Esin ERSOĞAN
708191001

BLG 513E Image Processing
Homework 2 Report

01.12.2019

1) Implementation

In this homework, a bird counting algorithm is implemented with the below steps:

1.1. Gaussian Filter:

Gaussian filter is applied to blur the image by removing the noise. 5*5 size Gaussian filter is applied with 1.4 sigma value. The blurred image after the filter is in the below, right one.

$I = I \times GaussFilter_{5x5}$ (x: Convolution operator)

1.2. Thresholding to Produce a Binary Image:

To analyze the image, I converted it to a binary image using thresholding.
In the analysis of a binary image, background should have zero values and foreground objects should have one values.

```
for i=1:size(grad, 1)
   for j=1:size(grad, 2)
       if grad(i,j) > 75
           grad(i,j)=0; %To convert the background to zeros
       else
           grad(i,j)=255; %To convert the foreground to ones
       end
   end
end
```

1.3. Convert to Binary Image:

The image which is applied a threshold is converted to binary image using a built-in function in Matlab just as the below:

```
M=im2bw(grad);
```

1.4. Connected Component Labelling:

8-connected component algorithm is applied to the binary image so as to count the number of birds in the image.

In the connected component algorithm, I search the image pixels row by row.
I created a connected component matrix named as C.

```
C = zeros(size(M,1),size(M,2));
```

For each pixel in image matrix, M, I searched the four pixel in the connected component matrix: left one, left-upper one, upper one and right-upper one.
In each four pixel search, I also checked the conflicts between these neighbors and I added bigger ones to the equivalence list:

```
eqList = containers.Map('KeyType','double','ValueType','double');
```

For the left one:

```
if C(i,j-1)~=0
    C(i,j)=C(i,j-1);
end
```

For the left-upper one:

```
if C(i-1,j-1)~=0
    C(i,j)=C(i-1,j-1);
    if C(i,j-1)~=0
        if C(i,j-1)~=C(i-1,j-1)
            C(i,j)=min(C(i,j-1), C(i-1,j-1));
            eqList(max(C(i,j-1), C(i-1,j-1)))=min(C(i,j-1), C(i-1,j-1));
            % max olanı min olarak işaretle
        end
    end
end
```

For the upper one:

```
if C(i-1,j)~=0
    C(i,j)=C(i-1,j);
    if C(i,j-1)~=0
        if C(i,j-1)~=C(i-1,j)
            C(i,j)=min(C(i,j-1), C(i-1,j));
            eqList(max(C(i,j-1), C(i-1,j)))=min(C(i,j-1), C(i-1,j));
            % max olanı min olarak işaretle
        end
    end
    if C(i-1,j-1)~=0
        if C(i-1,j-1)~=C(i-1,j)
            C(i,j)=min(C(i-1,j-1), C(i-1,j));
            eqList(max(C(i-1,j-1), C(i-1,j)))=min(C(i-1,j-1), C(i-1,j));
            % max olanı min olarak işaretle
        end
    end
end
```

For the right-upper one:

```
if C(i-1,j+1)~=0
    C(i,j)=C(i-1,j+1);
     if C(i,j-1)~=0
        if C(i,j-1)~=C(i-1,j+1)
            C(i,j)=min(C(i,j-1), C(i-1,j+1));
            eqList(max(C(i,j-1), C(i-1,j+1)))=min(C(i,j-1), C(i-1,j+1));
            % max olanı min olarak işaretle
        end
    end
    if C(i-1,j-1)~=0
        if C(i-1,j-1)~=C(i-1,j+1)
            C(i,j)=min(C(i-1,j-1), C(i-1,j+1));
            eqList(max(C(i-1,j-1), C(i-1,j+1)))=min(C(i-1,j-1), C(i-1,j+1));
            % max olanı min olarak işaretle
        end
    end
    if C(i-1,j)~=0
        if C(i-1,j)~=C(i-1,j+1)
            C(i,j)=min(C(i-1,j), C(i-1,j+1));
            eqList(max(C(i-1,j), C(i-1,j+1)))=min(C(i-1,j), C(i-1,j+1));
            % max olanı min olarak işaretle
        end
    end
end
```

For the non-zero pixel values who does not have any neighbors, I added them as new regions:

```
if C(i,j)==0
    k=k+1;
    eqList(k)=k;
    C(i,j)=k;
end
```

To resolve the conflicts:

To resolve the conflicts between neighbors, I travel the equivalence list beginning from the biggest value:

```
for t=0:eqList.Count-1
    for i=1:size(M,1)
        for j=1:size(M,2)
            if C(i,j)==eqList.Count-t
                C(i,j)=eqList(C(i,j));
            end
        end
    end
end
```

