

MULTIPLE CIRCLE DETECTION IN IMAGES: A
SIMPLE EVOLUTIONARY ALGORITHM APPROACH
AND A NEW BENCHMARK OF IMAGES

Supplementary material

Authors:

Pending

Appendix

A Genetic algorithm	3
B Parameter tuning process	4
B.1 Number of generations	4
B.2 Crossover and mutation rates	5
B.2.1 Normality tests	6
B.2.2 Pairwise Wilcoxon test	7
B.3 Fitness threshold	10
C Selection criterion (detection process)	12
D Detection results	14
D.1 Synthetic images	15
D.2 Hand-drawn images	20
D.3 Real images	23
D.4 Real images, external dataset	28
E Metrics	35

A Genetic algorithm

This document presents a justification for the use of genetic algorithm (GA) through the explanation of how the crossover operator works for circle detection. Then the experiments related to the statistical test of the parameter tuning process are presented. As described in [1] Figure A.1-a) shows two different circles (red and cyan) each encoded by 3 positions (3 red dots and 3 cyan dots), as can be seen, the generated circles do not fit into the point cloud (edge map), however, in Figure A.1-b) by taking one of the three dots of red circle, and two dots of the three dots of the cyan circle, together they form a new green circle that fits in all points.

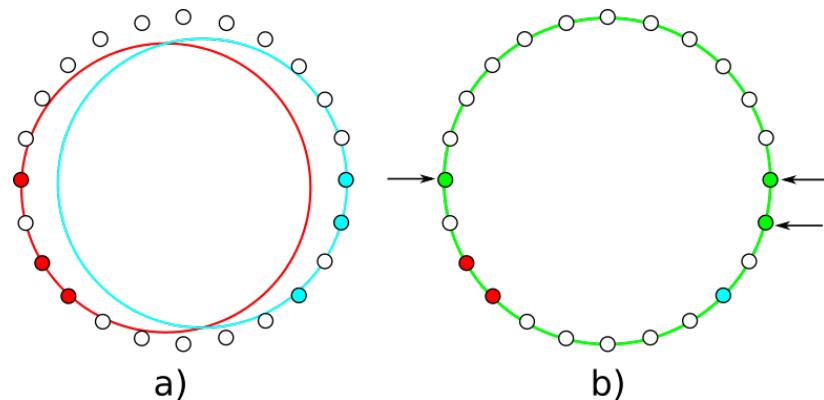


Figure A.1: Example of crossover operator in circle detection problem

In order to show how this process works, three individuals in a run of the algorithm over a synthetic circle is show in the following figure:

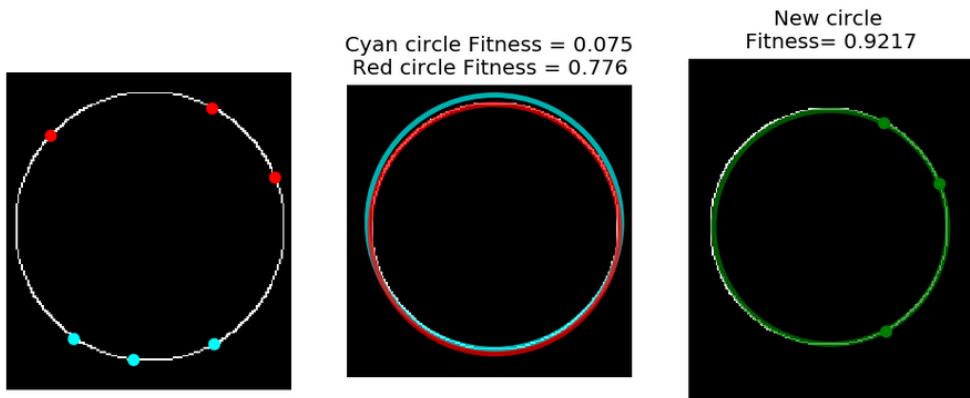


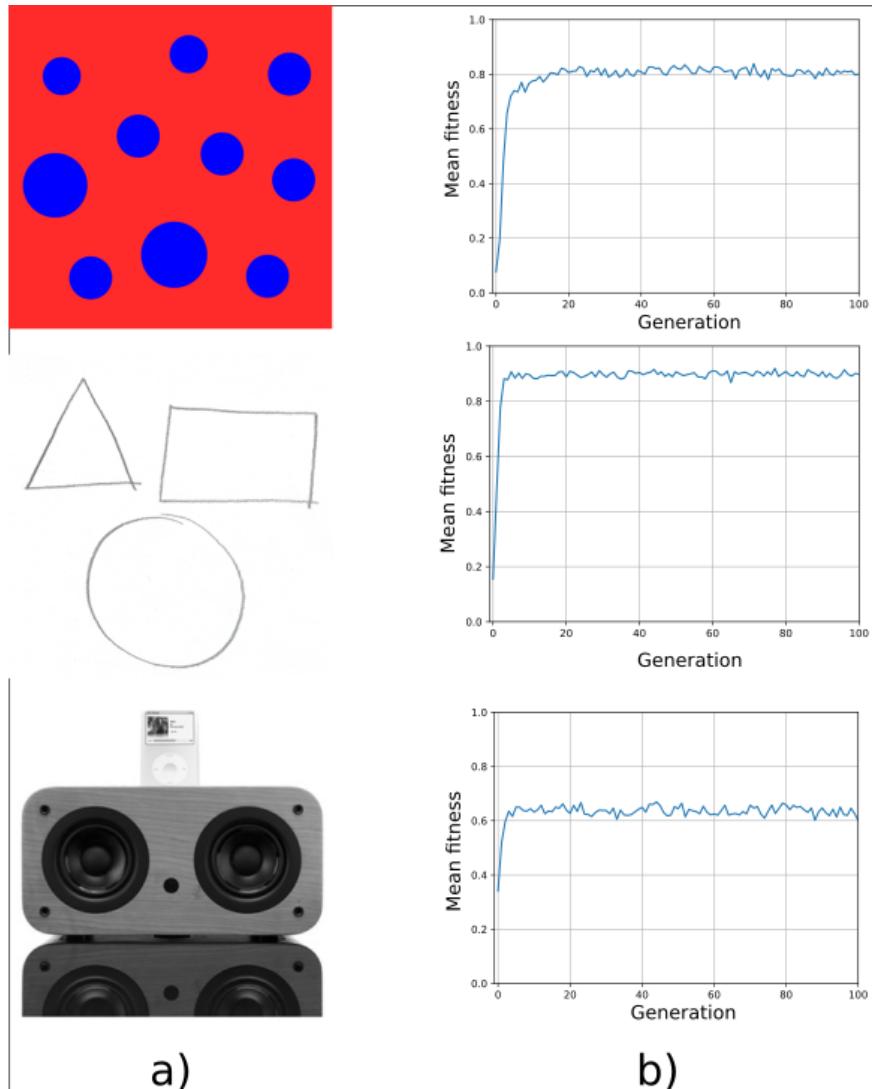
Figure A.2: Simulation of crossover operator in circle detection problem

As can be seen, the fitness of the red and cyan circles are 0.776 and 0.075, respectively; by applying a crossover operator a new better circle is formed, with a 0.9217 fitness value.

B Parameter tuning process

B.1 Number of generations

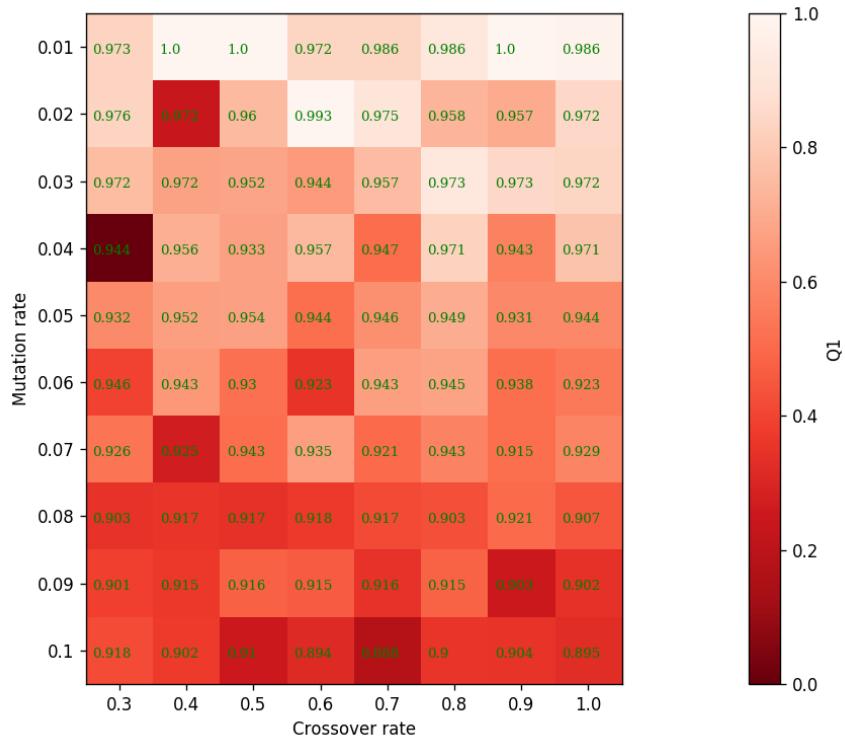
We present a plot of the mean fitness of the population in the evolution process; the input images are: synthetic, hand-drawn and real images. The algorithm is run 30 times.



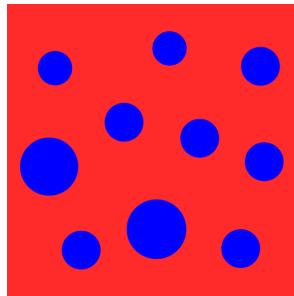
Population size, mutation and crossover rates used for this process were 70, 0.1 and 0.55, respectively, as they were used in the work of Ayala-Ramirez (reference [37] from the paper). In the figure, on column a) we present the input image used, and in column b) the mean fitness value as a function of the generation can be observed. As can be seen, we can conclude that fitness converges at generation 25, however, we suggest using 50 generations to have enough margin to be sure that the convergence is achieved in every case.

B.2 Crossover and mutation rates

We present an exhaustive grid search in order to establish the best combination of mutation and crossover rates. To this aim, we ran for 50 generations and a population size of 70 individuals, the experiment is repeated 100 times over 80 different combinations of mutation and crossover rates, the mean fitness value is computed. The mutation rate was explored in range [0.01,0.1], steps of 0.01, and the crossover rate in range [0.3,1] in steps of 0.1, resulting in 80 combinations. In order to show all combinations the following grid is presented:



where each cell presents the median value and the color corresponds to the first quartile sample. As can be seen, the combinations: (0.01,0.4), (0.01,0.5) and (0.01,0.9) produce the highest values, so we select the first one. The input image used was the following:

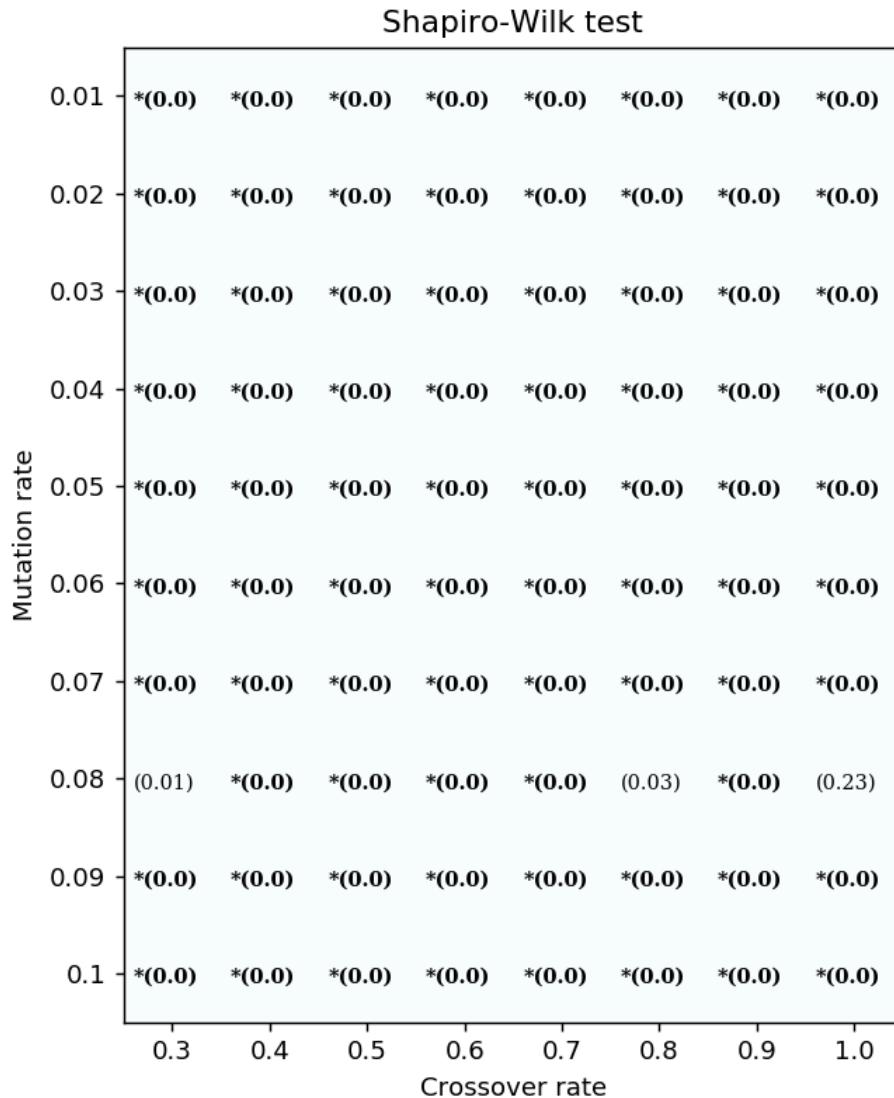


By following reference [2], we use the non-parametric Wilcoxon test, but first we determine whether or not the results follow a normal distribution, in this regard

a normality test is performed: Shapiro-Wilk. Remember that the null hypothesis establishes that the sample under test comes from a normal distribution, while the alternative hypothesis tells us that the sample does not.

B.2.1 Normality tests

Over each grid the resulting p values are presented for each test, the bold ones along with an * represents the ones that reject the null hypothesis with $\alpha = 0.01$, this value is used in all tests.



As can be seen, in Shapiro-Wilk test, only combinations: (0.08,0.3), (0.08,0.8) and (0.08,1) accept the null hypothesis.

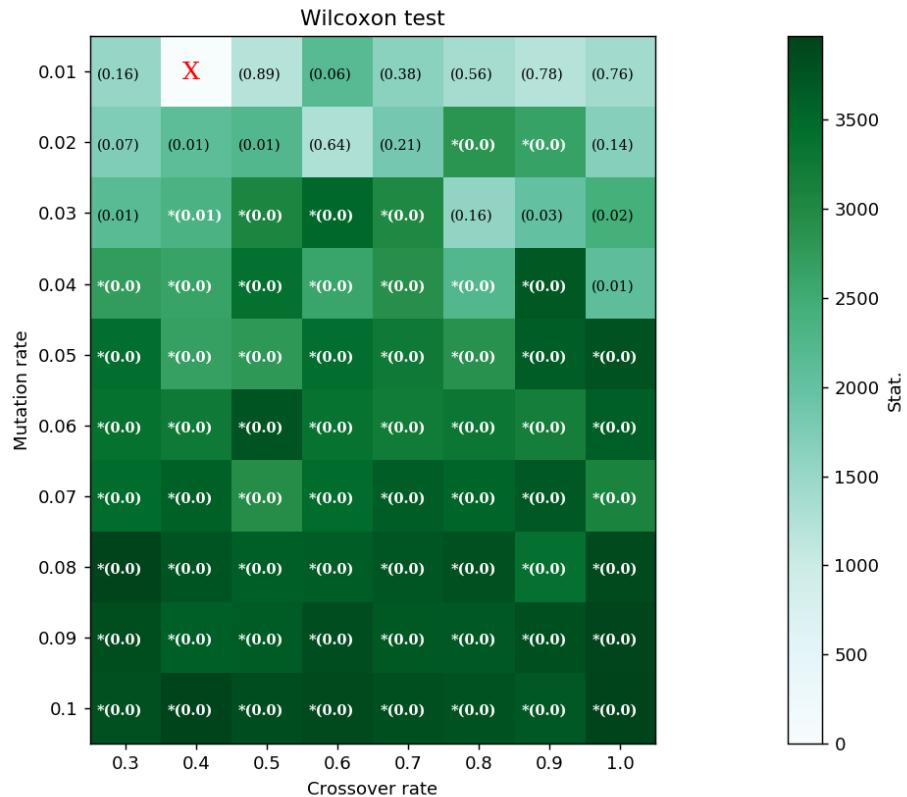
B.2.2 Pairwise Wilcoxon test

Given that the results did not follow a normal distribution we performed a pairwise Wilcoxon test, in order to establish whether there is significant difference between one of the combinations and the rest of them. This non-parametric test establish that:

$$H_0 : X - Y = 0$$

$$H_a : X - Y \neq 0, \quad H_a : X - Y > 0, \quad H_a : X - Y < 0,$$

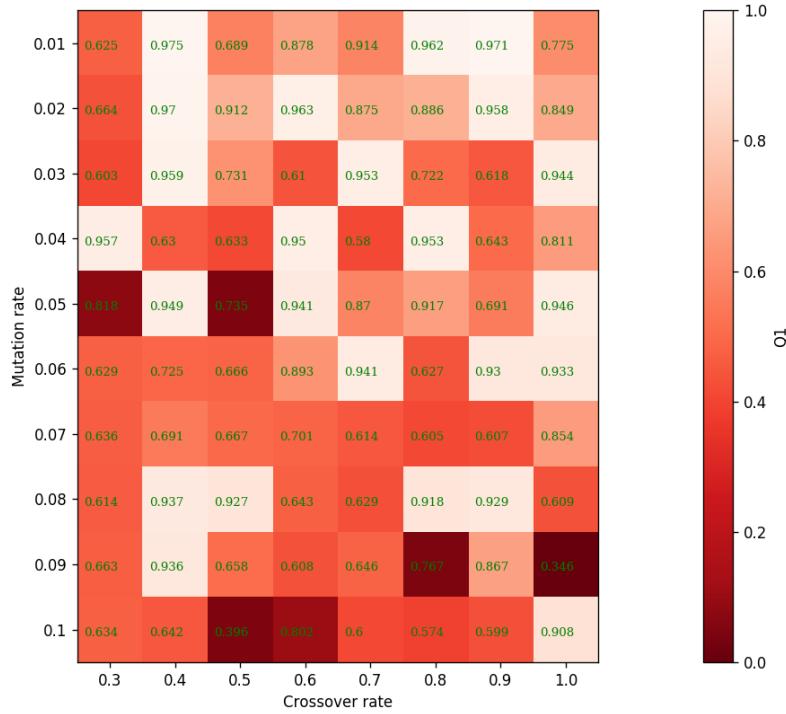
The sample to compare is the one mentioned above (0.01,0.4), the following grid shows the obtained results:



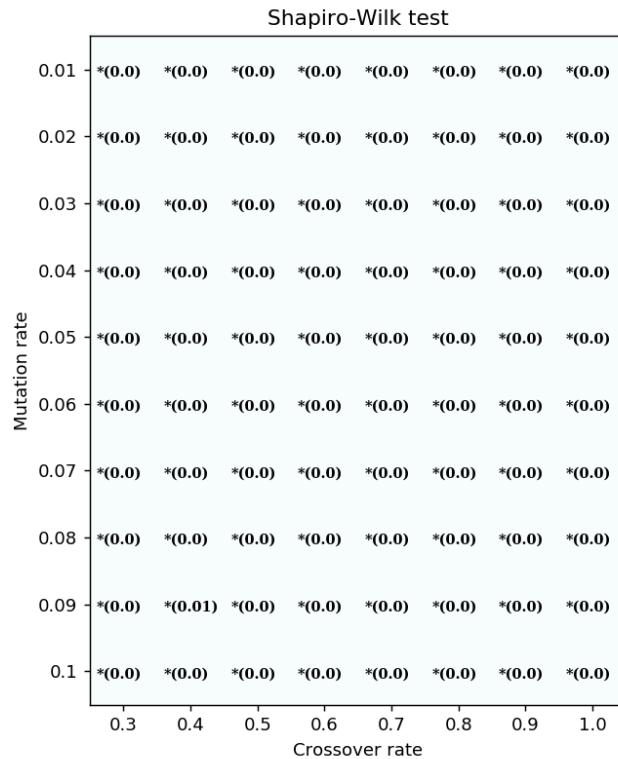
We can conclude that: there is no significant difference over the black values, while white values reject the null hypothesis. In order to extend these results we have performed the same experiment, using a different input image:



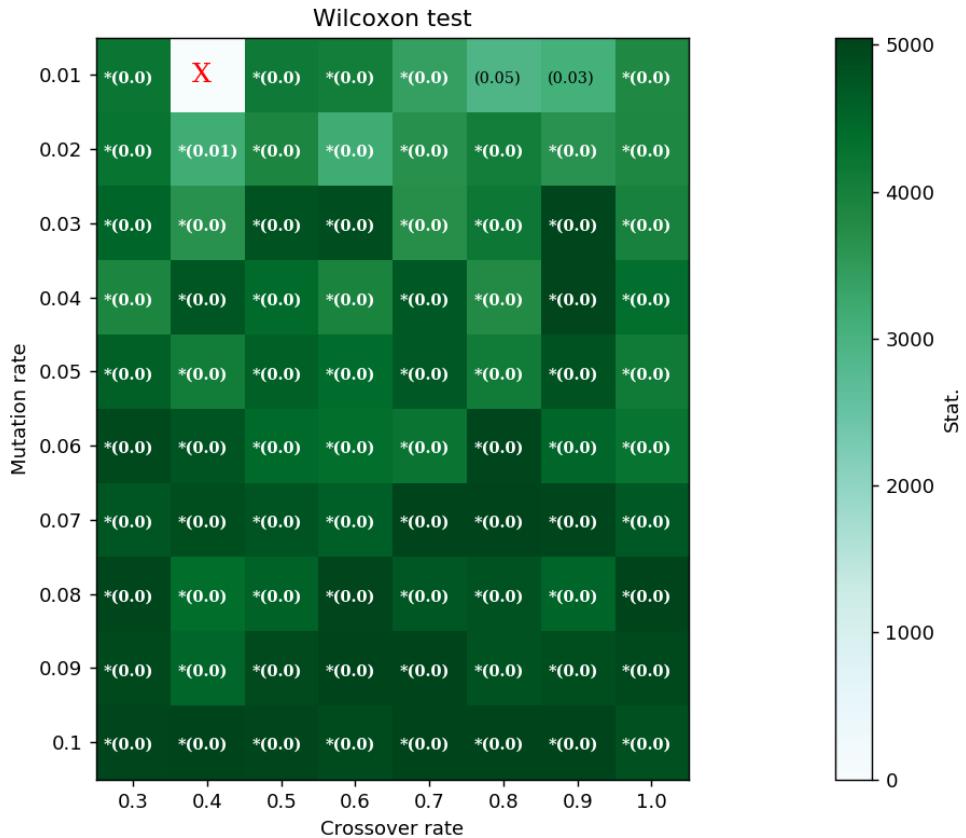
The resulting grid is the following:



We can also see that the combination of (0.01,0.4) is the one with the greatest median fitness value. The normality test results are presented in the following grids:



As can be seen, Shapiro-Wilk test rejects the null hypothesis in all cases. Last, we repeat the pairwise Wilcoxon test, the obtained results are the following:



Here we can observe that only 2 combinations (the black values) accept the null hypothesis, while for the other values there is a significant difference from the result produced by the combination (0.01,0.4). Concluding that the values for each rate are the following:

1. **Mutation rate** = 0.01
2. **Crossover rate** = 0.4

All statistical tests were performed in R with the following functions:

- wilcox.test and
- shapiro.test

B.3 Fitness threshold

The fitness threshold (used as a stopping criterion for detection process) is fixed to 0.5, we can see how this parameter works:

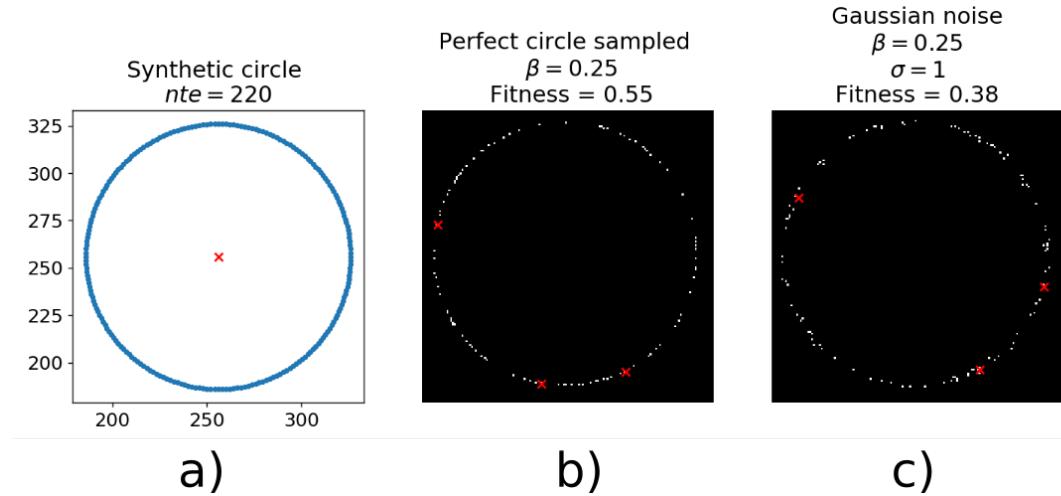


Figure a) shows a generated circle with the TP method by using the following parameters: $r = 70$, $\alpha = 0.5$ and $nte = 220$ (represents the factor that controls the number of points to generate). Figure b) presents also a TP generated circle with $r = 70$, but with $\alpha = 1$, $nte = 220$, and by randomly missing 75% of its points ($\beta = 0.25$), its corresponding fitness equals to 0.55; this fitness is computed from the individual (3 points) displayed in Figure b) as 3 red marks ("x"). Then, in Figure c) the same circle is presented but with Gaussian noise-added to mimic a deformation; this process consists of adding a random Gaussian 2D array with factor noise $\alpha = 1$ and a center mean value of zero, to the circle presented in Figure b); as can be seen the fitness drops to 0.38. With a fitness threshold of 0.5 we can recover circles with 75% of missing points from its perimeter. Notice that the β factor is not mentioned in the paper, this value was only used to illustrate what happens when the circles present missing points. Now, we present a synthetic image created with Inkscape with $r = 70$ px (Figure d)):

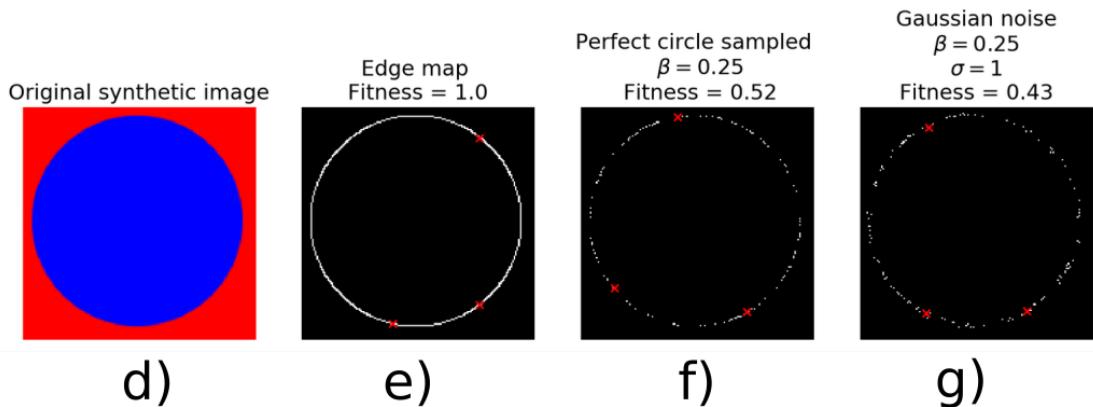
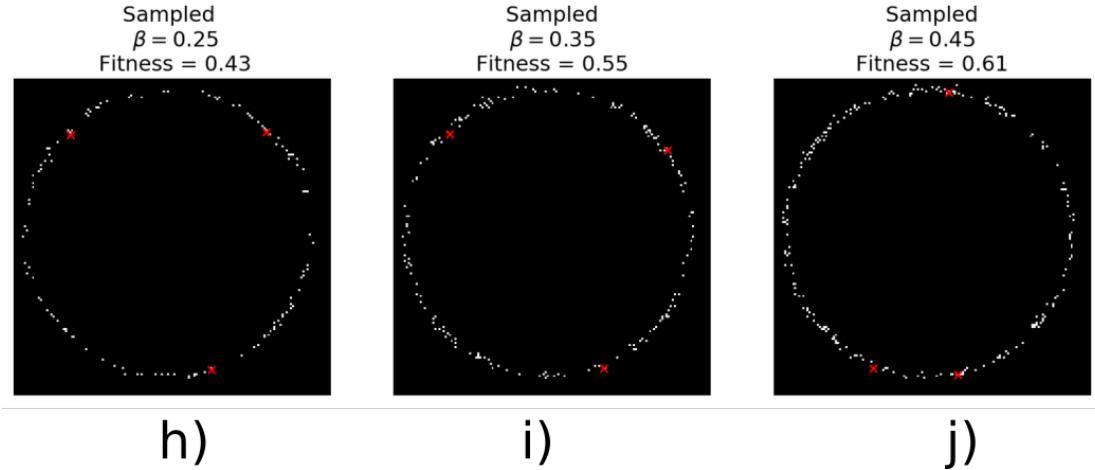


Figure e) presents the edge map obtained from Figure d) and the fitness of the individual displayed (red "x") is equal to 1. Figures f) and g) present the same conditions

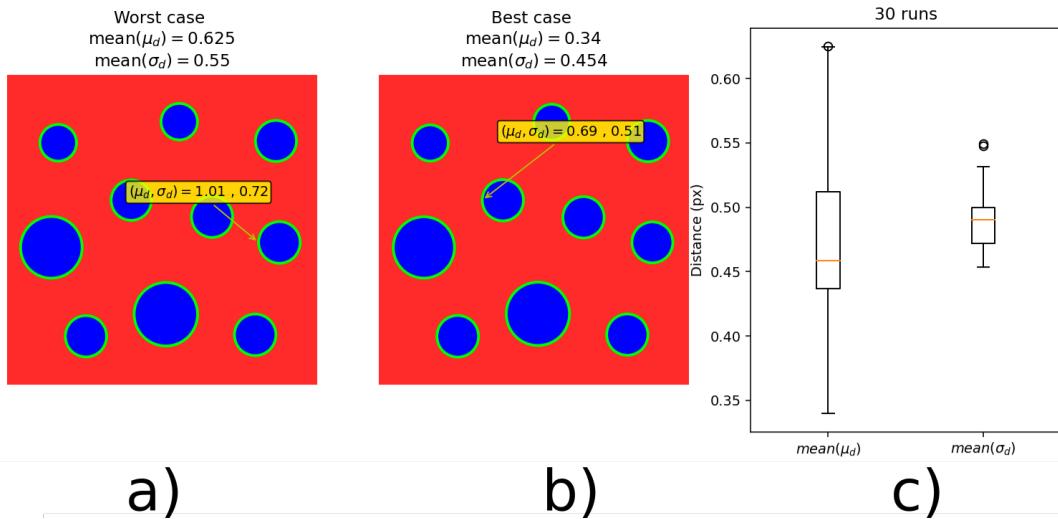
as in figures b) and c), respectively, but using the points from the edge map in Figure e). As can be seen, the fitness values are equal to 0.52 and 0.43, respectively. Concluding that a fitness threshold 0.5 can recover non deformed circles with 75% of missing points from its perimeter. Additionally, we present the following figures:



In figures h), i) and j), also the points from edge map in figure e) were used, and the same configuration noise as in Figure g) was used. We decrease the amount of missing points: from 75%, to 65% and to 55%, respectively ($\beta = 0.25, 0.35$ and 0.45). We can observe that circles with deformation (Gaussian noise-added) and between 55% and 65% of its perimeter points missing (as can be seen in figures i) and j)) can be recovered by using a fitness threshold of 0.5. However, circles that are defined by only 25% of the total points in the circumference will not be recovered.

C Selection criterion (detection process)

Since CDSGA is a stochastic method, we ran 30 times the algorithm on every image and chose the one with the lowest average of μ_d and σ_d from all circles. Remember that μ_d and σ_d are the mean and standard deviation distance (measured in pixels, px), respectively, between the detected circle and the closest edge pixels; we can see these values as a metric of the completeness of the detected circle. The following figures present a sample of each type image used in this work (synthetic, hand-drawn and real) with the worst and the best case of detection from the 30 runs. Figure a)

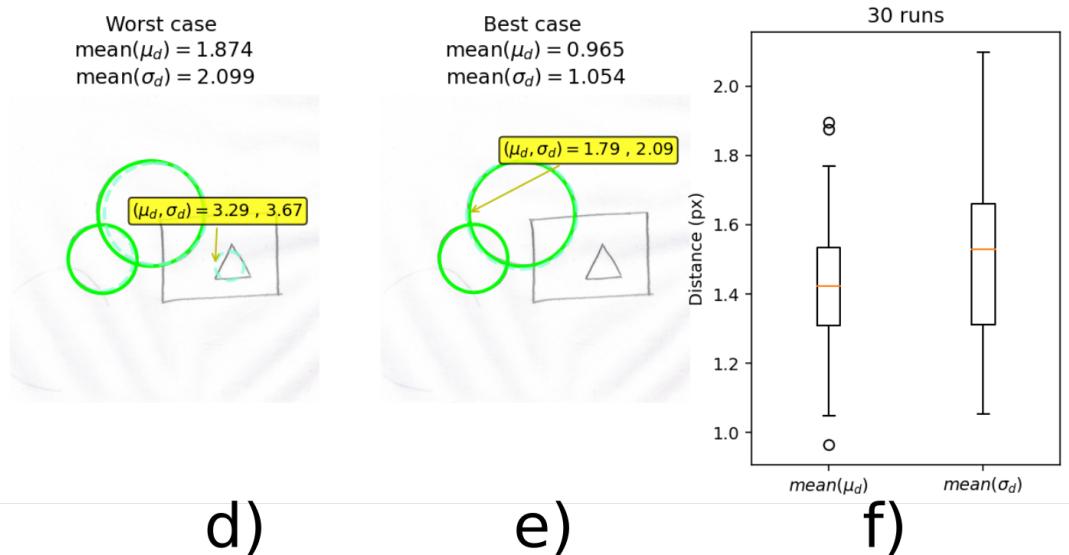


a)

b)

c)

and b) shows the worst and the best case, respectively, for a sample of synthetic images. In Figure a) we can see the worst circle detected marked, as well in Figure b). Figure c) shows a multiple box-plot for the 30 μ_d and σ_d for every run. Notice that the difference between the worst and the best case is less than 1 pixel.



d)

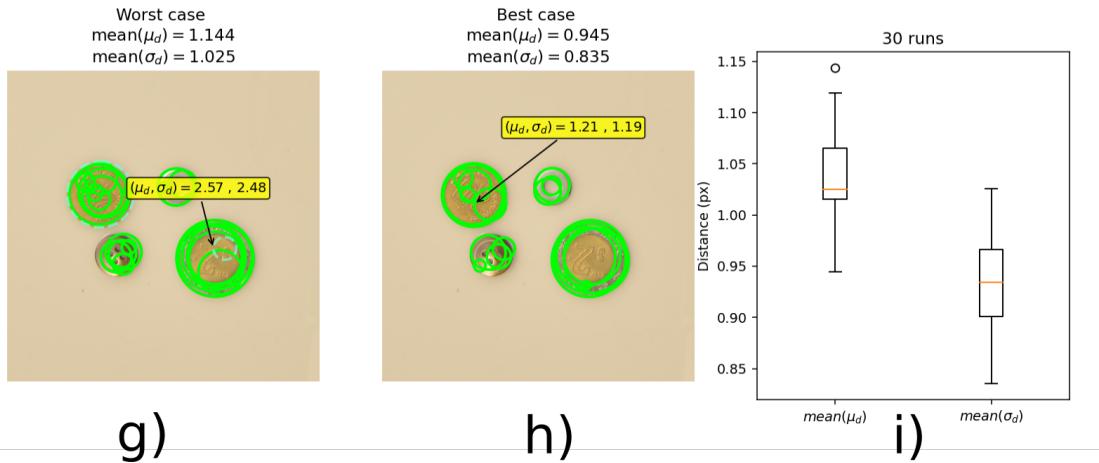
e)

f)

Figures d) and e) also show the worst and the best case, respectively, but in a sample of hand-drawn images. Figure f) shows a multiple box-plot with difference between

C SELECTION CRITERION (DETECTION PROCESS)

the worst and best case of less than 2 pixels. Notice that the worst circles detected in Figures d) and e) are detected false positives.



