

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1. Hasil Pengumpulan Data

Dataset merupakan gambar dari 3 lahan parkir di *Federal University of Parana*, Curitiba, Brasil. Dataset gambar berjumlah 12.416 gambar tersebut didapatkan dari (Paulo, Luiz, Alceu, & Eunelson, 2015) dengan cara mengunduh melalui website <https://public.roboflow.com/object-detection/pklot>. Pada penelitian ini, data gambar yang digunakan sebesar 502 gambar yang diambil dari salah satu dari tiga lahan parkir dalam dataset. Data gambar yang dipilih adalah gambar dengan kualitas jernih dan pencahayaan yang cukup terang. Salah satu contoh gambar yang tersedia dalam dataset dapat dilihat pada Gambar 4.1.



Gambar 4. 1 Gambar dalam dataset

#### 4.2. Hasil Cropping Data

Data yang sudah ditentukan kemudian dipotong untuk menghilangkan bagian lahan parkir yang terletak cukup jauh dari posisi kamera. Pemotongan gambar diperlukan untuk mengurangi kemungkinan kesalahan dalam pendeteksian objek yang terletak cukup jauh dikarenakan piksel objek yang terlalu kecil. Proses pemotongan gambar dilakukan dengan menggunakan aplikasi Adobe Photoshop CC 2019.



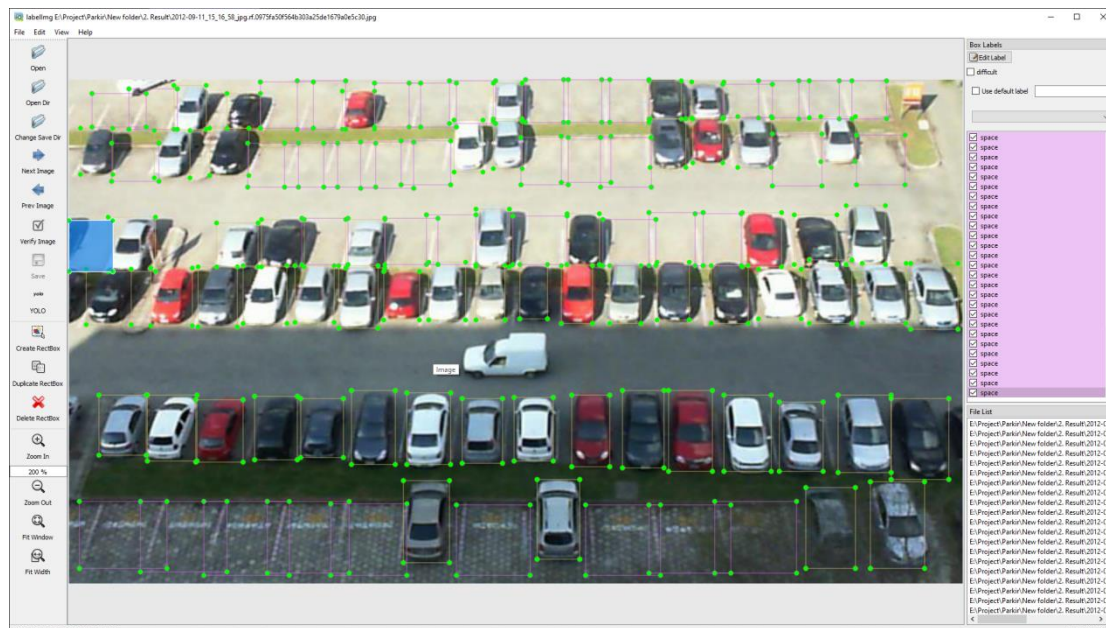
(a)

(b)

Gambar 4. 2 (a) Gambar Asli, (b) Gambar setelah dipotong

### 4.3. Hasil Pelabelan Objek

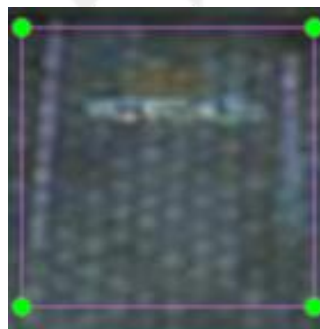
Hasil dari proses *cropping data* kemudian akan diberikan label. Proses pemberian label akan dilakukan untuk setiap objek yang terdapat dalam gambar yang akan digunakan. Label yang diberikan terbagi menjadi dua kelas yaitu “car” dan “space”. Setiap objek dengan kelas “car” akan diberikan kode kelas 0, sedangkan objek dengan kelas “space” akan diberikan kode kelas 1 seperti yang ditunjukkan pada Tabel 4.1. Proses pelabelan objek dilakukan dengan menggunakan aplikasi “LabelImg.py” seperti pada Gambar 4.3.



Gambar 4. 3 Label yang diberikan pada objek

Label diberikan pada setiap objek dengan kriteria-kriteria sebagai berikut:

- Objek “Space” adalah semua lahan parkir yang ada dalam kondisi kosong. Lahan parkir kosong ditandai dengan dua garis parkir dan tidak ada mobil diantara dua garis tersebut. Contoh label pada kelas objek “Space” dapat dilihat pada Gambar 4.4.



Gambar 4. 4 Contoh label kelas "Space"

- Objek “Car” adalah semua mobil yang ada dalam lahan parkir. Mobil yang diberikan label hanya mobil yang berada diantara garis parkir. Contoh label yang diberikan pada kelas objek “Car” dapat dilihat pada Gambar 4.5.



Gambar 4. 5 Contoh label kelas "Car"

- Mobil yang ada diluar lahan parkir tidak akan diberikan label sebagai objek “Car”. Contoh mobil yang tidak termasuk dalam kelas objek “Car” dapat dilihat pada Gambar 4.6.



