

OPTIMIZING YOLOV8 FOR AUTONOMOUS DRIVING: BATCH SIZE FOR BEST MEAN AVERAGE PRECISION (MAP)

Zaid Syarif Hidayat^{*1}, Yudhistira Arie Wijaya², Rudi Kurniawan³

^{1,2}Informatics Engineering, STMIK IKMI Cirebon, Indonesia

³Software Engineering, STMIK IKMI Cirebon, Indonesia

Email: ¹zaid.syarief@gmail.com, ²yudhistira010471@gmail.com, ³rudi226ikmi@gmail.com

(Article received: December 21, 2023; Revision: January 15, 2024; published: July 29, 2024)

Abstract

Artificial intelligence (AI), especially computer vision, has made rapid progress in recent years. One of the rapidly developing fields in computer vision is object detection. The ability to detect objects accurately and quickly is essential for the development of autonomous driving technology or vehicles that can operate automatically without human intervention. However, the development of autonomous driving technology is still facing various challenges, especially related to the accuracy and speed of object detection by the system. The purpose of this study is to analyze the performance based on the mean average precision (mAP) value of the results of adjusting the number of epochs, batch size, and image size on one of the emerging object detection methods, YOLOv8, in the context of autonomous driving. The analysis focuses on the batch size hyperparameter on the object detection performance of YOLOv8. The research was conducted with an experimental approach where the YOLOv8 hyperparameters were modified and their performance was evaluated using the driver simulation dataset from the CARLA simulator. Object detection performance was evaluated using the mean average precision (mAP) metric. The research results with the highest mAP value are found in scheme VIII with an mAP value of 98.2% and a training time of 59.45 minutes. For scheme III, it gets the fastest training time of 36.25 minutes. Based on the mAP results, modifications to the number of batch sizes and the use of high image sizes can affect the performance and performance of object detection for autonomous driving.

Keywords: Autonomous Driving, Batch Size, Object Detection, YOLOv8, mAP

OPTIMISASI YOLOV8 UNTUK AUTONOMOUS DRIVING: BATCH SIZE UNTUK MEAN AVERAGE PRECISION (MAP) TERBAIK

Abstrak

Teknologi kecerdasan buatan (*artificial intelligence*) khususnya *computer vision* tengah mengalami kemajuan pesat dalam beberapa tahun terakhir. Salah satu bidang yang tengah berkembang pesat dalam *computer vision* adalah metode deteksi objek. Kemampuan mendeteksi objek dengan akurat dan cepat sangat dibutuhkan dalam pengembangan teknologi *autonomous driving* atau kendaraan yang dapat beroperasi secara otomatis tanpa campur tangan manusia. Namun demikian, pengembangan teknologi *autonomous driving* masih menghadapi berbagai tantangan terutama terkait akurasi dan kecepatan deteksi objek oleh sistem. Tujuan dari penelitian ini adalah untuk melakukan analisis performa berdasarkan nilai mean average precision (mAP) dari hasil pengaturan jumlah *epoch*, *batch size*, *image size* terhadap salah satu metode deteksi objek yang tengah berkembang yaitu YOLOv8 dalam konteks *autonomous driving*. Analisis difokuskan pada *hyperparameter batch size* terhadap performa deteksi objek oleh YOLOv8. Penelitian dilakukan dengan pendekatan eksperimental di mana *hyperparameter* YOLOv8 di modifikasi dan performanya dievaluasi menggunakan dataset simulasi pengemudi dari CARLA simulator. Performa deteksi objek dievaluasi dengan metrik *mean average precision (mAP)*. Hasil penelitian dengan nilai mAP yang paling tinggi terdapat pada skema VIII dengan nilai mAP 98.2% dan waktu training 59.45 menit. Untuk skema III mendapatkan waktu training yang paling cepat yaitu selama 36.25 menit. berdasarkan hasil mAP tersebut, modifikasi terhadap jumlah *batch size* dan penggunaan ukuran *image size* yang tinggi dapat mempengaruhi performa dan kinerja deteksi objek untuk *autonomous driving*.

Kata kunci: Autonomous Driving, Batch Size, Object Detection, YOLOv8, mAP.

