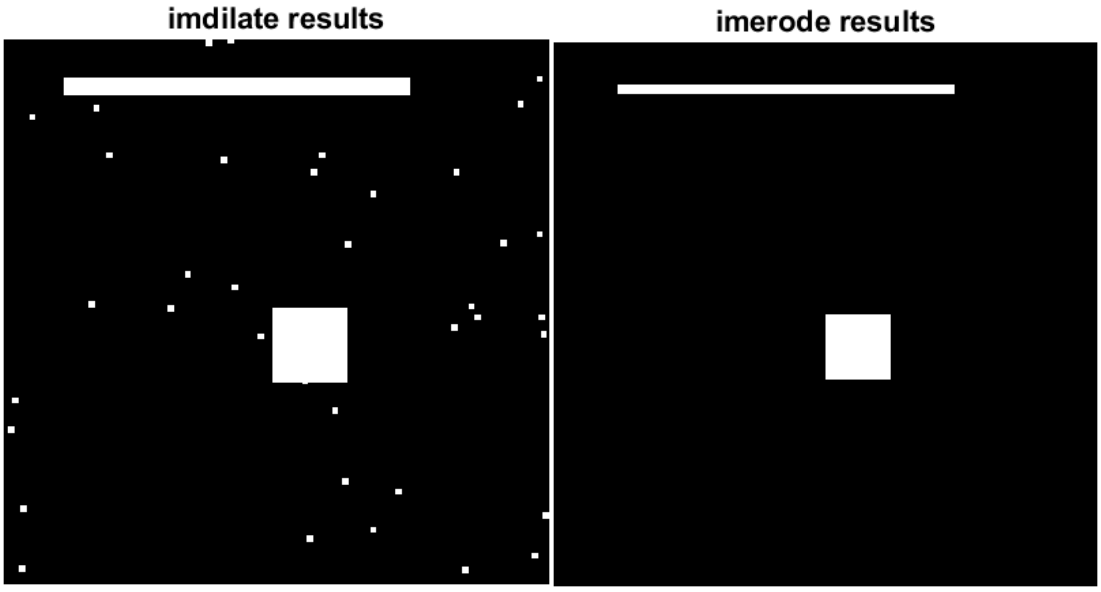
Output: SE = [1 1 1;1 1 1;1 1 1]



a)



b)

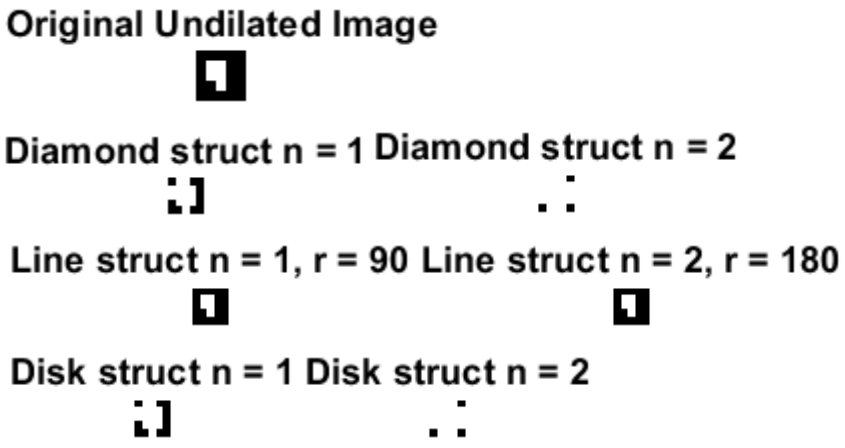


Results look the same between the built in MATLAB function and our own.

Q2.2 Code:

```
%Emmanuel Mati
%Summer 2020
%Sensors and Vision Systems
%Assignment 5-2.2

%Clearing previous results
close all;
clear all;
clc;

%User input for SE
Image=[0 0 0 0 0;0 1 1 0 0;0 1 1 0 0;0 0 1 0 0;0 0 0 0 0]


%structing elements
diaStruct1 = strel('diamond', 1)
diaStruct2 = strel('diamond', 2)

lineStruct1 = strel('line', 1, 90)
lineStruct2 = strel('line', 2, 180)

diskStruct1 = strel('disk', 1)
diskStruct2 = strel('disk', 2)

%applying dialation
figure;imshow(Image);title('Original Undilated Image');
figure;imshow(imdilate(Image,diaStruct1));title('Diamond struct n = 1');
figure;imshow(imdilate(Image,diaStruct2));title('Diamond struct n = 2');
figure;imshow(imdilate(Image,lineStruct1));title('Line struct n = 1, r =
90');
figure;imshow(imdilate(Image,lineStruct1));title('Line struct n = 2, r =
180');
figure;imshow(imdilate(Image,diskStruct1));title('Disk struct n = 1');
figure;imshow(imdilate(Image,diskStruct2));title('Disk struct n = 2');
```

