e-ISSN : 2621 – 5934
P-ISSN : 2621 – 7112

Komparasi Algoritma You Only Look Once (YOLO) Dan Real-Time Detection Transformer (RT-DETR) Untuk Menghitung Jumlah Kendaraan Bermotor

Deden Fadhli[1], Wargijono Utomo[2], Nur Hikmah[3]

[1,2,3] Jurusan Teknik Informatika, Fakultas Teknik, Universitas Krisnadwipayana, Jakarta

Abstrak. Masalah lalu lintas menjadi tantangan utama dalam manajemen kota, terutama di negara berkembang sebagai perhatian global. Peningkatan jumlah kendaraan menyebabkan peningkatan trafik di jalan raya yang tidak diimbangi dengan perluasan jalan, sehingga menyebabkan kemacetan, terutama di dalam kota. Penelitian ini membandingkan kinerja dua algoritma deteksi objek, yaitu *You Only Look Once* (YOLO) dan *Real-Time Detection Transformer* (RT-DETR), dalam menghitung jumlah kendaraan bermotor seperti, motor, mobil, bus, truk, dan pickup. Penelitian ini diuji dengan menggunakan data video berdurasi 1 menit 11 detik yang diambil dari atas jembatan penyeberangan orang (JPO) pada 3 tempat yang berbeda di Bekasi (Caman Raya, Stadiun Bekasi, dan Mall Living Plaza) dengan kondisi jalan ramai lancar dan waktu yang berbeda (pagi, siang, dan malam). Hasil penelitian menunjukkan bahwa algoritma RT-DETR memiliki akurasi deteksi lebih tinggi dibandingkan YOLO. RT-DETR berhasil mengungguli YOLO dalam menghitung kendaraan di berbagai kondisi yang diuji terutama dalam pengujian kondisi pencahayaan yang berbeda. RT-DETR memiliki rata-rata akurasi deteksi untuk seluruh jenis kendaraan sebesar 0,86 sedangkan YOLO sebesar 0,79. RT-DETR juga menunjukkan performa yang lebih konsisten pada berbagai jenis kendaraan yang diuji. Berdasarkan hasil ini, RT-DETR lebih unggul dalam menghitung kendaraan bermotor dibandingkan YOLO terhadap berbagai kondisi yang telah diuji.

Kata kunci— YOLO, RT-DETR, *Counting Vehicles, Object Detection, Computer Vision*

1. PENDAHULUAN

Masalah lalu lintas adalah tantangan utama dalam manajemen kota, terutama di negara berkembang, dengan kemacetan menjadi perhatian global. Ahli, pemerintah, teknisi, dan peneliti tertarik mencari solusi untuk masalah ini [1]. Peningkatan jumlah kendaraan menyebabkan meningkatnya trafik di jalan raya, sementara lebar jalan tidak bertambah signifikan, sehingga mengakibatkan kemacetan terutama di dalam kota. Untuk mengatasi kemacetan ini, perlu dirumuskan strategi rekayasa trafik yang efektif. Rekayasa trafik membutuhkan data kondisi jalan dan kendaraan, termasuk jumlah kendaraan pada suatu ruas jalan dari waktu ke waktu [2]. Untuk menentukan jumlah kendaraan, bila mengandalkan perhitungan manual rawan kesalahan serta membutuhkan banyak waktu dan tenaga [3]. Dengan kemajuan teknologi, para peneliti mencoba mengembangkan alat pendeteksi kendaraan otomatis untuk memudahkan dalam menghitung jumlah kendaraan. Salah satu inovasi tersebut adalah alat pendeteksi kendaraan yang menggunakan algoritma *Deep Learning* pada jaringan pendeteksinya [4].

Di era teknologi yang pesat, deteksi objek dalam *computer vision* memungkinkan identifikasi objek dalam gambar atau video. *Computer vision* mempelajari bagaimana komputer dapat melihat dan menganalisis objek dalam gambar. Algoritma deteksi objek yang umum digunakan saat ini adalah algoritma *Deep Learning* [5]. Pada penelitian ini akan menyoroti dua algoritma utama yaitu, YOLO dan RT-DETR.

