



Simulasi deteksi golongan kendaraan pada gerbang tol menggunakan YOLOv4

Reny Medikawati Taufiq¹, Sunanto², Yoze Rizki³, M. Rizki Amanda Pratama⁴

Email: ¹renymedikawati@umri.ac.id, ²sunanto@umri.ac.id, ³yozerizki@umri.ac.id, ⁴160401077@student.umri.ac.id

¹²³⁴Teknik Informatika, Fakultas Ilmu Komputer, Universitas Muhammadiyah Riau

Diterima: 04 Agustus 2022 | Direvisi: 17 Agustus 2022 | Disetujui: 17 Agustus 2022

©2020 Program Studi Teknik Informatika Fakultas Ilmu Komputer,
Universitas Muhammadiyah Riau, Indonesia

Abstrak

Jalan tol merupakan infrastruktur vital yang membuat pelayanan distribusi barang dan jasa menjadi lebih produktif dan efisien. Namun pada kenyataannya, di kota besar tingginya kemacetan juga terjadi di jalan tol. Salah satu titik rawan kemacetan adalah di gerbang tol. Kemacetan ini tidak hanya terjadi pada jam sibuk, tetapi juga terjadi sepanjang hari. Kemacetan disebabkan waktu tunggu pada proses pembayaran. Kemacetan yang terjadi di Gerbang Tol Otomatis (GTO) Multi Kendaraan salah satunya disebabkan oleh adanya proses penentuan golongan kendaraan secara manual oleh petugas pada *control room*. Kendaraan yang menggunakan jalan tol digolongkan kedalam 5 golongan berdasarkan jumlah gandar. Petugas melihat satu persatu kendaraan yang melintas dan menentukan golongan kendaraan tersebut, biaya tol yang harus dibayar oleh pengguna jalan tol adalah berdasarkan golongan kendaraan yang digunakan. Kemacetan pada jalan tol menimbulkan dampak negatif seperti seperti pemborosan bahan bakar dan waktu, dan juga dampak lingkungan yang dapat menyebabkan kerugian secara ekonomi. Oleh sebab itu, pada penelitian ini akan dilakukan simulasi deteksi golongan kendaraan pada gerbang tol dengan menggunakan teknologi *computer vision* dan *deep learning*, dengan algoritma YOLOv4. Dengan adanya pendeteksian golongan kendaraan secara otomatis maka diharapkan waktu tunggu pada gerbang tol dapat memenuhi Standar Pelayanan Minimal (SPM) Jalan Tol yaitu maksimal 5 detik. Dataset berupa 650 gambar golongan kendaraan, setelah di augmentasi menjadi 1547 gambar. Proses training dilakukan menggunakan *Google Colabs*. Video rekaman lalu lintas kendaraan yang sedang berjalan akan menjadi inputan pada pengujian implementasi aplikasi Python. Dari pengujian yang telah dilakukan dapat dilihat bahwa model sudah dapat mendeteksi golongan kendaraan dengan cukup efektif.


Kata kunci: deteksi kendaraan, YOLOv4, deteksi golongan kendaraan, deep learning, computer vision

Simulation of vehicle class detection at toll gate using YOLOv4

Abstract

The toll road is a vital infrastructure that makes the distribution of goods and services more productive and efficient. But in reality, in big cities, high congestion also occurs on toll roads. One of the congestion spot is at the toll gate. This congestion does not only occur during peak hours, but also occurs throughout the day. Congestion due to waiting time in the payment process. At the Multi-Vehicle Automatic Toll Gate (GTO), congestion is caused by the process of determining the class of vehicles manually by officers in the control room. Vehicles using toll roads are classified into 5 groups based on the number of axles. Officers look at the passing vehicles one by one and determine the class of the vehicle, the toll fees that must be paid by toll road users are based on the class of vehicles. Congestion on toll roads has negative impacts such as waste of fuel and time, as well as environmental impacts that can cause economic losses. Therefore, in this study, a simulation of vehicle class detection at toll gates will be carried out using computer vision and deep learning technology, with the YOLOv4 algorithm. With the automatic detection of vehicle classes, it is expected that the waiting time at toll gates can meet the Toll Road Minimum Service Standards (SPM), which is a maximum of 5 seconds. The dataset is 650 images of vehicle groups, after being augmented it becomes 1547 images. The training process is carried out using Google Colabs. Video recordings of vehicle traffic is used as input for testing the implementation of Python applications. From the measurement that have been carried out, it can be conclude that the model is able to detect the class of vehicles.

