Hacettepe University
Department of Computer Engineering
BBM418 Computer Vision Laboratory
Assignment 1

Atakan Ayyıldız
21516681

January 24, 2020

Abstract

In this assignment I am getting familiar with edge detection methods and Hough Transform. I can obviously say that Hough transform is not the best method to find a licence plate. Hough transform is more troublesome to use when Contour methods are available. In order to write briefly what has been done:

1 Read images from the given folder.(images)

img = cv2.imread(file_path, 0) is to obtain gray image. imgGry = cv2.medianBlur(img, 3) blur the unused corners with median

2 Preprocess an image before using it. Firstly

```
img0=cv2.imread(file_path)
img = cv2.imread(file_path, 0)
imgGry = cv2.medianBlur(img, 3)
edges = cv2.Canny(imgGry, 150, 200, apertureSize=3)
```

3 Use Canny edge detection method. It is a multi-stage algorithm.

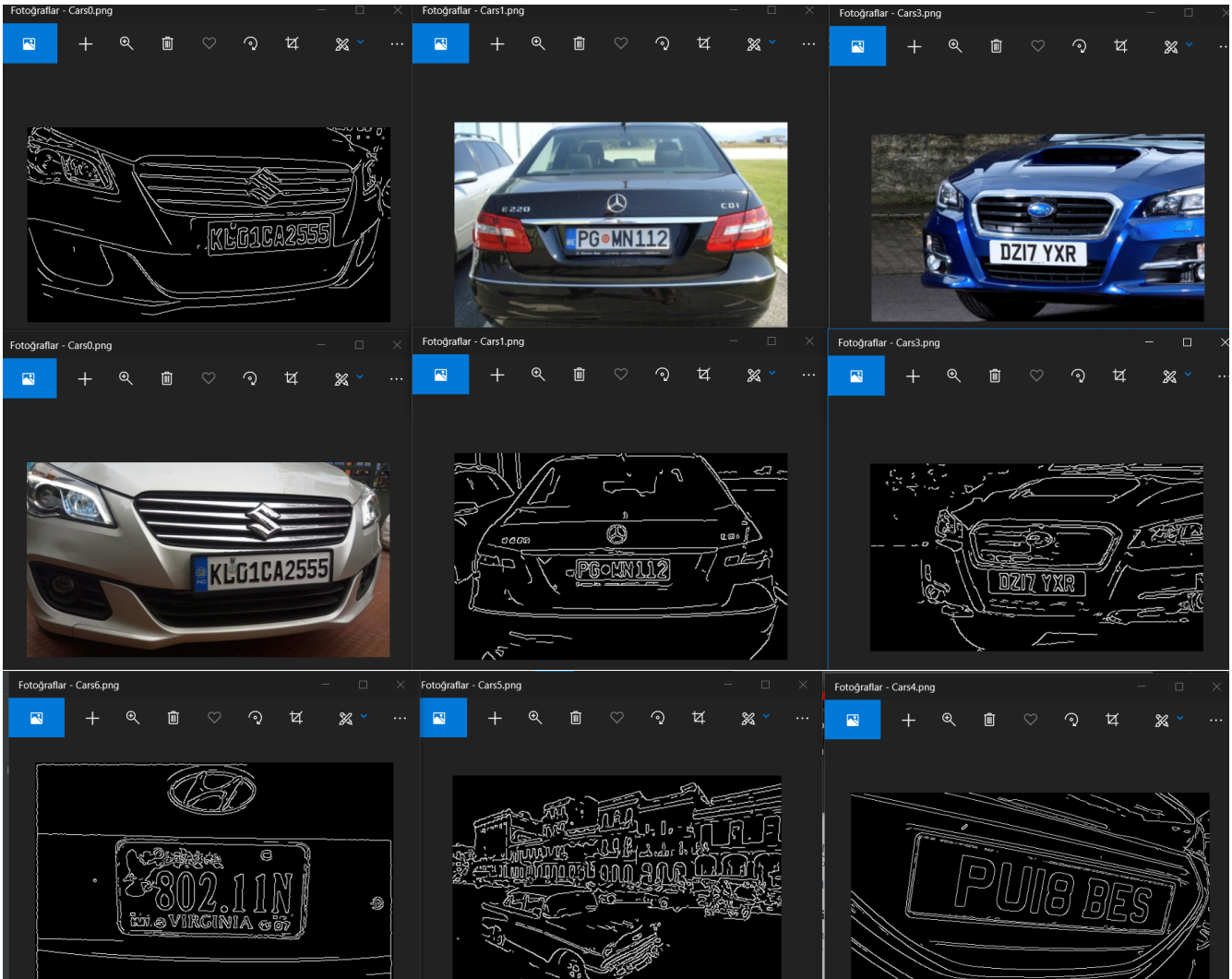
3.1 Noise reduction is used because edge detection results are highly sensitive to image noise. It is mainly based on derivatives.