You Only Look Once (YOLO) adalah sebuah algoritma yang diperkenalkan oleh Joseph Redmon pada tahun 2016 yang digunakan sebagai alat mengolah deteksi objek. YOLO bisa digunakan sebagai alat memperoses deteksi objek secara *real time*. Apabila algoritma ini dibedakan dengan sistem deteksi objek *real time* yang lain, YOLO mempunyai mAP dan FPS yang lebih besar dibandingkan yang lain [6]. *Real-Time Detection Transformer* (RT-DETR) yang diperkenalkan pada tahun 2023, merupakan kerangka kerja deteksi objek *end-to-end* yang inovatif berdasarkan arsitektur transformer. RT-DETR baru-baru ini menghasilkan hasil yang luar biasa. RT-DETR memanfaatkan mekanisme uniknya untuk unggul dalam menangani informasi global, yang menunjukkan potensi besar untuk menjadi pendekatan baru dalam deteksi cacat. Selain

meningkatkan kualitas kueri dan perubahan kecepatan yang dapat disesuaikan tanpa pelatihan ulang, RT-DETR mengelola fitur multi-skala dengan baik. Dukungan untuk menggunakan CLI/PIP semakin menekankan betapa mudahnya penggunaannya [7].

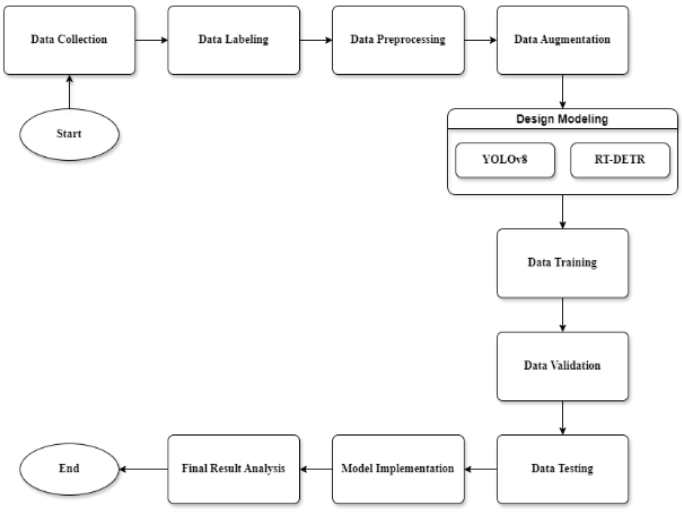
Untuk implementasinya dengan *learning rates* dan *epochs* yang konsisten ditetapkan untuk semua model, serta *batch size* yang disesuaikan. Metrik kinerja menunjukkan bahwa YOLOv8 unggul dengan *mean Average Precision* (mAP) sebesar 92,6 dan RT-DETR dengan 86,4 [8]. YOLO merupakan pilihan yang sangat baik untuk aplikasi yang memerlukan deteksi waktu nyata dengan fokus pada kecepatan, sehingga cocok untuk aplikasi seperti analisis video dan pelacakan objek langsung. Di sisi lain, RT-DETR unggul dalam tugas-tugas yang menuntut akurasi yang lebih baik dan menangani interaksi yang kompleks antara objek, yang mungkin sangat penting dalam bidang-bidang seperti pencitraan medis dan deteksi objek berbutir halus, serta mengklaim bahwa kinerja yang unggul baik dalam kecepatan maupun akurasi dibandingkan dengan semua detektor YOLO dengan skala yang sama [9].

Penelitian tentang penghitungan kendaraan bermotor penting karena pengaruhnya terhadap lalu lintas jalan. Memperbandingkan antara algoritma seperti YOLO dan RT-DETR perlu dilakukan karena keduanya memiliki pendekatan yang berbeda terkait kecepatan dan akurasi deteksi. Oleh karena itu, akan dilakukan perbandingan secara spesifik yaitu, dalam konteks pemantauan lalu lintas untuk pengambilan hasil penghitungan kendaraan secara *real-time*.