Keywords: vehicle detection, YOLOv4, vehicle class detection, deep learning, computer vision

 doi: <https://doi.org/10.37859/coscitech.v3i2.3928>

1. PENDAHULUAN

Jalan tol merupakan infrastruktur vital yang diperlukan untuk mobilitas sosial dan ekonomi. Dalam sistem perekonomian keberadaan jalan tol dapat meningkatkan pelayanan distribusi barang dan jasa. Dengan adanya jalan tol arus keluar masuk barang antar wilayah menjadi lebih cepat dan lebih mudah, sehingga aktivitas bisnis dapat dilakukan dengan lebih produktif dan efisien.

Namun pada kenyataannya, di kota besar tingginya kemacetan juga terjadi di jalan tol. Kemacetan pada jalan tol terjadi pada gerbang masuk, ruas jalan dan gerbang keluar. Kemacetan ini tidak hanya terjadi pada jam sibuk saat pagi dan sore hari, namun hampir terjadi sepanjang hari. Kemacetan pada jalan tol menimbulkan berbagai dampak negatif bagi pengguna jalan, yaitu pemborosan bahan bakar, kerusakan lingkungan yang disebabkan oleh polusi kendaraan bermotor, waktu tunggu yang terbuang percuma [1]. Secara ekonomi besaran kerugian yang terjadi akibat kemacetan di jalan tol telah dihitung yaitu berdasarkan Biaya Operasional Kendaraan (BOK) sebesar Rp.134.725,51/km. Besaran kerugian berdasarkan biaya waktu perjalanan sebesar Rp.1.907.353,40/jam. Dan besaran kerugian berdasarkan biaya emisi adalah sebesar Rp.598.593,03/jam. Dapat dilihat bahwa biaya waktu perjalanan adalah aspek kerugian yang paling tinggi, karena aspek ini merepresentasikan produktivitas manusia [1].

Salah satu titik rawan kemacetan adalah pada gerbang tol keluar. Kemacetan disebabkan karena transaksi pembayaran masih memakan waktu. Gerbang Tol Otomatis (GTO) dengan mekanisme pembayaran non tunai menggunakan *e-Toll Card* digunakan untuk memberikan kemudahan, kenyamanan dan kelancaran transaksi di gerbang tol. Teknologi ini diharapkan mampu mempercepat waktu transaksi sehingga dapat memenuhi Standar Pelayanan Minimal (SPM) Jalan Tol yang tertuang pada Permen PU No. 16/PRT/M/2014. Menurut SPM Jalan Tol ditetapkan bahwa waktu layanan maksimal pada gardu tol transaksi adalah maksimal 5 detik untuk setiap kendaraan [2]. Pada kenyataannya sistem GTO ini juga belum mampu mengatasi permasalahan kemacetan di gerbang tol karena dibutuhkan proses penentuan jenis golongan kendaraan sebagai dasar penentuan perhitungan besaran biaya tol, dan kendaraan juga harus berhenti sejenak untuk menempelkan kartu pada pembaca elektronik.

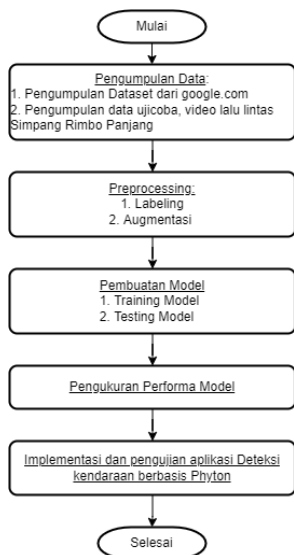
Besarnya biaya tol yang harus dibayar tergantung pada golongan kendaraan yang digunakan, oleh karena itu sebelum transaksi pembayaran dilakukan perlu ditentukan golongan kendaraan terlebih dahulu. Terdapat dua jenis GTO yaitu, (1) GTO Manual yang hanya dapat digunakan oleh kendaraan mobil kecil Golongan 1 Non Bus, (2) GTO Multi Golongan yang dapat dilalui oleh golongan kendaraan I s/d V seperti truk, mobil biasa maupun bus, karena tidak adanya *gantry* untuk membatasi tinggi kendaraan sebelum masuk gardu [3]. Menurut penjelasan Marketing & Communication Department Head Jasa Marga Transjawa Tollroad Regional Division, Corry Annelia Ponti, sistem pemilahan golongan pada GTO Multi Golongan masih dilakukan secara manual oleh petugas pada *control room* [3]. Proses pengecekan golongan kendaraan secara manual ini menyebabkan antrian kemacetan di beberapa gerbang tol.

