BBM 418 - Computer Vision Laboratory
2022 Spring

Murat Çelik
21827263
Department of Computer Engineering
Hacettepe University
Ankara, Turkey
b21827263@cs.hacettepe.edu.tr

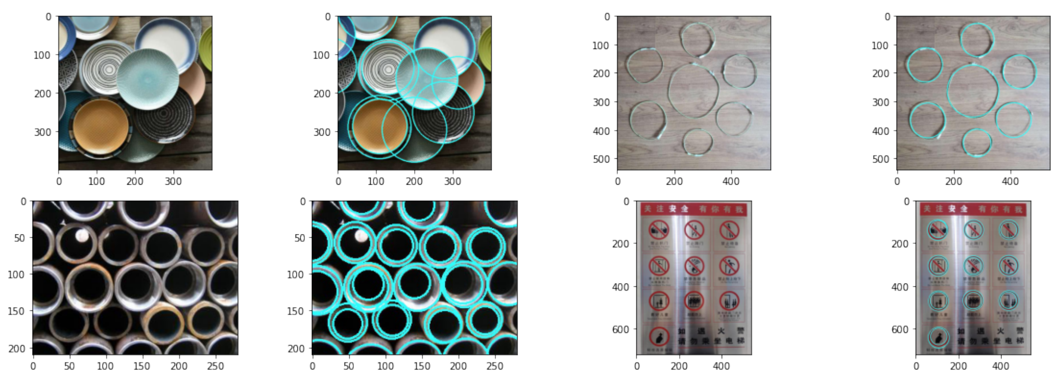
1 Introduction

In this assignment, work was done on edge detection methods and Hough Transform. Edge detection is an image processing technique that attempts to locate the boundaries of objects within images by detecting discontinuities in brightness. The Hough transform is a technique that can be used to detect features of a particular shape in an image. In this assignment, we defined this shape as a circle.

2 DataSet

Dataset consists of sample images including circular objects. For each image, there is also a txt file provided for ground-truth information showing where the circular objects are placed on the image. The information that determines the circles on the images is given in the "GT" folder in the dataset.

Figure 1: Below are examples of original images and images which showing circles in the original image.



This information is displayed on the original image with the help of opencv library.

3 Edge Map

An edge map is an image that indicated where edges are in the image. There are different techniques such as Sobel Edge detector, Laplacian Edge detector, Canny Edge detector.

3.1 Sobel Edge Detector

The Sobel filter uses two 3 x 3 cores. One of these analyze the horizontal direction and one the vertical direction in images.

Figure 2: Frequently used Sobel filter example.

-1	0	+1
-2	0	+2
-1	0	+1

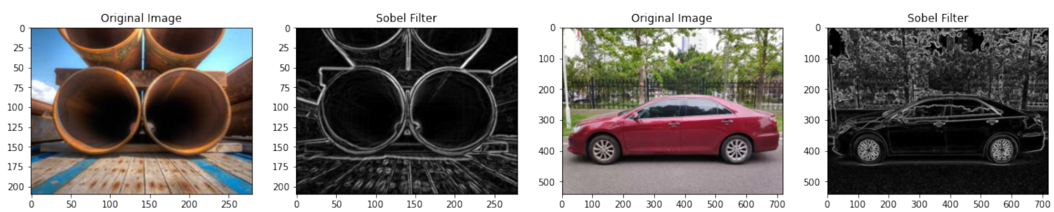
Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

These can be combined with $|G| = \sqrt{(G_x^2 + G_y^2)}$ or $|G| = |G_x| + |G_y|$ to find the absolute magnitude of the transition at each point and the direction of that transition .

Figure 3: Edge Map Examples with Sobel Edge Detector Method



3.2 Laplacian Edge Detector

It is one of the edge detection methods that highlights regions with rapid density change.Generally, the following kernels are used:

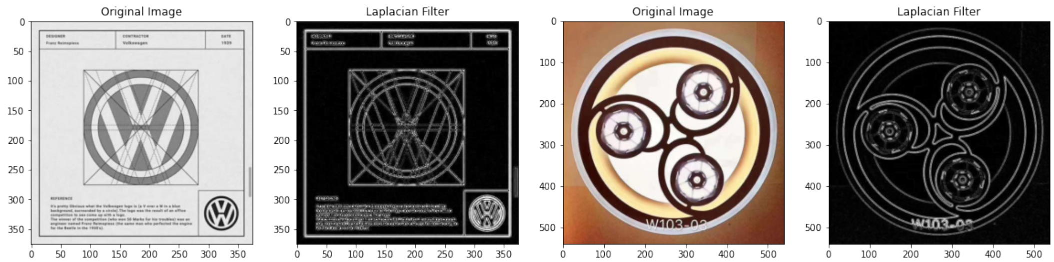
Figure 4: Frequently used Laplacian filter example.