Pemilihan YOLO dan RT-DETR sebagai subjek penelitian didasarkan atas keunggulan dan kelemahan keduanya yang berbeda-beda, yang membuatnya menarik untuk diperbandingkan dan diteliti lebih lanjut dalam konteks spesifik seperti penghitungan kendaraan. Dengan membandingkan kedua algoritma ini, penelitian ini diharapkan memberikan wawasan tentang pengoptimalan algoritma deteksi objek untuk mendukung pengembangan sistem transportasi yang lebih cerdas dan efektif.

2. Metode

Metode penelitian yang akan digunakan untuk mencapai tujuan penelitian telah dirancang dan terdiri dari beberapa tahapan yang jelas. Gambar 1 adalah rincian metode lengkapnya:



Gambar 1 Diagram Alir Metodologi Penelitian

A. Pengumpulan Data (Data Collection)

Pengumpulan data (*Data Collection*) adalah suatu proses sistematis untuk mengumpulkan dan mengukur informasi dari berbagai sumber untuk memperoleh gambaran yang lengkap dan akurat tentang suatu fenomena yang diminati. Tahapan pertama dalam penelitian ini adalah mengumpulkan data yang diperlukan untuk melatih dan menguji model dari algoritma YOLO dan RT-DETR. Data yang digunakan terdiri dari video dan gambar yang diambil dari lingkungan jalan raya, yang mengandung berbagai jenis kendaraan bermotor seperti mobil, motor, bus, pickup, dan truk. Data berupa video digunakan sebagai bahan uji untuk dijadikan program

penghitungan kendaraan sedangkan data berupa gambar digunakan sebagai dataset yang nantinya akan menjadi sebuah model algoritma.

Data yang diambil merupakan data dengan sudut pandang yang terlihat dari atas karena jangkauan penglihatannya akan terlihat lebih luas agar memperlihatkan seluruh arus jalan dan menghindari terjadinya tumpang tindih pada objek. Data ini diperoleh dengan mengumpulkan secara langsung dari lapangan, yaitu dari atas Jembatan Penyebrangan Orang (JPO). Kumpulan data ini akan dijadikan sebagai dataset sekaligus bahan uji. Jumlah data yang dikumpulkan untuk dijadikan sebagai dataset yaitu sebanyak 1150 gambar.

B. Pelabelan Data (Data Labeling)

Setelah data terkumpul, langkah selanjutnya adalah memberi label pada data tersebut. Pelabelan data (*Data Labeling*) adalah proses memberi label atau anotasi pada data mentah untuk membuatnya lebih bermakna dan dapat digunakan dalam model pembelajaran mesin (*machine learning*) atau analisis data lainnya. Pelabelan data sangat penting dalam pengembangan model pembelajaran yang memerlukan data yang terstruktur dan teranotasi untuk pelatihan dan evaluasi.

Proses *labeling* dilakukan untuk menentukan lokasi (*bounding box*) dan jenis setiap kendaraan yang muncul dalam gambar atau video. *Labeling* dapat dilakukan secara manual atau menggunakan alat bantu semi-otomatis. Alat yang digunakan dalam tahap *labeling* ini menggunakan roboflow dengan bantuan teknologi Autodistill. Pemberian label yang akurat sangat penting karena data berlabel ini akan digunakan sebagai dasar untuk melatih dan menguji model deteksi objek.

C. Pelabelan Data (Data Labeling)

Selanjutnya, dilakukan preprocessing data yang meliputi beberapa tahap, seperti pembersihan data, transformasi data, dan pembagian data. Proses ini membantu meningkatkan kualitas data, mengurangi kesalahan, dan memastikan bahwa data siap digunakan untuk tahap analisis selanjutnya agar dapat digunakan secara efektif dalam analisis atau model pembelajaran mesin (machine learning).

Pembersihan data dilakukan dengan mengidentifikasi atau memperbaiki data yang salah atau tidak konsisten, seperti kesalahan pada pelabelan objek. Transformasi data dilakukan dengan menerapkan ukuran gambar sebesar 640x640 agar ukuran gambar lebih kecil dan pelatihan lebih cepat, auto-orientation untuk standarisasi pengurutan pixel, dan auto-adjust contrast untuk meningkatkan normalisasi dan deteksi garis dalam berbagai kondisi pencahayaan. Pembagian data dilakukan dengan 70% data training, 20% data validation, dan 10% data testing.

