Metehan Kaya
21401258
CS 484 - Fall 2018

Homework 1

Question 1

dilation.m (Second definition)

```
function binary_image = dilation( source_image , struct_el )
    [ n , m ] = size( source_image );
    [ h , w ] = size( struct_el );
    a = ( h - 1 ) / 2;
    b = ( w - 1 ) / 2;
    binary_image = zeros( n , m );
    for i = 1 : n
        for j = 1 : m
            if source_image( i , j ) == 1
                for r = -a : a
                    if i + r >= 1 && i + r <= n
                        for c = -b : b
                            if j + c >= 1 && j + c <= m
                                if struct_el( r + a + 1 , c + b + 1 ) == 1
                                    binary_image( i + r , j + c ) = 1;
                                end
                            end
                        end
                    end
                end
            end
        end
    end
end
```

erosion.m (Second definition)

```
function binary_image = erosion( source_image , struct_el )
    [ n , m ] = size( source_image );
    [ h , w ] = size( struct_el );
    a = ( h - 1 ) / 2;
    b = ( w - 1 ) / 2;
    binary_image = ones( n , m );
    for i = 1 : n
        for j = 1 : m
            for r = -a : a
                if i + r >= 1 && i + r <= n
                    for c = -b : b
                        if j + c >= 1 && j + c <= m
                            if struct_el( r + a + 1 , c + b + 1 ) == 1 &&
source_image( i + r , j + c ) == 0
                                binary_image( i , j ) = 0;
                            end
                        end
                    end
                end
            end
        end
    end
end
```

Question 2

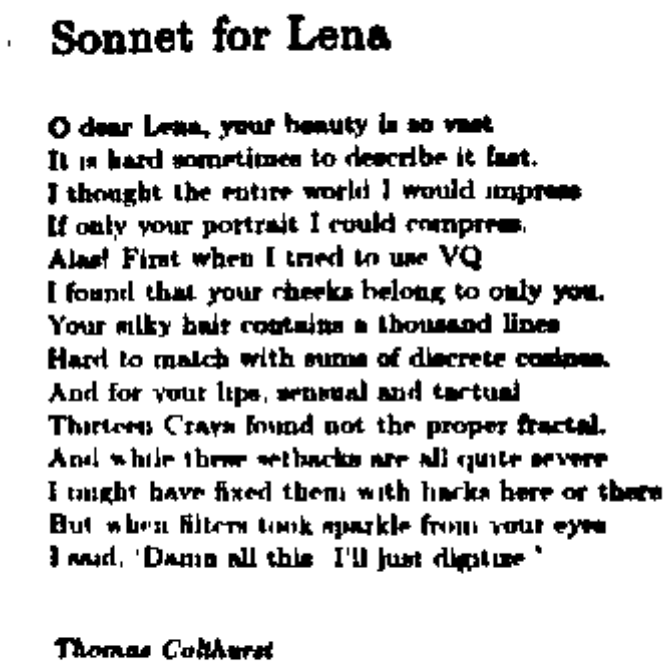
Description

I calculated average matrix by using convolution with 53x41 matrix. Then, applied a threshold value, and got the binary image. Later on, applied erosion and dilation respectively, with both 1x3 and 3x1 matrices.

Discussion

I worked on small matrices since the letters are close to each other. Therefore, I used 1x3 and 3x1 matrices instead of a 3x3 matrix filled with 1s. Since using the second matrix after the result of first one could be worse, I used these matrices seperately and ORed them to get a better result. When I change the threshold value, the letters get disappeared or more noises appear.

Image



Sonnet for Lena

O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

Thomas Colthurst

Figure 1. Q2_output.png

Q2.m

```
input_image = imread( 'data\\sonnet.png' );
[ n, m, numBands ] = size( input_image );

sum = zeros( n , m );
cnt = zeros( n , m );
```

```
h = ones(53,41);
h = h./2173;
avg = conv2( input_image , h , 'same' );

temp_image = ( input_image > avg - 7.5 );
figure; imshow( temp_image );
temp_image = ~temp_image;

struct_el_1_3_1 = ones(1,3);
struct_el_3_1_1 = ones(3,1);

output_image_1 = temp_image;
output_image_1 = erosion( output_image_1 , struct_el_1_3_1 );
output_image_1 = dilation( output_image_1 , struct_el_1_3_1 );

output_image_2 = temp_image;
output_image_2 = erosion( output_image_2 , struct_el_3_1_1 );
output_image_2 = dilation( output_image_2 , struct_el_3_1_1 );

output_image = ( output_image_1 | output_image_2 );

output_image = ~output_image;
figure; imshow( output_image );
imwrite( output_image , 'Q2_output.png' );
```