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

The main formula is as follows: $Laplace(f) = \dfrac{\partial^2 f}{\partial x^2} + \dfrac{\partial^2 f}{\partial y^2}$

Figure 5: Edge Map Examples with Laplacian Edge Detector Method



3.3 Canny Edge Detector

The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images. Step :

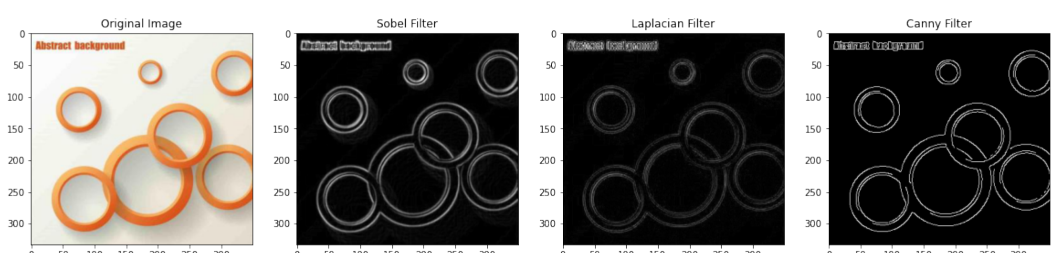
1. Noise Reduction using Gaussian filter.
2. Get help from the Sobel kernel. At this point our formulas is: $Edge\_Gradient\ (G) = \sqrt{G_x^2 + G_y^2}$
   $Angle\ (\theta) = \tan^{-1}\left(\frac{G_y}{G_x}\right)$.
3. Gradient direction is always perpendicular to edges, so it is rounded to vertical or horizontal or two diagonal directions which four angles representing
4. Non-maximum Suppression
5. Hysteresis Thresholding

Figure 6: Edge Map Examples with Canny Edge Detector Method

3.4 Edge Detectors Comparision

Canny Edge Detector was chosen as the choice with its thicker and cleaner lines.