D. Augmentasi Data (Data Augmentation)

Augmentasi data (data augmentation) adalah teknik yang digunakan untuk meningkatkan jumlah dan keragaman data pelatihan tanpa mengumpulkan data baru. Ini sangat berguna dalam pembelajaran mesin, terutama dalam domain pengenalan pola seperti computer vision dan pengolahan bahasa alami. Tujuan utama dari data augmentation adalah untuk mengatasi masalah overfitting, meningkatkan kinerja model, dan membuat model lebih umum sehingga dapat bekerja dengan baik pada data yang belum pernah dilihat sebelumnya.

Augmentasi data pada tahapan ini dilakukan teknik berupa flipping (horizontal dan vertikal) untuk membantu model agar menjadi tidak peka terhadap orientasi subjek, rotation (-15o sampai 15o) untuk membantu model agar lebih tahan terhadap rol kamera, 90o rotation (clockwise, counter-clockwise, dan upside down) untuk membantu model agar menjadi tidak peka terhadap orientasi kamera, cropping (0% - 20% zoom) untuk membantu model agar lebih tahan terhadap terjemahan subjek dan posisi kamera, shearing (10o horizontal dan 10o vertikal) untuk membantu model agar lebih tahan terhadap kamera dan jarak subjek, grayscaling (15%) untuk menerapkan skala abu-abu secara probabilistik dan membuat model agar lebih cepat dan tidak peka terhadap warna subjek, brightness (-15% sampai 15%) untuk membantu model agar lebih tahan terhadap perubahan pencahayaan dan pengaturan kamera, saturation (-25% sampai 25%) untuk menyesuaikan kecerahan warna pada gambar secara acak, dan exposure (-10% sampai -10%) untuk membantu model agar lebih tahan warna terhadap perubahan pencahayaan dan pengaturan kamera.

E. Perancangan Model (Design Modeling)

Tahap berikutnya adalah perancangan model algoritma YOLO dan RT-DETR. Kedua model algoritma ini dirancang dengan Google Colab menggunakan *library* ultralytics dan roboflow. Setelah perancangan, masing-masing model algoritma akan melakukan latihan, validasi, dan pengujian menggunakan *dataset* yang telah disiapkan sebelumnya untuk memastikan hasil perbandingan yang pasti.

F. Pelatihan Data (Data Training)

Data training adalah proses di mana model pembelajaran mesin (machine learning) dilatih menggunakan dataset yang telah disiapkan agar model dapat belajar dan membuat prediksi atau keputusan berdasarkan pola dalam data tersebut. Data yang dipakai sebanyak 70% (805 gambar).

Proses training melibatkan pengaturan hyperparameter yang optimal, seperti batch size = 16, epochs = 25, image size = 640, dan learning rates = 0,01. Selama proses training, GPU digunakan untuk mempercepat komputasi, mengingat bahwa training model deteksi objek merupakan tugas yang sangat memerlukan sumber daya komputasi tinggi.

G. Validasi Data (Data Validation)

Data validation adalah proses evaluasi data untuk memastikan bahwa data tersebut akurat, konsisten, dan sesuai dengan kriteria atau aturan yang telah ditentukan sebelum digunakan dalam analisis atau setelah melakukan pelatihan model pembelajaran mesin (machine learning). Data validation penting untuk memastikan bahwa data yang digunakan dapat dipercaya dan valid, sehingga hasil analisis atau model yang dihasilkan juga dapat diandalkan. Pada tahapan ini akan menghasilkan metrik evaluasi seperti, precision, recall, F1-Score, dan accuracy. Dari metrik tersebut akan dilakukan evaluasi kinerja untuk mengetahui keakuratan model yang telah di buat agar mampu mendapatkan hasil yang memuaskan pada tahap pengujian nantinya. Data yang dipakai sebanyak 20% (230 gambar).

