

Pendeteksi Mobil Berdasarkan Merek dan Tipe Menggunakan Algoritma YOLO

Car Detector by Brand and Type Using the YOLO Algorithm

Al Riansyah^{1*}
A. Haidar Mirza²

^{1,2}Teknik Informatika, Universitas Bina Darma, Indonesia
¹alriansyah1218@gmail.com, ²haidar.mirza06@gmail.com

***Penulis Korespondensi:**
Al Riansyah
alriansyah1218@gmail.com

Riwayat Artikel:

Diterima : 15 Agustus 2022
Direview : 22 Agustus 2022
Disetujui : 26 September 2022
Terbit : 30 Juni 2023

Abstrak

Sistem pemerintah di suatu wilayah atau kota memiliki peraturan dalam mengatur aktivitas penduduknya, terutama yang berkaitan berkendara di jalan raya. Pelanggaran lalulintas kerap terjadi di jalan perkotaan maupun jalan raya. Hal ini dapat memicu kecelakaan akibat melanggar peraturan lalulintas. Hal tersebut mendorong pemerintah untuk mengambil tidak tegas bagi para pengendara yang melanggar peraturan. Maka dari itu diperlukan sebuah sistem yang mampu membantu memantau kondisi lalu lintas. Sehingga penelitian ini bertujuan untuk membuat sistem yang mampu mendeteksi kendaraan mobil berdasarkan merek dan tipenya dengan tingkat akurasi pendeteksian yang tinggi, sehingga dapat mempermudah dalam mengenali objek mobil yang ada di sekitar. Sistem pendeteksi akan dikembangkan menggunakan Algoritma YOLO (You Only Look Once), karena YOLO adalah salah satu metode paling cepat dan akurat dalam pendeteksian objek bahkan mampu melebihi hingga 2 kali kemampuan algoritma lain. Algoritma YOLO (You Only Look Once) merupakan arsitektur dari deep learning dan sebuah algoritma yang dikembangkan untuk mendeteksi sebuah objek secara real-time. Pendeteksian dilakukan pada gambar, dan ketika akses webcam laptop yang didalamnya terdapat objek mobil menggunakan model dari dataset yang telah di training menggunakan framework Darknet. Objek mobil yang terdeteksi akan terdapat *bounding box* di area objek tersebut, dan terdapat keterangan nama mobil dan tipe serta tahun produksinya pada area *bounding box* tersebut. Berdasarkan pengujian kinerja klasifikasi dari data yang telah dilakukan training menunjukkan bahwa nilai akurasi mencapai 92% sehingga dapat disimpulkan bahwa sistem dapat bekerja dengan baik.

Kata Kunci: Algoritma YOLO, Darknet, Deep Learning, Pendeteksi objek

Abstract

The government system in an area or city has regulations in regulating the activities of its residents, especially those related to driving on the highway. Traffic violations often occur on urban roads and highways, this can trigger accidents due to violating traffic regulations. This has prompted the government to take firm measures against motorists who violate regulations. Therefore we need a system that can help monitor traffic conditions. So, the aims this research is create a system that is able to detect car vehicles based on the brand and type with a high level of detection accuracy, so that it can make it easier to recognize car objects around. The detection system will be developed using the YOLO (You Only Look Once) Algorithm, because YOLO is one of the fastest and most accurate methods of object detection and is even capable of exceeding 2 times the capabilities of other algorithms. The YOLO (You Only Look Once) algorithm is an architecture of Deep Learning and an algorithm developed to detect an object in real-time. Detection is

carried out on the image, and when accessing a laptop webcam, which contains a car object, uses a model from a dataset that has been trained using the Darknet framework. The detected car object will have a bounding box in the object area, and there will be a description of the car name and type and year of production in the bounding box area. Based on the classification performance test of the data that has been trained, it shows that the accuracy value reaches 92% so it can be concluded that the system can work well.