Agar pelayanan sesuai dengan Standar Pelayanan Minimal (SPM) Jalan Tol dan menghindari kerugian ekonomi maka perlu dilakukan percepatan waktu pemrosesan transaksi di gerbang tol. Pertumbuhan infrastruktur khususnya di jalan tol sepatutnya diiringi oleh perkembangan teknologi. Penelitian ini berfokus pada alternatif solusi pendeteksian golongan kendaraan secara otomatis dengan menggunakan teknologi *computer vision* dan *deep learning*. Beberapa penelitian mengenai deteksi golongan kendaraan berbasis citra telah dilakukan dengan menggunakan beberapa pendekatan yaitu, penelitian Safitri dengan pendekatan pengolahan citra menggunakan metode *background subtraction* dan *template matching* [4], klasifikasi golongan kendaraan berdasarkan Fitur Histogram of Oriented Gradients (HOG) dengan metode K-Nearest Neighbors (K-NN) [5], Pengenalan Golongan Jenis Kendaraan Bermotor Menggunakan CNN [6], Klasifikasi Jenis Menggunakan Metode Yolo (You Only Look Once) [7] dan menggunakan metode Yolo v3 [8]. Dalam penelitian ini akan digunakan metode Yolo v4 yang merupakan pengembangan dari versi sebelumnya. Yolov4 memiliki performa dua kali lebih cepat dan efisien dibandingkan dengan Yolo v3 [9]. Perbedaan Yolov3 dengan versi sebelumnya terletak pada arsitektur darknet yang digunakan, yaitu Darknet-53 [10]. Yolov4 juga menggunakan arsitektur Darknet-53 yaitu CSPDarknet-53 atau CSPNet yang menyediakan backbone baru yang dapat meningkatkan performa learning capability dari CNN [11]

Dengan adanya deteksi otomatis golongan kendaraan berbasis citra petugas di *control room* tidak perlu lagi memantau golongan kendaraan satu persatu, hal ini diharapkan dapat mempercepat transaksi di gerbang tol. Pada penelitian ini akan dilakukan simulasi pendeteksian golongan kendaraan dengan data uji coba berupa video yang diambil dari jalan raya Rimbo Panjang Pekanbaru.

2. METODE PENELITIAN

Dalam penelitian ini akan dilakukan simulasi deteksi golongan kendaraan secara otomatis berdasarkan citra menggunakan *computer vision* dan *deep learning*. Algoritma deteksi yang digunakan adalah Yolo v4. Dataset citra kendaraan diambil dari google sebanyak 650 data dari beberapa sumber. Adapun tahapan penelitian yang dilakukan dapat dilihat pada Gambar 1.



Gambar 1. Tahapan penelitian

3. HASIL DAN PEMBAHASAN

3.1. Pembuatan Dataset

Dataset citra golongan kendaraan dikumpulkan dari berbagai sumber melalui google.com. Dataset yang digunakan sebanyak 650 gambar. Golongan kendaraan yang dideteksi adalah Golongan 1 (Sedan, Jip, Pick Up/Truk Kecil, dan Bus), Golongan 2 (Truk dengan 2 gandar), Golongan 3 (Truk dengan 3 gandar). Contoh dataset golongan kendaraan dapat dilihat pada Gambar 2. Karena penggolongan kendaraan memperhatikan jumlah gandar maka dataset gambar yang dikumpulkan dari google adalah gambar mobil tampak samping, yang memperlihatkan jumlah gandarnya.