Question 3

Description

I constructed binary image by comparing blue band with an exact value, red band and green band. Then, applied erosion and dilation on the binary matrix with 3 structuring elements and ORed the resulting matrices. Later on, instead of using the functions provided by MATLAB to find connected components, implemented DSU algorithm with 2 optimizations that make the time complexity better. One optimization is that I link root of the tree with less height to the other one, with higher height. The other optimization is that during the finding the root of a given vertex, assign the root as parent of each vertex of the chain from the given node to the root. There was a long path which was assumed as a component. Tested if DSU works correct by implementing dfs and counting the number of components. Removed that component from my final image by applying a threshold value on the number of pixels for each component. Finally, assigned 20 different colors to the connected components.

Discussion

Because of the shapes of airplane wings, I thought that applying erosions dilations by using small diagonal matrices could be reasonable. Because of the shadows, different tones of colos, and sizes, using single threshold on the whole image may not be sufficient. There are some gaps between the components. If I apply dilations further, noises will become larger and small planes at the left of the image will become closer, which are not desired. I removed a long path from the image. Similarly, some noises can be removed by checking the number of pixels. However, still there will be some remaining noises like the short path colored to red at the top left corner.

Image

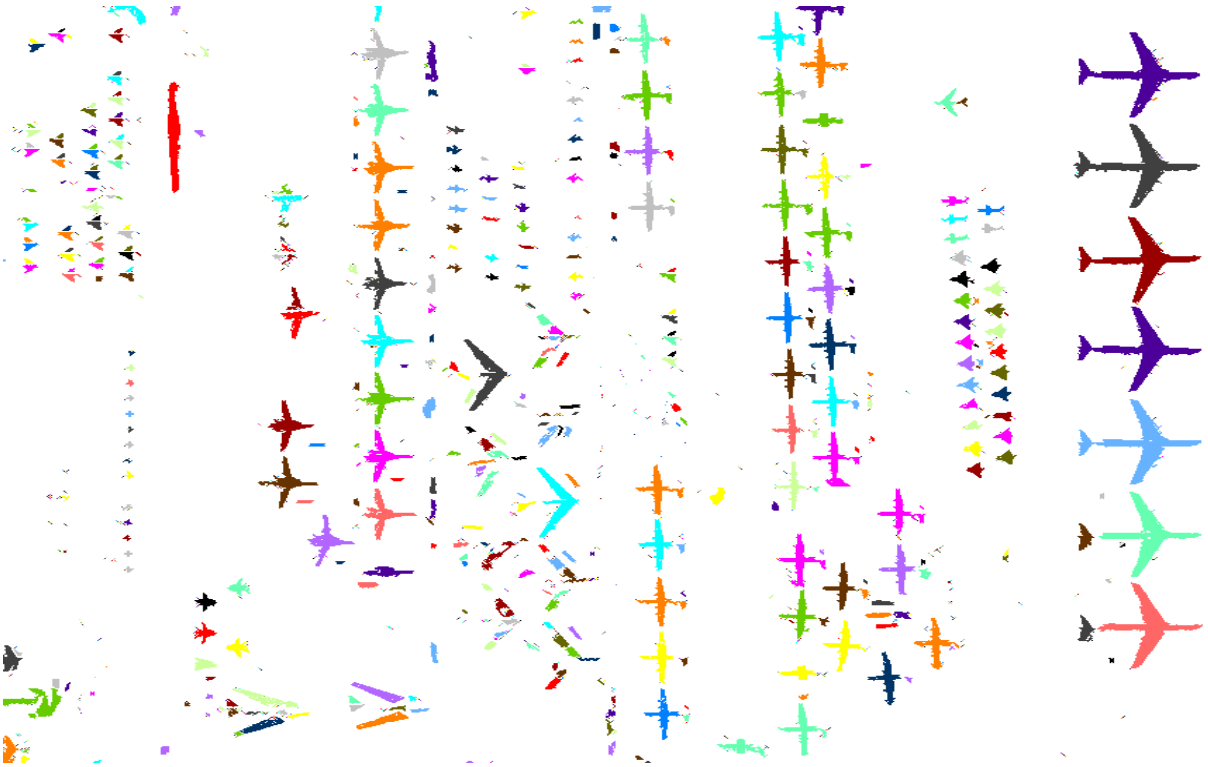


Figure 2. Q3_output.png

Q3.m

```
input_image = imread( 'data\\airplane_graveyard.jpg' );
[ n, m, numBands ] = size( input_image );

red = input_image( :, :, 1 );
green = input_image( :, :, 2 );
blue = input_image( :, :, 3 );

binary_image = ( blue >= 170 & blue >= green * 0.85 & blue >= red * 0.95 );
figure; imshow( binary_image );

struct_el_3_3_1 = [ 1 0 0 ; 0 1 0 ; 0 0 1 ];
binary_image_1 = binary_image;
binary_image_1 = erosion( binary_image_1 , struct_el_3_3_1 );
binary_image_1 = dilation( binary_image_1 , struct_el_3_3_1 );

struct_el_3_3_2 = [ 0 0 1 ; 0 1 0 ; 1 0 0 ];
binary_image_2 = binary_image;
binary_image_2 = erosion( binary_image_2 , struct_el_3_3_2 );
binary_image_2 = dilation( binary_image_2 , struct_el_3_3_2 );

struct_el_1_3_1 = ones(1,3);
binary_image_3 = binary_image;
binary_image_3 = erosion( binary_image_3 , struct_el_3_3_2 );
binary_image_3 = dilation( binary_image_3 , struct_el_3_3_2 );

binary_image = (binary_image_1 | binary_image_2 | binary_image_3);
figure; imshow( binary_image );

mark = zeros( n , m );
ncc = 0;
link_cc = [];

for r = 1 : n
    for c = 1 : m
        if binary_image(r,c) == 1
            valueTop = -1 ;
            valueLeft = -1;
            if( r > 1 && binary_image(r-1,c) == 1 )
                valueTop = mark(r-1,c);
            end
            if( c > 1 && binary_image(r,c-1) == 1 )
                valueLeft = mark(r,c-1);