The same structure is the same for the sample of real images (figures g), h) and i)). Figure i) shows that the difference between the best and worst cases is less than 2 pixels. The worst detected circle in Figure g) is a real false positive, while in Figure h) passes as a detected circle.

D Detection results

The details of every image are presented in this section. Each figure is displayed with the following structures: (a) the manually annotated image (ground truth), (b) its corresponding edge map, (c) the EDCircles results, (d) the CHT results, and (e) our proposed method results; we also present a table at the end of each image category with all metric results for each circle (Radius and Position). Additionally, Table D.1 shows the input parameters used in CHT for each image.

Image	Mininmum Distances between circles	Canny high threshold	Accumulator threshold
Synthetic Figure 9			
1)	10	50	60
2)	10	50	150
3)	10	50	60
4)	10	50	50
Real images figures 11 and 13			
1)	10	55	96
2)	10	47	190
3)	10	50	70
4)	10	50	20
5)	10	255	200
Real images external data set figures 14 and 15			
1)	10	100	51
2)	10	35	50
3)	10	50	120
4)	10	50	90
5)	10	46	70
6)	10	54	150
Hand-draw images Figure 10			
1)	10	50	70
2)	10	50	60
Gradual noise added Figure 18			
1 %	170	50	50
10 %	170	50	50
20 %	170	200	30

Table D.1: HoughCircle() input parameters for every image compared.