Gambar 2. Dataset golongan kendaraan. (a) Golongan 1, (b) Golongan 2, (c) Golongan 3

3.2. Labeling dan Augmentasi Data

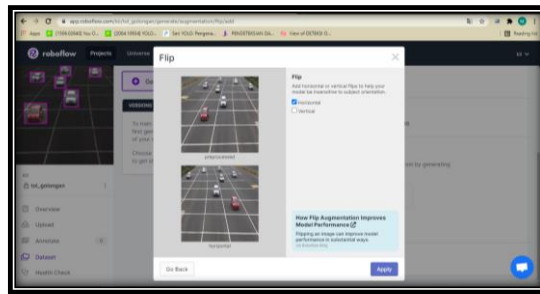
Preprocessing terdiri dari tahapan labeling dan augmentasi gambar. Pelabelan berfungsi untuk memberi identitas objek pada gambar, agar objek tersebut memiliki identitas sesuai dengan golongan. Augmentasi adalah proses penggandaan data sehingga dataset menjadi lebih banyak. Teknik Augmentasi yang digunakan adalah flip, 90° rotate, crop, rotation, shear, blur, noise. Aplikasi yang digunakan pada preprocessing adalah app.roboflow.com. Jumlah data setelah dilakukan augmentasi adalah 1547 gambar.

1. Labelling data : Pada proses ini gambar/data set di beri identitas nya masing-masing. Proses labelling dapat dilihat pada Gambar 3.



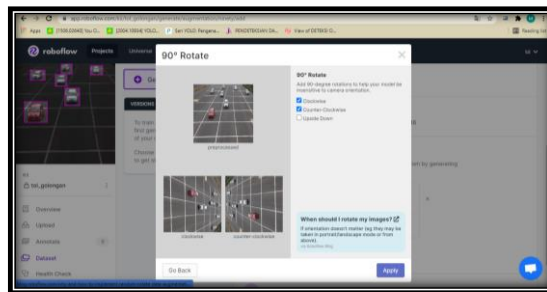
Gambar 3. Proses Labelling

2. Flip: Pada proses ini gambar di balikkan horizontal dan vertikal. Proses flip dapat dilihat pada Gambar 4.



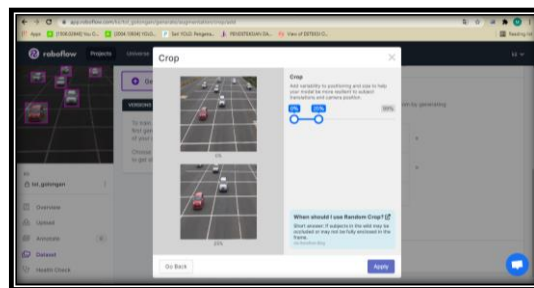
Gambar 4. Proses Flip

3. 90° Rotate: Pada proses ini gambar gambar di putar dari arah kiri ke kanan dan sebaliknya sebesar seperempat putaran. Proses 90° Rotate dapat dilihat pada Gambar 5.