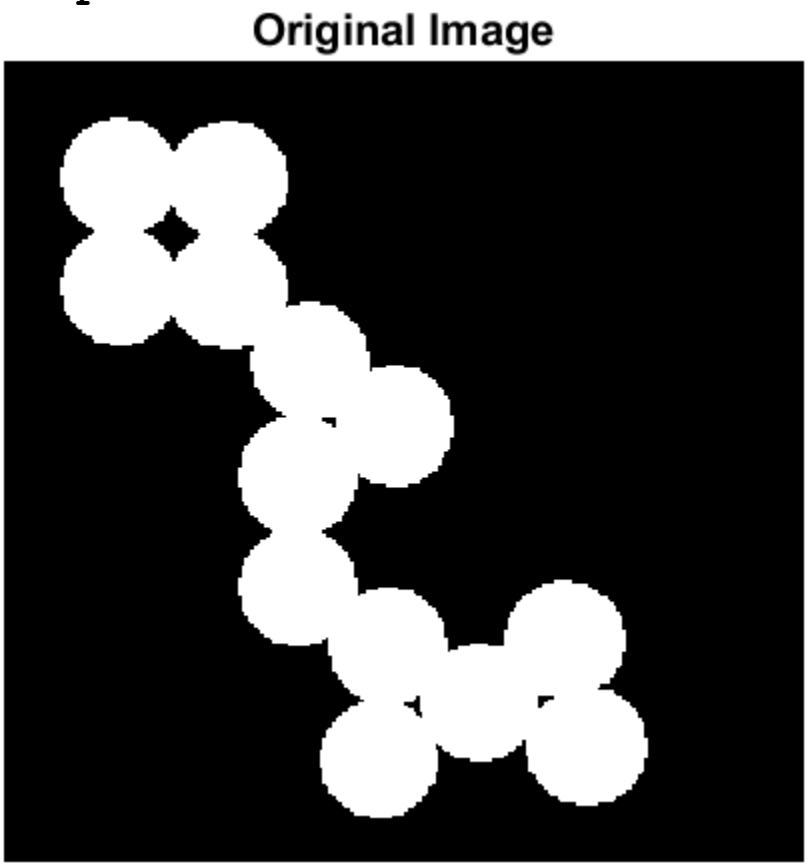
Output:



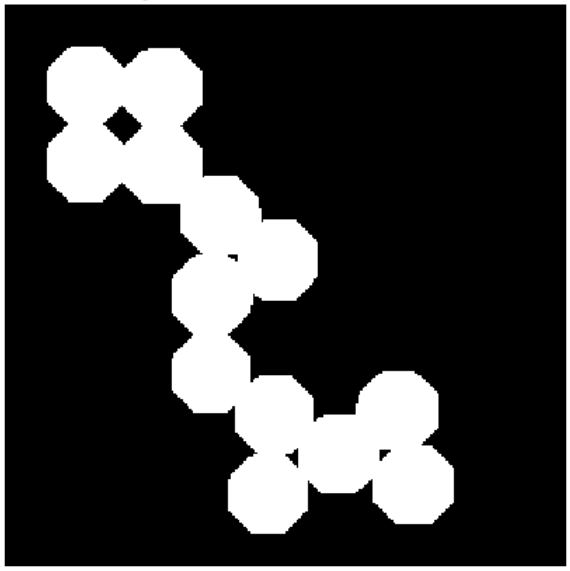
Q2.3 Code:

```
%Emmanuel Mati
%Summer 2020
%Sensors and Vision Systems
%Assignment 5-2.3

%Clearing previous results
close all;
clear all;
clc;

Image= im2bw(imread('binaryImage.png'));
SE1 = strel('disk',15); %struct used
SE2 = [1 0 0;0 1 0;0 0 1];
SE3 = [0 0 0;1 0 0;0 0 0];

%Output
figure;imshow(Image);title('Original Image');
figure;imshow(imopen(Image, SE1));title('a. Imopen with disk struct r = 15');
%a
figure;imshow(imclose(Image, SE1));title('b. Imclose with disk struct r =
15'); %b
figure;imshow(bwhitmiss(Image, SE2, SE3));title('C. bwhitmiss with SE =
[100;010;001] & [000;100;000]'); %c
```