Keywords: *YOLO Algorithm, Darknet, Deep Learning, Object detection*

1. Pendahuluan

Sistem pemerintah di suatu wilayah atau kota memiliki peraturan dalam mengatur aktivitas penduduknya, terutama yang berkaitan berkendara di jalan raya. Pelanggaran lalulintas kerap terjadi di jalan perkotaan maupun jalan raya, hal ini dapat memicu kecelakaan akibat melanggar peraturan lalulintas. Hal tersebut mendorong pemerintah untuk mengambil tidak tegas bagi para pengendara yang melanggar peraturan. Oleh karena itu, diperlukan adanya sebuah sistem yang dapat membantu memantau kondisi lalu lintas[1].

Perkembangan teknologi saat ini semakin pesat, banyak sekali teknologi baru dan canggih di berbagai fasilitas. Perkembangan sistem identifikasi di lingkungan masyarakat atau instansi sangat pesat salah satunya adalah sistem pengenalan citra digital mobil dengan kecepatan dan akurasi yang tinggi. Setiap mobil di dunia memiliki bentuk bodi yang berbeda-beda. Oleh sebab itu, citra bentuk mobil digunakan sebagai penanda agar mudah dikenali oleh sistem.

Banyaknya merek dan tipe mobil membuat pengawasan lalulintas kesulitan untuk mengenali merek dan tipenya. Menanggapi permasalahan tersebut penulis memiliki tujuan untuk membuat sistem yang mampu mendeteksi kendaraan mobil berdasarkan merek dan tipenya dengan tingkat akurasi pendeteksian yang tinggi, sehingga dapat mempermudah dalam mengenali objek mobil yang ada di sekitar. Setiap kendaraan mobil memiliki bentuk citra yang berbeda. Maka dari itu citra mobil tersebut digunakan sebagai pengenalan data yang nantinya akan diolah menjadi dataset.

Sistem pendeteksi akan dibuat menggunakan Algoritma YOLO (You Only Look Once), karena YOLO adalah salah satu metode pendeteksian objek yang paling cepat dan akurat, bahkan memiliki kemampuan yang dapat mengungguli algoritma lain hingga 2 kali lipat[2]. Algoritma YOLO (You Only Look Once) merupakan arsitektur dari Deep Learning[3].

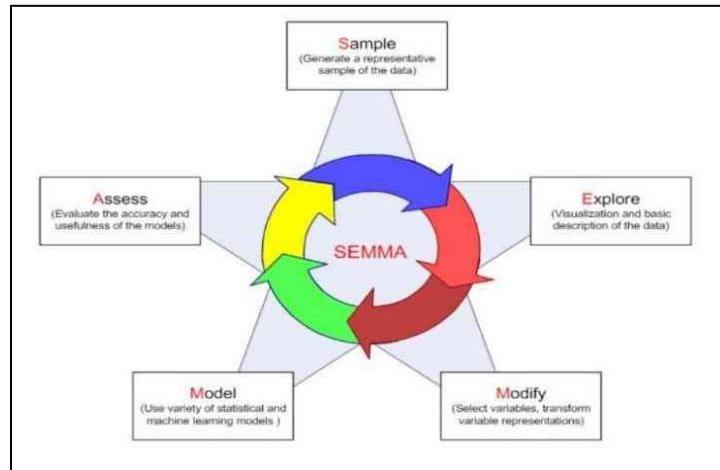
You Only Look Once (YOLO) adalah algoritma yang dikembangkan untuk mendeteksi objek secara real time. Sistem dijalankan menggunakan pengklasifikasi atau pencari lokasi yang diatur ulang untuk mendeteksi[4].

2. Metode Penelitian

Dalam penelitian ini, metode yang digunakan adalah SEMMA. Metode ini digunakan sebagai pengembangan sistem dan metode observasi selama proses pengumpulan data. SEMMA adalah singkatan dari Sample, Explore, Modify, Model, dan Assess. Akronim yang mengacu pada proses menjalankan objek data mining[5]. Adapun tahapan dalam penelitian ini antara lain sebagai berikut :

Pengumpulan data (Sample)

Pada fase ini dilakukan proses pengumpulan data, alat, dan bahan yang diperlukan untuk penelitian. Pengumpulan data dilakukan dengan menggunakan studi pustaka[6], dimana dalam hal ini peneliti melakukan pencarian data dari sumber internet. Data yang digunakan dalam penelitian ini adalah data mobil dengan tampilan sisi depan, sisi samping kanan dan kiri serta sisi belakang. Data masukan dalam penelitian ini adalah gambar dalam format .jpg.