Gambar 4. 6 Contoh mobil yang tidak diberikan label

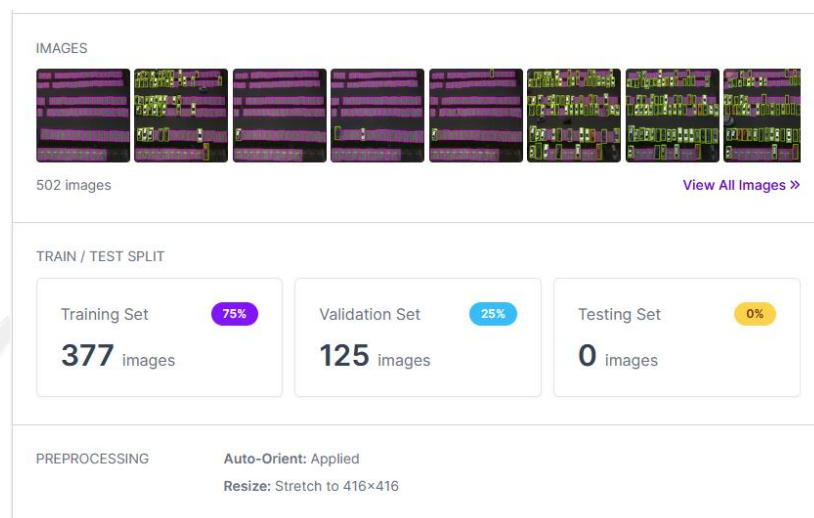
Proses pelabelan objek dilakukan pada setiap gambar dalam dataset yang akan digunakan dalam penelitian. Label yang diberikan pada setiap gambar kemudian disimpan dalam format YOLO. Proses pelabelan objek akan menghasilkan data index berupa kode kelas, koordinat titik pusat label ( $x$ ,  $y$ ), lebar (*width*) dan tinggi (*height*) dari setiap label yang diberikan. Hasil pelabelan objek pada Gambar 4.3 dapat dilihat pada Tabel 4.1.

Tabel 4. 1 Hasil pelabelan objek

Kode Kelas	Titik Koordinat X	Titik Koordinat Y	Width	Height
0	0.953548	0.713303	0.064516	0.160550
0	0.890323	0.702982	0.059355	0.153670
0	0.819355	0.708716	0.049032	0.137615
...				
1	0.025806	0.331422	0.049032	0.098624

#### 4.4. Hasil *Training Dataset*

Setelah mendapatkan data index objek, dataset dibagi untuk melakukan proses *training dataset*. Dataset yang berjumlah 502 data gambar dibagi menjadi 377 gambar (75%) sebagai *training set* dan 125 gambar (25%) sebagai *validation set*. Data dibagi secara acak dengan menggunakan bantuan aplikasi roboflow (<https://app.roboflow.com/>) seperti pada Gambar 4.7.



Gambar 4. 7 Pembagian data menggunakan aplikasi Roboflow

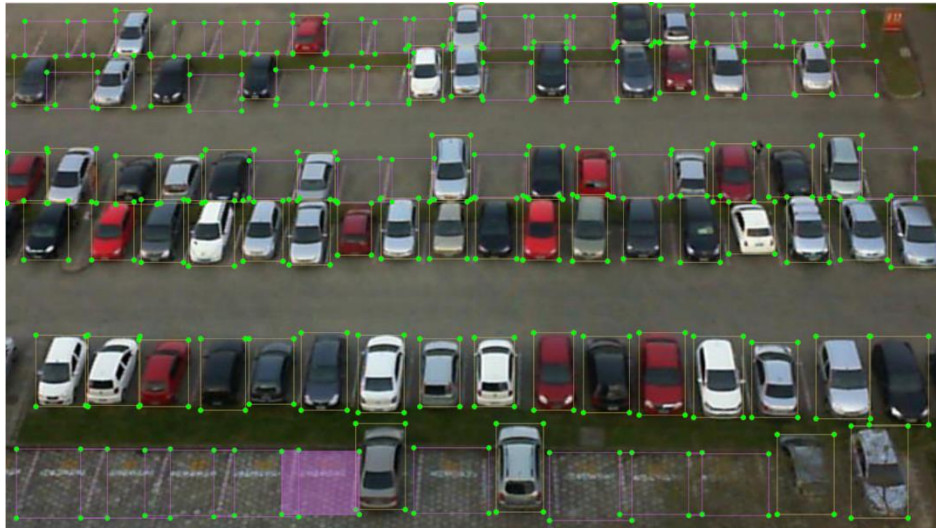
Data *training set* dan *validation set*, yang sudah didapatkan kemudian akan digunakan untuk melakukan proses *training dataset*. Proses *training dataset* dilakukan dengan menggunakan Google Colabs. Google Colabs menyediakan perangkat GPU Nvidia Tesla K80 24GB GDDR5 untuk membantu mempercepat proses *training dataset*. Proses *training dataset* dilakukan dengan menggunakan modul bawaan dari YOLOv5. Proses *training dataset* menghasilkan *dataset model*. *Dataset model* berisi sekumpulan data yang kemudian akan dijadikan acuan untuk melakukan proses deteksi objek.

#### 4.5. Hasil Pengujian

Pada penelitian ini, pengujian dilakukan terhadap salah satu gambar lahan parkir. Pengujian dilakukan untuk menentukan jumlah dan posisi lahan parkir kosong yang tersedia. Pengujian dilakukan pada gambar untuk setiap *dataset model* (*Epoch* 100, *Epoch* 200 dan *Epoch* 300) yang dihasilkan pada proses *training dataset*.

Gambar lahan parkir yang diuji memiliki 39 lahan parkir kosong dan 67 mobil didalamnya. Gambar lahan parkir yang digunakan dalam proses pengujian dapat dilihat pada Gambar 4.8.



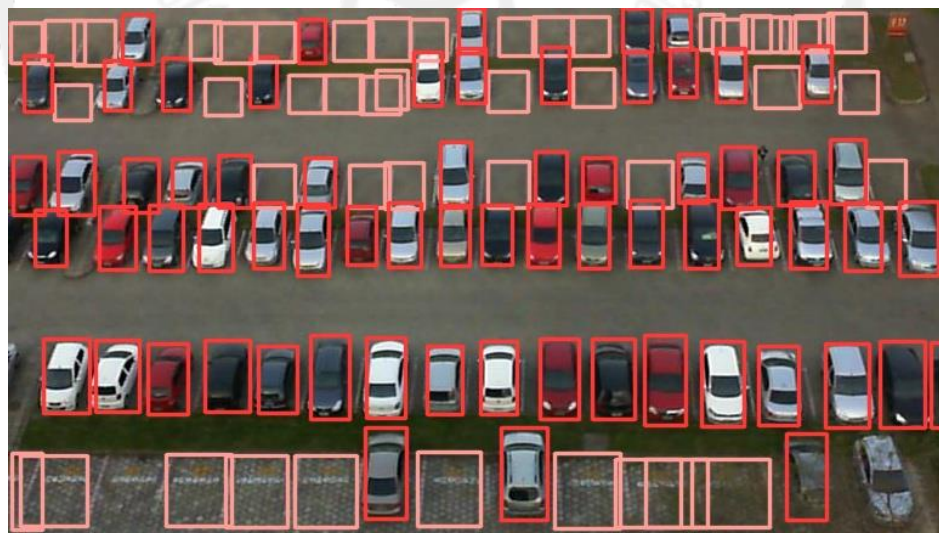


Gambar 4. 8 Gambar yang diuji

Skenario pengujian dilakukan dengan beberapa *confidence threshold* berbeda mulai dari 0.5 sampai 0.75 dengan peningkatan 0.05. Pada bagian ini diperlihatkan hasil analisa dari 3 buah skenario pengujian yaitu:

#### 4.5.1. Pengujian pada skenario *dataset model* dengan 100 epoch dan *confidence threshold* 0.5

Pada bagian ini dijabarkan hasil deteksi objek pada gambar dengan menggunakan *dataset model* yang dilatih dengan 100 epoch sebagai acuan. Sistem akan menampilkan objek yang terdeteksi dengan nilai *confidence* diatas 0.5. Hasil dari deteksi objek pada skenario ini dapat dilihat pada Gambar 4.9



Gambar 4. 9 Hasil deteksi skenario *dataset model* dengan 100 epoch dan *confidence threshold* 0.5

Sebelum menentukan nilai *mAP*, sistem menentukan *intersection over union* (*IoU*) untuk menentukan hasil prediksi yang dilakukan oleh sistem benar (*True Positive*

/ *TP*) atau salah (*False Positive / FP*). Hasil prediksi dan nilai *precision-recall* untuk kelas “Car” pada skenario ini terdapat pada Tabel 4.2.

Tabel 4. 2 Prediksi kelas “Car” skenario dataset model dengan 100 epoch dan confidence threshold 0.5

Bounding Box No	Result	Presicion	Recall
1	<i>TP</i>	1	0,01
2	<i>TP</i>	1	0,03
3	<i>TP</i>	1	0,04
4	<i>TP</i>	1	0,06
5	<i>TP</i>	1	0,07
6	<i>TP</i>	1	0,09
7	<i>TP</i>	1	0,1
8	<i>TP</i>	1	0,12
9	<i>TP</i>	1	0,13
10	<i>TP</i>	1	0,15
11	<i>TP</i>	1	0,16
12	<i>TP</i>	1	0,18
13	<i>TP</i>	1	0,19
14	<i>TP</i>	1	0,21
15	<i>TP</i>	1	0,22
16	<i>TP</i>	1	0,24
17	<i>TP</i>	1	0,25
18	<i>TP</i>	1	0,27
19	<i>TP</i>	1	0,28
20	<i>TP</i>	1	0,3
21	<i>TP</i>	1	0,31
22	<i>TP</i>	1	0,33
23	<i>TP</i>	1	0,34
24	<i>TP</i>	1	0,36
25	<i>TP</i>	1	0,37
26	<i>TP</i>	1	0,39
27	<i>TP</i>	1	0,4
28	<i>TP</i>	1	0,42
29	<i>TP</i>	1	0,43
30	<i>TP</i>	1	0,45
31	<i>TP</i>	1	0,46
32	<i>TP</i>	1	0,48
33	<i>TP</i>	1	0,49
34	<i>TP</i>	1	0,51
35	<i>TP</i>	1	0,52
36	<i>TP</i>	1	0,54
37	<i>TP</i>	1	0,55
38	<i>TP</i>	1	0,57
39	<i>TP</i>	1	0,58
40	<i>TP</i>	1	0,6
41	<i>TP</i>	1	0,61

Bounding Box No	Result	Presicion	Recall
42	<i>TP</i>	1	0,63
43	<i>TP</i>	1	0,64
44	<i>TP</i>	1	0,66
45	<i>TP</i>	1	0,67
46	<i>TP</i>	1	0,69
47	<i>TP</i>	1	0,7
48	<i>TP</i>	1	0,72
49	<i>TP</i>	1	0,73
50	<i>TP</i>	1	0,75
51	<i>TP</i>	1	0,76
52	<i>TP</i>	1	0,78
53	<i>TP</i>	1	0,79
54	<i>TP</i>	1	0,81
55	<i>TP</i>	1	0,82
56	<i>TP</i>	1	0,84
57	<i>TP</i>	1	0,85
58	<i>TP</i>	1	0,87
59	<i>TP</i>	1	0,88
60	<i>TP</i>	1	0,9
61	<i>TP</i>	1	0,91
62	<i>TP</i>	1	0,93
63	<i>TP</i>	1	0,94
64	<i>TP</i>	1	0,96
65	<i>TP</i>	1	0,97
66	<i>TP</i>	1	0,99
67	<i>FP</i>	0,99	0,99

Pada Tabel 4.2 untuk setiap objek dengan kelas “Car” yang terdeteksi oleh sistem, dilakukan perhitungan nilai *precision* dan *recall*. Contoh perhitungan *precision* dan *recall* pada *bounding box* ke-13 yang ditemukan oleh sistem adalah sebagai berikut:

$$Precision_{13} = \frac{\text{Total tp dari 13 bounding box}}{13}$$

$$Precision_{13} = \frac{13}{13}$$

$$Precision_{13} = 1$$

$$Recall_{13} = \frac{\text{Total tp dari 13 bounding box}}{\text{Total objek "Car"}}$$

$$Recall_{13} = \frac{13}{67}$$

$$Recall_{13} = 0,19$$

Sedangkan hasil prediksi dan nilai *precision-recall* untuk kelas “Space” pada skenario ini terdapat pada Tabel 4.3

Tabel 4. 3 Prediksi kelas “Space” skenario dataset model dengan 100 epoch dan confidence threshold 0.5

Bounding Box No	Result	Presicion	Recall
1	TP	1	0,03
2	FP	0,5	0,03
3	TP	0,67	0,05
4	FP	0,5	0,05
5	TP	0,6	0,08
6	FP	0,5	0,08
7	TP	0,57	0,1
8	FP	0,5	0,1
9	TP	0,56	0,13
10	FP	0,5	0,13
11	TP	0,55	0,15
12	FP	0,5	0,15
13	TP	0,54	0,18
14	FP	0,5	0,18
15	TP	0,53	0,21
16	FP	0,5	0,21
17	TP	0,53	0,23
18	FP	0,5	0,23
19	TP	0,53	0,26
20	FP	0,5	0,26
21	TP	0,52	0,28
22	FP	0,5	0,28
23	TP	0,52	0,31
24	FP	0,5	0,31
25	TP	0,52	0,33
26	FP	0,5	0,33