Figure 7: Edge Detector Methods comparison examples



3.5 More examples of Canny Edge Detector

Figure 8: Edge Map Examples with Canny Edge Detector Method

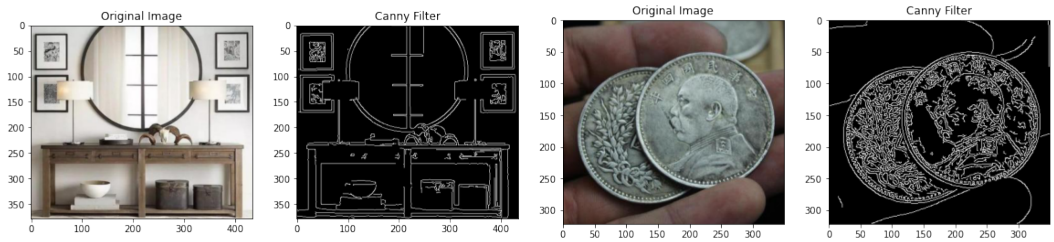


Figure 9: Edge Map Examples with Canny Edge Detector Method

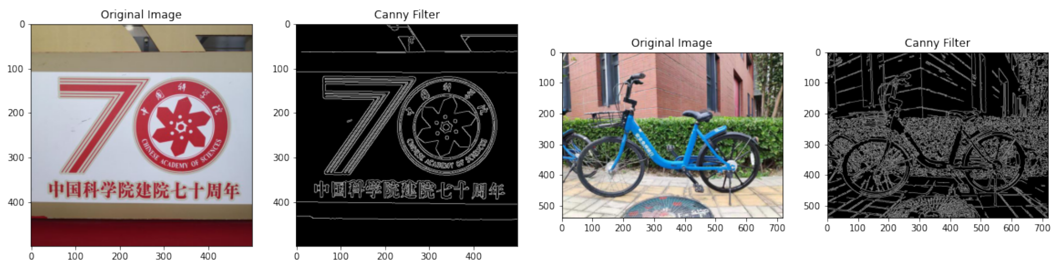


Figure 10: Edge Map Examples with Canny Edge Detector Method

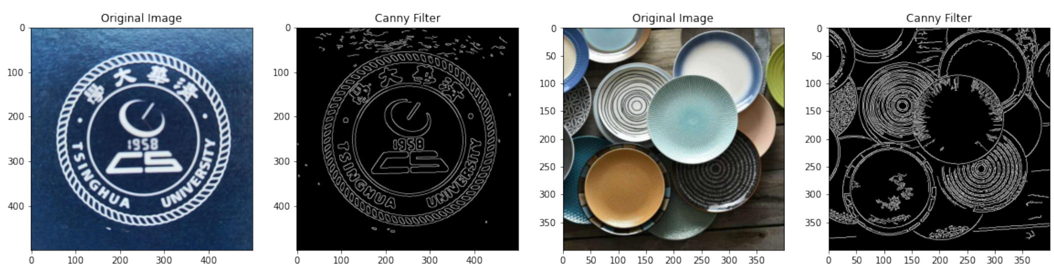
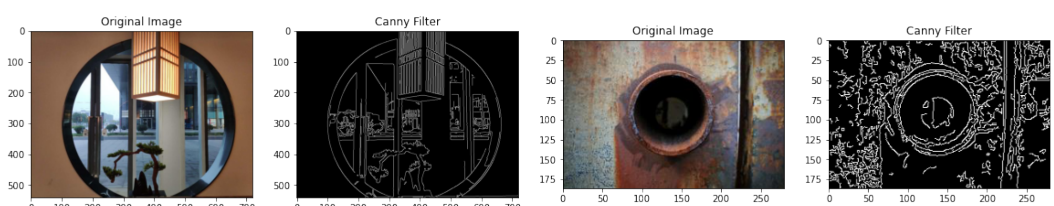


Figure 11: Edge Map Examples with Canny Edge Detector Method



4 Hough Transform

The Hough transform is a method designed to find and represent the properties of a particular shape in an image.

1. Blur and grayscale processing is applied to a image.
2. The resulting image is given to the canny edge detector to create an edge map.
3. The prepared edge map is scanned to find the circle. Searching for circles formed by clustering points that are at a certain distance from a certain center point.
4. Estimated circles detected are recorded.
5. If some detected circles are observed to be almost identical to each other, those predictions are eliminated.
6. We save the estimated values of the grunt truth values.

The hyperparameters are determined as follows:

- Radius (r_change) = 2 This value represents the radius change after each calculation.
- Theta number(num_of_thetas) = 120 This value represents how many angular values are looked at in each calculation.
- Threshold (bin_threshold) = 0.5 This value represents what percentage of the circle must be seen to produce a circle shape.

4.1 Intersection over Union (IoU)

The IOU (Intersection over Union) is the score that determines the overlap of two areas with each other. The higher the intersection, the higher this value.

In this task, the formulas specified in the article "CircleNet: Anchor-free Detection with Circle Representation" were used to calculate the IoU score for the circles.

$$IoU = \frac{Area(A \cap B)}{Area(A \cup B)}$$

In this assignment, IoU score calculation is done only between intersecting circles.

5 Experiment Result

The results obtained in a wide variety of data sets seem sufficient at the first stage. These results are shared in the table below. It was added to the test results with the image samples.

Table 1: The data of 10 sample images are shared below. These data are also retained for all values in the dataset.

Pictures Name	Intersection over Union Score	Num of Circle in Original	Num of Circle in Predicted	Num of intersecting Circle
5.txt	77.29	13	11	11
27.txt	74.50	20	10	20
46.txt	79.69	7	8	8
64.txt	93.51	2	1	2
68.txt	92.24	2	2	2
69.txt	62.27	16	13	26
81.txt	91.12	2	3	2
101.txt	90.01	1	1	1
102.txt	83.35	31	33	31
113.txt	90.01	1	1	1

5.1 Successful Results

Successful results are given below. 10 of these results are shared below with their data. These results show us that circles with sharp lines are detected more effectively by the algorithm.

Figure 12: Below is the image named "5.txt". IoU score is 77.29% .