1.5. Convert to RGB Image the Labeled Image:

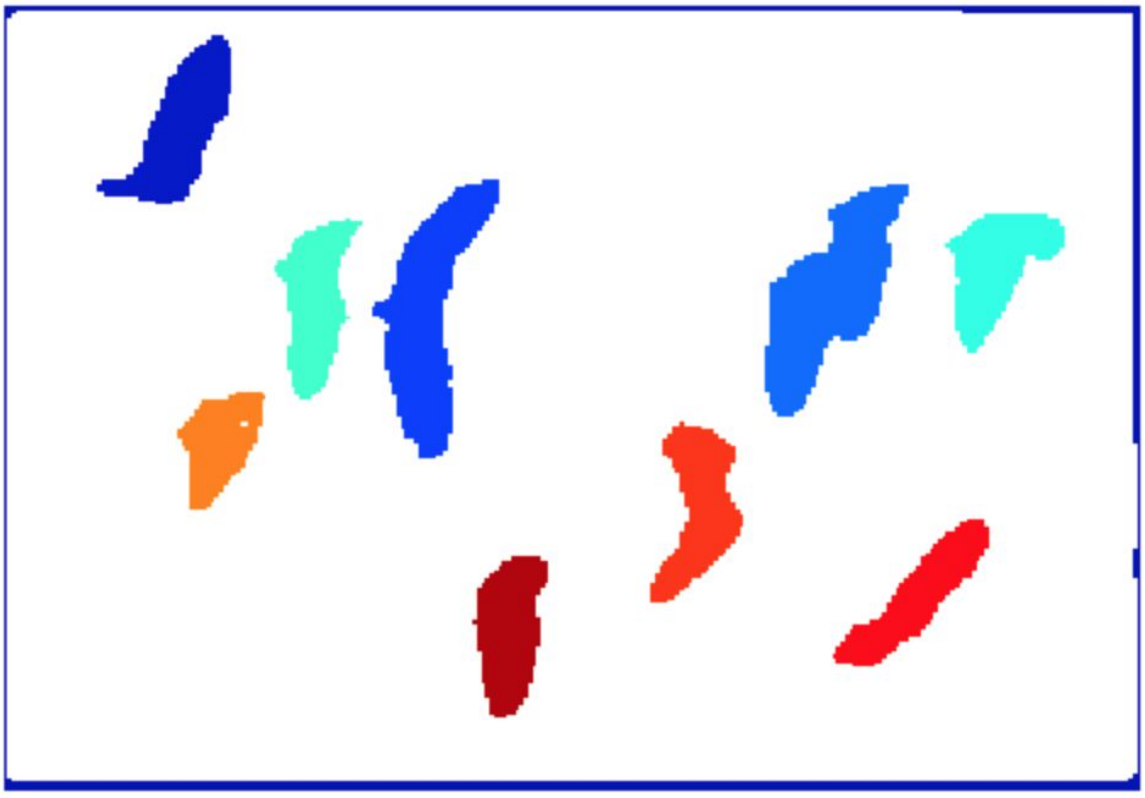
The labelled image is converted to RGB image according to its labels:

```
rgb = label2rgb(C);
```

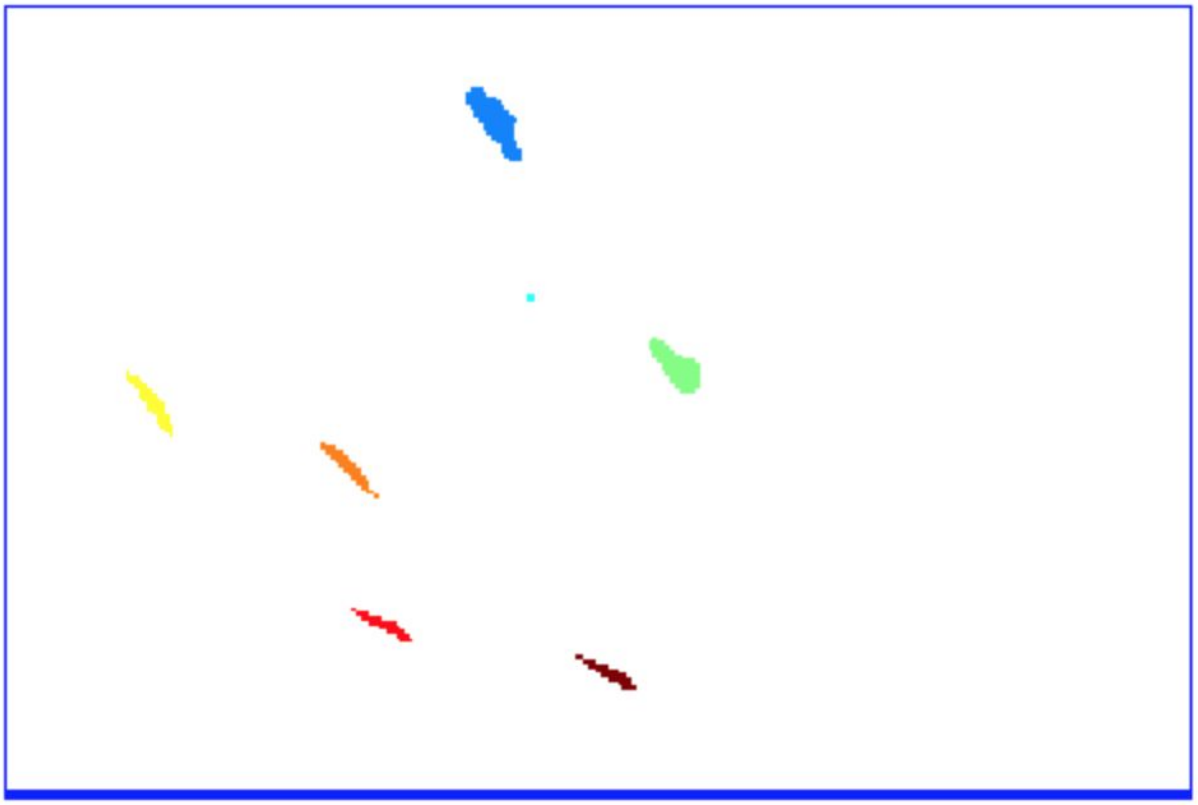
2) Results

The results obtained with the Bird Counting Algorithm are below:

'bird 1.jpg'



'bird 2.jpg'



'bird 3.bmp'