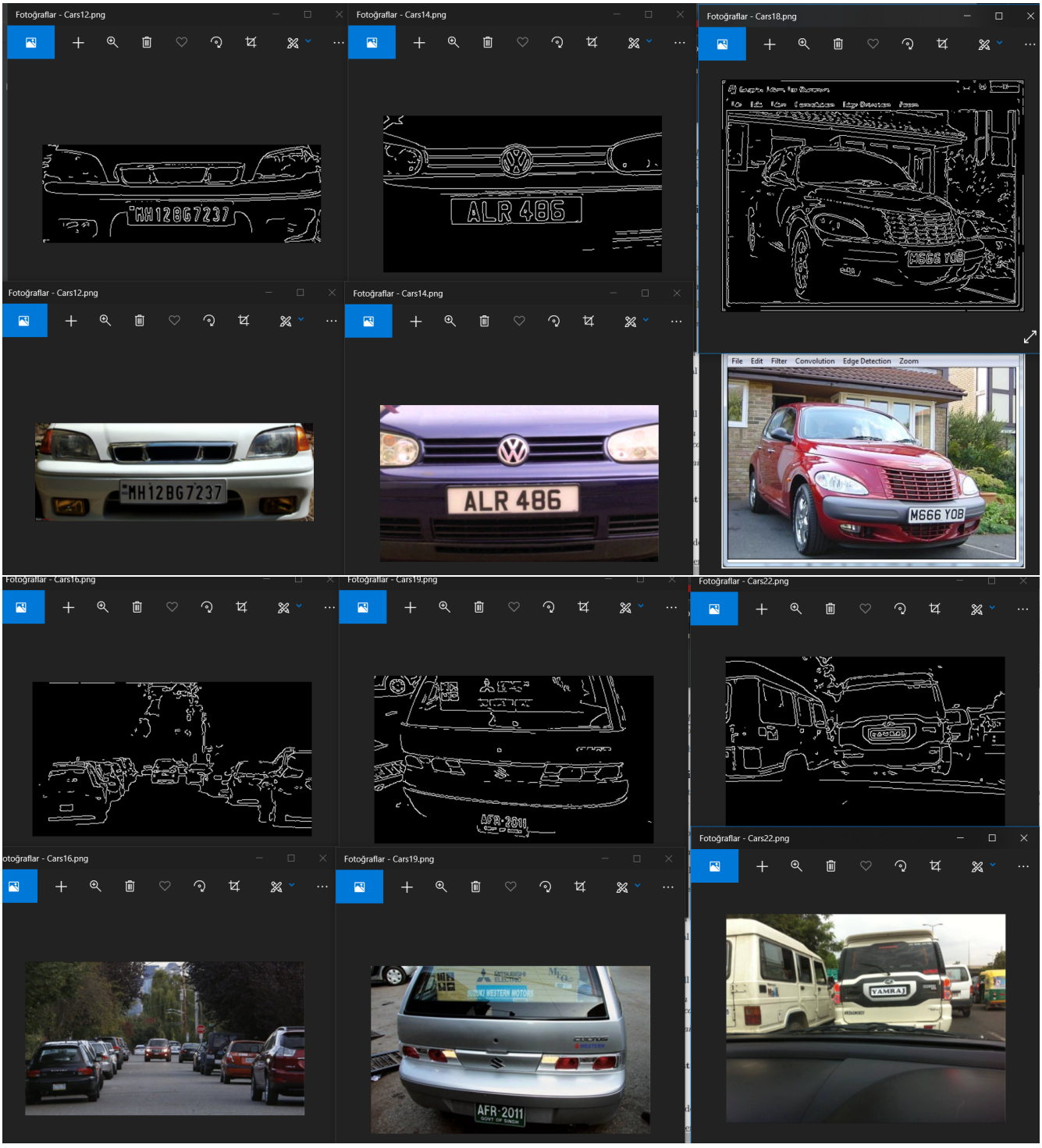
3.2 Gradient Calculation detects the edge intensity. Sobel filters are also used in Canny edge detection method. So we can say that Canny edge detection is more complicated than Sobel filters. Calculation of Gradient and edge direction is handled after the Sobel filters. At the end of this step we can see some edges are thin some others are much dense.