Setelah model dilatih, dilakukan validasi menggunakan subset data yang tidak digunakan selama training. Validasi ini bertujuan untuk mengukur kinerja awal model dalam mendeteksi kendaraan bermotor pada data yang belum pernah dilihat sebelumnya. Pada proses ini menggunakan beberapa parameter, seperti image size = 640, batch size = 16, threshold confidence = 0,25, dan IoU = 0,75. Hasil validasi digunakan untuk mengevaluasi apakah model sudah cukup baik atau perlu dilakukan penyesuaian lebih lanjut.

H. Pengetesan Data (Data Testing)

Data testing adalah proses evaluasi model pembelajaran mesin menggunakan dataset yang belum pernah dilihat oleh model selama pelatihan. Tujuannya adalah untuk mengukur kinerja model pada data baru dan memastikan bahwa model dapat menggeneralisasi dengan baik di luar data pelatihan. Data yang dipakai sebanyak 10% (115 gambar).

Data testing memberikan gambaran tentang seberapa baik model akan berfungsi di dunia nyata ketika dihadapkan dengan data yang tidak dikenal. Proses yang dilakukan adalah melakukan prediksi model dengan menggunakan beberapa parameter, seperti image size = 640, threshold confidence = 0,25, dan IoU = 0,75.

I. Implementasi Model (Model Implementation)

Setelah berhasil mendapatkan model algoritma dari hasil perancangan, langkah selanjutnya adalah mengimplementasikan model algoritma tersebut dalam program penghitungan kendaraan. Implementasi program ini akan memanfaatkan dua library utama, yaitu OpenCV dan Ultralytics. OpenCV, yang merupakan singkatan dari Open Source Computer Vision Library akan digunakan untuk pemrosesan gambar dan video, seperti deteksi tepi, transformasi gambar, dan pelacakan objek. Sementara itu, Ultralytics, yang dikenal dengan model YOLO (You Only Look Once) untuk deteksi objek real-time, akan digunakan untuk mendeteksi dan menghitung jumlah kendaraan secara efisien dan akurat. Kombinasi dari kedua library ini diharapkan dapat menghasilkan program yang tidak hanya cepat dalam melakukan penghitungan kendaraan, tetapi juga memiliki tingkat akurasi yang tinggi dalam berbagai kondisi lingkungan.

Pada tahapan ini akan dilakukan pengujian lebih lanjut menggunakan video dengan ditambahkan program penghitungan kendaraan. Data yang digunakan merupakan video yang diambil dari 3 tempat yang berbeda (Caman Raya, Stadiun Bekasi, dan Mall Living Plaza) dengan kondisi lalu lintas ramai lancar dan waktu yang berbeda (pagi, siang, dan malam). Dari pengujian ini akan didapatkan hasil berupa keakuratan

berdasarkan jumlah penghitungan kendaraan dari setiap video yang di uji dengan masing-masing model algoritma.

J. Analisis Hasil Akhir

Setelah evaluasi kinerja dilakukan, langkah selanjutnya adalah analisis hasil untuk mendapatkan insight yang lebih mendalam dari data yang diperoleh. Analisis ini melibatkan perbandingan kinerja antara YOLO dan RT-DETR berdasarkan metrik yang telah dijelaskan sebelumnya.

Hasil dari evaluasi kinerja dijelaskan secara rinci untuk memberikan gambaran yang jelas tentang kelebihan dan kekurangan masing-masing algoritma dalam sebuah kesimpulan. Hasil deteksi juga divisualisasikan dalam bentuk gambar berupa grafik yang menunjukkan deteksi objek kendaraan bermotor, sehingga dapat memberikan visualisasi yang lebih intuitif tentang performa model.

Setelah mengikuti tahapan-tahapan diatas secara terperinci, diharapkan penelitian ini dapat memberikan perbandingan yang komprehensif antara algoritma YOLO dan RT-DETR dalam konteks menghitung kendaraan bermotor, serta memberikan wawasan yang berharga bagi pengembangan sistem deteksi kendaraan di masa depan.