Gambar 1. Tahapan metode SEMMA.

(Sumber: <https://3.bp.blogspot.com/-5kZcCc3J3Ko/UhIqDpX7W11/AAAAAAAAACzA/bdh7uzErRHQ/s1600/SEMMA.JPG>)

Deskripsi data (Explore)

Pada fase ini, pencarian data dilakukan dengan mencari di situs market yaitu *website olx* dan *facebook marketplace*. Spesifikasi data akan menjelaskan gambaran tentang informasi data yang akan digunakan. Visualisasi data akan menampilkan informasi dari data secara visual[7].

Transformasi data (Modify)

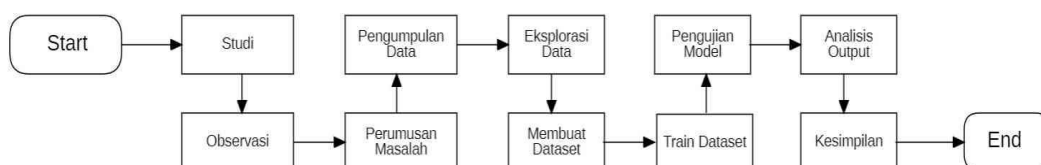
Pada fase ini dilakukan proses memodifikasi data dengan mendapatkan, menyeleksi dan mentransformasi variabel-variabel untuk fokus pada proses pemilihan model.

Pemodelan data (Model)

Data yang telah dimodifikasi akan di lakukan proses pemodelan data dengan suatu perangkat lunak untuk secara otomatis mencari kombinasi data yang dapat diandalkan untuk memprediksi hasil yang diinginkan. Data yang ada akan dilakukan proses anotasi. Proses anotasi dilakukan dengan cara memberi label pada gambar dengan menambahkan kotak pembatas. Selanjutnya melakukan pelatihan data, yang bertujuan untuk melatih komputer dengan mengolah data beranotasi tersebut menjadi suatu model pendeteksian untuk mencapai prediksi. Proses selanjutnya yaitu mengimplementasikan algoritma You Look Only Once (YOLO) untuk pendeteksian. Implementasi dilakukan dengan mengimplementasikan model pelatihan pada perangkat lunak menggunakan bahasa pemrograman Python[8].

Evaluasi data (Assess)

Pada tahap ini akan dilakukan evaluasi terhadap pemodelan yang telah dibuat. Pengujian dilakukan secara real time menggunakan webcam dan pengujian pada gambar. Pengujian dilakukan untuk menentukan seberapa akurat subjek dengan model yang baru dilatih[9]. Peneliti akan mendapatkan hasil berdasarkan data yang sudah diproses pada tahapan pertama pengumpulan data sampai pada tahap pemodelan, hasil akhir mengenai tingkat accuracy, precision dan confusion matrix.