3.3 Non-Maximum Suppression: Visit every point in the gradient intensity matrix to find to max value edge directions. At the final moment return the non-max suppression algorithm with thin edges calculated by two main criteria (edge direction in radiant, and pixel intensity)

3.4 Double threshold: There are 3 pixel types. Strong ones include the pixels that they have high intensity. Weak ones are we can not say them strong to include absolutely but we can not also say we can not include. Other ones not included.

3.5 Edge Tracking by Hysteresis: If there is a weak pixel but it has a neighbor processed pixel that is strong pixel, then hysteresis transform that weak pixel to strong pixel. So Canny edge is done.





4 Licence plate detection

Firstly we should find the vertical lines with Hough line method. It takes two degrees to find the corner vertical lines(–5,0). Using the Canny edge array's strong edges it starts to find.

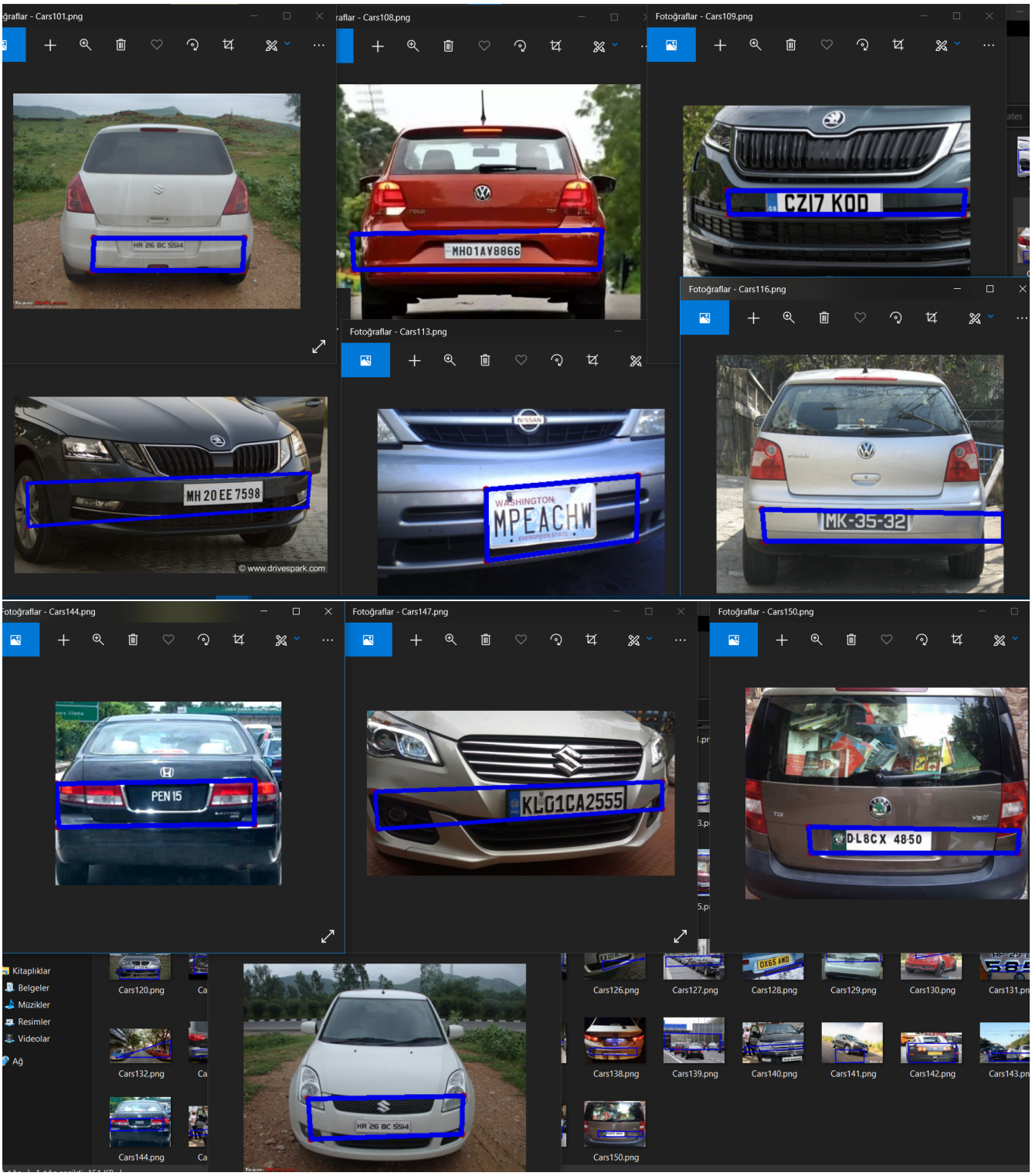
```python
def hough_line(img, degree, degree2):
    # Rho and Theta ranges
    thetas = np.deg2rad(np.arange(degree, degree2))
    height, width = img.shape
    diag_len = int(np.ceil(np.sqrt(width * width + height * height)))  # max_dist
    rhos = np.linspace(-diag_len, diag_len, int(diag_len * 2) + 1)#create rhos length of a diagonal

    #store them to not calculate everytime
    cos_t = np.cos(thetas)
    sin_t = np.sin(thetas)
    num_thetas = len(thetas)

    accumulator = np.zeros((int(diag_len * 2), num_thetas), dtype=np.uint64)
    y_idxs, x_idxs = np.nonzero(img)  # (row, col) indexes to edges

    # Voting is started here for each rho and theta pairs
    for i in range(len(x_idxs)):
        x = x_idxs[i]
        y = y_idxs[i]
        for t_idx in range(num_thetas):
            # Calculate rho. diag_len is added for a positive index
            rho = int(round(x * cos_t[t_idx] + y * sin_t[t_idx]) + int(diag_len))
            accumulator[rho, t_idx] += 1
    return accumulator, thetas, rhos
```