D.1 Synthetic images

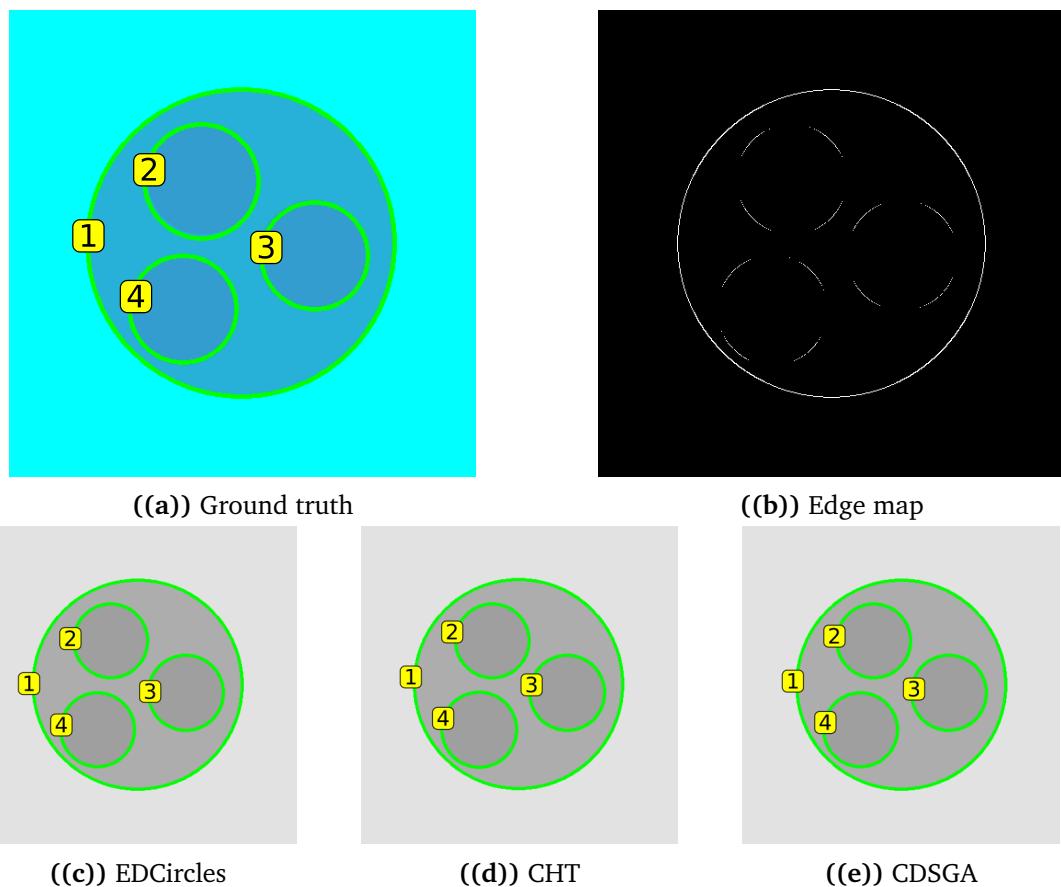


Figure D.1: Figure 9-1 from the paper

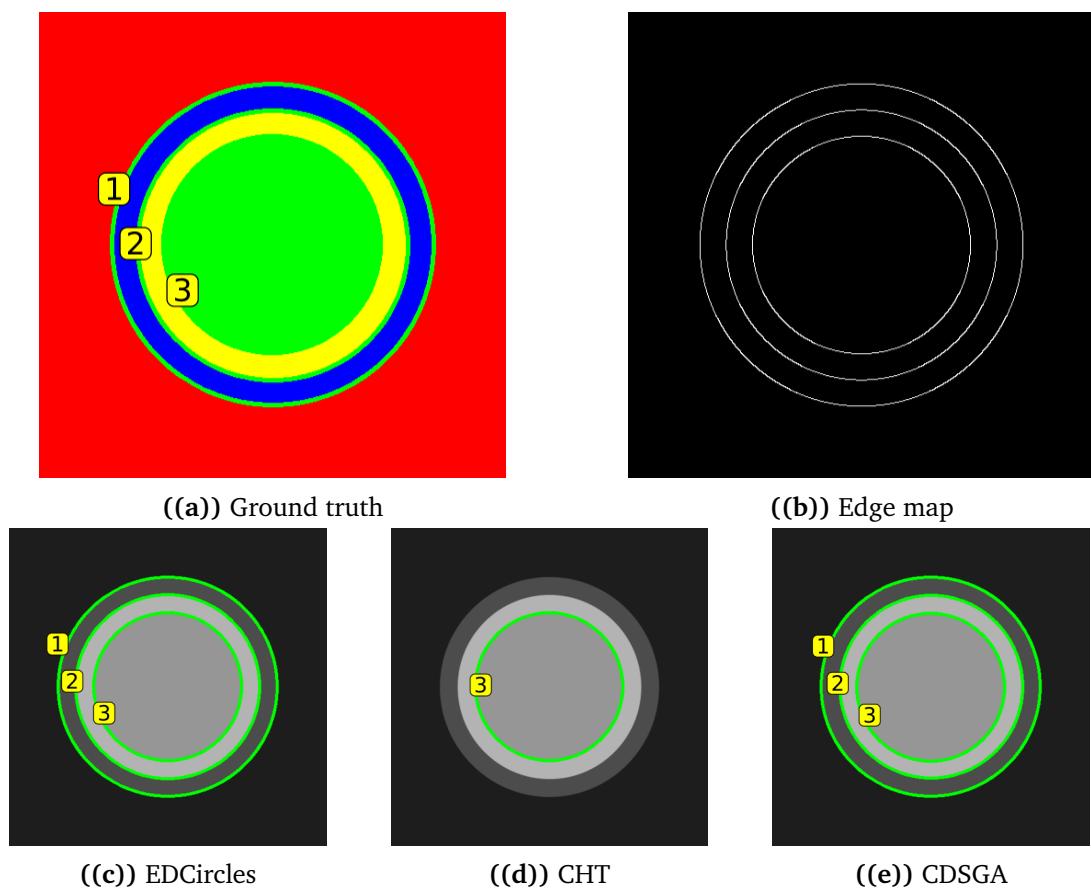


Figure D.2: Figure 9-2 from the paper

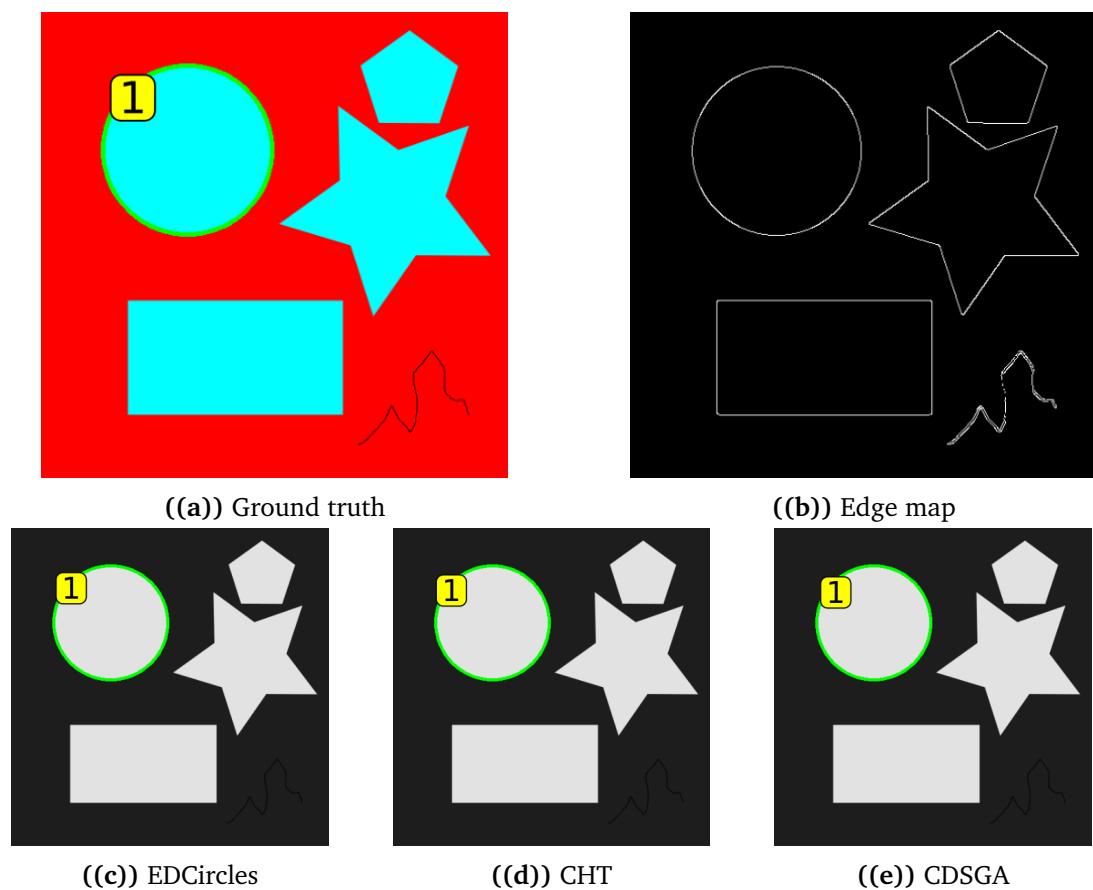


Figure D.3: Figure 9-3 from the paper

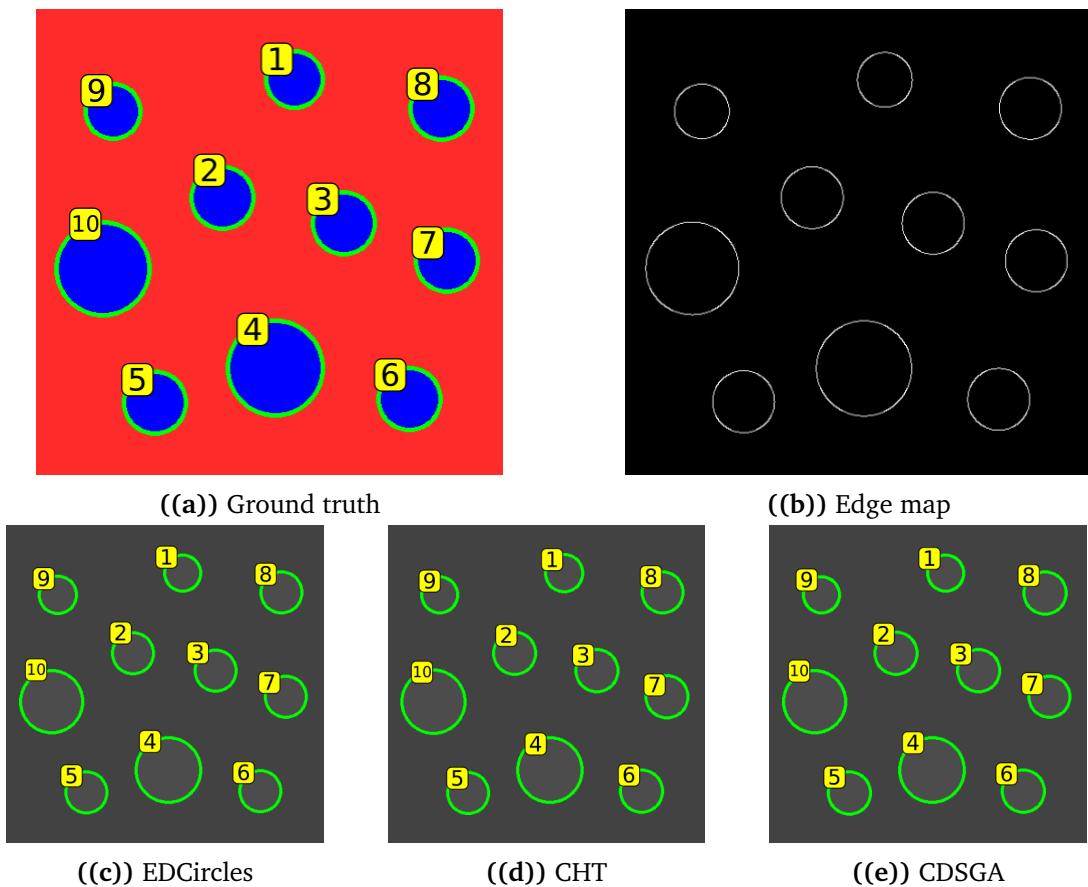


Figure D.4: Figure 9-4 from the paper

Image	Circle	EDCircles		CHT		CDSGA	
		Position center	Radius	Position center	Radius	Position center	Radius
Figure D.1	1	0.0016	0.0028	0.0023	0.0005	0.0034	0.0034
	2	0.0087	0.0379	0.0098	0.0391	0.0076	0.0391
	3	0.0049	0.0196	0.0045	0.0193	0.0057	0.0198
	4	0.0017	0.0247	0.0006	0.0248	0.0019	0.0212
Figure D.2	1	0.0022	0.0033			0.0021	0.0047
	3	0.0030	0.0018			0.0014	0.0038
	2	0.0013	0.0021	0.0013	0.0008	0.0007	0.0002
Figure D.3	1	0.0019	0.0007	0.0022	0.0056	0.0021	0.0006
Figure D.4	1	0.0028	0.0424	0.0026	0.0355	0.0050	0.0492
	2	0.0022	0.0132	0.0035	0.0007	0.0038	0.0092