3. Hasil

Tabel 1 Hasil Pengujian Model

	YOLO	RT-DETR
Precision	All = 0,802 Bus = 0,718 Car = 0,877 Motorcycle = 0,931 Pickup = 0,753 Truck = 0,733	All = 0,846 Bus = 0,756 Car = 0,896 Motorcycle = 0,929 Pickup = 0,809 Truck = 0,839
Recall	All = 0,783 Bus = 0,735 Car = 0,891 Motorcycle = 0,766 Pickup = 0,683 Truck = 0,84	All = 0,8 Bus = 0,706 Car = 0,902 Motorcycle = 0,881 Pickup = 0,714 Truck = 0,798
F1-Score	All = 0,792 Bus = 0,726 Car = 0,883 Motorcycle = 0,84 Pickup = 0,716 Truck = 0,782	All = 0,822 Bus = 0,73 Car = 0,898 Motorcycle = 0,9 Pickup = 0,758 Truck = 0,817
Accuracy	All = 0,796 Bus = 0,68 Car = 0,93 Motorcycle = 0,82 Pickup = 0,69 Truck = 0,86	All = 0,868 Bus = 0,79 Car = 0,96 Motorcycle = 0,96 Pickup = 0,75 Truck = 0,88

Precision

a. Secara umum, RT-DETR memiliki *Precision* yang lebih tinggi (0,846) dibandingkan dengan YOLO (0,802).
b. Untuk setiap kategori kendaraan, RT-DETR cenderung memiliki nilai *Precision* yang lebih tinggi, terutama pada kategori Truk (0,839) dibandingkan dengan YOLO (0,733). YOLO hanya unggul sedikit pada kategori Motor saja.

Recall

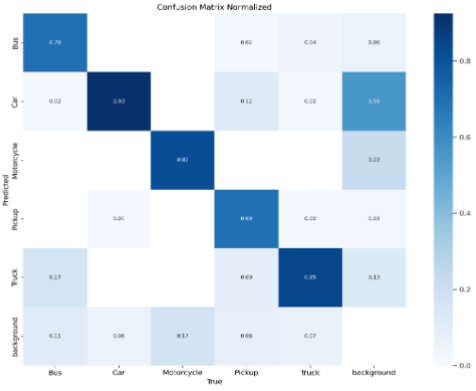
a. RT-DETR memiliki nilai *Recall* yang lebih tinggi (0,783) dibandingkan dengan YOLO (0,8) untuk semua kategori.
b. Namun, pada kategori Truk dan Bus, YOLO menunjukkan nilai *Recall* yang lebih tinggi.

F1-Score

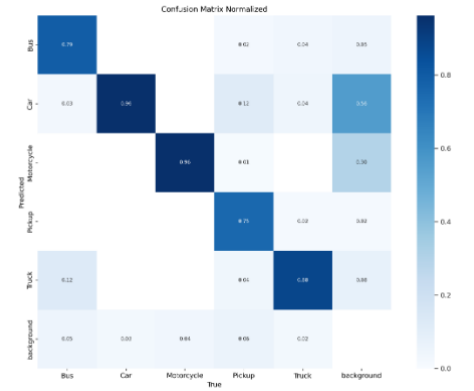
a. *F1-Score* RT-DETR secara keseluruhan lebih tinggi (0,822) dibandingkan YOLOv8 (0,792).
b. RT-DETR unggul dalam kategori dalam semua kategori kendaraan dibandingkan YOLO.

Accuracy

a. RT-DETR memiliki *Accuracy* keseluruhan yang lebih tinggi (0,868) dibandingkan dengan YOLO (0,796).
b. Kategori Motor menunjukkan perbedaan yang mencolok, di mana RT-DETR memiliki *Accuracy* 0,96, sementara YOLO hanya 0,82. RT-DETR unggul dalam kategori dalam semua kategori kendaraan.



