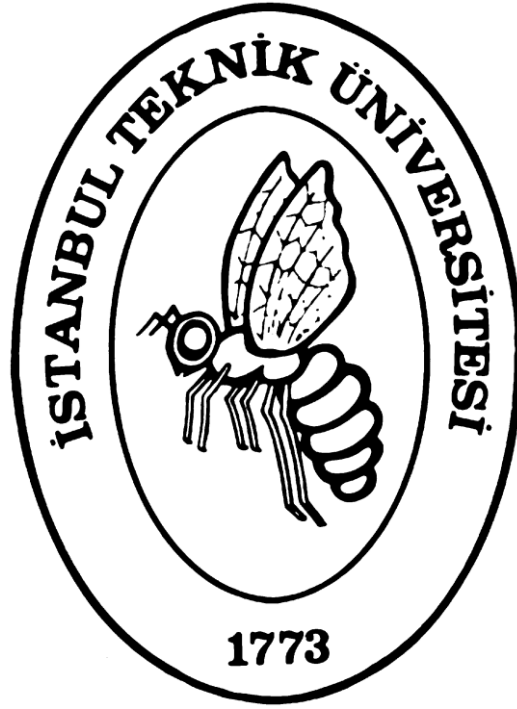


BLG477E



# Multimedia Computing

## Homework 03

**STUDENT NAME : HÜSEYİN ERDOĞAN**  
**STUDENT NO : 040100054**

**CRN : 22539**

## 1 Environment

I used Microsoft Visual Studio 2013 and OpenCV 2.4.11 for the homework. I used Microsoft Visual Studio in Debug Mode for Win32.

When program is running, first number of founded birds will be shown then use have to press a button to see go next bird image.

## 2 Counting Birds

First Otsu threshold values are used on bird images. Otsu threshold algorithm is taken from my homework 2 source code. Aim of using Otsu threshold algorithm on image is removing background from image. Thanks to Otsu threshold algorithm, we can concentrate on bird.

After using Otsu threshold algorithm on image, some part of birds can be separate from bird or lap over other birds. We can use Morphological filters to correct these problems.

I used Opening filter of Morphological filters because it is useful for removing small objects, it can be assumed that the objects are bright on a dark foreground. Left letter is before using Closing Morphological filter and right letter is after using Closing Morphological filter at image at below. It can be seen that small spaces in the corners of the letter tend to disappear.

```
dst = open(src, element) = dilate(erode(src, element))
```



In my project, I use Closing Morphological filter two times on bird images because of increasing of its effectiveness.

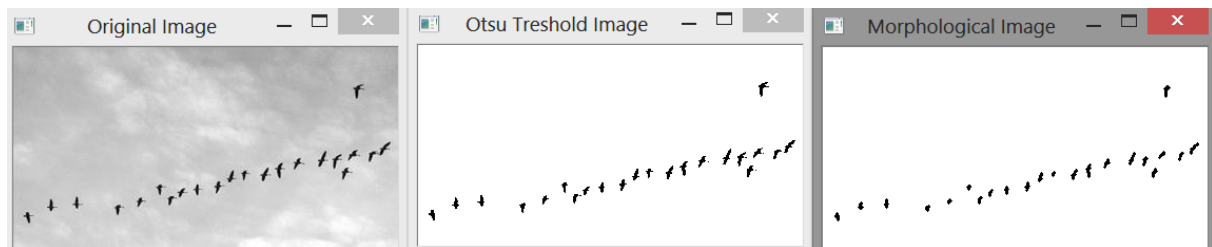
- “bird1.jpg” image after Closing Morphological filter with 7 sized cross filter.



- “bird2.jpg” image after Closing Morphological filter with 7 sized cross filter.



- “bird3.bmp” image after Closing Morphological filter with 3 sized cross filter.