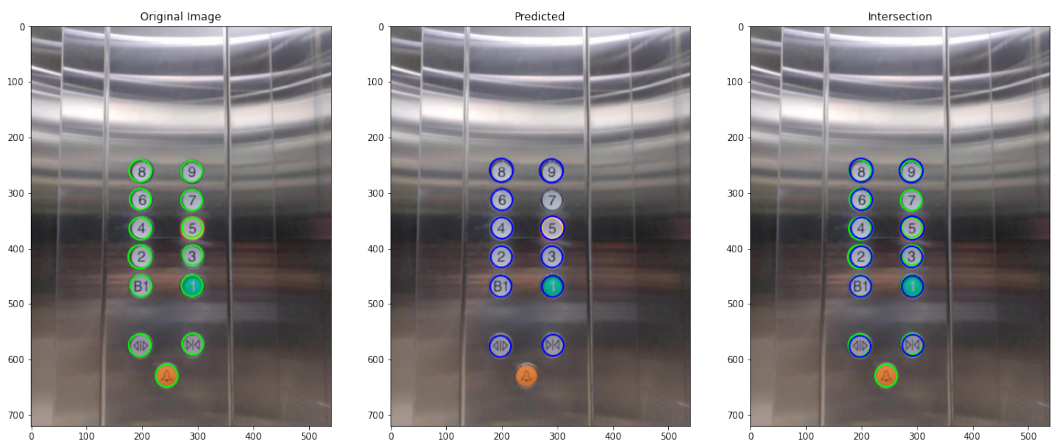


Figure 13: Below is the image named "27.txt". IoU score is 74.50% .

Figure 14: Below is the image named "46.txt". IoU score is 76.69% .

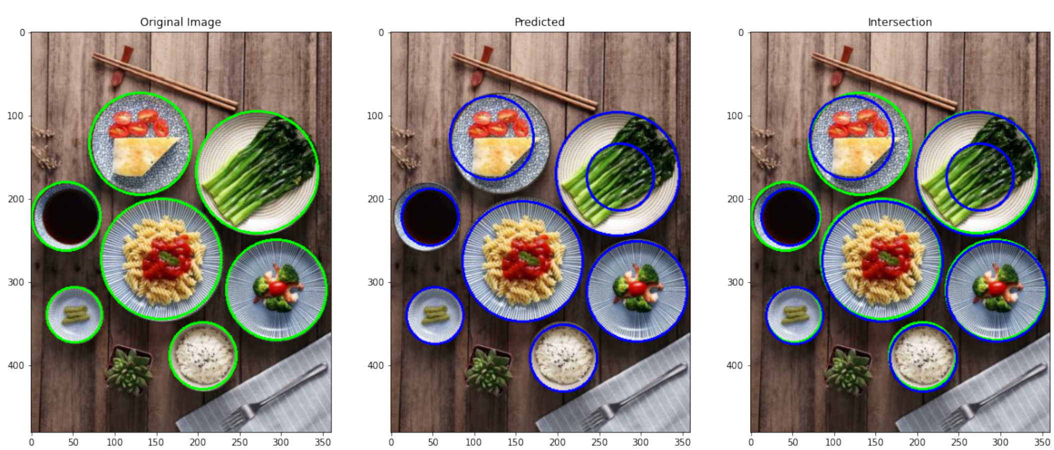


Figure 15: Below is the image named "64.txt". IoU score is 93.51% .

Figure 16: Below is the image named "68.txt". IoU score is 92.24% .

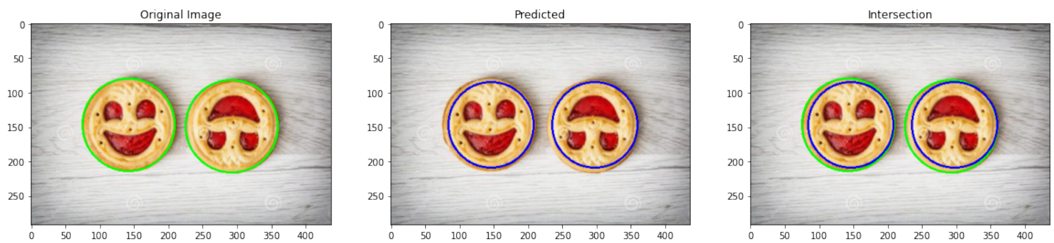


Figure 17: Below is the image named "69.txt". IoU score is 62.67% .