1. PENDAHULUAN

Teknologi Kecerdasan Buatan atau Artificial Intelligence (AI) tengah mengalami perkembangan yang cepat sejalan dengan kemajuan zaman [1]. Salah satu aspek dalam AI yakni computer vision khususnya dalam hal Deteksi Objek terus mengalami perkembangan dengan berbagai fitur yang dirancang untuk mendukung kebutuhan industri dan mempermudah tugas manusia dalam menjalankan pekerjaan tertentu [2]. Pengenalan objek atau deteksi objek dapat diaplikasikan pada berbagai sistem salah satunya autonomous driving [3]. Autonomous driving adalah sebutan untuk kendaraan yang mampu mengendalikan kendaraan tanpa kendali manusia. Autonomous driving juga merupakan sebuah sistem cerdas kompleks yang terdiri dari beberapa subsistem yang menangani banyak sub tugas dari fase persepsi hingga fase kontrol [4]. Algoritma pendeteksi objek berfungsi untuk mengidentifikasi dan memprediksi objek disekitar kendaraan [5]. Algoritma objek deteksi yang digunakan adalah YOLOv8. YOLOv8 adalah versi terbaru dari YOLO yang dibuat oleh Ultralytics. YOLO adalah model algoritma deteksi satu tahap (one-stage detection) yang memiliki kecepatan deteksi yang sangat tinggi [6], dengan kecepatan deteksi yang sangat tinggi ini merupakan hal penting untuk melakukan deteksi secara real-time pada kendaraan otonom.

Dalam era perkembangan teknologi autonomous driving yang semakin pesat, muncul beberapa permasalahan krusial yang perlu dipecahkan dalam konteks deteksi objek untuk autonomous driving [7]. Salah satu tantangan utama dalam pengembangan kendaraan otonom adalah meningkatkan akurasi dan kecepatan sistem deteksi objek [8]. Sistem deteksi objek yang akurat dan cepat sangat penting untuk memastikan keamanan dan kenyamanan pengguna kendaraan otonom. Kendaraan otonom perlu memiliki kemampuan untuk mendeteksi objek-objek dengan tingkat akurasi yang tinggi dalam waktu nyata [9]. Kendaraan otonom dapat beroperasi secara aman dan efisien di berbagai situasi yang mungkin terjadi dalam dunia nyata [10]. Dalam konteks penelitian ini, permasalahan-permasalahan ini menjadi sangat relevan mengingat pengembangan autonomous driving sedang mengalami pertumbuhan yang signifikan [11]. Oleh karena itu, perlu adanya analisis terhadap metode deteksi objek seperti YOLOv8. YOLOv8 digunakan untuk mengatasi tantangan-tantangan ini dan memastikan bahwa autonomous driving dapat memiliki akurasi yang tinggi dalam mendeteksi objek dan dapat melakukan deteksi secara real-time. YOLOv8 dipilih karena merupakan versi terbaru dari YOLO yang memiliki arsitektur jaringan yang ringan, metode penggabungan fitur yang efektif, dan hasil deteksi yang lebih akurat [12].

Studi – studi sebelumnya dalam bidang pengolahan citra dan deteksi objek untuk kendaraan

otonom telah melakukan deteksi objek untuk kendaraan otonom. Misalnya, penelitian oleh [3] pada penelitian ini menggunakan metode Complex YOLOv4 dan menggunakan dataset 3D point cloud LIDAR dari KITTI Vision Benchmark. Pada penelitian ini, telah berhasil dilakukan penyesuaian terhadap hyperparameter dengan hasil yang lebih unggul dibandingkan dengan model Complex YOLOv4 asli. Hasil kinerja paling optimal, seperti yang diukur dengan nilai rata-rata presisi per area (mAP) dalam berbagai skenario pengujian yang telah dilakukan, ditemukan pada model dalam skenario ketiga dengan mAP mencapai 58.3%. Penelitian selanjutnya oleh [5] pada penelitian ini menggunakan YOLOv4, untuk dataset yang digunakan adalah dataset KITTI dan BDD100k. Dalam hasil eksperimen yang dilakukan, terlihat bahwa model 1 yang telah diperbaharui menunjukkan tingkat akurasi yang lebih tinggi dibandingkan dengan YOLOv4 aslinya dalam tugas deteksi objek. Peningkatan ini terutama tercermin dalam peningkatan mean average precision (mAP) sebesar 2,06% pada set validasi KITTI dan 2,95% pada set validasi BDD. Selain itu, mAP50 dari model 2 juga mengalami peningkatan sebesar 1,73%, sementara kecepatan inferensi model tersebut meningkat sebesar 4,83 fps. Penelitian selanjutnya oleh [8] penelitian ini menggunakan YOLOv4 dengan menggunakan beberapa fitur diantaranya Weighted-Residual-Connections, Cross-Stage-Partial-connections, Cross mini-Batch Normalization, Self-adversarial-training, Mish-activation, dan CioU loss. Dengan menggabungkan beberapa fitur tersebut mendapatkan hasil 43,5% AP(65,7% AP50) untuk dataset MS COCO.

Pada penelitian ini bertujuan untuk menganalisis performa YOLOv8 dalam melakukan deteksi objek untuk autonomous driving. Penelitian ini akan melakukan modifikasi terhadap hyperparameter YOLOv8. Hyperparameter yang digunakan pada penelitian ini adalah epoch, batch size, dan image size [13]. Pada penelitian ini diharapkan dapat memberikan wawasan mengenai pengaruh dari pengaturan hyperparameter pada metode YOLOv8 dalam mendeteksi objek khususnya dalam konteks autonomous driving. Pada penelitian ini juga diharapkan dapat memberikan hasil pengaturan hyperparameter yang optimal.