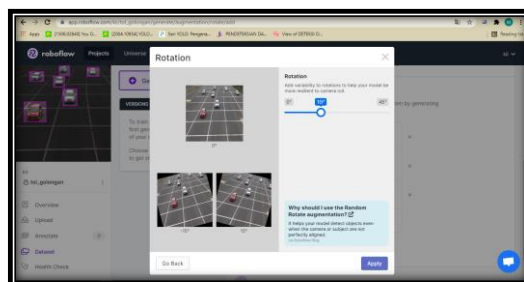
Gambar 5. Proses 90° Rotate

4. Crop: Pada proses ini gambar di potong 20% dari gambar asli nya. Proses crop dapat dilihat pada Gambar 6.



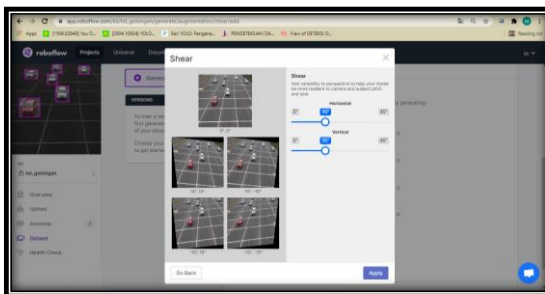
Gambar 6. Proses Crop

5. Rotation: Pada proses ini gambar di memutar beberapa beberapa arah. Proses *rotation* dapat dilihat pada Gambar 7.



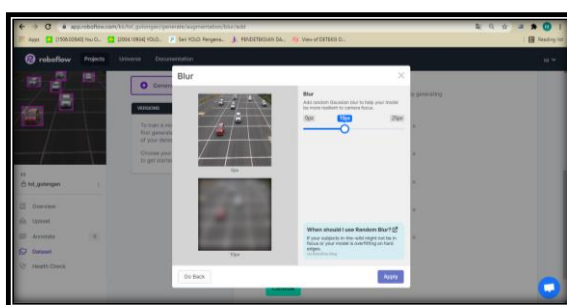
Gambar 7. Proses Rotation

6. Shear: Pada proses ini gambar di geser sudut nya sebesar 15 derajat sehingga menghasilkan 4 gambar baru. Proses shear dapat dilihat pada Gambar 8.



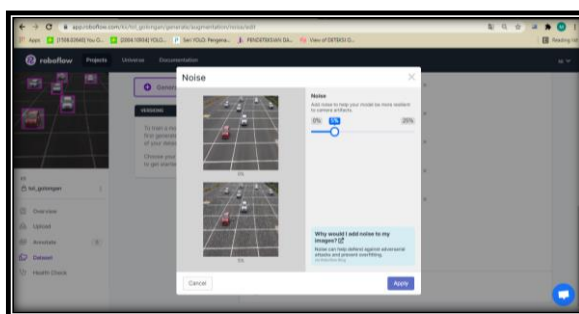
Gambar 8. Proses Shear

7. Blur: Pada proses ini gambar hasil jadi jadi tidak focus cenderung menjadi abu-abu. Proses blur dapat dilihat pada Gambar 9.



Gambar 9. Proses Blur

8. Noise: Pada proses ini gambar muncul butiran-butiran warna yang mengganggu pada gambar sebesar 5%. Proses Noise dapat dilihat pada Gambar 10.



Gambar 10. Proses Noise

3.3. Training

Pada proses ini akan di lakukan tahapan Training YOLO, semua data gambar pada dataset akan akan di pelajari oleh YOLO. Karena akan memproses data gambar dalam jumlah yang cukup banyak, maka pada proses training sangat di butuhkan sebuah GPU (*Graphic Processing Unit*). Oleh karena itu training dilakukan dengan menggunakan tools Google Colab. Hasil dari tahapan training adalah berupa file .cfg (weight & coco.name). Tahapan-tahapan training dapat dilihat pada Tabel 1.

Tabel 1. Tahapan Training

No.	Keterangan	Fungsi
1.	konfigurasi cuDNN	Pada tahapan ini untuk mengecek hardware GPU pada google colab
2.	Installing Darknet	Pada proses ini google colab akan menginstallkan darknet di folder google drive.
3.	ekstrak datasets	Proses ini melakukan ekstrak data yang ada pada roboflow

4.	konfigurasi <i>training</i>	Pada tahap ini ditentukan jumlah pengulangan pada gambar.
5.	<i>Training</i> YOLO	Pada proses ini dihasilkan file berupa <i>cfg</i> (<i>weight & coco.name</i>)

3.4. Pengukuran Performa Model

Pengukuran performa model dilakukan dengan menghitung *Average Precision (AP)*, *Training Loss* dan kinerja klasifikasi dengan *Confusion Matrix*.

Dalam melakukan multi-klasifikasi gambar, *confusion matrix* menampilkan setiap hasil kelas kemudian membandingkan hasil klasifikasi dengan nilai yang sebenarnya. Selain menganalisis *accuracy*, proses *confusion matrix* juga menghitung indeks lain, seperti *precision*, *recall*, dan *f1-score* [12]

Presisi merupakan rasio dari jumlah objek yang terdeteksi dengan benar atau *True Positive* dibandingkan dengan seluruh data yang diprediksi positif [10]. Rumus presisi dapat dilihat pada Rumus 1 [13].

$$Presisi = \frac{TP}{TP+FP} \times 100\% \tag{1}$$

Recall merupakan rasio dari jumlah objek yang terdeteksi dengan benar atau *True Positive* dibandingkan dengan seluruh data yang positif [10]. Rumus presisi dapat dilihat pada Rumus 2 [14].

$$Recall = \frac{TP}{TP+FN} \times 100\% \tag{2}$$

F1 Score merupakan perbandingan rata-rata dari nilai presisi dan recall [10]. Rumus F1 score dapat dilihat pada Rumus 3.