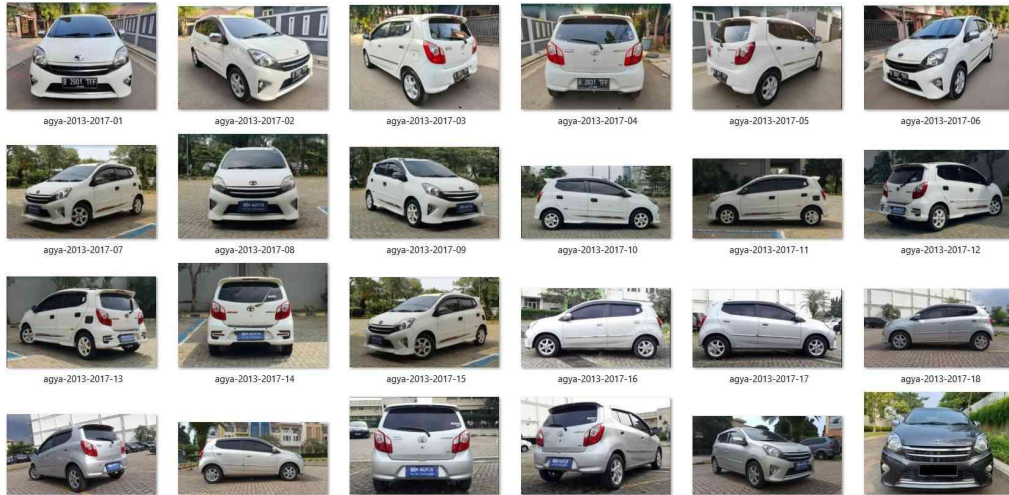
Gambar 2. Alur Penelitian

3. Hasil dan Pembahasan

Hasil

Sample

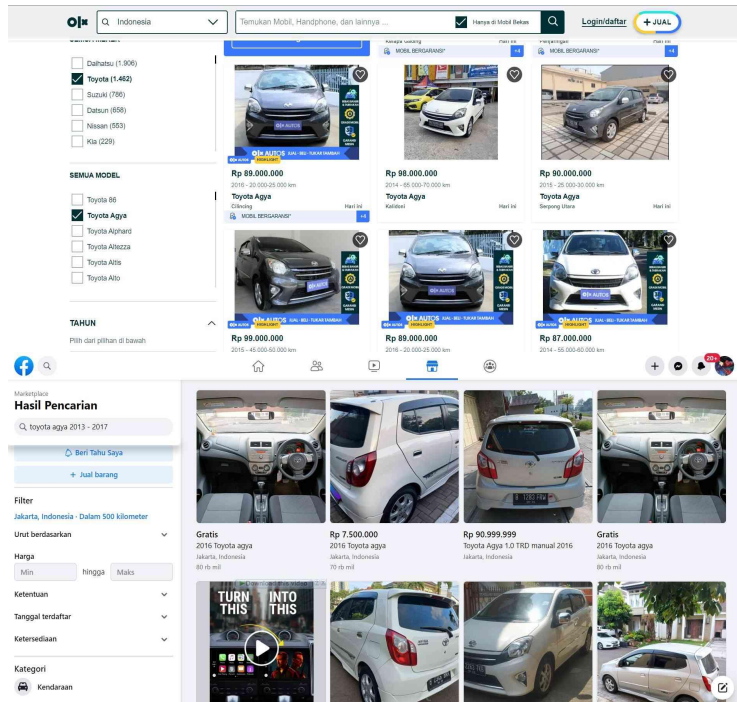
Langkah awal yang dilakukan penulis dalam merancang sistem yaitu dengan mengumpulkan dataset gambar mobil yang akan digunakan sebagai dataset untuk data train dan data uji[10]. Data yang digunakan dalam penelitian ini berupa data mobil dengan tampilan dari setiap sisinya. Data gambar mobil tersebut akan masuk kedalam dataset sistem.