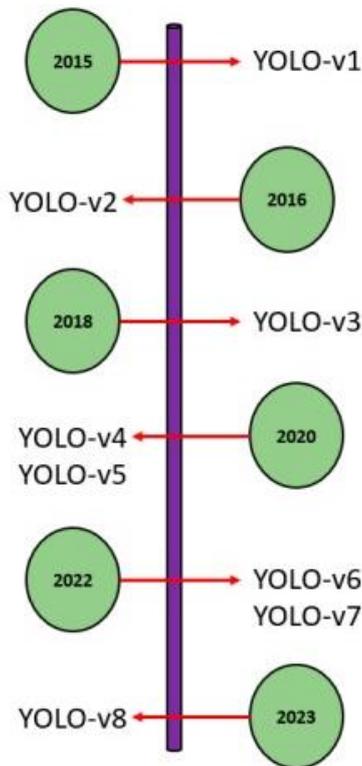
2. KAJIAN TEORI

2.1. *Autonomous Driving*

Autonomous driving adalah istilah yang digunakan untuk kendaraan yang dapat beroperasi tanpa kendali manusia. Fitur utama dari pengemudi otomatis adalah kemampuannya untuk mendeteksi dengan akurat dan cepat kendaraan lain, pejalan kaki, dan objek lain di sekitarnya. Kemampuan deteksi ini sangat penting untuk mencegah terjadinya kecelakaan lalu lintas.

2.2. You Only Look Once

Kemampuan yang sangat penting dalam mencegah kecelakaan lalu lintas pada autonomous driving adalah kemampuan deteksi objek yang akurat dan cepat. Salah satu metode yang digunakan untuk mendeteksi objek secara real-time adalah You Only Look Once (YOLO)[14] Seiring dengan kemajuan teknologi, terdapat beberapa versi terbaru dari metode YOLO yang bertujuan untuk meningkatkan kualitasnya seperti pada Gambar 1. YOLOv8, sebagai versi terbaru, menonjol di antara versi-versi lainnya dengan performansi deteksi dan kecepatan deteksi yang paling unggul.



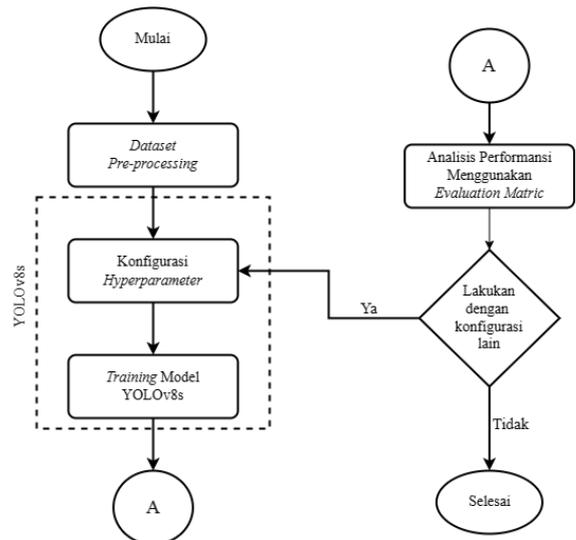
Gambar 1. YOLO Evaluation Timeline

2.3. YOLOv8

Pada Januari 2023, Ultralytics, perusahaan yang mengembangkan YOLOv5, merilis YOLOv8 dengan memperkenalkan lima versi yang berbeda dalam skala: YOLOv8n (nano), YOLOv8s (kecil), YOLOv8m (sedang), YOLOv8l (besar), dan YOLOv8x (ekstra besar). YOLOv8 dapat digunakan untuk berbagai tugas visi komputer, termasuk deteksi objek, segmentasi, estimasi pose, pelacakan, dan klasifikasi. YOLOv8 mempertahankan struktur dasar yang mirip dengan YOLOv5 namun mengalami perubahan pada CSPLayer, yang sekarang disebut sebagai modul C2f. Modul C2f ini (cross-stage partial bottleneck dengan dua konvolusi) menggabungkan fitur tingkat tinggi dengan informasi kontekstual untuk meningkatkan akurasi deteksi [9], [15]. Pada penelitian ini YOLOv8 yang digunakan adalah YOLOv8s.