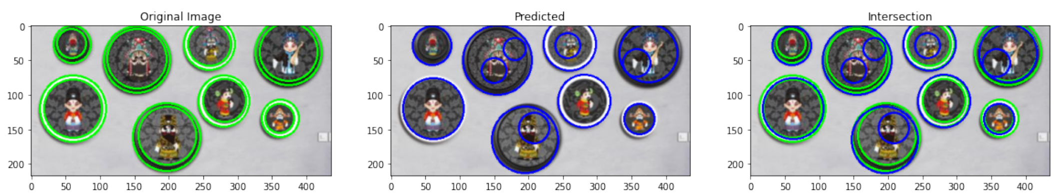
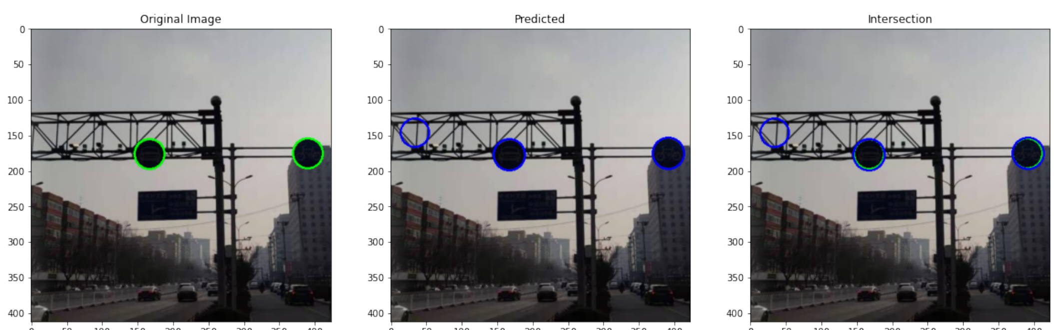


Figure 18: Below is the image named "81.txt". IoU score is 91.12% .



It can be said that if the algorithm is optimized, it will achieve better results.

Figure 19: Below is the image named "101.txt". IoU score is 90.01% .

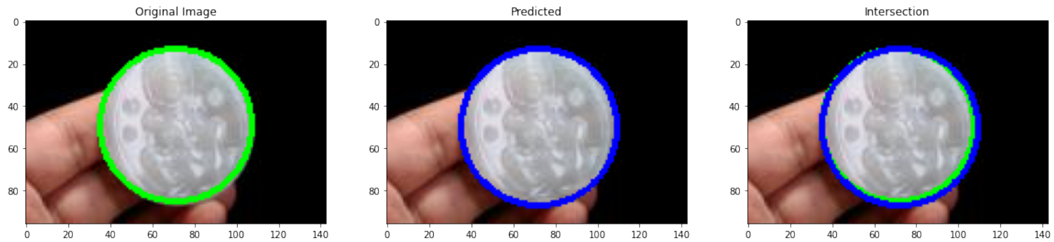


Figure 20: Below is the image named "102.txt". IoU score is 83.35% .

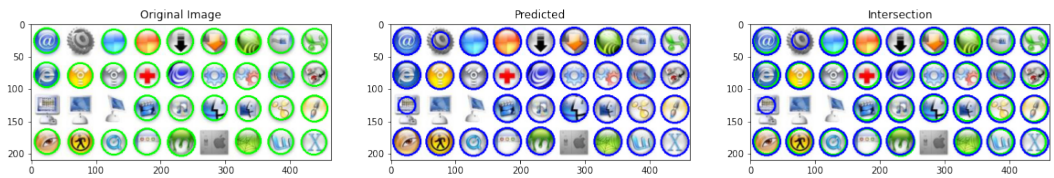
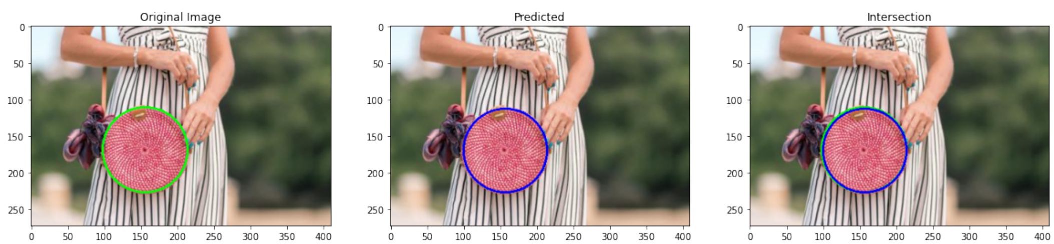


Figure 21: Below is the image named "113.txt". IoU score is 90.01% .