            end
            if valueTop == -1 && valueLeft == -1
                ncc = ncc + 1;
                mark(r,c) = ncc;
            end
            if valueTop ~= -1 && valueLeft == -1
                mark(r,c) = valueTop;
            end
            if valueTop == -1 && valueLeft ~= -1
                mark(r,c) = valueLeft;
            end
            if valueTop ~= -1 && valueLeft ~= -1
                mark(r,c) = valueTop;
                if valueTop ~= valueLeft
                    ccs = [valueTop , valueLeft];
                    link_cc = [link_cc ; ccs];
```

```
                end
            end
        end
    end
end

par = dsu( link_cc , ncc );

colorsRGB = [ 153 0 0 ; 255 0 0 ; 255 102 102 ; 102 51 0 ; 255 128 0 ; 102
102 0 ; 255 255 0 ; 102 204 0 ; 204 255 153 ; 102 255 178 ; 0 255 255 ; 0
51 102 ; 0 128 255 ; 102 178 255 ; 76 0 153 ; 178 102 255 ; 255 0 255 ; 64
64 64 ; 192 192 192 ; 0 0 0 ];
[ colors_no, colors_bands ] = size( colorsRGB );

tree_size = zeros( ncc , 1 );
group_id = zeros( n , m );
root_id = zeros( n , m );

for i = 1 : n
    for j = 1 : m
        if binary_image(i,j) == 1
            group_id(i,j) = mark(i,j);
            root_id(i,j) = par( group_id(i,j) , 1 );
            tree_size( root_id(i,j) , 1 ) = tree_size( root_id(i,j) , 1 ) +
1;
        end
    end
end

imageRGB_red = zeros( n , m );
imageRGB_green = zeros( n , m );
imageRGB_blue = zeros( n , m );

for i = 1 : n
    for j = 1 : m
        imageRGB_red(i,j) = 255;
        imageRGB_green(i,j) = 255;
        imageRGB_blue(i,j) = 255;
        if binary_image(i,j) == 1 && tree_size( root_id(i,j) , 1 ) <= 2000
            color_id = mod( root_id(i,j) - 1 , colors_no ) + 1;
            imageRGB_red(i,j) = colorsRGB( color_id , 1 );
            imageRGB_green(i,j) = colorsRGB( color_id , 2 );
            imageRGB_blue(i,j) = colorsRGB( color_id , 3 );
        end
    end
end

imageRGB = zeros( n , m , 3 );
imageRGB( :, :, 1 ) = imageRGB_red;
imageRGB( :, :, 2 ) = imageRGB_green;
imageRGB( :, :, 3 ) = imageRGB_blue;
figure; imshow( uint8( imageRGB ) );
imwrite( uint8( imageRGB ), 'Q3_output.png' );

function par = dsu( link_cc , ncc )

    par = [];
    for i = 1 : ncc
        par = [par ; i];
    end
```

```
    depth = zeros( ncc , 1 );
    [ link_sz , link_width ] = size( link_cc );

    for i = 1 : link_sz
        a = link_cc( i , 1 );
        b = link_cc( i , 2 );
        [a , par] = findPar( a , par );
        [b , par] = findPar( b , par );
        if a == b
            continue
        end
        if depth( a , 1 ) == depth( b , 1 )
            par( a , 1 ) = b;
            depth( b , 1 ) = depth( a , 1 ) + 1;
        elseif depth( a , 1 ) < depth( b , 1 )
            par( a , 1 ) = b;
        else
            par( b , 1 ) = a;
        end
    end

    for i = 1 : ncc
        a = i;
        [a , par] = findPar( a , par );
    end

end

function [par_id , par] = findPar( node , par )
    par_id = 0;
    if par(node,1) == node
        par_id = node;
    else
        [par_id , par] = findPar( par(node,1) , par );
        par(node,1) = par_id;
    end
end
```