3. METODE PENELITIAN

Dalam penelitian ini, akan dievaluasi kinerja deteksi objek menggunakan metode YOLOv8s untuk autonomous driving. Penelitian ini mencakup eksperimen dengan memodifikasi tiga hyperparameter metode YOLOv8s, yaitu *epoch*, *batch size*, dan *image size*. Hasil dari berbagai skenario pengujian akan dianalisis dengan mengukur metrik evaluasi masing-masing model. Secara keseluruhan, alur kerja sistem penelitian ini dapat dilihat dalam diagram alir seperti yang tergambar pada Gambar 2 dibawah ini.



Gambar 2. Tahapan Metode YOLOv8s

3.1. Dataset Preprocessing

Dataset yang digunakan adalah data citra skema lalu lintas dari CARLA [16] yang di peroleh dari roboflow universe [17]. Pada dataset yang digunakan terdapat 1211 data sampel untuk *training*, 343 data sampel untuk *validation*, dan 165 data sampel untuk *testing*. Dataset yang digunakan terdapat beberapa kelas. untuk daftar kelas dapat dilihat pada Tabel 1.

Tabel 1. Daftar Kelas Dataset

NO	Kelas	NO	Kelas
1	Vehicle	6	Traffic light red
2	Moto bike	7	Traffic sign 30
3	Bike	8	Traffic sign 60
4	Traffic light green	9	Traffic sign 90
5	Traffic light orange	10	Person

3.2. Konfigurasi Hyperparameter

Pada penelitian ini menggunakan skema konfigurasi *hyperparameter* diantaranya berupa *epoch*, *batch size*, dan *image size*. Konfigurasi hyperparameter dilakukan untuk memperoleh skema pengujian terbaik dalam mendeteksi objek secara *real-time* dan akurat. Penelitian ini menggunakan konfigurasi *batch size* dengan nilai 16, 32, dan 64. Untuk semua skema pengujian serta konfigurasi *epoch* dengan jumlah 100. Sedangkan untuk konfigurasi *image size* ukuran yang digunakan antara

lain adalah 416×416 , 512×512 dan 640×640 . Pada penelitian ini terdapat beberapa skema konfigurasi seperti yang terlihat pada Tabel 2.

Tabel 2. Skema Konfigurasi Hyperparameter

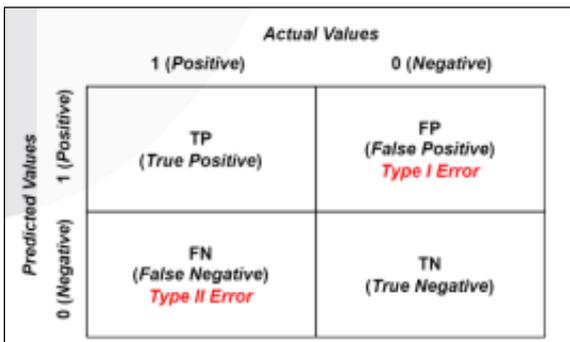
Skema	Epoch	Batch Size	Image Size
I	100	16	416 x 416
II	100	32	416 x 416
III	100	64	416 x 416
IV	100	16	512 x 512
V	100	32	512 x 512
VI	100	64	512 x 512
VII	100	16	640 x 640
VIII	100	32	640 x 640
IX	100	64	640 x 640

3.3. Training Model YOLOv8s

Pada tahap ini dilakukan proses *training* model YOLOv8s menggunakan Google Colab. Proses ini dilakukan untuk mengetahui kinerja dari setiap skema pengujian. Tahap pertama dilakukan *mount to* Google Drive yang bertujuan untuk mengakses file yang tersimpan pada Google Drive. Setelah dilakukan *mount to* Google Drive, tahap selanjutnya adalah melakukan perintah *Git Clone* dan *Install Requirements* untuk membuat salinan repositori dalam Google Drive dan menginstall beberapa persyaratan untuk menjalankan proses training skema pengujian menggunakan YOLOv8. Lalu dilakukan proses training berdasarkan skema pengujian yang digunakan dan dilakukan evaluasi hasil performansi menggunakan *evaluation metrics*. Berdasarkan skema pengujian yang telah dirancang, proses training model dilakukan sebanyak 27 percobaan seperti yang terlihat pada Tabel 2. Setelah dilakukan proses *training* model, selanjutnya akan dilakukan analisis performansi menggunakan *evaluation metrics*.

3.4. Evaluation Metrics

Pada penelitian ini digunakan *evaluation metrics* sebagai cara untuk mengukur seberapa baik model machine learning yang sudah dirancang. Pada penelitian ini *evaluation metrics* yang digunakan antara lain adalah *Precision*, *Recall*, *Average Precision* (AP), dan *Mean Average Precision* (mAP) [18]. Selain itu, terdapat *confusion matrix* atau *error matrix*.



Gambar 3. Confusion Matrix [19]

Confusion matrix memberikan informasi perbandingan hasil klasifikasi model dengan hasil klasifikasi sebenarnya. Matriks kesalahan ini berupa tabel yang menggambarkan kinerja suatu model seperti pada Gambar 3.