Rho=x.cos(theta)+ y.sin(theta) is the main criteria. Firstly create an 2D accumulator with “Rhos” are rows the and “thetas” are the columns. Size is dependent on what we expect from the Hough Lines method. In my function it is 0-180. So My function’s accumulator has 180 columns. For the maximum value of the Rho is the diagonal length of the images. For the every pair of the (Rho,theta) we increment Accumulator[rho,theta]++. (This is voting). At the end the method brings me 3 variable. Accumulator 2d numpy array that contains votes of each pair of rho-theta. Thetas and rhos arrays. In the returning accumulator vertical lines contain less votes. So that i choose the if 20 ¡ acc[y][x] ¡ 30: Then sorting function used with the smallest 10 votes. Some operation is done for the opposite side of the car image. First vertical accumulator is created for the left of the image other is created for the right part of the image. Calculate the cos and sin values according to their Rho and theta. Then to create the line 4 points are calculated. After calculating all the lines we need to chose the correct rectangle to define the plate. There is a list called vertical_1, vertical_2, horizontals to chose the correct lines for the plate. I chose the horizontal lines near the bottom of the picture. Usually the plates are in the lower sections. For the vertical ones are the hardest ones. I still haven’t found the absolute correct vertical lines. After the choosing appropriate lines I found the intersection of them then I created the rectangle of plates.