Gambar 3. Contoh objek mobil

Explore

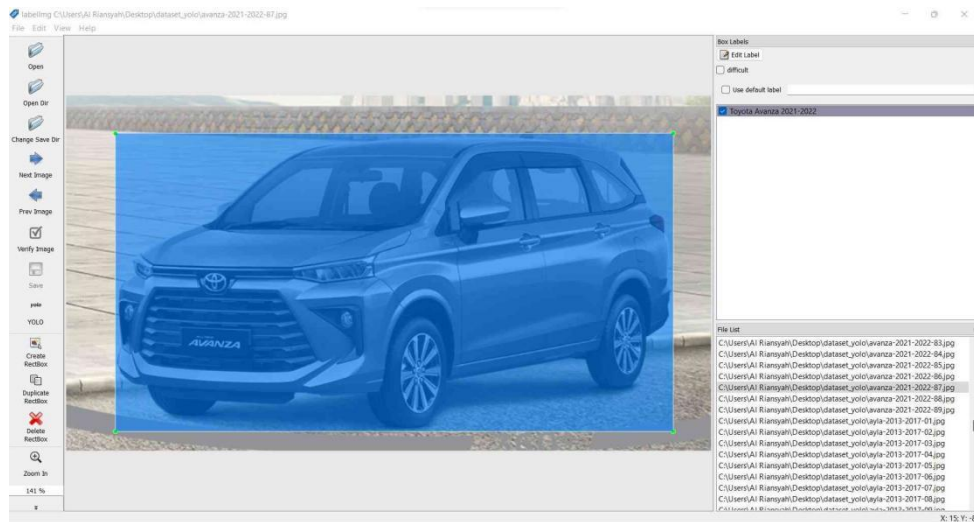
Pada tahap ini penulis melakukan eksplorasi data, dengan cara mencari dan mempelajari data yang dibutuhkan dalam penelitian ini dari sumber internet. Data yang dibutuhkan yaitu data gambar mobil berdasarkan merek dan tipe serta tahun produksinya. Setelah itu penulis melakukan penyeleksian data yang dibutuhkan tersebut pada *website olx* dan *marketplace facebook*[11].



Gambar 4. Seleksi data pada website olx dan marketplace facebook

Modify

Melakukan modifikasi terhadap semua data yang telah dikumpulkan dengan, melakukan pelabelan kelas pada data gambar mobil berdasarkan merek dan tipe serta tahun produksinya dengan menggunakan tool *LabelImg*[12]. Dengan cara membuat area *rectangle bounding box* khusus format *YOLO* pada gambar mobil tersebut.



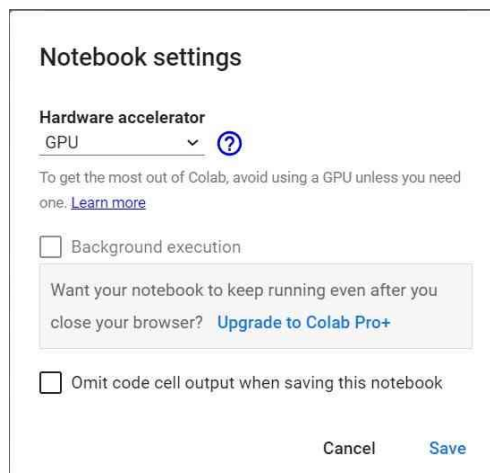
Gambar 5. Membuat area *rectangle bounding box*

Model

Pada tahap pemodelan ini, dataset mobil berdasarkan merek dan tipe serta tahun produksinya akan dilakukan proses train pemodelan. Proses train pemodelan dataset akan dikembangkan menggunakan *framework darknet* sebagai alat pemodelan dan *website google colab* sebagai lingkungan pengembangan system, dan memanfaatkan *library* dan fitur alat yang ada di *website google colab* antara lain yaitu *library opencv*. GPU online dan *libray* bahasa pemrogramman *python*[13]. Berikut adalah listing program proses pemodelan dataset yang dilakukan pada lingkungan pengembang di *website google colab* :

Mengaktifkan GPU

Dalam proses train dataset akan sangat membutuhkan GPU untuk mempercepat proses train dataset[14]. Maka dari itu harus mengaktifkan GPU terlebih dahulu pada menu edit, lalu pilih menu *Notebook settings* dan aktifkan GPU, lalu simpan settingan.