Precision adalah perbandingan antara *True Positive* (TP) dengan jumlah keseluruhan data yang diprediksi sebagai *True Positive* (TP) dan *False Positive* (FP), sebagaimana dijelaskan dalam persamaan 1. Presisi menggambarkan sejauh mana tingkat keakuratan antara data yang diharapkan dengan hasil prediksi yang diberikan oleh model.

$$Precision = \frac{TP}{TP+FP} \tag{1}$$

Recall adalah rasio antara prediksi *True Positive* (TP) dan jumlah keseluruhan data yang seharusnya diprediksi sebagai *True Positive* (TP) dan *False Negative* (FN) sebagaimana dijelaskan dalam persamaan 2. *Recall* mencerminkan tingkat kesuksesan model dalam mengidentifikasi kembali suatu informasi[20].

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

Average Precision (AP) adalah metode untuk menggambarkan nilai presisi dan *recall* secara keseluruhan dalam satu nilai, mencerminkan rata-rata presisi secara umum atau rata-rata akurasi dari model[21]. Perhitungan AP dapat dilihat pada persamaan 3.

$$AP = \int_0^1 p(r)dr \tag{3}$$

Mean Average Precision (mAP) merupakan nilai rata-rata *Average Precision* (AP) dari suatu objek pada setiap kelas. Perhitungan mAP dapat dilihat pada persamaan 4[22].

$$mAP = \frac{1}{k} \sum_{i=1}^k AP_i \times 100\% \tag{4}$$

4. HASIL DAN PEMBAHASAN

Berdasarkan hasil pengujian metrics evaluasi, seperti *Precision*, *Recall*, *Average Precision* (AP), dan *Mean Average Precision* (mAP), model yang akan dijadikan acuan adalah model yang telah melalui proses pelatihan. Hal ini dikarenakan metrics evaluasi yang diperoleh merupakan hasil modifikasi dan optimalisasi dari pelatihan model YOLOv8. Pada setiap pelatihan, jumlah epoch yang digunakan adalah 100, dan ukuran batch size yang digunakan adalah 16, 32, dan 64. Selain itu, akan dilakukan modifikasi pada konfigurasi ukuran jaringan pada setiap epoch dengan ukuran 416×416 , 512×512 dan 640×640 . Model dengan nilai mAP mendekati 100% dianggap sebagai yang terbaik. Untuk hasil dari semua pelatihan pada penelitian ini dapat dilihat pada Tabel 3.

Tabel 3. Hasil Eksperimen

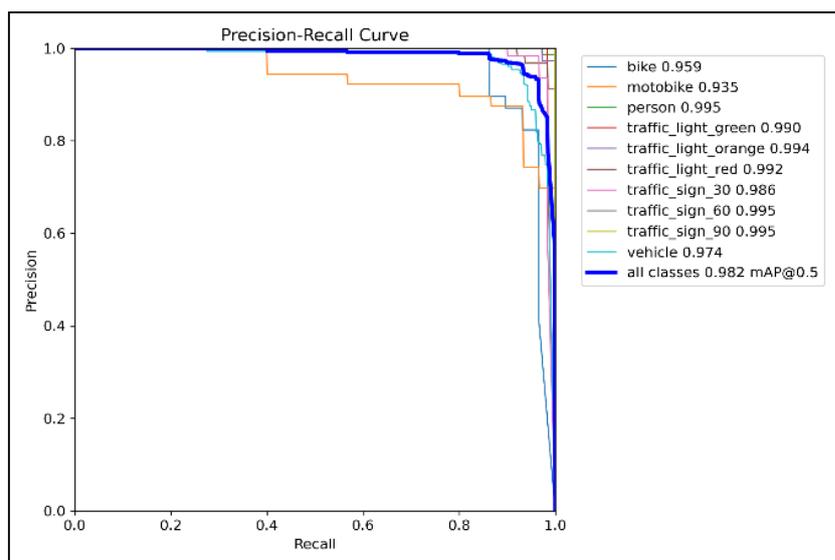
Skema	Precision	Recall	mAP50	mAP 50-95	Time (minute)
I	97%	91.4%	96.5%	74.6%	43.44
II	95.4%	92.8%	95.8%	74.8%	40.39
III	95.4%	93.5%	96.4%	74.6%	36.25
IV	93.5%	94.4%	97.3%	76%	61.03
V	94.9%	94.8%	97.6%	76.2%	52.26
VI	94.9%	93.6%	97.3%	75.5%	48.28
VII	95.3%	95.5%	97.5%	77%	64.51
VIII	95.5%	95.4%	98.2%	76.8%	59.45
IX	95.1%	95.6%	97.5%	77.1%	60.33

Hasil training Skema menunjukkan performa yang menjanjikan dengan variasi konfigurasi image size dan batch size. Secara umum, skema dengan *image size* yang lebih besar cenderung meraih akurasi (*precision*) dan mAP (*mean Average Precision*) yang lebih tinggi. Skema VII (640 x 640) mencapai akurasi tertinggi 95.3% dan Skema VIII (640 x 640) mencapai mAP tertinggi 98.2% sebagaimana terdapat pada Gambar 4. Namun, peningkatan *image size* juga berdampak pada waktu training yang lebih lama, dengan Skema VII membutuhkan waktu 64.51 menit dan Skema VIII 59.45 menit.