Bounding Box No	Result	Presicion	Recall
27	TP	0,52	0,36
28	FP	0,5	0,36
29	TP	0,52	0,38
30	FP	0,5	0,38
31	TP	0,52	0,41
32	FP	0,5	0,41
33	TP	0,52	0,44
34	FP	0,5	0,44
35	TP	0,51	0,46
36	FP	0,5	0,46
37	TP	0,51	0,49
38	FP	0,5	0,49
39	TP	0,51	0,51
40	FP	0,5	0,51
41	TP	0,51	0,54
42	FP	0,5	0,54
43	TP	0,51	0,56
44	FP	0,5	0,56
45	TP	0,51	0,59
46	FP	0,5	0,59
47	TP	0,51	0,62
48	FP	0,5	0,62
49	TP	0,51	0,64
50	FP	0,5	0,64
51	TP	0,51	0,67
52	FP	0,5	0,67

Tabel 4.3 Lanjutan

Bounding Box No	Result	Presicion	Recall
53	TP	0,51	0,69
53	TP	0,51	0,69
54	FP	0,5	0,69
55	TP	0,51	0,72
56	FP	0,5	0,72
57	TP	0,51	0,74
58	FP	0,5	0,74
59	TP	0,51	0,77
60	FP	0,5	0,77
61	TP	0,51	0,79
62	FP	0,5	0,79
63	TP	0,51	0,82
64	FP	0,5	0,82
65	TP	0,51	0,85
66	FP	0,5	0,85
67	TP	0,51	0,87
68	FP	0,5	0,87
69	TP	0,51	0,9
70	FP	0,5	0,9
71	FP	0,49	0,9

Bounding Box No	Result	Presicion	Recall
72	FP	0,49	0,9
73	TP	0,49	0,92
74	FP	0,49	0,92
75	FP	0,48	0,92
76	FP	0,47	0,92
77	TP	0,48	0,95
78	FP	0,47	0,95
79	FP	0,47	0,95
80	FP	0,46	0,95
81	TP	0,47	0,97
82	FP	0,46	0,97
83	FP	0,46	0,97
84	FP	0,45	0,97
85	FP	0,45	0,97
86	FP	0,44	0,97
87	FP	0,44	0,97
88	FP	0,43	0,97
89	FP	0,43	0,97
90	FP	0,42	0,97

Pada Tabel 4.3 untuk setiap objek dengan kelas “Space” yang terdeteksi oleh sistem, dilakukan perhitungan nilai *precision* dan *recall*. Contoh perhitungan *precision* dan *recall* pada *bounding box* ke-13 yang ditemukan oleh sistem adalah sebagai berikut:

$$Precision_{13} = \frac{\text{Total tp dari 13 bounding box}}{13}$$

$$Precision_{13} = \frac{7}{13}$$

$$Precision_{13} = 0,54$$

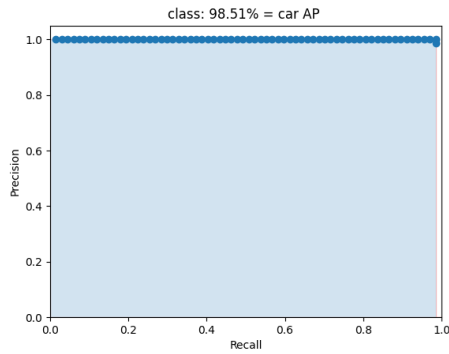
$$Recall_{13} = \frac{\text{Total tp dari 13 bounding box}}{\text{Total objek "Space"}}$$

$$Recall_{13} = \frac{7}{39}$$

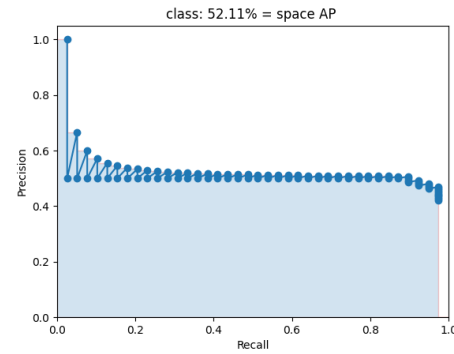
$$Recall_{13} = 0,18$$

Setelah mendapatkan hasil prediksi dan nilai *precision-recall*, sistem membuat kurva *precision-recall* untuk setiap kelas objek yang dideteksi. Hasil kurva *precision-recall* untuk kelas objek “Car” dapat dilihat pada Gambar 4.10 sedangkan untuk kelas objek “Space” pada Gambar 4.11.





Gambar 4. 10 Kurva precision-recall kelas  
"Car"



Gambar 4. 11 Kurva precision-recall kelas  
"Space"

Setelah mendapatkan kurva *precision-recall* untuk semua kelas objek yang dideteksi, sistem dapat menghitung nilai *average precision (AP)*. Nilai AP didapatkan dari rata-rata nilai *interpolated precision (inter\_p)* untuk setiap *recall (r)* dari kurva *precision-recall*. Nilai *interpolated precision* didapatkan dari nilai *precision* tertinggi pada nilai *recall* yang bersangkutan. Perhitungan nilai AP untuk kelas objek "Car" dan kelas objek "Space" adalah sebagai berikut:

$$AP_{Car} = \frac{1}{n} \sum_r P_{inter_p}(r)$$

$$AP_{Car} = \frac{1}{67} * 66$$

$$AP_{Car} = 0,9851 \text{ atau } 98,51\%$$

$$AP_{space} = \frac{1}{n} \sum_r P_{inter_p}(r)$$

$$AP_{space} = \frac{1}{39} * 20,95$$

$$AP_{space} = 0,5211 \text{ atau } 52,11\%$$

Setelah mendapatkan nilai AP untuk setiap kelas objek, ditentukan nilai *mean average precision (mAP)* dengan menghitung rata-rata nilai AP semua kelas objek. Karena terdapat 2 kelas objek pada penelitian maka perhitungan nilai *mAP* adalah sebagai berikut:

$$mAP = \frac{AP_{Car} + AP_{Space}}{2}$$

$$mAP = \frac{98,51+52,11}{2}$$

$$mAP = 75,31\%$$

Dengan demikian didapatkan nilai *mean average precision (mAP)* untuk skenario *dataset model* dengan 100 epoch dan *confidence threshold* 0.5 sebesar 75,31%

#### 4.5.2. Pengujian pada skenario *dataset model* dengan 200 epoch dan *confidence threshold* 0.55

Pada bagian ini dijabarkan hasil deteksi objek pada gambar dengan menggunakan *dataset model* yang dilatih dengan 200 epoch sebagai acuan. Sistem akan menampilkan objek yang terdeteksi dengan nilai *confidence* diatas 0.55. Hasil dari deteksi objek pada skenario ini dapat dilihat pada Gambar 4.12



Gambar 4. 12 Hasil deteksi skenario dataset model dengan 200 epoch dan confidence threshold 0.55

Sebelum menentukan nilai *mAP*, sistem menentukan *intersection over union* (*IoU*) untuk menentukan hasil prediksi yang dilakukan oleh sistem benar (*True Positive / TP*) atau salah (*False Positive / FP*). Hasil prediksi dan nilai *precision-recall* untuk kelas “Car” pada skenario ini terdapat pada Tabel 4.4.