5.2 Failed Results

Below are the unsuccessful results and their reasons. These reasons are supported by visual examples.

Figure 22: Because some parts of the pipes, which appear as circles in the figure below, are not visible in the image, the algorithm is unable to detect this.

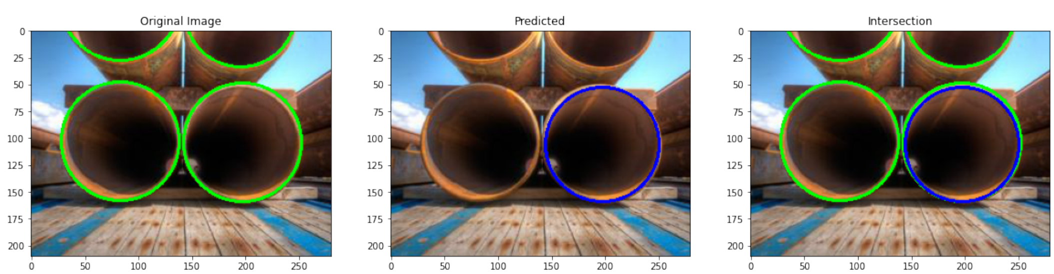


Figure 23: In the image below, the algorithm detects a circle in the noise cloud, due to the high noise (especially tree branches).

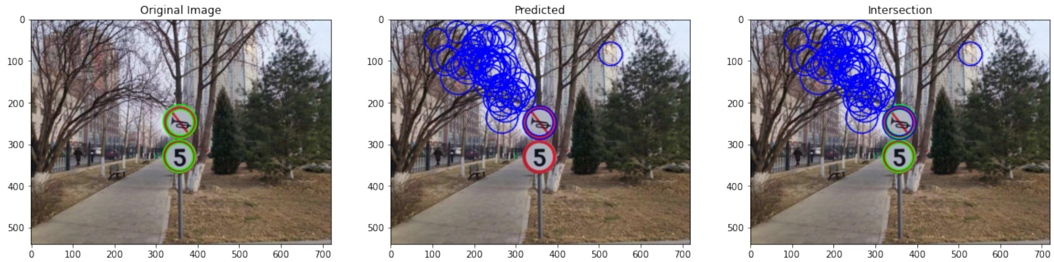


Figure 24: There are some important details in the image below. As seen in the image, the algorithm has difficulty recognizing some circles because they are not a regular circle shape. It is also observed that the detection of some colors in the image is better than other colors. Finally, it is observed that there are circles that are not detected in the main data and are detected by the algorithm.