Gambar 2 Confusion Matrix YOLO



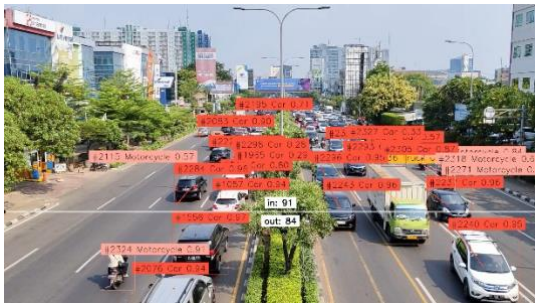
Gambar 3 Confusion Matrix RT-DETR

Berdasarkan hasil evaluasi kinerja dari tabel dan confusion matrix tersebut, beberapa poin analisis dapat diambil:

a. RT-DETR memiliki *precision* yang lebih tinggi dibandingkan YOLO di semua kategori kecuali motor.
b. RT-DETR memiliki *recall* keseluruhan yang lebih tinggi dibandingkan YOLO. Beberapa menunjukkan bahwa YOLO lebih baik dalam mendeteksi kendaraan truk dan bus.
c. RT-DETR juga memiliki *F1-Score* keseluruhan yang lebih tinggi dibandingkan YOLO, yang menunjukkan keseimbangan yang lebih baik antara *precision* dan *recall*.
d. RT-DETR menunjukkan akurasi yang lebih tinggi dibandingkan YOLO di semua kategori kendaraan.

Berdasarkan hasil evaluasi dari tabel tersebut, RT-DETR menunjukkan kinerja yang lebih unggul dibandingkan dengan YOLO. Hal ini terlihat dari semua metrik evaluasi utama seperti *Precision, Recall, F1-Score, dan Accuracy* yang lebih tinggi pada RT-DETR. Secara keseluruhan, RT-DETR adalah algoritma yang lebih unggul dalam tugas deteksi dan peng-hitungan kendaraan. Berikut adalah program untuk menghitung jumlah kendaraan:

Gambar 4 Hasil Penghitungan Kendaraan Pada YOLO



Gambar 5 Hasil Penghitungan Kendaraan Pada RT-DETR

4. Simpulan

Berdasarkan hasil penelitian yang telah dilakukan mengenai perbandingan performa algoritma YOLO dan RT-DETR dalam menghitung jumlah kendaraan bermotor, dapat disimpulkan sebagai berikut:

a. Rekomendasi Algoritma

Hasil evaluasi kinerja sudah dapat dipastikan perbandingannya dari masing-masing algoritma. RT-DETR memiliki kinerja yang lebih unggul dari pada YOLO. Dalam hal ini, RT-DETR merupakan algoritma yang terbaik dalam penghitungan kendaraan dibandingkan dengan YOLO. Oleh karena itu, RT-DETR bisa menjadi sebuah rekomendasi yang tepat untuk kasus seperti ini.

b. Keakuratan Deteksi

Algoritma RT-DETR menunjukkan performa yang lebih tinggi dibandingkan dengan YOLO dalam hal keakuratan deteksi kendaraan bermotor. Contohnya pada nilai akurasi deteksi pada RT-DETR untuk label mobil adalah 0.96 dan motor 0.96 lebih tinggi dibandingkan dengan YOLO yang masing-masing memiliki nilai 0.93 dan 0.92.

c. Efektivitas pada Berbagai Jenis Kendaraan

F1-Score menunjukkan bahwa RT-DETR memiliki performa yang konsisten lebih baik untuk semua jenis kendaraan yang diuji. *F1-Score* untuk motor dan mobil lebih tinggi dibandingkan dengan kendaraan lain, mungkin karena data motor dan mobil lebih banyak dan bervariasi, sehingga lebih mudah diklasifikasikan .

d. Performa pada Berbagai Kondisi

RT-DETR menunjukkan keunggulan dalam kondisi yang telah diuji terutama dalam kondisi pencahayaan yang berbeda-beda.

Secara keseluruhan, penelitian ini menyimpulkan bahwa meskipun kedua algoritma menunjukkan kinerja yang baik, RT-DETR lebih efektif dan akurat dalam mendeteksi kendaraan dibandingkan YOLO berdasarkan hasil pengujian yang telah dilakukan.