	3	0.0018	0.0137	0.0023	0.0004	0.0017	0.0055
	4	0.0011	0.0055	0.0012	0.0006	0.0009	0.0058
	5	0.0027	0.0159	0.0019	0.0280	0.0023	0.0011
	6	0.0008	0.0047	0.0007	0.0057	0.0003	0.0006
	7	0.0014	0.0084	0.0009	0.0059	0.0017	0.0100
	8	0.0014	0.0174	0.0028	0.0280	0.0008	0.0145
	9	0.0043	0.0303	0.0038	0.0414	0.0016	0.0356
	10	0.0023	0.0201	0.0019	0.0128	0.0022	0.0257

Table D.2: Results of all detected circles from synthetic images.

D.2 Hand-drawn images

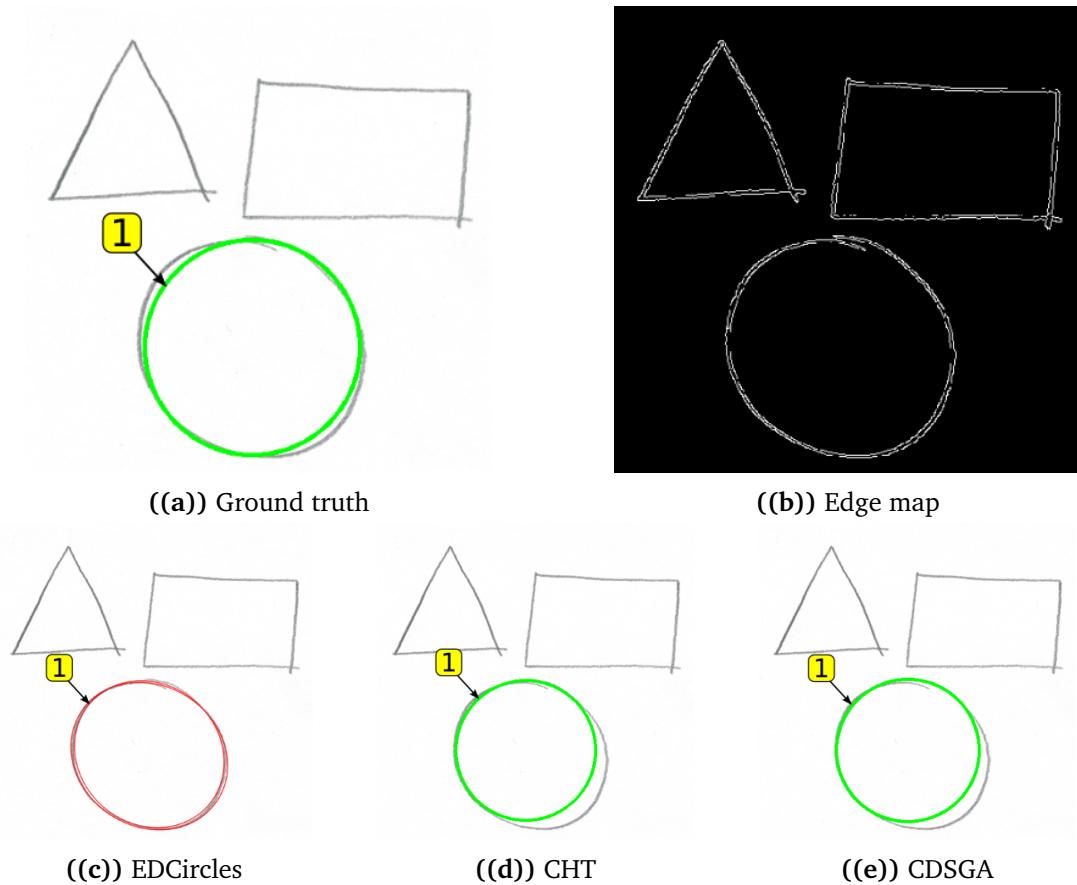


Figure D.5: Figure 10-1 from the paper

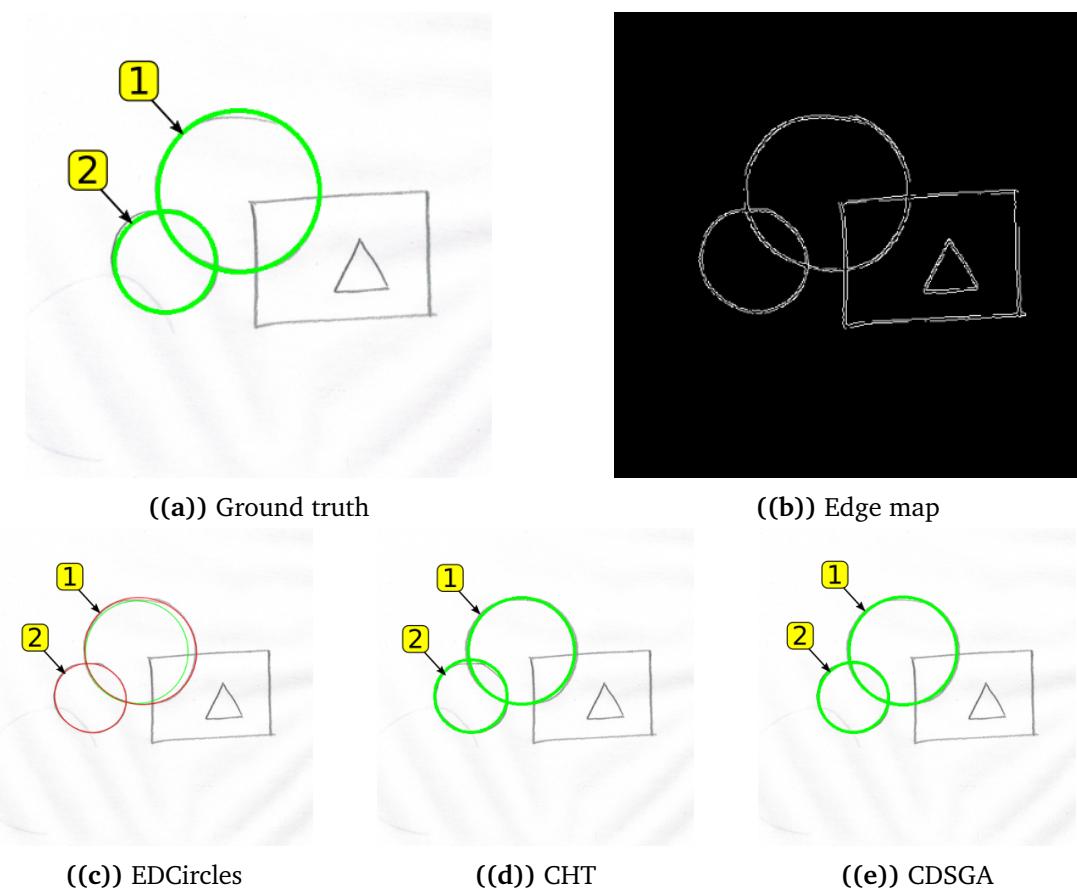


Figure D.6: Figure 10-2 from the paper

Image	Circle	EDCircles		CHT		CDSGA	
		Position center	Radius	Position center	Radius	Position center	Radius
Figure D.5	1	0.0050	0.0400	0.0267	0.0491	0.0248	0.0296
Figure D.6	1	0.0203	0.0660	0.0046	0.0338	0.0037	0.0223
	2	0.0050	0.0350	0.0126	0.0449	0.0054	0.0012

Table D.3: Results of all detected circles from **hand-drawn** images.