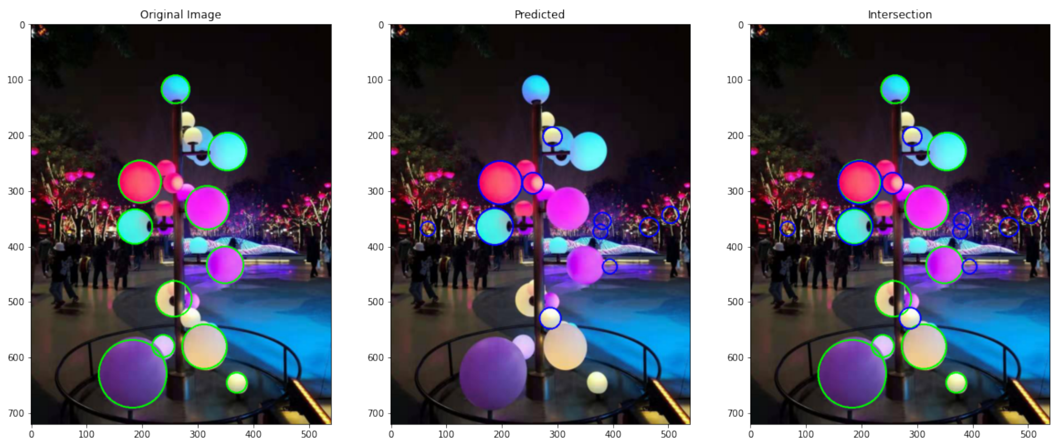
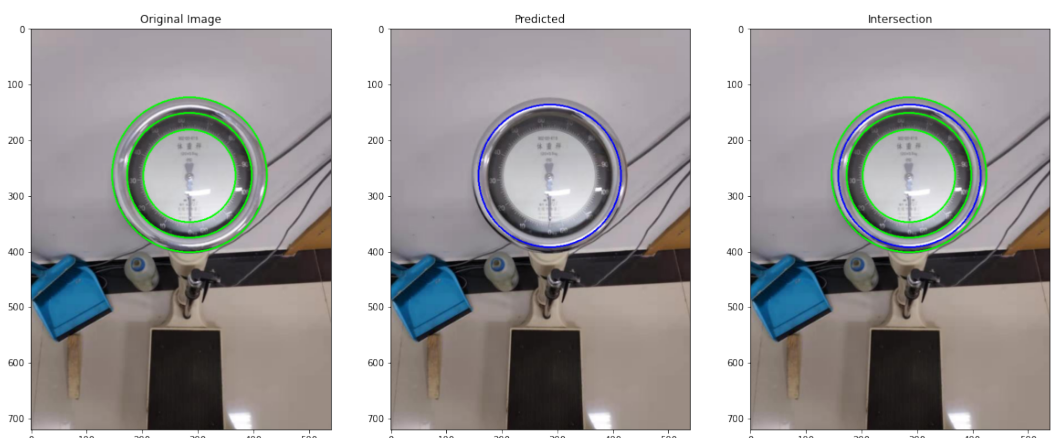
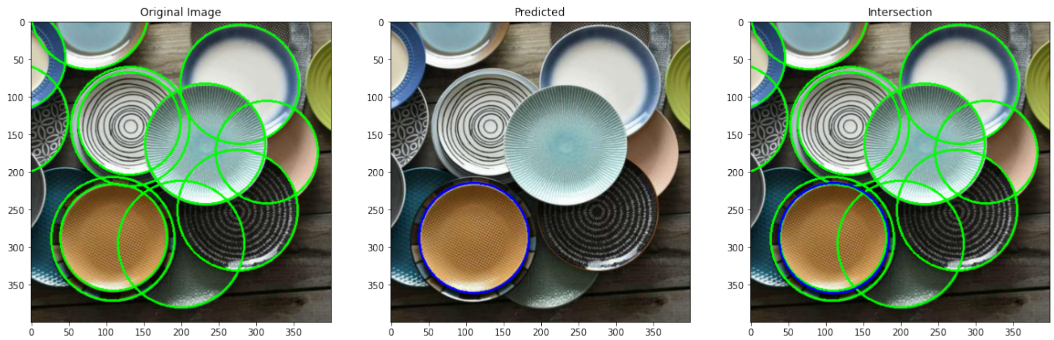


Figure 25: It is observed that the algorithm predicts the nested circles as a single circle in the image below.



Another unsuccessful result of this algorithm is that it is not successful in terms of time cost.

Figure 26: In the image below, which is thought to be easy to detect at first, things get difficult for the algorithm. The main reason for this is that when you pay attention to the plates, it has been observed that there are circular motifs in the plate and the algorithm cannot make a good decision at this point.



6 Conclusion

In this assignment, Edge Detectors and Hough Transform are examined. Information on these subjects was given, and analyzes were made with their codes. In these analyzes, visual and numerical data are shared above.

When we look at the results, it is seen that a simple algorithm can recognize the circles on the image. It is observed that the higher the number of noise in the image, the more errors the algorithm gives. It has been observed that the algorithm has difficulty in finding semi-circles, and there are intersections in some circles that are irrelevant to the subject. However, in general, it can be said that the results can be interpreted positively.

References

[1] Haichun Yang and Ruining Deng and Yuzhe Lu and Zheyu Zhu and Ye Chen and Joseph T. Roland and Le Lu and Bennett A. Landman and Agnes B. Fogo and Yuankai Huo. CircleNet: Anchor-free Detection with Circle Representation., 2020.

[2] Assoc. Prof. Dr. Nazlı İkizler Cinbiş, BBM 418 - Computer Vision Laboratory Programming Assignment-1, 2022

[3] Qi Han and Kai Zhao and Jun Xu and Ming-Ming Cheng. Deep Hough Transform for Semantic Line Detection, 2020.