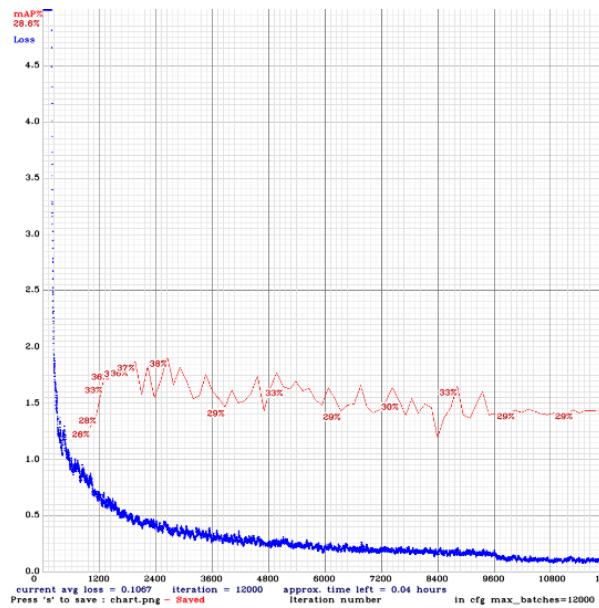
$$F1\ Score = 2 \times \frac{Recall \times Presisi}{Recall + Presisi} \times 100\% \tag{3}$$

Hasil pengukuran *confusion matrix* dapat dilihat pada Tabel 2.

Tabel 2. Tahapan Training

Parameter	Hasil
<i>True Positive (TP)</i>	108
<i>False Positive (FP)</i>	139
<i>False Negative (FN)</i>	64
<i>Precision</i>	0,44
<i>Recall</i>	0,63
<i>F1-Score</i>	0,52

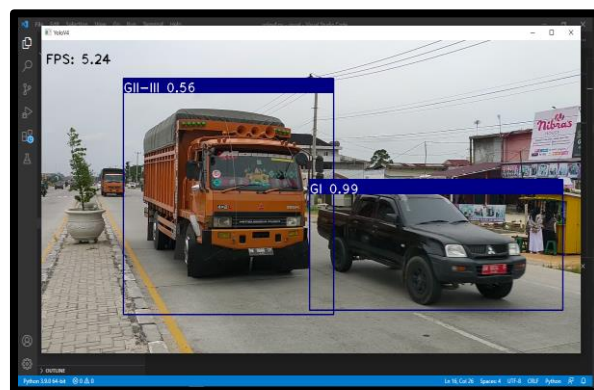
Pada Gambar 11 dapat dilihat bahwa, dari hasil training 12000 *epoch* didapatkan nilai *average loss* sebesar 0.1067. Dapat dilihat pula pada gambar, bahwa model yang dirancang memiliki mean Average Precision rendah pada angka 28.6%. hal ini dapat terjadi dikarenakan terlalu beragamnya jenis dan ukuran kendaraan yang dideteksi, juga terjadinya efek blur pada objek bergerak pada citra yang ditangkap berupa kendaraan berkecepatan tinggi yang dapat mengubah morfologi objek tersebut. Rendahnya nilai *average precision* berpengaruh pada tidak konsistennya pengenalan objek oleh sistem, sehingga mengurangi kualitas model pembelajaran yang dibangun. Penggunaan kamera berresolusi tinggi dan secara signifikan memperbanyak jumlah dataset dari berbagai macam varian kendaraan dapat mengurangi resiko rendahnya nilai mean average precision.



Gambar 11. Pengukuran Performa Model

3.5. Implementasi Aplikasi Deteksi Golongan Kendaraan dengan berbasis Python

Setelah tahap proses *training* kemudian didapat file berupa *cfg*, *weight* dan *coco.names* lalu tahap selanjutnya adalah melakukan pembuatan kode pemrograman menggunakan Bahasa *python* untuk implementasi *Yolov4* dalam pendeteksian golongan kendaraan dengan inputan video. Hasil uji *runtime* deteksi golongan kendaraan dapat dilihat pada Gambar 11.