Tabel 4. 4 Prediksi kelas “Car” skenario dataset model  
dengan 200 epoch dan confidence threshold 0.55

Bounding Box No	Result	Presicion	Recall
1	TP	1	0,01
2	TP	1	0,03
3	TP	1	0,04
4	TP	1	0,06
5	TP	1	0,07
6	TP	1	0,09
7	TP	1	0,1
8	TP	1	0,12
9	TP	1	0,13
10	TP	1	0,15
11	TP	1	0,16
12	TP	1	0,18
13	TP	1	0,19
14	TP	1	0,21
15	TP	1	0,22
16	TP	1	0,24
17	TP	1	0,25
18	TP	1	0,27
19	TP	1	0,28
20	TP	1	0,3
21	TP	1	0,31
22	TP	1	0,33
23	TP	1	0,34
24	TP	1	0,36
25	TP	1	0,37
26	TP	1	0,39
27	TP	1	0,4
28	TP	1	0,42
29	TP	1	0,43
30	TP	1	0,45
31	TP	1	0,46
32	TP	1	0,48
33	TP	1	0,49
34	TP	1	0,51
35	TP	1	0,52
36	TP	1	0,54
37	TP	1	0,55
38	TP	1	0,57
39	TP	1	0,58

Bounding Box No	Result	Presicion	Recall
40	TP	1	0,6
41	TP	1	0,61
42	TP	1	0,63
43	TP	1	0,64
44	TP	1	0,66
45	TP	1	0,67
46	TP	1	0,69
47	TP	1	0,7
48	TP	1	0,72
49	TP	1	0,73
50	TP	1	0,75
51	TP	1	0,76
52	TP	1	0,78
53	TP	1	0,79
54	TP	1	0,81
55	TP	1	0,82
56	TP	1	0,84
57	TP	1	0,85
58	TP	1	0,87
59	TP	1	0,88
60	TP	1	0,9
61	TP	1	0,91
62	TP	1	0,93
63	TP	1	0,94
64	TP	1	0,96
65	TP	1	0,97
66	TP	1	0,99
67	TP	1	1
68	FP	0,99	1

Pada Tabel 4.4 untuk setiap objek dengan kelas “Car” yang terdeteksi oleh sistem, dilakukan perhitungan nilai *precision* dan *recall*. Contoh perhitungan *precision* dan *recall* pada *bounding box* ke-20 yang ditemukan oleh sistem adalah sebagai berikut:

$$Precision_{20} = \frac{\text{Total tp dari 20 bounding box}}{20}$$

$$Precision_{20} = \frac{20}{20}$$

$$Precision_{20} = 1$$

$$Recall_{20} = \frac{\text{Total tp dari 20 bounding box}}{\text{Total objek "Car"}}$$

$$Recall_{20} = \frac{20}{67}$$

$$Recall_{20} = 0,3$$

Sedangkan hasil prediksi dan nilai *precision-recall* untuk kelas “Space” pada skenario ini terdapat pada Tabel 4.5.

Tabel 4. 5 Prediksi kelas “Space” skenario dataset model dengan 200 epoch dan confidence threshold 0.55

Bounding Box No	Result	Presicion	Recall	Bounding Box No	Result	Presicion	Recall
1	TP	1	0,03	22	TP	1	0,56
2	TP	1	0,05	23	FP	0,96	0,56
3	TP	1	0,08	24	TP	0,96	0,59
4	TP	1	0,1	25	TP	0,96	0,62
5	TP	1	0,13	26	TP	0,96	0,64
6	TP	1	0,15	27	TP	0,96	0,67
7	TP	1	0,18	28	TP	0,96	0,69
8	TP	1	0,21	29	TP	0,97	0,72
9	TP	1	0,23	30	TP	0,97	0,74
10	TP	1	0,26	31	TP	0,97	0,77
11	TP	1	0,28	32	TP	0,97	0,79
12	TP	1	0,31	33	TP	0,97	0,82
13	TP	1	0,33	34	FP	0,94	0,82
14	TP	1	0,36	35	TP	0,94	0,85
15	TP	1	0,38	36	TP	0,94	0,87
16	TP	1	0,41	37	FP	0,92	0,87
17	TP	1	0,44	38	TP	0,92	0,9
18	TP	1	0,46	39	TP	0,92	0,92
19	TP	1	0,49	40	FP	0,9	0,92
20	TP	1	0,51	41	TP	0,9	0,92
21	TP	1	0,54				



Pada Tabel 4.5 untuk setiap objek dengan kelas “Space” yang terdeteksi oleh sistem, dilakukan perhitungan nilai *precision* dan *recall*. Contoh perhitungan *precision* dan *recall* pada *bounding box* ke-20 yang ditemukan oleh sistem adalah sebagai berikut:

$$Precision_{20} = \frac{\text{Total tp dari 20 bounding box}}{20}$$

$$Precision_{20} = \frac{20}{20}$$

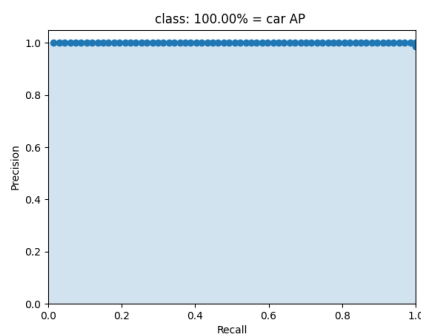
$$Precision_{20} = 1$$

$$Recall_{20} = \frac{\text{Total tp dari 20 bounding box}}{\text{Total objek "Space"}}$$

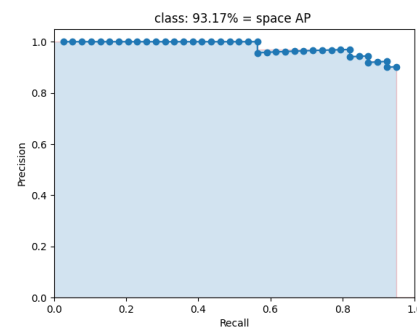
$$Recall_{20} = \frac{20}{39}$$

$$Recall_{20} = 0,51$$

Setelah mendapatkan hasil prediksi dan nilai *precision-recall*, sistem membuat kurva *precision-recall* untuk setiap kelas objek yang dideteksi. Hasil kurva *precision-recall* untuk kelas objek “Car” dapat dilihat pada Gambar 4.13 sedangkan untuk kelas objek “Space” pada Gambar 4.14.