Skema dengan *batch size* yang lebih besar juga cenderung berprestasi lebih baik. Skema I (*batch size* 16, *image size* 416 x 416) mencapai akurasi tertinggi 97.0% dan Skema IV (*batch size* 16, *image size* 512 x 512) mencapai mAP tertinggi 97.3%. Namun, sama seperti *image size*, peningkatan *batch size* juga berdampak pada waktu *training* yang lebih lama, dengan Skema I membutuhkan waktu 43.44 menit dan Skema IV 61.03 menit.

Menariknya, Skema III (*batch size* 64, *image size* 416 x 416) memiliki waktu *training* tercepat (36.25 menit) namun tetap menunjukkan performa yang baik dengan akurasi 95.4% dan mAP 96.4%. Ini menjadikannya pilihan menarik jika efisiensi waktu training diprioritaskan tanpa mengorbankan performa secara signifikan.

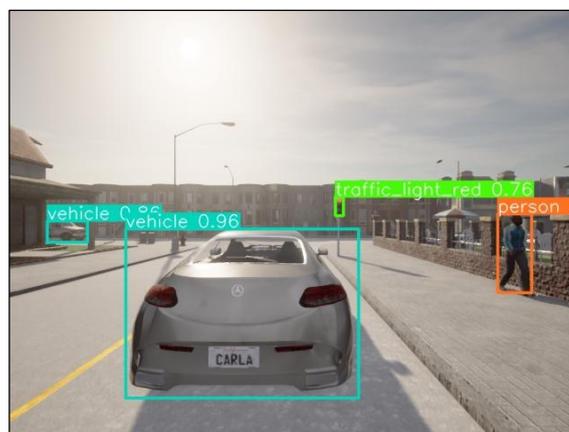
Secara keseluruhan, hasil training Skema menunjukkan trade-off antara akurasi, kecepatan training, dan konfigurasi. Pilihan skema terbaik bergantung pada prioritas, apakah mengejar akurasi tertinggi seperti pada skema VIII dan Gambar 4, waktu training tercepat, atau keseimbangan keduanya.



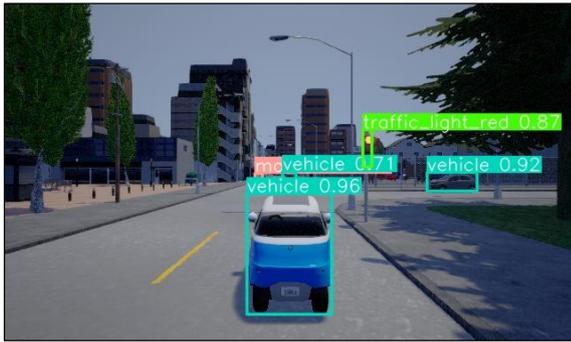
Gambar 4. Grapik Precision Recal Curve Skema VII

Seperti yang terlihat pada *Precision-Recall Curve* diatas maka dapat disimpulkan performa model terlihat baik pada setiap kelas. Pada setiap kelas mendapatkan nilai rata - rata AP diatas 90% dan untuk semua kelas memiliki nilai mAP50 sebesar 98.2%.

Pada skema VIII juga dilakukan *testing* dengan menggunakan data *testing* yang berbeda dengan data *training*. Skema VIII dipilih karena memiliki nilai mAP yang paling tinggi dari semua skema training. Berikut ini beberapa hasil dari *testing* menggunakan model skema VIII. Untuk hasil *testing* dapat dilihat pada Gambar 5 dan 6 dibawah ini.



Gambar 4. Testing 1 Skema XXIV



Gambar 5. Testing 2 Skema XXIV

Berdasarkan hasil testing menggunakan dataset yang berbeda dari data training, dapat disimpulkan bahwa model hasil dari skema training VIII dapat melakukan deteksi dengan baik.