Gambar 12. Tampilan Runtime Aplikasi Python untuk Deteksi Golongan Kendaraan

4. KESIMPULAN

Berdasarkan pengujian yang dilakukan pada aplikasi, dengan data inputan berupa video kondisi lalu lintas di persimpangan Jalan Rimbo Panjang, dapat dilihat bahwa model yang dibangun sudah dapat melakukan deteksi kendaraan dengan baik. Dapat diambil kesimpulan bahwa:

1. Performa pada *yolo v4* cukup efektif untuk mendeteksi jenis golongan kendaraan dan bisa juga dilakukan menggunakan video lalu lintas jalan Rimbo Panjang.
2. Tingkat kepercayaan deteksi semakin baik jika video diambil dari sisi samping (atau menyerong), hal ini disebabkan karena dataset gambar yang dilakukan *training* adalah gambar kendaraan dari arah samping atau menyerong.

DAFTAR PUSTAKA

- [1] G. Eristian and Nahdalina, "ANALISIS BIAYA KEMACETAN PADA RUAS JALAN TOL JAKARTA-CIKAMPEK KM. 26," *J. Ilm. Desain dan Konstr.*, vol. 20, no. 1, pp. 70–83, 2021.
- [2] A. M. S. Sufanir, "Efektivitas Gardu Tol Otomatis (Gto) Buah Batu Ditinjau Dari Kecepatan Transaksi Rata-Rata," *Pros. Simp. II - UNIID*, no. September, pp. 338–342, 2017.
- [3] A. P. Niagara, "Ternyata, Begini Cara Gerbang Tol Deteksi Golongan Kendaraan," *KumparanOTO*, 20-Nov-2020. [Online]. Available: <https://kumparan.com/kumparanoto/ternyata-begini-cara-gerbang-tol-deteksi-golongan-kendaraan-luciqdxxli8/full>. [Accessed: 31-Jul-2022].
- [4] K. Safitri and S. H. Sitorus, "Penentuan Golongan Kendaraan Roda Empat Keatas Menggunakan Metode Background Subtraction Dan Metode

- Template Matching,” vol. 09, no. 01, 2021.
- [5] L. N. Akbar, F. Utamingrum, P. Studi, T. Komputer, F. I. Komputer, and U. Brawijaya, “Klasifikasi Golongan Kendaraan Berdasarkan Fitur Histogram of Oriented Gradients (HOG) Menggunakan metode K-Nearest Neighbors (K-NN) Berbasis Raspbery PI 3,” vol. 3, no. 9, pp. 8531–8538, 2019.
- [6] R. Herwanto, K. Gunadi, and E. Setyati, “Pengenalan Golongan Jenis Kendaraan Bermotor pada Ruas Jalan Tol Menggunakan CNN,” *J. Infra Petra*, vol. 8, no. 1, pp. 196–202, 2020.
- [7] Z. Abidin, “KLASIFIKASI JENIS KENDARAAN PADA GERBANG TOL MENGGUNAKAN METODE YOLO (YOU ONLY LOOK ONCE),” 2021.
- [8] A. Amwin, “Deteksi Dan Klasifikasi Kendaraan Berbasis Algoritma You Only Look Once (YOLO),” 2021.
- [9] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” 2020.
- [10] T. A. A. H. Kusuma, K. Usman, and S. Saidah, “People Counting for Public Transportations Using You Only Look Once Method,” *J. Tek. Inform.*, vol. 2, no. 1, pp. 57–66, 2021.
- [11] C. Y. Wang, H. Y. Mark Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh, and I. H. Yeh, “CSPNet: A new backbone that can enhance learning capability of CNN,” *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2020-June, no. June 2021, pp. 1571–1580, 2020.
- [12] A. Rahim, K. Kusriani, and E. T. Luthfi, “Convolutional Neural Network untuk Kalasifikasi Penggunaan Masker,” *Inspir. J. Teknol. Inf. dan Komun.*, vol. 10, no. 2, p. 109, 2020.
- [13] Aditya Quantano Surbakti, Regiolina Hayami, and Januar Al Amien, “Analisa Tanggapan Terhadap Psbb Di Indonesia Dengan Algoritma Decision Tree Pada Twitter,” *J. CoSciTech (Computer Sci. Inf. Technol.*, vol. 2, no. 2, pp. 91–97, 2021.
- [14] G. I. Hayami Regiolina, Soni Soni, “Klasifikasi Jamur Menggunakan Algoritma Naïve Bayes,” *J. Comput. Sci. Inf. Technol.*, vol. 3, no. 2, pp. 1–19, 2020.