Gambar 4. 13 Kurva precision-recall kelas "Car"



Gambar 4. 14 Kurva precision-recall kelas "Space"

Setelah mendapatkan kurva *precision-recall* untuk semua kelas objek yang dideteksi, sistem dapat menghitung nilai *average precision (AP)*. Nilai AP didapatkan dari rata-rata nilai *interpolated precision (inter\_p)* untuk setiap *recall (r)* dari kurva *precision-recall*. Nilai *interpolated precision* didapatkan dari nilai *precision* tertinggi pada nilai *recall* yang bersangkutan. Perhitungan nilai AP untuk kelas objek “Car” dan kelas objek “Space” adalah sebagai berikut:

$$AP_{Car} = \frac{1}{n} \sum_r P_{inter_p}(r)$$

$$AP_{Car} = \frac{1}{67} * 67$$

$$AP_{Car} = 1 \text{ atau } 100\%$$

$$AP_{space} = \frac{1}{n} \sum_r P_{interp}(r)$$

$$AP_{space} = \frac{1}{39} * 36,33$$

$$AP_{space} = 0,9317 \text{ atau } 93,17\%$$

Setelah mendapatkan nilai *AP* untuk setiap kelas objek, ditentukan nilai *mean average precision (mAP)* dengan menghitung rata-rata nilai *AP* semua kelas objek. Karena terdapat 2 kelas objek pada penelitian maka perhitungan nilai *mAP* adalah sebagai berikut:

$$mAP = \frac{AP_{car} + AP_{space}}{2}$$

$$mAP = \frac{100 + 93,17}{2}$$

$$mAP = 96,58\%$$

Dengan demikian didapatkan nilai *mean average precision (mAP)* untuk skenario *dataset model* dengan 200 epoch dan *confidence threshold* 0.55 sebesar 96,58%

#### 4.5.3. Pengujian pada skenario *dataset model* dengan 300 epoch dan *confidence threshold* 0.6

Pada bagian ini dijabarkan hasil deteksi objek pada gambar dengan menggunakan *dataset model* yang dilatih dengan 300 epoch sebagai acuan. Sistem akan menampilkan objek yang terdeteksi dengan nilai *confidence* diatas 0.6. Hasil dari deteksi objek pada skenario ini dapat dilihat pada Gambar 4.15



Gambar 4. 15 Hasil deteksi skenario *dataset model* dengan 300 epoch dan *confidence threshold* 0.6

Sebelum menentukan nilai *mAP*, sistem menentukan *intersection over union (IoU)* untuk menentukan hasil prediksi yang dilakukan oleh sistem benar (*True Positive*)

/ *TP*) atau salah (*False Positive / FP*). Hasil prediksi dan nilai *precision-recall* untuk kelas “Car” pada skenario ini terdapat pada Tabel 4.6.

Tabel 4. 6 Prediksi kelas “Car” skenario dataset model  
dengan 300 epoch dan confidence threshold 0.6

Bounding Box No	Result	Presicion	Recall
1	<i>TP</i>	1	0,01
2	<i>TP</i>	1	0,03
3	<i>TP</i>	1	0,04
4	<i>TP</i>	1	0,06
5	<i>TP</i>	1	0,07
6	<i>TP</i>	1	0,09
7	<i>TP</i>	1	0,1
8	<i>TP</i>	1	0,12
9	<i>TP</i>	1	0,13
10	<i>TP</i>	1	0,15
11	<i>TP</i>	1	0,16
12	<i>TP</i>	1	0,18
13	<i>TP</i>	1	0,19
14	<i>TP</i>	1	0,21
15	<i>TP</i>	1	0,22
16	<i>TP</i>	1	0,24
17	<i>TP</i>	1	0,25
18	<i>TP</i>	1	0,27
19	<i>TP</i>	1	0,28
20	<i>TP</i>	1	0,3
21	<i>TP</i>	1	0,31
22	<i>TP</i>	1	0,33
23	<i>TP</i>	1	0,34
24	<i>TP</i>	1	0,36
25	<i>TP</i>	1	0,37
26	<i>TP</i>	1	0,39
27	<i>TP</i>	1	0,4
28	<i>TP</i>	1	0,42
29	<i>TP</i>	1	0,43
30	<i>TP</i>	1	0,45
31	<i>TP</i>	1	0,46
32	<i>TP</i>	1	0,48
33	<i>TP</i>	1	0,49

Bounding Box No	Result	Presicion	Recall
34	<i>TP</i>	1	0,51
35	<i>TP</i>	1	0,52
36	<i>TP</i>	1	0,54
37	<i>TP</i>	1	0,55
38	<i>TP</i>	1	0,57
39	<i>TP</i>	1	0,58
40	<i>TP</i>	1	0,6
41	<i>TP</i>	1	0,61
42	<i>TP</i>	1	0,63
43	<i>TP</i>	1	0,64
44	<i>TP</i>	1	0,66
45	<i>TP</i>	1	0,67
46	<i>TP</i>	1	0,69
47	<i>TP</i>	1	0,7
48	<i>TP</i>	1	0,72
49	<i>TP</i>	1	0,73
50	<i>TP</i>	1	0,75
51	<i>TP</i>	1	0,76
52	<i>TP</i>	1	0,78
53	<i>TP</i>	1	0,79
54	<i>TP</i>	1	0,81
55	<i>TP</i>	1	0,82
56	<i>TP</i>	1	0,84
57	<i>TP</i>	1	0,85
58	<i>TP</i>	1	0,87
59	<i>TP</i>	1	0,88
60	<i>TP</i>	1	0,9
61	<i>TP</i>	1	0,91
62	<i>TP</i>	1	0,93
63	<i>TP</i>	1	0,94
64	<i>TP</i>	1	0,96
65	<i>TP</i>	1	0,97
66	<i>TP</i>	1	0,99

Pada Tabel 4.6 untuk setiap objek dengan kelas “Car” yang terdeteksi oleh sistem, dilakukan perhitungan nilai *precision* dan *recall*. Contoh perhitungan *precision* dan *recall* pada *bounding box* ke-30 yang ditemukan oleh sistem adalah sebagai berikut:

$$Precision_{30} = \frac{\text{Total tp dari 30 bounding box}}{30}$$

$$Precision_{30} = \frac{30}{30}$$

$$Precision_{30} = 1$$

$$Recall_{30} = \frac{\text{Total tp dari 30 bounding box}}{\text{Total objek "Car"}}$$

$$Recall_{30} = \frac{30}{67}$$

$$Recall_{30} = 0,45$$

Sedangkan hasil prediksi dan nilai *precision-recall* untuk kelas “Space” pada skenario ini terdapat pada Tabel 4.7