5. DISKUSI

Pada penelitian sebelumnya yang membahas mengenai analisis performansi deteksi objek pada metode *complex YOLOv4* untuk *autonomous driving*. Pada penelitian ini, telah berhasil dilakukan penyesuaian terhadap *hyperparameter* dengan hasil yang lebih unggul dibandingkan dengan model *Complex YOLOv4* asli. Hasil kinerja paling optimal, seperti yang diukur dengan nilai mAP dalam berbagai skenario pengujian yang telah dilakukan, ditemukan pada model skenario ketiga mendapatkan mAP mencapai 58.3% [3]. Pada penelitian ini juga memiliki tujuan yang sama yaitu dengan melakukan modifikasi pada *hyperparameter* metode YOLO, tetapi yang membedakan antaran penelitian ini dengan penelitian sebelumnya adalah terletak pada *hyperparameter* yang digunakan. Pada penelitian sebelumnya berfokus pada *hyperparameter* yang digunakan adalah *epoch* dan *network size* (pada penelitian ini disebut *image size*). Sedangkan untuk penelitian ini penulis berfokus pada *hyperparameter batch size*, dan *image size* dan juga pada penelitian ini menghitung berapa lama waktu training berjalan pada setiap skema. Perbedaan lainnya adalah versi YOLO yang digunakan, pada penelitian sebelumnya menggunakan *Complex YOLOv4* sedangkan pada penelitian ini menggunakan YOLOv8 atau lebih tepatnya YOLOv8s. Perbedaan lainnya terdapat pada dataset yang digunakan pada dataset penelitian sebelumnya menggunakan dataset KITTI sedangkan pada penelitian ini menggunakan dataset Carla simulator. Hasil penelitian ini menunjukkan bahwa YOLOv8 dengan pengaturan *hyperparameter epoch* 100, *batch size* 32, dan *image size* 640 x 640 memiliki performa yang paling baik berdasarkan nilai mAP yaitu sebesar 98,2% dan mAP50-95 sebesar 76,8%. Hasil tersebut menunjukkan bahwa model YOLOv8 mampu mendeteksi objek dengan akurasi yang tinggi. Selain itu, hasil penelitian juga menunjukkan bahwa YOLOv8 dengan pengaturan *hyperparameter epoch* 100, *batch size* 64, dan *image size* 416 x 416 memiliki waktu training yang paling cepat, yaitu 36,25 menit

serta memiliki nilai mAP50 sebesar 95.5% dan mAP50-95 sebesar 71,1%.

6. KESIMPULAN

Penelitian ini menunjukkan bahwa modifikasi *hyperparameter* dapat meningkatkan performa model YOLO. Baik itu hasil dari penelitian sebelumnya yang menggunakan *Complex YOLOv4* yang mendapatkan peningkatan pada nilai mAP di dibandingkan dengan model *Complex YOLOv4* asli. Hasil kinerja paling optimal, seperti yang diukur dengan mAP dalam berbagai skenario pengujian yang telah dilakukan, ditemukan pada model dalam skenario ketiga dengan mAP mencapai 58.3%. Adapun pada penelitian ini yang menggunakan YOLOv8 mendapatkan nilai terbaik pada skema ke delapan dengan nilai mAP 98.2% dan untuk waktu *training* paling cepat didapatkan pada skema ketiga dengan waktu 36.25 menit. Pada Penelitian ini memberikan informasi tentang waktu training yang dibutuhkan untuk YOLOv8 dengan berbagai pengaturan *hyperparameter*. Secara keseluruhan, penelitian ini telah berhasil menganalisis pengaruh *hyperparameter batch size*, dan *image size* terhadap performa YOLOv8 dalam mendeteksi objek untuk *autonomous driving*. Hasil training berbagai skema juga menunjukkan *trade-off* antara akurasi, kecepatan training, dan konfigurasi. Pilihan skema terbaik bergantung pada prioritas, apakah mengejar akurasi tertinggi, waktu training tercepat, atau keseimbangan keduanya. Pada penelitian ini perlu dilanjutkan dengan menggunakan versi lain dari model YOLO atau melakukan perbandingan dengan metode objek deteksi lain.

DAFTAR PUSTAKA

- [1] K. R. Chowdhary, *Fundamentals of artificial intelligence*. Springer India, 2020. doi: 10.1007/978-81-322-3972-7.
- [2] N. Jannah, A. Wibowo, and S. Siadari, 'Eksplorasi Fitur Untuk Peningkatan Kinerja Deteksi Objek Berbasis Pada Pesawat Tanpa Awak', *e-Proceeding of Engineering*, vol. 8, p. 2943, 2022.
- [3] H. Nurhadiati, A. S. Wibowo, and A. Pratondo, 'Analisis Performansi Deteksi Objek Pada Metode Complex YOLOv4 Untuk Autonomous Driving', Bandung, Dec. 2022.
- [4] O. Natan and J. Miura, 'End-to-End Autonomous Driving With Semantic Depth Cloud Mapping and Multi-Agent', *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 557–571, 2023, doi: 10.1109/TIV.2022.3185303.
- [5] R. Wang *et al.*, 'A Real-Time Object Detector for Autonomous Vehicles Based on YOLOv4', *Comput Intell Neurosci*, vol.