D.3 Real images

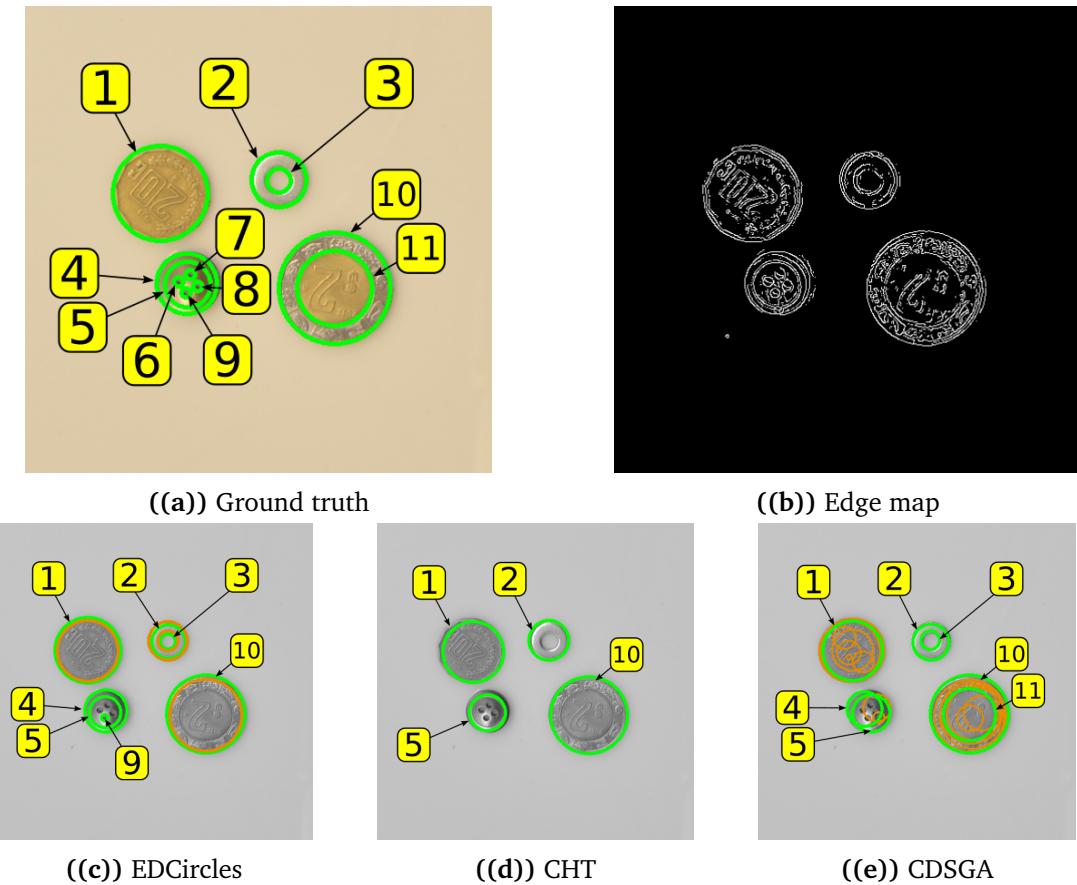


Figure D.7: Figure 11-1 from the paper

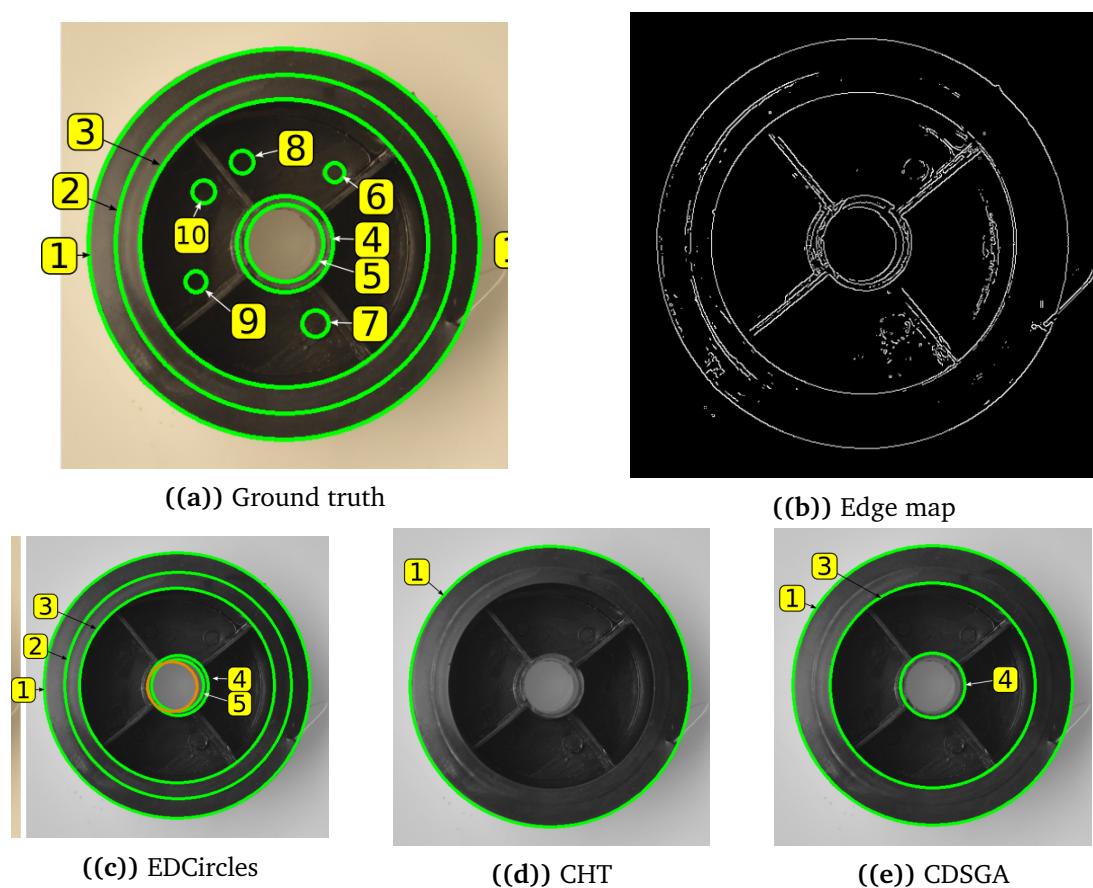


Figure D.8: Figure 11-2 from the paper

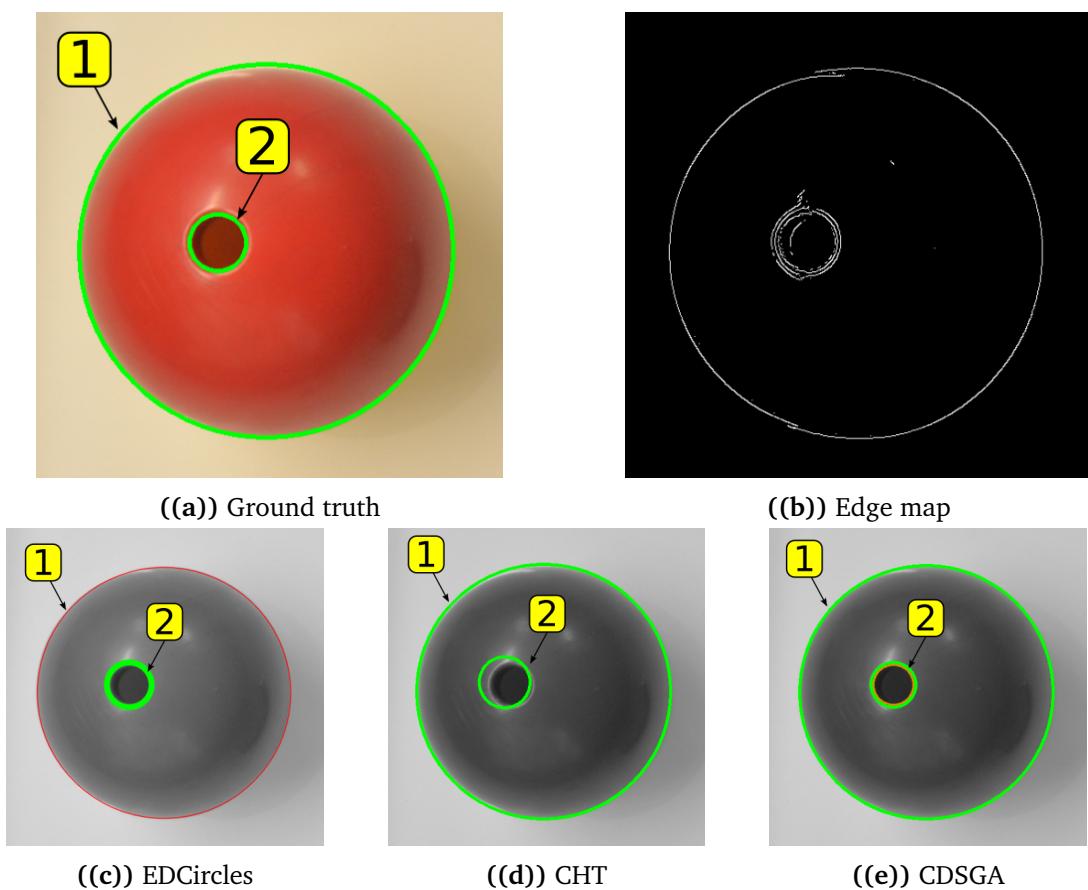


Figure D.9: Figure 13-3 from the paper

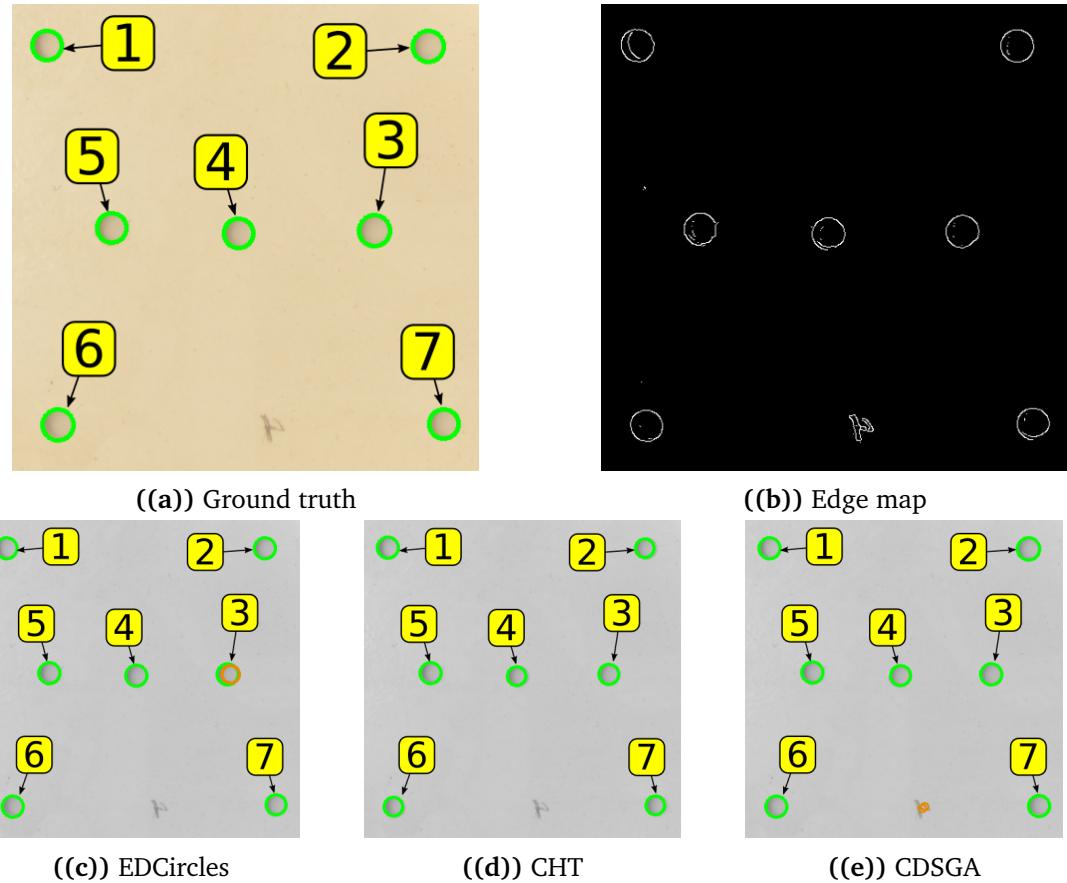


Figure D.10: Figure 13-4 from the paper

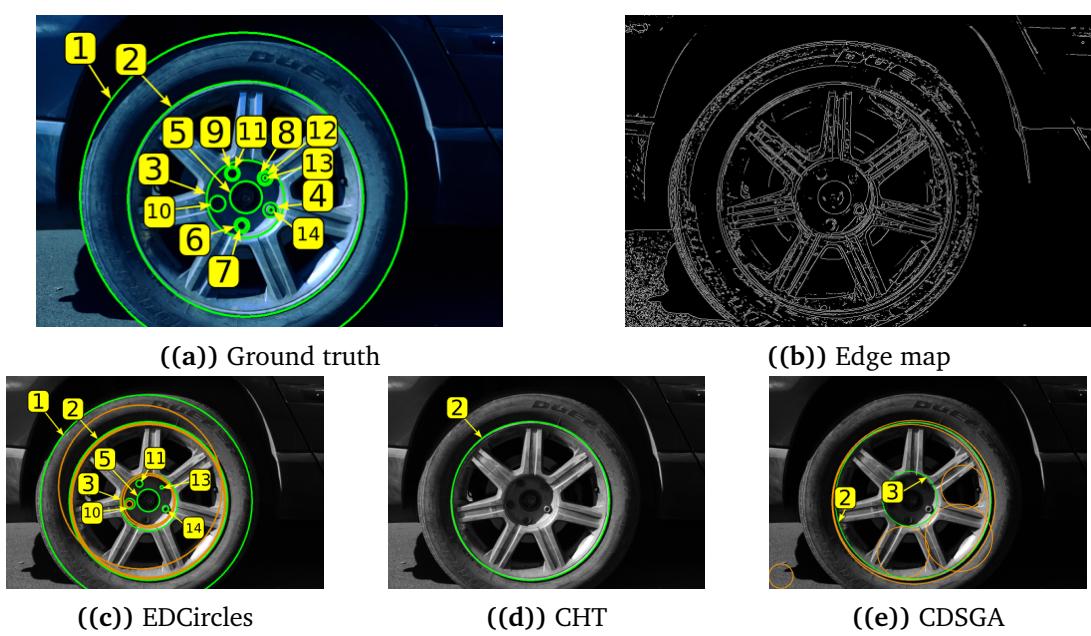


Figure D.11: Figure 13-5 from the paper

Image	Circle	EDCircles		CHT		CDSGA	
		Position center	Radius	Position center	Radius	Position center	Radius
Figure D.7	1	0.0060	0.0218	0.0269	0.0559	0.0109	0.0659
	2	0.0021	0.1193	0.0055	0.0368	0.0037	0.0384
	3	0.0021	0.0781			0.0059	0.1665
	4	0.0036	0.0152			0.0203	0.1534
	5	0.0016	0.0334			0.0129	0.2283
	9	0.0012	0.0225				
	10	0.0016	0.0238	0.0050	0.0141	0.0019	0.0202
	11					0.0011	0.0217
	Figure D.8	1	0.0034	0.0057	0.0071	0.0125	0.0015
	2	0.0025	0.0120			0.0011	0.0025
	3	0.0025	0.0022			0.0035	0.0541
Figure D.9	4	0.0025	0.0557				
	5	0.0029	0.0034				
Figure D.10	1	0.0050	0.0070	0.0029	0.0012	0.0043	0.0072
	2	0.0074	0.1049	0.0348	0.3331	0.0053	0.1879
	3	0.0097	0.0536	0.0099	0.0180	0.0147	0.0406
	4	0.0022	0.0072	0.0067	0.1062	0.0007	0.0198
	5	0.0024	0.0051	0.0026	0.0457	0.0013	0.0082
	6	0.0007	0.0398	0.0058	0.0777	0.0048	0.0211
	7	0.0071	0.0466	0.0096	0.0234	0.0051	0.0619
Figure D.11	8	0.0024	0.0069	0.0057	0.0850	0.0015	0.0084
	9	0.0023	0.0137	0.0046	0.1102	0.0010	0.0079
	10	0.0099	0.0468				
	11	0.0031	0.0099	0.0028	0.0132	0.0097	0.0249
	12	0.0036	0.0814			0.0174	0.0375
	13	0.0022	0.0260				
	14	0.0031	0.0142				

Table D.4: Results of all detected circles from real images.