Tabel 4. 7 Prediksi kelas “Space” skenario dataset model dengan 300 epoch dan confidence threshold 0.6

Bounding Box No	Result	Presicion	Recall
1	TP	1	0,03
2	TP	1	0,05
3	TP	1	0,08
4	TP	1	0,1
5	TP	1	0,13
6	TP	1	0,15
7	TP	1	0,18
8	TP	1	0,21
9	TP	1	0,23
10	TP	1	0,26
11	TP	1	0,28
12	TP	1	0,31
13	TP	1	0,33
14	TP	1	0,36
15	TP	1	0,38
16	TP	1	0,41
17	TP	1	0,44
18	TP	1	0,46
19	TP	1	0,49
20	TP	1	0,51

Bounding Box No	Result	Presicion	Recall
21	TP	1	0,54
22	TP	1	0,56
23	TP	1	0,59
24	TP	1	0,62
25	TP	1	0,64
26	TP	1	0,67
27	TP	1	0,69
28	TP	1	0,72
29	TP	1	0,74
30	TP	1	0,77
31	TP	1	0,79
32	TP	1	0,82
33	TP	1	0,85
34	TP	1	0,87
35	TP	1	0,9
36	TP	1	0,92
37	TP	1	0,95
38	TP	1	0,97
39	TP	1	1

Pada Tabel 4.7 untuk setiap objek dengan kelas “Space” yang terdeteksi oleh sistem, dilakukan perhitungan nilai *precision* dan *recall*. Contoh perhitungan *precision* dan *recall* pada *bounding box* ke-30 yang ditemukan oleh sistem adalah sebagai berikut:



$$Precision_{30} = \frac{\text{Total tp dari 30 bounding box}}{30}$$

$$Precision_{30} = \frac{30}{30}$$

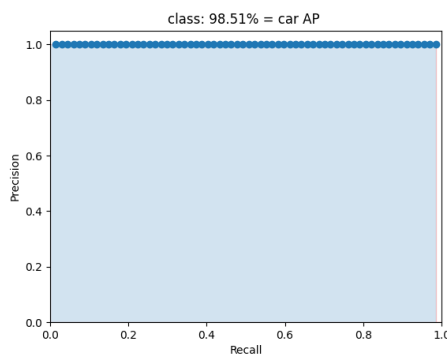
$$Precision_{30} = 1$$

$$Recall_{30} = \frac{\text{Total tp dari 30 bounding box}}{\text{Total objek "Space"}}$$

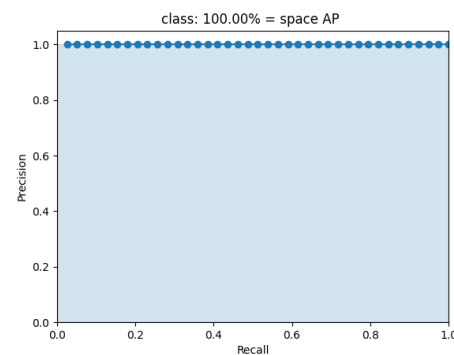
$$Recall_{30} = \frac{30}{39}$$

$$Recall_{30} = 0,77$$

Setelah mendapatkan hasil prediksi dan nilai *precision-recall*, sistem membuat kurva *precision-recall* untuk setiap kelas objek yang dideteksi. Hasil kurva *precision-recall* untuk kelas objek “Car” dapat dilihat pada Gambar 4.16 sedangkan untuk kelas objek “Space” pada Gambar 4.17



Gambar 4. 16 Kurva precision-recall kelas "Car"



Gambar 4. 17 Kurva precision-recall kelas "Space"

Setelah mendapatkan kurva *precision-recall* untuk semua kelas objek yang dideteksi, sistem dapat menghitung nilai *average precision (AP)*. Nilai AP didapatkan dari rata-rata nilai *interpolated precision (inter\_p)* untuk setiap *recall (r)* dari kurva *precision-recall*. Nilai *interpolated precision* didapatkan dari nilai *precision* tertinggi pada nilai *recall* yang bersangkutan. Perhitungan nilai AP untuk kelas objek “Car” dan kelas objek “Space” adalah sebagai berikut:

$$AP_{Car} = \frac{1}{n} \sum_r P_{inter_p}(r)$$

$$AP_{Car} = \frac{1}{67} * 66$$

$$AP_{Car} = 0,9851 \text{ atau } 98,51\%$$

$$AP_{space} = \frac{1}{n} \sum_r P_{interp}(r)$$

$$AP_{space} = \frac{1}{39} * 39$$

$$AP_{space} = 1 \text{ atau } 100\%$$

Setelah mendapatkan nilai *AP* untuk setiap kelas objek, ditentukan nilai *mean average precision (mAP)* dengan menghitung rata-rata nilai *AP* semua kelas objek. Karena terdapat 2 kelas objek pada penelitian maka perhitungan nilai *mAP* adalah sebagai berikut:

$$mAP = \frac{AP_{car} + AP_{Space}}{2}$$

$$mAP = \frac{98,51 + 100}{2}$$

$$mAP = 99,25\%$$

Dengan demikian didapatkan nilai *mean average precision (mAP)* untuk skenario *dataset model* dengan 300 epoch dan *confidence threshold* 0.6 sebesar 99,25%. Hasil perhitungan nilai *average precision (AP)* pada seluruh skenario diperlihatkan pada Tabel 4.8 (kelas “Car”) dan Tabel 4.9 (Kelas “Space”)

Tabel 4. 8 AP kelas "Car"

Confidenc e	Epoch		
	100	200	300
0,5	98,51	100	98,51
0,55	98,51	100	98,51
0,6	98,51	100	98,51
0,65	98,51	52,3	98,51
0,7	98,51	100	98,51
0,75	95,52	97,01	98,51

Tabel 4. 9 AP kelas "Space"

Confidence	Epoch		
	100	200	300
0,5	52,11	93,17	97,44
0,55	97,05	93,17	100
0,6	94,67	93,17	100
0,65	92,24	48,92	97,44
0,7	84,62	86,12	92,31
0,75	61,54	78,77	82,05

Untuk hasil perhitungan *mean average precision (mAP)* pada seluruh skenario pengujian dapat dilihat pada Tabel 4.10