Question 4

Description

I constructed greyscale image of each picture by taking average of 3 bands. Then, applied background substraction from each one. After using some threshold values, constructed binary images. Then, applied erosion and dilation with small structuring elements a few times. Finally, used the functions provided by MATLAB for connected components labeling, instead of implementing DSU algorithm.

Discussion

Commonalities: One issue is related to threshold values. If a small value is used, then parts which belongs to the same component get closer but existing noises become larger and new ones emerge. Otherwise, parts which belongs to the same component get farther. Another issue is that whenever the shadows become darker, I cannot separate them.

Q4_output_copyMachine_1.png: I used disks with size of 5 units as stel since faced with medium-sized noises. However, head of the man at the left of the image was not connected to his body. Therefore, applied further dilations. Copy machine is connected to body of the man since they are close in respect of distance and value of grey band. Also, showing the pants that the man is wearing is not easy because of the value of grey band of the image.

Q4_output_copyMachine_2.png: Similar problems occur but in this case, parts that belong to the same component are closer. Therefore, there is no need to apply additional dilations.

Q4_output_copyMachine_3.png: Part of the image that hair roots are visible has different values of grey band. Therefore, this part is filled with 0s in the binary image. Since this part is bigger than what I expected, I applied dilation with a small stel. Actually, more dilations or a bigger stel is needed to change 0s to 1s. However, this leads to a bigger component which is not desired. I could not separate the shadow from the component. I did not face with this problem in previous images because in this case, shadows are darker which make it different from the other ones.

Q4_output_station_1.png: Applied erosion and dilation respectively to remove some noises. Then, applied dilation and erosion respectively to connect a few pixel to the component.

Q4_output_station_2.png: Collar of the coat that man is wearing has different background now. It seperates parts of the same component. Also, the woman at the back of the mirror (or barrier, whatever) is visible partially. Because of her scarf and background, parts of the same component are disjoint. Also, a hand of a person that is not visible is detected with ease thanks to the background.

Q4_output_station_3.png: It is nearly impossible to separate 2 women since they are so close to each other. Also, it became harder to detect the man since his surrounding changed. The bag is colored to diferent colors since the parts of the bag have different backgrounds which make them disjoint. Also, the coat that man is wearing is colorful which makes merging components even harder.