5 About Intersection over Union

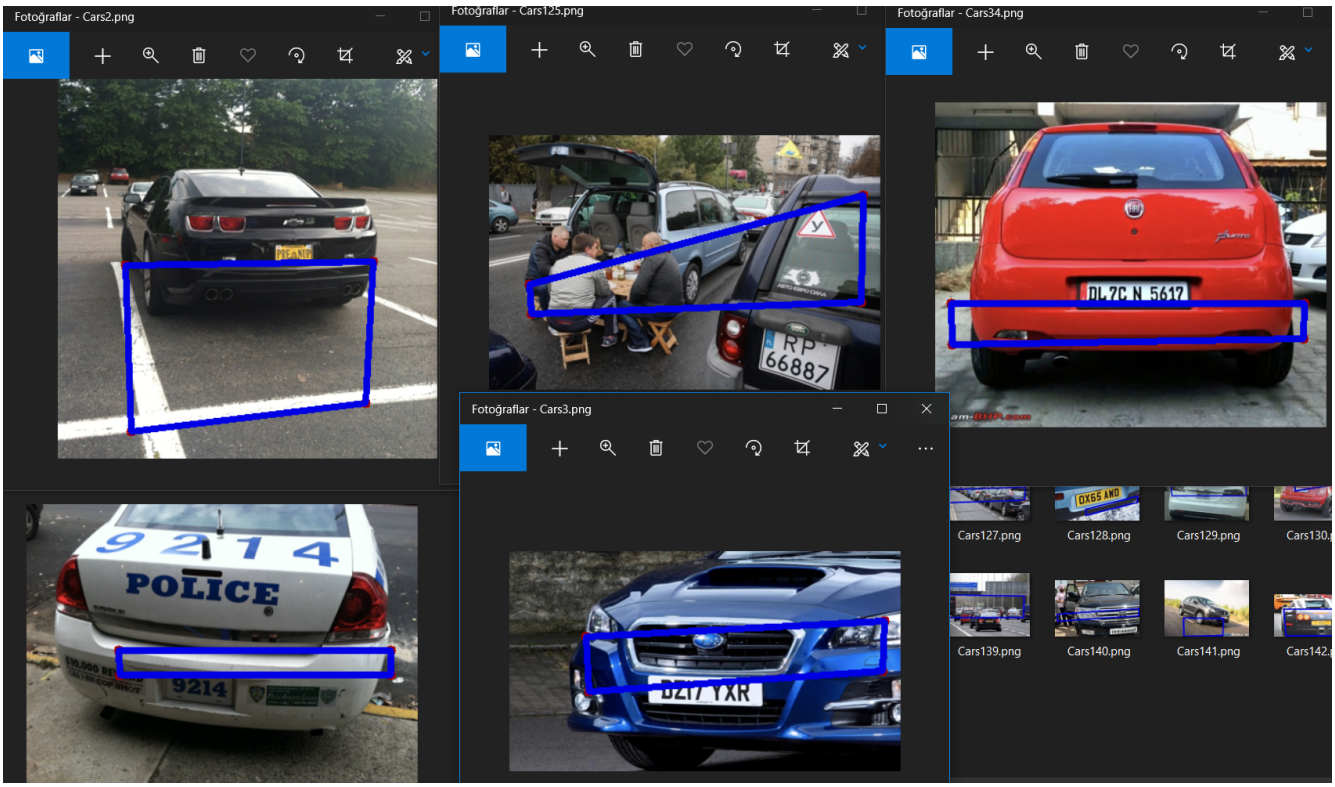
Intersection over Union is an evaluation metric used to measure the accuracy of an object detector on a particular data set. In order to apply IOU we need the ground truth boxes and predicted bounding boxes. First one comes as xml format and second one comes from our code. There are two boxes' coordinates for

the function. In the function we compute the overlap are of the two boxes,then divide it to the area of union. If the result is less than %40 (0.4) we can say poor. If the result is about %70 (0.7) the result is good. If the result is about %90 (0.9) it is excellent. In my code it is not good**. My average iou is: %9.9925** But this is not just about the written code. Input is the main problem. Also my edge detections are not good enough.

```
Cars0.png: 0.2610
Cars1.png: 0.2461
Cars10.png: 0.2916
Cars100.png: 0.0944
Cars101.png: 0.0000
Cars102.png: 0.1022
Cars103.png: 0.0170
Cars104.png: 0.0997
Cars105.png: 0.1349
Cars106.png: 0.0000
```

My output plate rectangles are bigger than the desired plate rectangles. So that is why my results are very poor. My output plates are equal to the Area of Union. That means my predicted rectangle already contains the desired plate.

6 Some images for the miss-calculated images. Fails.



The failure of the miss-calculation is simple. There are lots of combination for the plates. Some are rectangle, some are square. Their place is an another calculation option, color makes it harder to detect. Some may have considered as incorrect input . But the most important is choosing the wrong points. Some plates are not smooth enough. Or there are much sometimes strong lines can be considered as plate lines like car bumpers. Also we can considered detection method thresholds. Some plate lines can not be considered as strong lines.