DAFTAR PUSTAKA

[1] R. Nurhawanti, “Sistem Pendeteksi Sepeda Motor Pelanggar Marka Jalan Menggunakan Metode Convolutional Neural Networks (CNNs),” *Universitas Pendidikan Indonesia*, May 2019.

[2] F. Rofii, G. Priyandoko, M. I. Fanani, and A. Suraji, "Vehicle Counting Accuracy Improvement By Identity Sequences Detection Based on Yolov4 Deep Neural Networks," *TEKNIK*, vol. 42, no. 2, pp. 169–177, Aug. 2021, doi: 10.14710/teknik.v42i2.37019.
[3] M. Leriansyah, "Deteksi dan Perhitungan Kendaraan untuk Mengetahui Arus Kepadatan Lalu Lintas secara Otomatis Menggunakan YOLO-V3," *Universitas Islam Indonesia*, 2020.
[4] Y. Pratama and E. Rasywir, "Eksperimen Penerapan Sistem Traffic Counting dengan Algoritma YOLO (You Only Look Once) V.4.," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 5, no. 4, p. 1438, Oct. 2021, doi: 10.30865/mib.v5i4.3309.
[5] D. Setiawan, Y. Riti, and N. Trisuwita, "Perbandingan Performa Model SSD Mobilenet V2 dan FPNLite dalam Deteksi Helm Pengendara Sepeda Motor," *Komputika : Jurnal Sistem Komputer*, vol. 13, no. 1, May 2024, doi: 10.34010/komputika.v13i1.10333.
[6] S. Jupiyandi, F. R. Saniputra, Y. Pratama, M. R. Dharmawan, and I. Cholissodin, "Pengembangan Deteksi Citra Mobil untuk Mengetahui Jumlah Tempat Parkir Menggunakan CUDA dan Modified YOLO," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 4, p. 413, Jul. 2019, doi: 10.25126/jtiik.2019641275.
[7] Y. Zhao *et al.*, "DETRs Beat YOLOs on Real-time Object Detection," *ArXiv*, Apr. 2023.
[8] J. E. Guisiano, D. Barretta, É. Moulines, T. Lauvaux, and J. Sublime, "OBJECT DETECTION MODELS SENSITIVITY & ROBUSTNESS TO SATELLITE-BASED ADVERSARIAL ATTACKS," in *IEEE International Symposium on Geoscience and Remote Sensing (IGARSS)*, Athens, Greece, Jul. 2024. [Online]. Available: https://hal.science/hal-04561852
[9] F. P. Rustamy, "DEtection TRansformer (DETR) vs. YOLO for object detection," Medium. Accessed: Jun. 28, 2024. [Online]. Available: https://medium.com/@faheemrustamy/detection-transformer-detr-vs-yolo-for-object-detection-baeb3c50bc3
[10] F. Jalled and I. Voronkov, "Object Detection using Image Processing," Nov. 2016.
[11] R. Munir, *Pengolahan Citra Digital Dengan Pendekatan Algoritmik*. Bandung: INFORMATIKA, 2004.
[12] F. Q. Syuhaila, "SISTEM DETEKSI PLAT NOMOR OTOMATIS MENGGUNAKAN ALGORITMA YOLO V3 DENGAN FRAMEWORK DARKNET," *ITS*, 2020.
[13] L. F. Basuki, "Implementasi Metode Histogram Of Oriented Gradients Dengan Optimasi Algoritma Frei-Chen Untuk Deteksi Citra Manusia," *UNIKOM*, Nov. 2016.
[14] M. Nada, "Penerapan Deep Learning Menggunakan Convolutional Neural Network (CNN)," *Medium*, Jun. 2019.
[15] W. Suartika, A. Y. Wijaya, and R. Soelaiman, "Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) pada Caltech 101," *JURNAL TEKNIK ITS*, vol. 5, no. 1, 2016.
[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91.
[17] L. Zhang *et al.*, "CCDN-DETR: A Detection Transformer Based on Constrained Contrast Denoising for Multi-Class Synthetic Aperture Radar Object Detection," *Sensors*, vol. 24, no. 6, p. 1793, Mar. 2024, doi: 10.3390/s24061793.