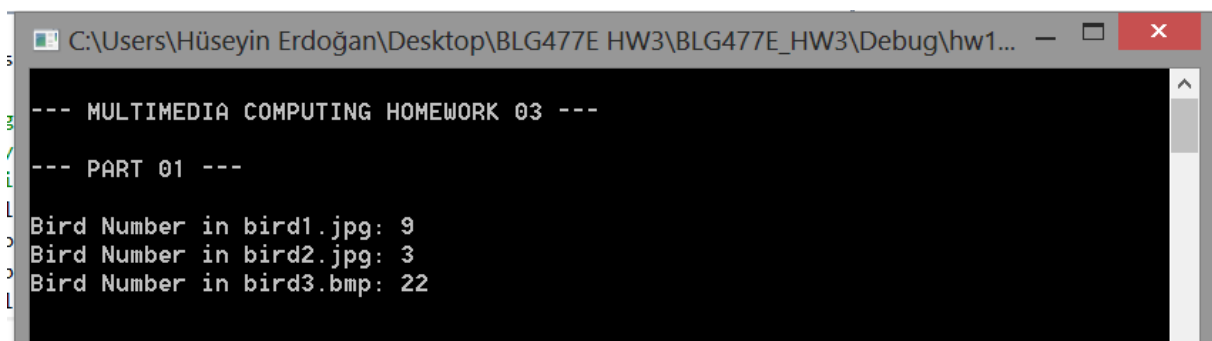


Lastly I used Connected Component Analysis for counting birds. I used DFS search algorithm for labelling connected regions recursively. Then I counted these labels. Code can be seen at below.

```
//Used for labelling neighbours of selected pixel
void DfsForConnectedComponent(int i, int j, int _currentLabel){
    if (i < 0 || i == morphologicalImage.rows || j < 0 || j == morphologicalImage.cols) return;
    if (labelsOfRegions[i][j] != 0 || morphologicalImage.at<uchar>(i, j) == 255) return;
    labelsOfRegions[i][j] = _currentLabel;
    for (int _direction = 0; _direction < 4; ++_direction){
        DfsForConnectedComponent(i + directionXNeighbours[_direction], j + directionYNeighbours[_direction], _currentLabel);
    }
}

//Used for finding region number from labeled image
int UseConnectedComponentFunctionOnImage(){
    for (int i = 0; i < morphologicalImage.rows; ++i){
        for (int j = 0; j < morphologicalImage.cols; ++j){
            if (labelsOfRegions[i][j] == 0 && morphologicalImage.at<uchar>(i, j) == 0){
                DfsForConnectedComponent(i, j, ++numberOfObjects);
            }
        }
    }
    return numberOfObjects;
}
```

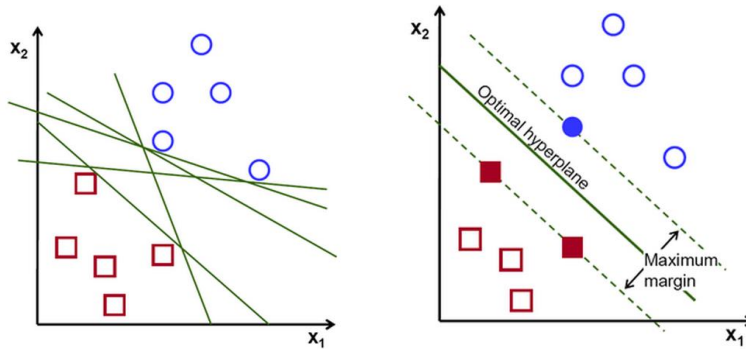
In the end program shows bird number for images.



Program finds correct bird number for “bird2.jpg” and “bird3.bmp” images. But in the “bird1.jpg”, there are two birds that are overlapping each other. Because of this, program finds one less bird number at “bird1.jpg” image.

### 3 Colour Classification

In this part, I decided to use Support Vector Machine method for machine learning. For these part, I used OpenCV's SVM library. Working principle of SVM can be seen at below.



I created a label array for every colour. In this array, labels of images are stored. SVM training algorithm was planned to run for every colour. But in these part, program gives me an error and I think these error is caused by taken arguments of SVM's train function. But I did not find a solution for these problem. However I wrote prediction function for this part. I cannot test it because problem that mentioned at above. I added these part's codes to homework.