Output:

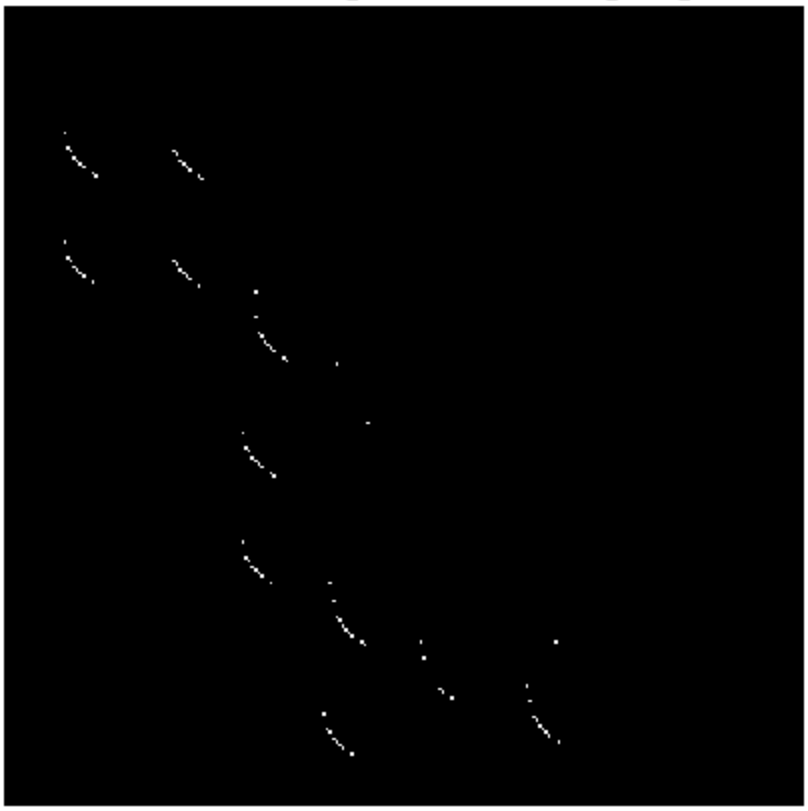


a. Imopen with disk struct r = 15



b. Imclose with disk struct r = 15

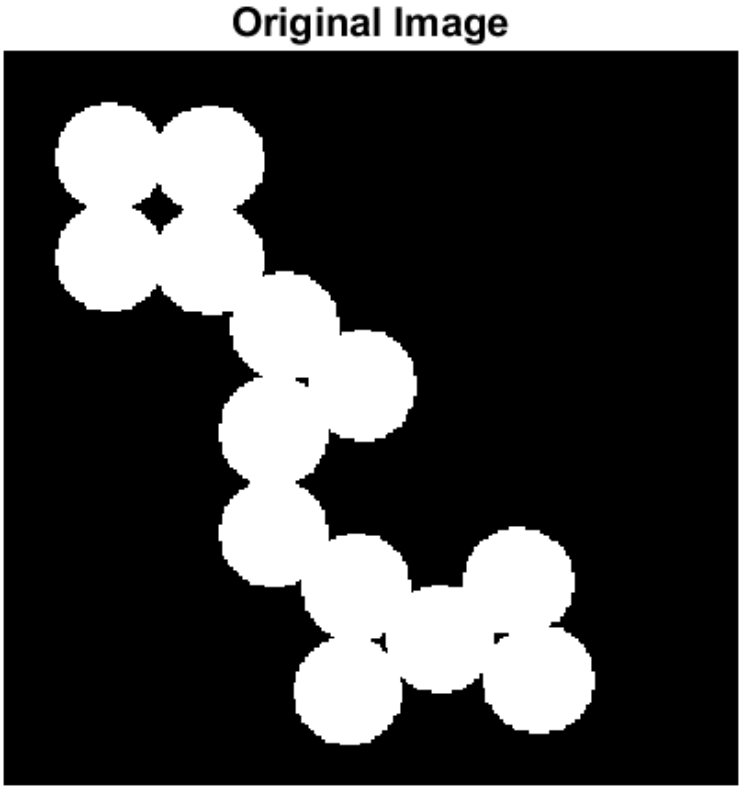
C. bwhitmiss with SE = [100;010;001] & [000;100;000]



Q2.4 Code:

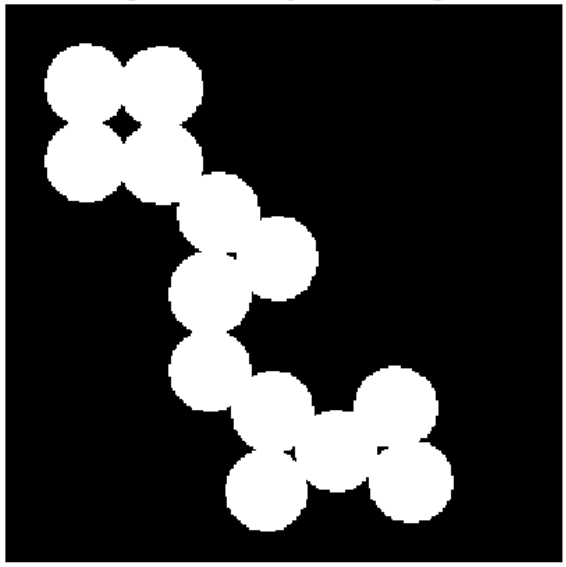
```
%Emmanuel Mati
%Summer 2020
%Sensors and Vision Systems
%Assignment 5-2.4

%%a
%Clearing previous results
close all;
clear all;
clc;

Image= im2bw(imread('binaryImage.png'));
[x, connectedObjects]= bwlabel(Image,4);
bwSelectImage = bwselect(Image);
%Output
figure;imshow(Image);title('Original Image');
figure;imshow(x);title(['This image has ',num2str(connectedObjects),' objects
using bwlabel']);
figure;imshow(bwSelectImage);title('bwselect image');

%%b
figure;imshow(bwmorph(Image,'skel',Inf));title('4b-b Image using skel');
figure;imshow(bwmorph(Image,'shrink',Inf));title('4b-c Image using shrink');
figure;imshow(bwmorph(Image,'remove',Inf));title('4b-d Image using remove');
```

Output:

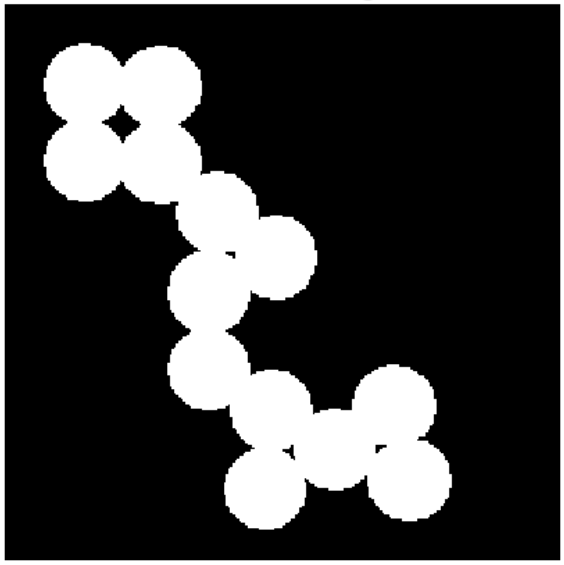


Part A:

This image has 1 objects using bwlabel



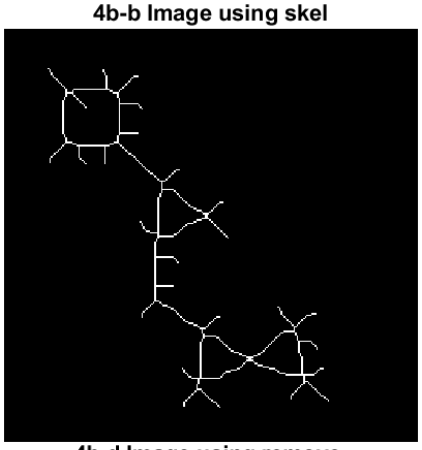
bwselect image



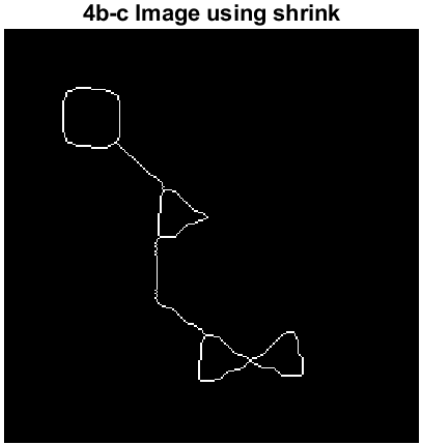
Note: *because we were given an image with connected edges, bwlabel and bwselect treat them as one image. To correct this, we would need to erode the edges until we get each circle by itself. However, this is not what the questions asked.*

Part B:

4b-b Image using skel



4b-c Image using shrink



4b-d Image using remove