D.4 Real images, external dataset

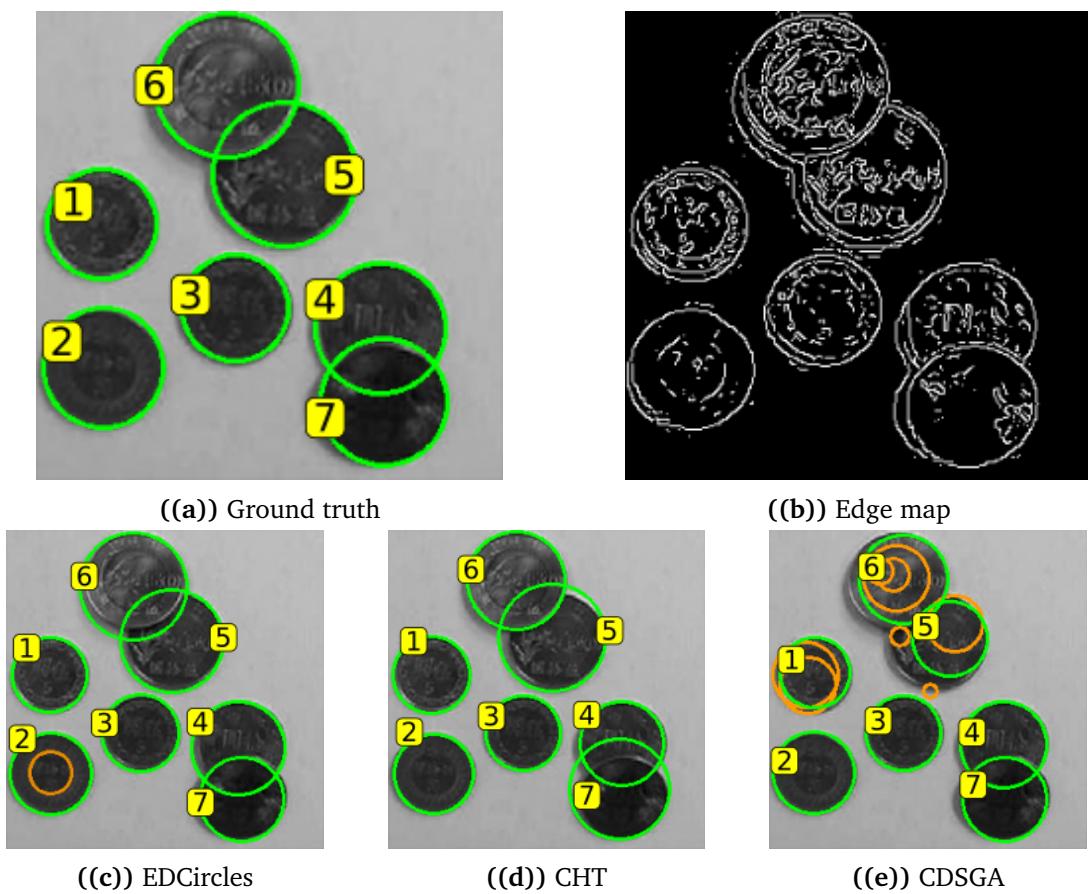


Figure D.12: Figure 14-1 from the paper

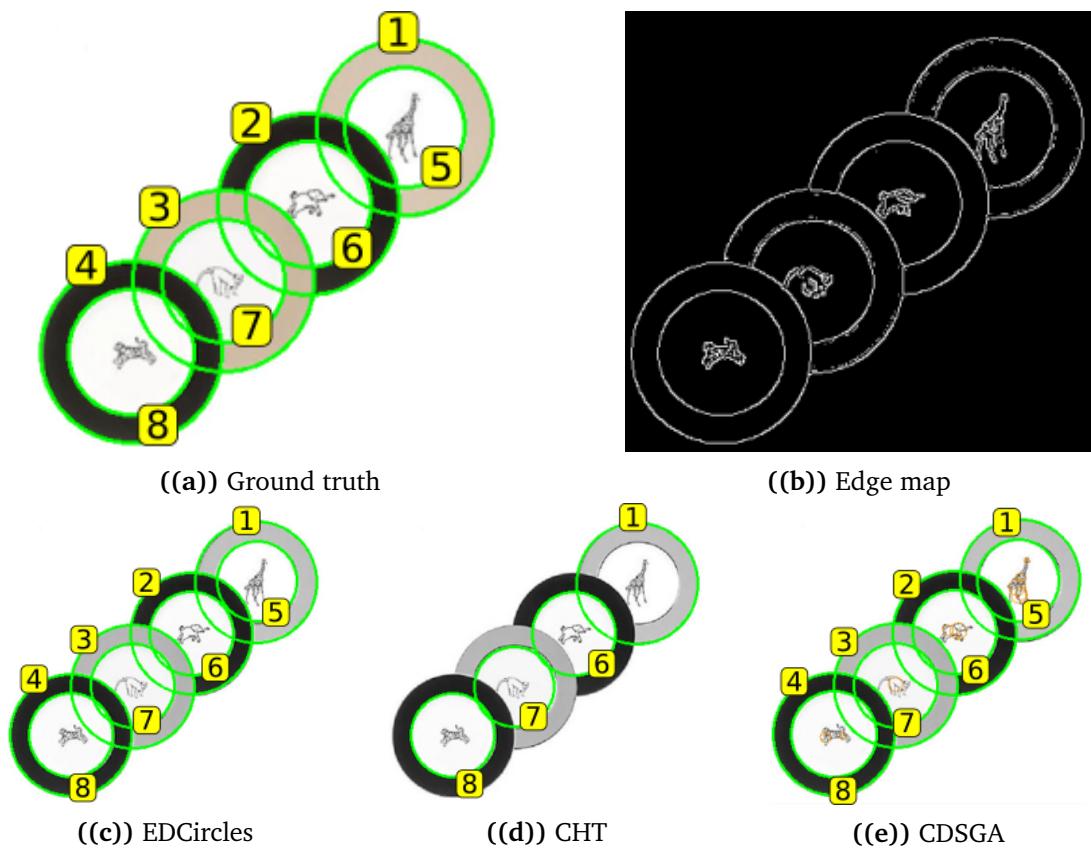


Figure D.13: Figure 14-3 from the paper

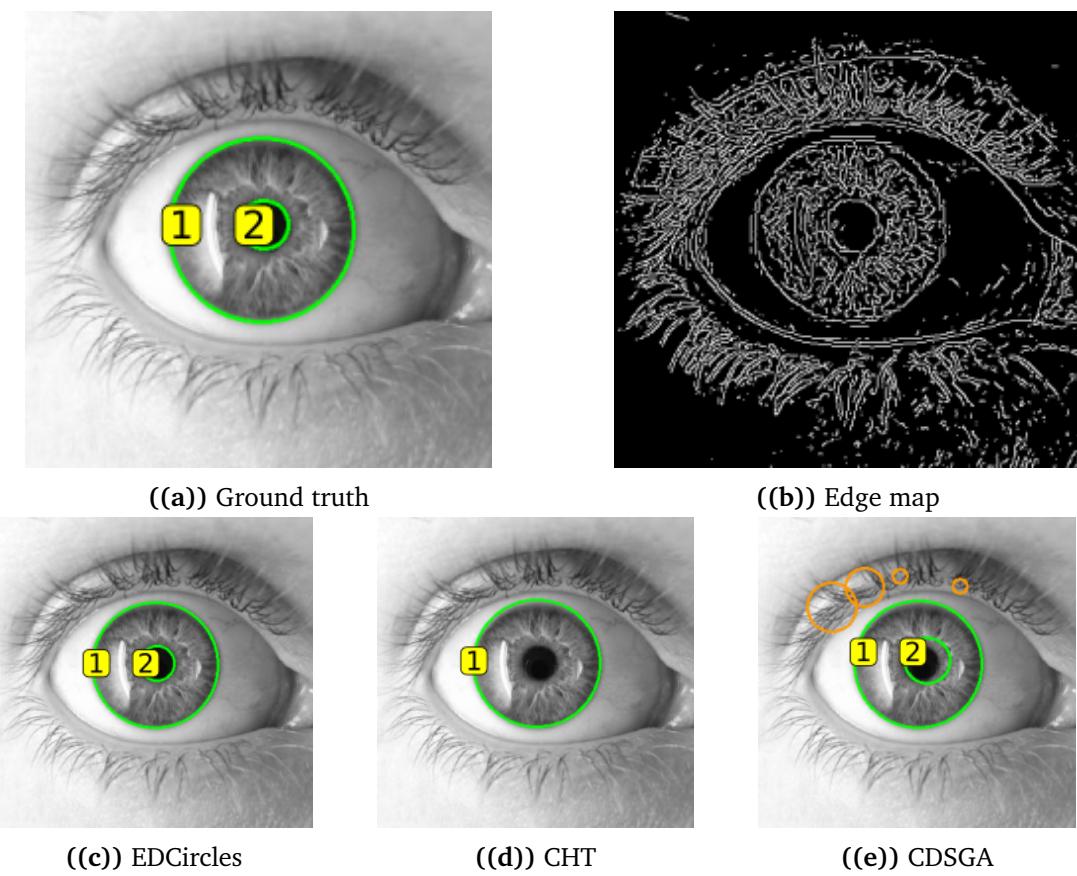


Figure D.14: Figure 14-3 from the paper

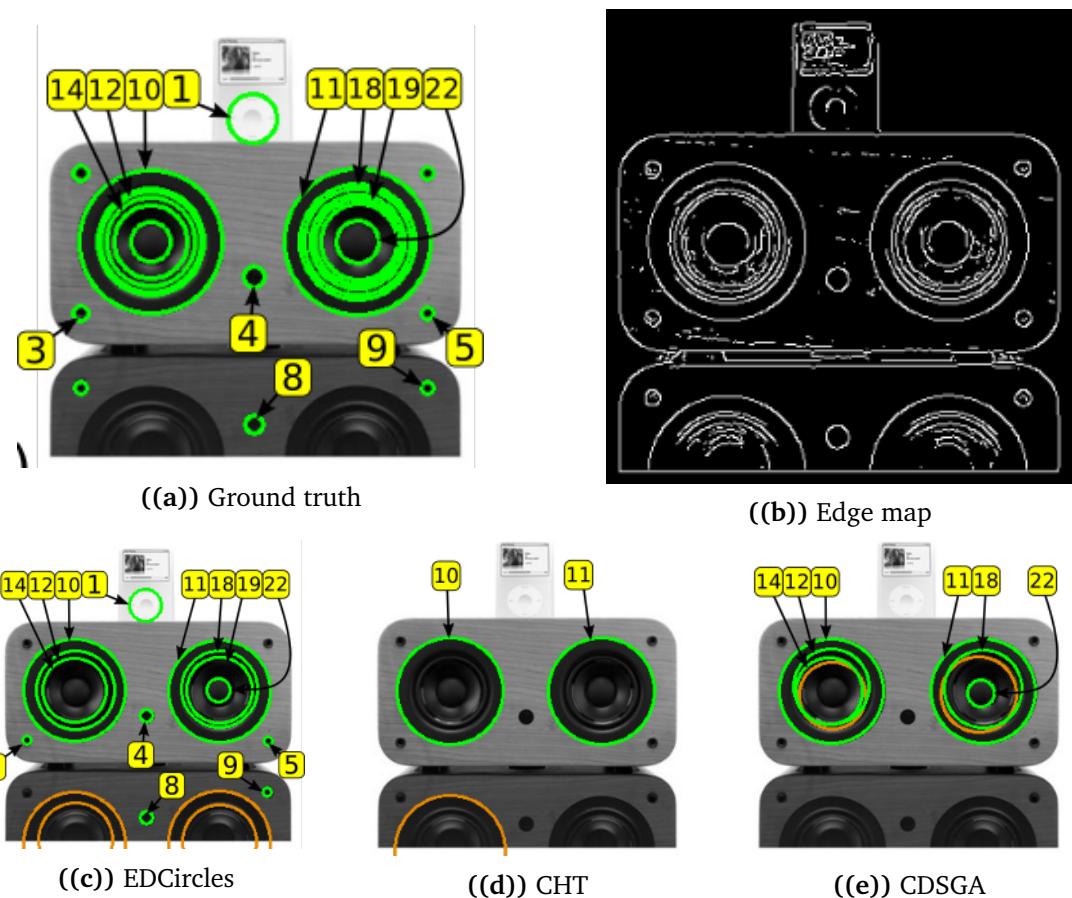


Figure D.15: Figure 15-4 from the paper

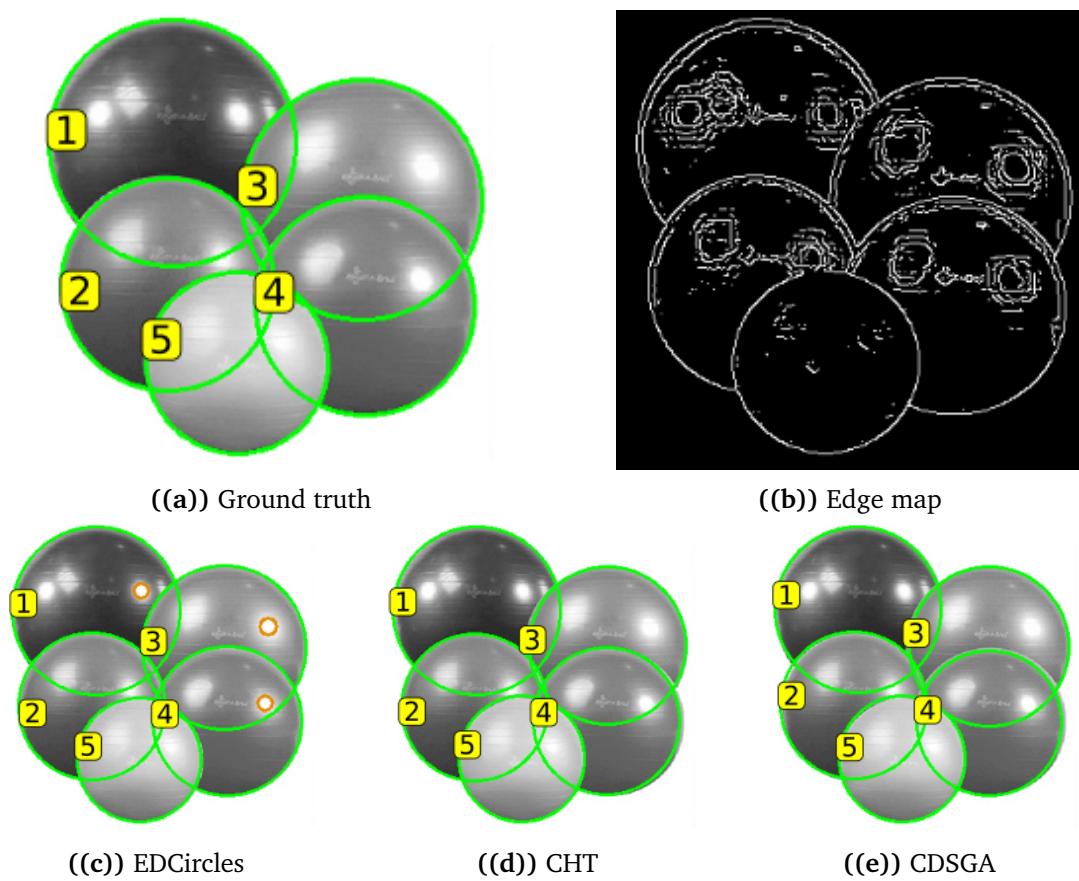


Figure D.16: Figure 15-5 from the paper



Figure D.17: Figure 15-6 from the paper

Image	Circle	EDCircles		CHT		CDSGA	
		Position center	Radius	Position center	Radius	Position center	Radius
Figure D.12	1	0.0059	0.0055	0.0083	0.0151	0.0216	0.0631
	2	0.0065	0.0112	0.0052	0.0173	0.0058	0.0094
	3	0.0063	0.0371	0.0028	0.0239	0.0047	0.0211
	4	0.0085	0.0468	0.0063	0.0449	0.0052	0.0180
	5	0.0169	0.0482	0.0312	0.1008	0.0499	0.2159
	6	0.0416	0.0926	0.0090	0.0012	0.0366	0.0938
	7	0.0100	0.0050	0.0240	0.1743	0.0083	0.0246
Figure D.13	1	0.0032	0.0116	0.0051	0.0168	0.0029	0.0034
	2	0.0024	0.0022			0.0013	0.0046
	3	0.0033	0.0135			0.0045	0.0022
	4	0.0017	0.0073			0.0012	0.0102
	5	0.0053	0.0264			0.0073	0.0009
	6	0.0034	0.0204	0.0057	0.0197	0.0033	0.0324
	7	0.0055	0.0197	0.0037	0.0500	0.0023	0.0138
Figure D.14	8	0.0016	0.0088	0.0012	0.0200	0.0022	0.0060
	1	0.0054	0.0096	0.0102	0.0240	0.0076	0.0116
Figure D.15	2	0.0035	0.0817			0.0266	0.3881
	1	0.0041	0.0282				
	3	0.0029	0.1073				
	4	0.0037	0.0338				
	5	0.0008	0.0859				
	8	0.0040	0.0077				
	9	0.0033	0.1108				
	10	0.0059	0.0072	0.0046	0.0171	0.0046	0.0044
	11	0.0022	0.0037	0.0009	0.0131	0.0019	0.0062
	12	0.0043	0.0312			0.0206	0.0298
	14	0.0116	0.0352			0.0180	0.0435
	18	0.0060	0.0172			0.0067	0.0019
	19	0.0043	0.0084				
	22	0.0062	0.1848			0.0122	0.1013
Figure D.16	1	0.0052	0.0012	0.0084	0.0015	0.0160	0.0109
	2	0.0051	0.0037	0.0022	0.0066	0.0106	0.0024
	3	0.0086	0.0063	0.0164	0.0434	0.0110	0.0267
	4	0.0017	0.0001	0.0066	0.0064	0.0075	0.0220
	5	0.0032	0.0024	0.0053	0.0097	0.0071	0.0166
Figure D.17	1	0.0041	0.0022	0.0023	0.0018	0.0267	0.0567
	2	0.0095	0.0183			0.0108	0.0168
	3	0.0111	0.0598			0.0023	0.0024

Table D.5: Results of all detected circles from real images (external dataset).

E Metrics

We have used the machine learning metrics for the circle detection problem. First, we explain the meaning of each variable for this problem:

1. True positives (TP): the number of correctly detected circles.
2. True negatives (TN): the number of non-circles shapes detected.
3. False positives (FP): the number of incorrectly detected circles.
4. False negatives (FN): the number of not detected circles.

Notices that TN can not be computed, due to circle detection problem only included circular shapes. Metrics used are the following:

$$Accuracy = \frac{TP}{TP + FP + FN} \quad (\text{E.1})$$

$$Precision = \frac{TP}{TP + FP} \quad (\text{E.2})$$

$$Recall = \frac{TP}{TP + FN} \quad (\text{E.3})$$

$$F1 = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} \quad (\text{E.4})$$

$$Specificity = \frac{TN}{TN + FP} \quad (\text{E.5})$$

$$FPR = \frac{FP}{FP + TN} \quad (\text{E.6})$$

$$FNR = \frac{FN}{FN + TP} \quad (\text{E.7})$$

Since TN can not be computed for the circle detection problem, FPR and $Specificity$ were not included.

References

- [1] G. Roth and M. D. Levine. Geometric primitive extraction using a genetic algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):901–905, 1994. pages 3
- [2] S. García, D. Molina, M. Lozano, and F Herrera. A study on the use of non-parametric tests for analyzing the evolutionary algorithms’ behaviour: a case study on the cec’2005 special session on real parameter optimization. *J Heuristics*, 15:617–644, 2009. pages 5