Images

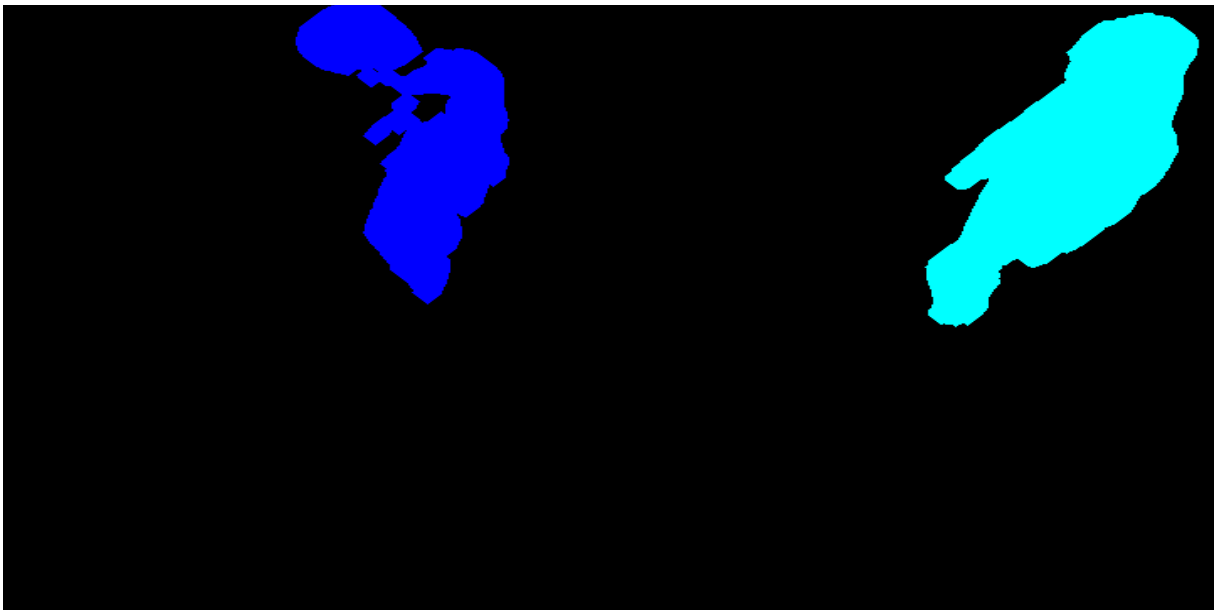


Figure 3. Q4_output_copyMachine_1.png

Figure 4. Q4_output_copyMachine_2.png

Figure 5. Q4_output_copyMachine_3.png

Figure 6. Q4_output_station_1.png

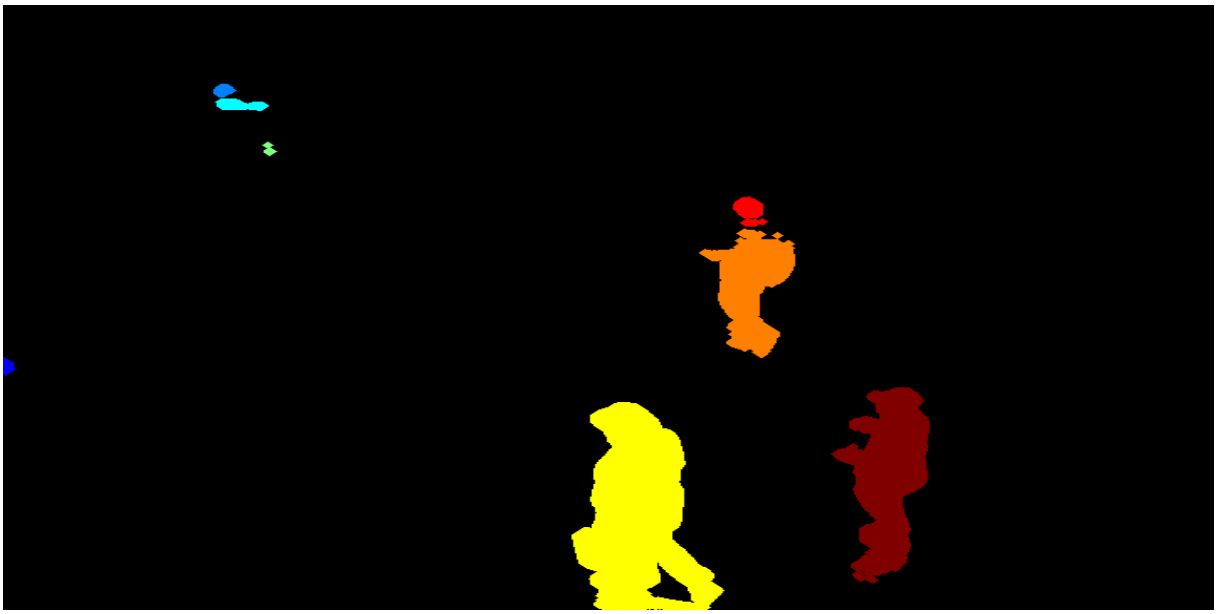


Figure 7. Q4_output_station_2.png

Figure 8. Q4_output_station_3.png

Q4.m

```
mystel_1 = [ 0,1,0 ; 1,1,1 ; 0,1,0 ];
mystel_2 = [ 0,0,1,0,0 ; 0,1,1,1,0 ; 1,1,1,1,1 ; 0,1,1,1,0 ; 0,0,1,0,0 ];
mystel_3 = [ 0,0,0,1,0,0,0 ; 0,0,1,1,1,0,0 ; 0,1,1,1,1,1,0 ; 1,1,1,1,1,1,1
; 0,1,1,1,1,1,0 ; 0,0,1,1,1,0,0 ; 0,0,0,1,0,0,0 ];
mystel_4 = ones(3,3);
mystel_5 = ones(5,5);

[ n, m, numBands ] = size( imread( 'data\\copyMachine\\0.png' ) );
grey = zeros( n , m , 4 );
tval = [ 0 ; 48 ; 42 ; 50 ];

for i = 1 : 4
    inputName = [ 'data\\copyMachine\\' , int2str(i-1) , '.png' ];
    input_image = imread( inputName );
    red = input_image(:,:,1);
```

```
    green = input_image(:,:,2);
    blue = input_image(:,:,3);
    grey(:,:,i) = ( red / 3.0 + green / 3.0 + blue / 3.0 );
    if i > 1
        diff = grey(:,:,1) - grey(:,:,i);
        binary_image = ( diff > tval(i,1) );
        if i == 2
            binary_image = erosion( binary_image , mystel_3 );
            binary_image = dilation( binary_image , mystel_3 );
            binary_image = dilation( binary_image , mystel_2 );
            binary_image = dilation( binary_image , mystel_2 );
        elseif i == 3
            binary_image = erosion( binary_image , mystel_2 );
            binary_image = dilation( binary_image , mystel_2 );
            binary_image = dilation( binary_image , mystel_2 );
            binary_image = erosion( binary_image , mystel_2 );
        elseif i == 4
            binary_image = erosion( binary_image , mystel_3 );
            binary_image = dilation( binary_image , mystel_3 );