Tabel 4. 10 nilai mAP

Confidence	Epoch		
	100	200	300
0,5	75,31	96,58	97,97
0,55	97,78	96,58	99,25
0,6	96,59	96,58	99,25
0,65	95,37	50,61	97,97
0,7	91,56	93,06	95,41
0,75	78,53	87,89	90,28

Berdasarkan nilai *mAP* yang didapatkan, hasil prediksi terbaik didapatkan pada skenario *dataset model* dengan 300 epoch dan *confidence threshold* 0,55 dan 0,6 dengan nilai *mAP* sebesar 99,25%

#### 4.6. Evaluasi

Hasil pada proses pengujian kemudian dievaluasi untuk menentukan ketepatan akurasi deteksi objek. Pada bagian ini diperlihatkan hasil evaluasi pada 3 skenario yaitu:

##### 4.6.1. Evaluasi hasil prediksi pada skenario *dataset model* dengan 100 epoch dan *confidence threshold* 0.5

Berdasarkan hasil prediksi deteksi objek yang ditampilkan pada Gambar 4.9 terdapat beberapa kesalahan deteksi objek. Pada skenario ini, sistem dapat mendeteksi 67 mobil dan 45 lahan parkir kosong. Dari 67 objek mobil yang terdeteksi, terdapat 1 objek bukan mobil yang terdeteksi sebagai mobil dan 1 objek mobil yang tidak dapat dideteksi. Kesalahan hasil prediksi objek mobil pada skenario ini dapat dilihat pada Gambar 4.18.



Gambar 4. 18 Kesalahan deteksi objek “Car” pada skenario dataset model dengan 100 epoch dan confidence threshold 0.5

(hijau) salah deteksi dan (kuning) tidak terdeteksi

Sedangkan dari 45 lahan parkir kosong yang terdeteksi terdapat 7 kesalahan deteksi dan 1 lahan parkir kosong yang tidak dapat dideteksi. Kesalahan hasil deteksi lahan parkir kosong dapat dilihat pada Gambar 4.19



Gambar 4. 19 Kesalahan deteksi objek “Space” pada skenario dataset model dengan 100 epoch dan confidence threshold 0.5

(hijau) salah deteksi dan (kuning) tidak terdeteksi

Berdasarkan analisa hasil deteksi pada Gambar 4.18 dan Gambar 4.19, sistem dapat mendeteksi 66 mobil dan 33 lahan parkir secara tepat.



#### 4.6.2. Evaluasi hasil prediksi pada skenario *dataset model* dengan 200 epoch dan *confidence threshold 0.55*

Berdasarkan hasil prediksi deteksi objek yang ditampilkan pada Gambar 4.12 terdapat beberapa kesalahan deteksi objek. Pada skenario ini, sistem dapat mendeteksi 68 mobil dan 41 lahan parkir kosong. Dari 68 objek mobil yang terdeteksi, sistem dapat mendeteksi seluruh mobil yang ada namun terdapat 1 objek bukan mobil yang terdeteksi sebagai mobil. Kesalahan hasil prediksi objek mobil pada skenario ini dapat dilihat pada Gambar 4.20



Gambar 4. 20 Kesalahan deteksi objek “Car” pada skenario dataset model dengan 200 epoch dan confidence threshold 0.55  
(hijau) salah deteksi

Sedangkan dari 41 lahan parkir kosong yang terdeteksi terdapat 3 kesalahan deteksi dan 1 lahan parkir kosong yang tidak dapat dideteksi. Kesalahan hasil deteksi lahan parkir kosong dapat dilihat pada Gambar 4.21



Gambar 4. 21 Kesalahan deteksi objek “Space” pada skenario dataset model dengan 200 epoch dan confidence threshold 0.55 (hijau) salah deteksi dan (kuning) tidak terdeteksi

Berdasarkan analisa hasil deteksi pada Gambar 4.20 dan Gambar 4.21, sistem dapat mendeteksi 67 mobil dan 37 lahan parkir secara tepat.

#### 4.6.3. Evaluasi hasil prediksi pada skenario *dataset model* dengan 300 epoch dan *confidence threshold 0.6*

Berdasarkan hasil prediksi deteksi objek yang ditampilkan pada Gambar 4.15 terdapat beberapa kesalahan deteksi objek. Pada skenario ini, sistem dapat mendeteksi 66 mobil dan 39 lahan parkir kosong. Dari 66 objek mobil yang terdeteksi, sistem gagal mendeteksi 1 objek mobil. Kesalahan hasil prediksi objek mobil pada skenario ini dapat dilihat pada Gambar 4.22



Gambar 4. 22 Kesalahan deteksi objek “Car” pada skenario dataset model dengan 300 epoch dan confidence threshold 0.6 (kuning) tidak terdeteksi

Sedangkan pada kelas objek “Space”, sistem dapat mendeteksi seluruh lahan parkir kosong tanpa ada kesalahan sama sekali. Dengan demikian, sistem dapat mendeteksi 66 mobil dan 39 lahan parkir secara tepat. Hasil pengujian pada skenario yang lain dapat dilihat pada bagian Lampiran. Jumlah objek yang terdeteksi benar dan nilai akurasi pada seluruh skenario dapat dilihat pada Tabel 4.11.

Tabel 4. 11 Akurasi objek yang terdeteksi

Epoch	Confidence Threshold	Mobil Terdeteksi	Lahan Parkir Terdeteksi	Mobil Terdeteksi Benar	Lahan Parkir Terdeteksi Benar	Kesalahan Deteksi	Akurasi (%)
100	50	67	45	66	33	15	86,607
	55	67	42	66	35	10	90,826
	60	67	40	66	35	9	91,589
	65	66	38	66	34	8	92,308
	70	66	33	66	33	7	92,929
	75	64	24	64	24	18	79,545
200	50	68	41	67	36	7	93,578
	55	68	41	67	37	6	94,495
	60	68	41	67	37	6	94,495
	65	67	40	67	36	6	94,393
	70	67	37	67	35	6	94,231
	75	65	32	65	32	9	90,722
300	50	68	40	67	37	4	96,296
	55	66	39	66	39	1	99,048
	60	66	39	66	39	1	99,048
	65	66	38	66	38	2	98,077
	70	66	36	66	36	4	96,078
	75	66	32	66	32	8	91,837

Dari Tabel 4.11 menunjukkan bahwa hasil deteksi terbaik terjadi pada skenario *dataset model* dengan 300 epoch dan *confidence threshold* 0,55 dan 0,6 dengan nilai akurasi sebesar 99,048%.