Gambar 6. Mengaktifkan GPU

Memvalidasi GPU

Mengecek apakah GPU berfungsi dengan baik pada lingkungan pengembangan pada *website google colab*. Berikut adalah listing program untuk mengecek apakah GPU sudah aktif ataupun tidak aktif.

```
import tensorflow as tf
device_list = tf.test.gpu_device_name()
device_list
if device_list != '/device:GPU:0':
    raise SystemError('GPU device not found')
print('Found GPU at: {}'.format(device_list))
```

Akses penyimpanan Google Drive

Untuk mengakses penyimpanan google drive dibutuhkan untuk melakukan izin akses, dengan tahap verifikasi ke akun google drive yang akan digunakan sebagai tempat penyimpanan berkas dataset dan file-file konfigurasi pada *framework Darknet*. Berikut adalah listing program untuk melakukan izin akses penyimpanan google drive[15] :

```
from google.colab import drive
drive.mount('/content/drive')
```

Clone framework Darknet

Melakukan *clone* atau mengunduh *framework darknet* ke lokasi direktori pada *google drive* yang akan digunakan dalam pengembangan system. File yang di *clone* berasal dari *website github* dalam bentuk file format zip, maka dari itu perlu dilakukan proses unzip pada file tersebut. Berikut adalah listing untuk melakukan *clone framework darknet* lalu meng unzip ke direktori pengembangan system[16] :

```
!git clone 'https://github.com/AlexeyAB/darknet' '/content/drive/My Drive/yolo_jurnal/darknet'
```

Mengunduh file konfigurasi train dataset

File konfigurasi dibutuhkan untuk proses train dataset menggunakan *framework darknet* tersebut. File yang di unduh merupakan file konfigurasi untuk mengakses GPU dan *library opencv* yang digunakan pada proses train dataset, dengan nama file yaitu Makefile. Berikut adalah listingnya :

```
!make
```

Memvalidasi framework Darknet

Sebelum melakukan training dataset, perlu mengecek terlebih dahulu apakah *framework darknet* tersebut berfungsi dengan baik. Pada tahap ini kita harus mengarahkan ke jalur direktori dimana *framework darknet* tersimpan. Berikut adalah listing untuk mengarahkan ke direktori dimana *framework darknet* tersimpan dan mengecek apakah *framework darknet* berjalan dengan baik :

```
%cd /content/drive/My Drive/yolo_jurnal/darknet
!./darknet
```

Train Dataset

Pada tahap ini dilakukan proses train dataset menggunakan *framework darknet*. Semua file konfigurasi yang dibutuhkan akan dijalankan pada jalur direktori yang sama dimana *framework darknet* tersimpan. Berikut adalah listing untuk melakukan train dataset :

```
!./darknet detector train cfg/labelled_data.data cfg/yolov3_custom.cfg weights/darknet53.conv.74 -dont_show
```

Assess

Evaluasi hasil pengujian

Pengujian dilakukan secara real-time menggunakan kamera webcam laptop dan pengujian pada contoh gambar, yang dijalankan menggunakan bahasa pemrograman python dan menggunakan aplikasi *visual studio code* sebagai teks editor *runtime*[17]. Berikut adalah hasil pengujian yang telah dilakukan :



Gambar 7. Pengujian menggunakan kamera webcam



Gambar 8. Pengujian pada gambar

Berikut adalah hasil pengujian dari 25 data mobil berdasarkan merek dan tipe serta tahun produksinya :

Tabel 1. Data hasil pengujian

No	Variable	Accuracy
1	Toyota Agya 2020 - 2022	97%
2	Toyota Avanza 2021 - 2022	98%
3	Toyota Calya 2022	97%
4	Toyota Raize 2022	98%
5	Daihatsu Sirion 2022	97%
6	Daihatsu Siga 2019 - 2022	96%
7	Daihatsu Ayla 2017 - 2020	92%
8	Daihatsu Roxy 2022	95%
9	Honda HR-V 2018 - 2021	90%
10	Toyota Avanza 2003 - 2005	84%
11	Toyota Avanza 2008 - 2011	87%
12	Toyota Agya 2017 - 2020	90%
13	Toyota Agya 2013 - 2017	89%
14	Toyota Calya 2019 - 2021	91%
15	Toyota Calya 2016 - 2019	91%
16	Daihatsu Sirion 2007 - 2009	90%
17	Daihatsu Sirion 2009 - 2011	89%
18	Daihatsu Sirion 2011 - 2015	92%
19	Daihatsu Sirion 2015 - 2018	91%
20	Daihatsu Siga 2016 - 2019	90%
21	Daihatsu Ayla 2013 - 2017	91%
22	Honda HR-V 2014 - 2018	93%
23	Toyota Avanza 2011 - 2015	92%
24	Toyota Avanza 2015 - 2019	90%
25	Daihatsu Sirion 2018 - 2021	90%
Rata-rata		92%