            binary_image = dilation( binary_image , mystel_1 );
        end
        c = bwlabel( binary_image , 4 );
        col = max( c(:) );
        c2 = label2rgb( c , jet( col ) , [ 0 0 0 ] );
        figure; imshow( c2 );
        outputName = [ 'Q4_output_copyMachine_' , int2str(i-1) , '.png' ];
        imwrite( c2 , outputName );
    end
end

[ n, m, numBands ] = size( imread( 'data\\station\\0.png' ) );
grey = zeros( n , m , 4 );
tval = [ 0 ; 53 ; 45 ; 35 ];

for i = 1 : 4
    inputName = [ 'data\\station\\' , int2str(i-1) , '.png' ];
    input_image = imread( inputName );
    red = input_image(:,:,1);
    green = input_image(:,:,2);
    blue = input_image(:,:,3);
    grey(:,:,i) = ( red / 3.0 + green / 3.0 + blue / 3.0 );
    if i > 1
        diff = grey(:,:,1) - grey(:,:,i);
        binary_image = ( diff > tval(i,1) );
        if i == 2
            binary_image = erosion( binary_image , mystel_3 );
            binary_image = dilation( binary_image , mystel_3 );
            binary_image = dilation( binary_image , mystel_1 );
            binary_image = erosion( binary_image , mystel_1 );
        elseif i == 3
            binary_image = erosion( binary_image , mystel_1 );
            binary_image = dilation( binary_image , mystel_1 );
            binary_image = dilation( binary_image , mystel_2 );
        elseif i == 4
            binary_image = erosion( binary_image , mystel_1 );
            binary_image = dilation( binary_image , mystel_1 );
            binary_image = dilation( binary_image , mystel_2 );
        end
        c = bwlabel( binary_image , 4 );
        col = max( c(:) );
        c2 = label2rgb( c , jet( col ) , [ 0 0 0 ] );
```

```
        figure; imshow( c2 );
        outputName = [ 'Q4_output_station_' , int2str(i-1) , '.png' ];
        imwrite( c2 , outputName );
    end
end
```