- 2021, 2021, doi: 10.1155/2021/9218137.
- [6] T. A. A. H. Kusuma, K. Usman, and S. Saidah, 'PEOPLE COUNTING FOR PUBLIC TRANSPORTATIONS USING YOU ONLY LOOK ONCE METHOD', *Jurnal Teknik Informatika (Jutif)*, vol. 2, no. 1, pp. 57–66, Feb. 2021, doi: 10.20884/1.jutif.2021.2.2.77.
- [7] J. Choi, D. Chun, H. Kim, and H.-J. Lee, 'Gaussian YOLOv3: An Accurate and Fast Object Detector Using Localization Uncertainty for Autonomous Driving', 2019.
- [8] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, 'YOLOv4: Optimal Speed and Accuracy of Object Detection', Apr. 2020, [Online]. Available: <http://arxiv.org/abs/2004.10934>
- [9] J. Terven and D. Cordova-Esparza, 'A Comprehensive Review of YOLO: From YOLOv1 and Beyond', *ArXiv*, Apr. 2023, [Online]. Available: <http://arxiv.org/abs/2304.00501>
- [10] Y. Cai *et al.*, 'YOLOv4-5D: An Effective and Efficient Object Detector for Autonomous Driving', *IEEE Trans Instrum Meas*, vol. 70, 2021, doi: 10.1109/TIM.2021.3065438.
- [11] A. Sarda, S. Dixit, and A. Bhan, 'Object detection for autonomous driving using YOLO [You only Look Once] algorithm', in *Proceedings of the 3rd International Conference on Intelligent Communication Technologies and Virtual Mobile Networks, ICICV 2021*, Institute of Electrical and Electronics Engineers Inc., Feb. 2021, pp. 1370–1374. doi: 10.1109/ICICV50876.2021.9388577.
- [12] H. Lou *et al.*, 'DC-YOLOv8: Small-Size Object Detection Algorithm Based on Camera Sensor', *Electronics (Switzerland)*, vol. 12, no. 10, May 2023, doi: 10.3390/electronics12102323.
- [13] B. Mahaur and K. K. Mishra, 'Small-object detection based on YOLOv5 in autonomous driving systems', *Pattern Recognit Lett*, vol. 168, pp. 115–122, Apr. 2023, doi: 10.1016/J.PATREC.2023.03.009.
- [14] L. Suroiyah, Y. Rahmawati, and R. Dijaya, 'FACEMASK DETECTION USING YOLO V5', *Jurnal Teknik Informatika (JUTIF)*, vol. 4, no. 6, pp. 1277–1286, 2023, doi: 10.52436/1.jutif.2023.4.6.1043.
- [15] I. Purwita Sary, E. Ucok Armin, S. Andromeda, E. Engineering, and U. Singaperbangsa Karawang, 'Performance Comparison of YOLOv5 and YOLOv8 Architectures in Human Detection Using Aerial Images', *Ultima Computing: Jurnal Sistem Komputer*, vol. 15, no. 1, 2023.
- [16] A. Dosovitskiy, G. Ros, F. Codevilla, A. López, and V. Koltun, 'CARLA: An Open Urban Driving Simulator', 2017.
- [17] FYP, 'FYP2022 Dataset'. Accessed: Jul. 08, 2023. [Online]. Available: <https://universe.roboflow.com/fyp-5uyrm/fyp2022>
- [18] Shah Deval, 'Mean Average Precision (mAP) Explained: Everything You Need to Know', <https://www.v7labs.com/blog/mean-average-precision>.
- [19] M. Hussain, 'YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection', *Machines*, vol. 11, no. 7. Multidisciplinary Digital Publishing Institute (MDPI), Jul. 01, 2023. doi: 10.3390/machines11070677.
- [20] R.-C. Chen, C. Dewi, Y.-C. Zhuang, and J.-K. Chen, 'Contrast Limited Adaptive Histogram Equalization for Recognizing Road Marking at Night Based on Yolo Models', *IEEE Access*, vol. 11, pp. 92926–92942, 2023, doi: 10.1109/ACCESS.2023.3309410.
- [21] J. Fan, T. Huo, and X. Li, 'A review of one-stage detection algorithms in autonomous driving', in *2020 4th CAA International Conference on Vehicular Control and Intelligence, CVCI 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 210–214. doi: 10.1109/CVCI51460.2020.9338663.
- [22] M. A. Yahya, S. Abdul-Rahman, and S. Mutalib, 'Object detection for autonomous vehicle with Lidar using deep learning', in *2020 IEEE 10th International Conference on System Engineering and Technology, ICSET 2020 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., Nov. 2020, pp. 207–212. doi: 10.1109/ICSET51301.2020.9265358.