Pembahasan

Agar diketahui seberapa akurat hasil dari sistem yang telah dijalankan, maka akan dilakukan pengujian terlebih dahulu untuk memvalidasi hasil pengujian pada tabel 1 yang mana terdiri dari 25 data mobil berdasarkan merek dan tipe serta tahun produksinya, yang terdiri dari 21 data positif dan 4 data negatif, dengan evaluasi dan validasi menggunakan perhitungan *confusion matrix* yang terdiri dari *accuracy*, *precision* dan *recall*[18]. Berdasarkan hasil uji coba dari sistem yang telah dibuat maka didapatkan[19] :

		<i>Observed</i>	
		<i>True</i>	<i>False</i>
<i>Predicted Class</i>	<i>True</i>	<i>True Positive (TP)</i>	<i>False Positive (FP)</i>
	<i>False</i>	<i>False Negative (FN)</i>	<i>True Negative (TN)</i>

Gambar 9. Tabel *convusion matrix*

Jadi accuracy, precision, dan recall dapat dirumuskan sebagai berikut[20] :

$$Precision = \frac{TP}{TP+FP} \qquad Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

$$Recall = \frac{TP}{TP+FN}$$

Tabel 2. Hasil confusion matrix

	True	False
True	20	1
False	1	3

$$Precision = \frac{TP}{TP+FP} = \frac{20}{20+1} = 0.95 \times 100\% = 95\%$$

$$Recall = \frac{TP}{TP+FN} = \frac{20}{20+1} = 0.95 \times 100\% = 95\%$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} = \frac{20+3}{20+3+1+1} = 0.92 \times 100\% = 92\%$$

Jadi, berdasarkan hasil perhitungan table confusion matrix didapatkan hasil dari sistem yang telah dibuat yaitu accuracy sebesar 92%, precision sebesar 95% dan recall sebesar 95%. Dengan tingginya hasil nilai accuracy, precision dan recall yang didapatkan, maka dapat dianggap bahwa sistem tersebut memiliki kinerja yang baik.

4. Penutup

Berdasarkan hasil penelitian yang telah dilakukan pada pembahasan yang telah diuraikan pada bab-bab sebelumnya, maka dapat disimpulkan bahwa Algoritma YOLO dapat dengan baik melakukan deteksi mobil berdasarkan merek dan tipe serta tahun produksinya dengan hasil nilai accuracy yang tinggi yaitu sebesar 92%, precision sebesar 95% dan recall sebesar 96%. Dilihat dari tingginya pencapaian nilai accuracy, precision dan recall yang diperoleh, bisa dianggap bahwa sistem tersebut memiliki kinerja yang baik karena hasil akurasi diatas 85%.

5. Referensi

- [1] A. R. Bakri, "Implementasi Intrusion Detection System (Ids) Menggunakan Telegram Sebagai Media Notifikasi," UPN Veteran Jawa Timur, 2020.
- [2] M. Harahap, J. Elfrida, P. Agusman, M. Rafael, R. Abram, and K. Andrianto, "Sistem Cerdas Pemantauan Arus Lalu Lintas Dengan DCN (Deep Convolutional Network)," in *Seminar Nasional Aptikom (SEMNASTIK) 2019*, 2019, pp. 367–376.
- [3] B. A. Wicaksono, I. Y. Purbasari, and Y. V. Via, "Deteksi Objek Mobil dan Motor pada Lalu Lintas Berbasis Deep Learning," *JIFOSI J. Inform. dan Sist. Inf.*, vol. 2, no. 2, pp. 334–342, 2021.
- [4] M. S. Hidayatulloh, "TA: Sistem Pengenalan Wajah Menggunakan Metode YOLO (You Only Look Once)," Universitas Dinamika, 2021.
- [5] T. I. Hermanto and Y. Muhyidin, "Analisis Data Sebaran Bandwidth Menggunakan Algoritma Dbscan Untuk Menentukan Tingkat Kebutuhan Bandwidth Di Kabupaten Purwakarta," *Rabit J. Teknol. dan Sist. Inf. Univrab*, vol. 5, no. 2, pp. 130–137, 2020.
- [6] W. Wahyudi, A. Hidayat, M. M. Fakhri, and others, "Penerapan Machine Learning Pada Mikrokontroler Arduino Mega PRO MINI ATmega2560-16AU," *J. Embed. Syst. Secur. Intell. Syst.*, vol. 3, no. 1, pp. 30–39, 2022.

- [7] N. P. S. Dewi, A. Y. A. Fianto, and W. Hidayat, "Penciptaan Buku Refensi Pura Mandhara Giri Semeru Agung Guna Mempopulerkan Wisata Religi Kabupaten Lumajang."
- [8] A. Amwin, "Deteksi dan Klasifikasi Kendaraan Berbasis Algoritma You Only Look Once (YOLO)," 2021.
- [9] M. R. Fauzan and A. P. W. Wibowo, "Pendeteksian Plat Nomor Kendaraan Menggunakan Algoritma You Only Look Once V3 Dan Tesseract," *J. Ilm. Teknol. Infomasi Terap.*, vol. 8, no. 1, pp. 57–62, 2021.
- [10] L. Rahma, H. Syaputra, A. H. Mirza, and S. D. Purnamasari, "Objek Deteksi Makanan Khas Palembang Menggunakan Algoritma YOLO (You Only Look Once)," *J. Nas. Ilmu Komput.*, vol. 2, no. 3, pp. 213–232, 2021.
- [11] D. Anggraini, S. A. Putri, and L. A. Utami, "Implementasi Algoritma Apriori Dalam Menentukan Penjualan Mobil Yang Paling Diminati Pada Honda Permata Serpong," *J. Media Inform. Budidarma*, vol. 4, no. 2, pp. 302–308, 2020.
- [12] N. H. Harani, C. Prianto, and M. Hasanah, "Deteksi Objek Dan Pengenalan Karakter Plat Nomor Kendaraan Indonesia Menggunakan Metode Convolutional Neural Network (CNN) Berbasis Python," *J. Tek. Inform.*, vol. 11, no. 3, pp. 47–53, 2019.
- [13] S. Ma'arif, T. Rohana, and K. Baihaqi, "Deteksi Jenis Beras Menggunakan Algoritma YOLOv3," *Sci. Student J. Information, Technol. Sci.*, vol. 3, no. 2, pp. 219–226, 2022.
- [14] M. A. BELLA, "Implementasi Algoritma Deep Learning Untuk Sistem Deteksi Kantuk Pada Pengemudi Menggunakan Yolo," 2021.
- [15] E. F. Fernanda, "LKP: Deteksi Penyakit Kanker Payudara Menggunakan Deep Learning," Universitas Dinamika, 2021.
- [16] K. Amoako, "Healthy Hive: A Beehive Health Management Tool," 2022.
- [17] M. Romzi and B. Kurniawan, "Implementasi Pemrograman Python Menggunakan Visual Studio Code," *J. Inform. dan Komput.*, vol. 11, no. 2, 2020.
- [18] H. Yun, "Prediction model of algal blooms using logistic regression and confusion matrix," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 3, p. 2407, 2021.
- [19] N. Amalina, "Uji akurasi aplikasi Augmented Reality pembelajaran huruf alfabet Bahasa Isyaratindonesia (BISINDO) pada Vuforia menggunakan Confusion Matrix," Universitas Islam Negeri Maulana Malik Ibrahim, 2019.
- [20] M. Hasnain, M. F. Pasha, I. Ghani, M. Imran, M. Y. Alzahrani, and R. Budiarto, "Evaluating trust prediction and confusion matrix measures for web services ranking," *IEEE Access*, vol. 8, pp. 90847–90861, 2020.